International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:13, No:5, 2019

Context Aware Anomaly Behavior Analysis for Smart Home Systems

Zhiwen Pan, Jesus Pacheco, Salim Hariri, Yiqiang Chen, Bozhi Liu

***Abstract*—**The Internet of Things (IoT) will lead to the development of advanced Smart Home services that are pervasive, cost-effective, and can be accessed by home occupants from anywhere and at any time. However, advanced smart home applications will introduce grand security challenges due to the increase in the attack surface. Current approaches do not handle cybersecurity from a holistic point of view; hence, a systematic cybersecurity mechanism needs to be adopted when designing smart home applications. In this paper, we present a generic intrusion detection methodology to detect and mitigate the anomaly behaviors happened in Smart Home Systems (SHS). By utilizing our Smart Home Context Data Structure, the heterogeneous information and services acquired from SHS are mapped in context attributes which can describe the context of smart home operation precisely and accurately. Runtime models for describing usage patterns of home assets are developed based on characterization functions. A threat-aware action management methodology, used to efficiently mitigate anomaly behaviors, is proposed at the end. Our preliminary experimental results show that our methodology can be used to detect and mitigate known and unknown threats, as well as to protect SHS premises and services.

***Keywords*—**Internet of Things, network security, context awareness, intrusion detection

I. Introduction

THE exponential growth of interconnections and interactions between ubiquitous devices has led to the development of the Internet services which evolve from computer and mobile networks to the IoT. It is expected that the number of IoT devices will reach more than 50 billion interconnected devices by 2020 [1]. By networking the devices to enable them to interact with data, computational tasks, and services, the IoT services revolutionize our way of living, working, communicating, educating, maintaining our health, and entertaining, etc. [2]-[4]. IoT applications, such as Smart Infrastructures (e.g., smart homes) are comprised of complex systems and characterized by interdependence, independence, cooperation, competition, and adaptation [5]-[7]. IoT services allow smart infrastructures to collaborate in order to offer customized services that satisfy people's needs. As one of the more important IoT applications, SHS utilize advanced automation and communication techniques to deliver sophisticated monitoring and control over the building's functions [8]. The major objectives of SHS are to achieve reliable and efficient operations, and to significantly reduce operational costs while improving the occupants' quality of living [9].

In the IoT environment, systems interact with each other using different levels of trust relationships, and consequently, require ultimate security solutions to protect information and processes. The exponential growths in the number of interconnected Internet-enabled devices expose the vulnerability of IoT applications such as SHS, to attackers. Even devices which are intended to operate only in local area networks sometimes get connected to the Internet due to careless configuration or to satisfy special needs (e.g., they need to be remotely monitored). There is a huge risk that these devices can be compromised and experience attacks with severe impacts (e.g., life threatening scenarios) [3], [7]. For instance, in 2014, a botnet of more than 100,000 home networking routers and other IP connected home devices such as smart TVs, multi-media centers, and refrigerators have been found to be involved in sending 750,000 malicious e-mails over a two-week period [10]. The security challenge of SHS consists of identifying security constraints on what should be protected. Current cybersecurity solutions are far from being satisfactory; most of them are focused on a single scenario instead of taking a holistic approach to cybersecurity [11]-[14].

One promising solution for the aforementioned problems in the smart home scenario is the Anomaly-based Behavior Analysis (ABA) approach. ABA is an information security mechanism that uses a baseline model to describe the normal behaviors of a system, so that malicious behaviors can be detected when a deviation from the baseline model is observed. Because of its ability for detecting novel threats, ABA has been actively studied by researchers to perform intrusion detection on cyber-attacks. However, since the information acquired from SHS systems can be from a variety of sources (e.g. asset configuration, user identity, network protocols, temporal and spatial information, sensors etc.), the existing ABA methodologies for protecting SHS which can merely analyzing no more than three kind of information (e.g. combine the network protocol information with one or two other types of information) cannot work properly when protecting a sophisticated system [15]-[17]. Hence, these existing ABA methodologies should be improved with a technique that can efficiently handle heterogeneous information and use the information to describe the normal behavior of a system in a more holistic manner. As a technique that is widely used in pervasive computing, the Context Aware computing aims at gathering the information that can be used to characterize the

Zhiwen. Pan and Yiqiang Chen are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology. Chinese Academy of Science, Beijing, China (e-mail: pzw@ict.ac.cn, yqchen@ict.ac.cn).

Jesus. Pacheco, Salim Hariri, and Bozhi Liu are with the Department of Electrical and Computer Engineering, the University of Arizona, Tucson, AZ 85719 (e-mail: {jpacheco, hariri}@email.arizona.edu, bozhiliu@email.arizona.edu).q

situation of a system in real time and adapt behaviors of the system accordingly [18]. This technique provides a mechanism to divide complex SHS behaviors into a finite set of contexts that can be analyzed by ABA methods such that their detection accuracy is improved significantly.

TABLE I
SHS NETWORK THREAT MODEL

Attack Categories	Impacts	Mitigation Mechanisms
Network Sniffing and Port Scanning	Attackers can gain knowledge of devices and resources, and then use them to plan active attacks.	Firewall, Authentication, Encryption, Network-based IDS
Packet Injection, Replay Attacks	Victim assets will be disrupted by responding to malicious packets.	Encryption, Time stamping, Authentication, Network -based IDS, CA-ABAS
Redirection	Messages in transit can be dropped and eavesdropped. Victim assets will be isolated from the network.	Secure Sockets Layer, Authentication, Network-based IDS, CA-ABAS
Man-In-the-Middle Attacks, Spoofing Attacks	Attackers can masquerade as authorized assets, thereby gaining local machine privilege to cause severe damage.	Firewall, Authentication, Encryption, Network-based IDS, CA-ABAS
Network Flooding	Unusually slow network performance and denial of access to assets or services.	Anti-jamming, Firewall, Authentication, Network-based IDS, CA-ABAS
Code injection	Data loss or corruption, lack of accountability, denial of access, or even administrator privileges takeover.	Input validation, Encryption, Authentication, Host-based IDS, CA-ABAS
Physical attacks	Loss or damage of computer and peripheral equipment, eavesdropping network traffic, denial of service.	Activity logging, Access control, CA-ABAS
Physical Damage, Human errors	Damage of computer and peripheral equipment, lack of accountability, denial of service.	Activity logging, Safety policy, CA-ABAS

In this paper, we introduce a novel Intrusion Detection System framework (IDS) named Context Aware Anomaly Behavior Analysis System (CA-ABAS) to protect SHS against known and unknown attacks or failures. In the framework, we defined a data structure for SHS which can be utilized to abstract and map the heterogeneous information collected from multiple resources into a formalized array of context. An anomaly behavior analysis is designed, specific to the contextual array, so that a baseline model which can explicitly describe the normal usage pattern of SHS assets under different contexts is generated. Our anomaly behavior analysis can be regarded as a one-class anomaly detection process, where the baseline model is generated only based on the normal operations of SHS assets. Consequently, the proposed IDS can detect a wide range of cyber or physical attacks against SHS, as long as these attacks result in state change of the home assets. We have evaluated our approach by launching several cyberattacks (e.g. User-identity Fraud, Replay, and Flooding attacks) against our Smart Home testbed developed at the University of Arizona Center for Cloud and Autonomic Computing. The results show that our CA-ABAS can be used to effectively protect SHS with one occupant.

The rest of the paper is organized as follows: The backgrounds of the anomaly behavior analysis, Smart Home vulnerabilities and impact levels are discussed in Section II. In Section III, a Smart Home Context Data Structure is introduced. In Section IV, our CA-ABAS is described in detail. Section V is devoted to present a summary of our experimental evaluation results. Finally, the paper is concluded in Section VI.

II. BACKGROUND

A. Related Works

In general, there are two kinds of behavior analysis techniques: anomaly-based and signature-based approaches [13]. The ABA defines a baseline model to describe all the normal behaviors so that outliers of this model can be considered abnormal behaviors. On the other hand, the signature-based behavior analysis uses a model to explicitly describe all the known abnormal behaviors. Compared with the signature-based approach, the advantage of the anomaly-based approach is its ability to detect unknown attacks. However, the major challenge of the anomaly-based approach is its high false-positive detection rates, especially when there is a big change in the normal operation of the target system.

The ABA approach is usually implemented as a detector within IDS. Because of its ability for detecting novel threats, ABA has been actively studied by researchers to perform intrusion detection on cyber-attacks. The existing works for protecting SHS with IDS are described as follows. In [15], Nobakht et al. proposed an anomaly-based IDS framework. The framework collects home asset usage through monitoring OpenFlow traffic and performs ABA by utilizing Support Vector Machine (SVM) algorithm. The anomaly-based IDS framework proposed in [16] collects SHS wireless signals through probes, and then performs anomaly detection by identifying the assets' physical location. In [17], Mariusz et al. deploy one local IDS within home gateway and one remote IDS on service provider side. In this way, the service provider can update the detection rules to the local IDS based on expert system. The information collected by the two IDS includes system resources, application logs and network traffic [17]. The drawbacks of these existing works are as follows: 1) The ABA algorithm utilized in [15] and [16] are two-class algorithms which means it needs both normal and anomaly data to generate baseline model. Since such a baseline model is trained to differentiate the patterns of normal and anomaly training data, the model's performance for detecting novel attacks (whose patterns are deviate from the patterns of anomaly training data) is questionable. 2) The ABA performed by [16] is only based on physical location information. The ABA performed by [15] can only analyze traffic between gateway and mobile apps. These works cannot protect SHS in a holistic manner. 3) The IDS in [17] rely heavily on expert knowledge, and the baseline

model of their expert system is not introduced in detail. Moreover, the authors in [17] did not validate their work with any experiments. 4) All the existing works did not take context information such as user identity, temporal behavior information, and time slot into consideration.

To detect anomaly usages that occurred in SHS with a sustained low false-positive rate, we propose an anomaly behavior analysis approach which can generate a baseline model to explicitly describe the normal usage pattern of home assets. The contributions of this paper are as follows: 1) Our ABA approach is designed to analyze a variety of kinds of context information including identity information, temporal behavior information, spatial information, and asset availability information. 2) We proposed a context modeling scheme which can aggregate and abstract the context information through mapping operations. 3) We proposed an anomaly characterization algorithm so that multiple baseline models which have one-to-one correspondence to behaviors under each context can be generated. During the real-detection, abnormality level of the current behavior is quantified based on baseline model of the current context. 4) To make the ABA approach achieve better detection performance for anomalies with higher potential threat, a threat-aware action management mechanism is introduced.

B. Vulnerabilities of SHS

According to a security analysis of home applications on the market [19], [20], smart homes are facing two vulnerabilities in general: 1) all the home applications are merely protected by simple username/password mechanisms, and 2) the web-services and smart applications installed in users' laptops and cell phones are usually designed to be over-privileged [21], which means that applications have ability to trigger unauthorized commands. For instance, a phone application which is merely designed to control air conditioner may potentially be used to send a command for controlling a gate. The threat vectors that can be launched against SHS network are shown in Table I. The information includes the category of the attack, the impacts if the attackers succeed, and the possible mitigation mechanisms. Those threats can cause disruption and malfunction of home assets or even create life threatening scenarios, especially when HVAC control and emergency power is affected. Moreover, the threats can be caused by both outsiders and insiders. All the attacks (except Network Sniffing and Port Scanning attack, which is a passive attack) are active attacks which lead to disruption, over-privilege or malfunction of home assets. All these consequences can be reflected by the anomaly behaviors invoked by the victim assets. Since our proposed IDS (CA-ABAS) aims at detecting the anomaly behaviors of home assets under formalized contexts and triggering mitigation actions accordingly, it can be utilized to address the active attacks listed in the threat model.

C. Impact Levels

The impact level is defined as the most severe consequence that a behavior can lead to [22]. Hence, it can be used as a scale to quantify the potential damages of anomaly behaviors that occurred within the SHS. The impact levels of SHS behaviors vary based on the type of involved home appliances and their change of status (e.g. from on to off) [23]. For instance, the impact level of opening the main door is higher than turning on a light. Moreover, switching on an oven at random time is more severe than switching it off, since switching on the oven could result in a fire accident, while switching off the oven can merely result in the food being uncooked. On the other hand, the impact levels of behaviors can be quantified by referring to the Federal Information Processing Standards Publication 199 (FIPS 199) impact level assessment criteria [22]. FIPS 199 characterizes the impact of threats as Low, Moderate or High. The Low impact represents the limited adverse effect on assets or individuals, the examples of such impact include minor damage to assets, minor financial loss, and disturbance of individuals. Moderate impact represents significant adverse effect, such impact include damage of property, privacy leaks, big financial loss, and minor harm to individuals. High impact represents severe or catastrophic adverse effect; such impact includes major financial losses, major damage to assets, and severe harm to individuals.

TABLE II
SHS IMPACT MODEL

Operations	Impact level	Consequence
Turn on lights, Turn on AC	Low	Minor financial loss, Minor disturbance of individuals
Turn off AC, turn off lights , turn off oven, close the garage door, close faucet, close front door	Low	Minor disturbance of individuals
Open garage door, open faucet	Moderate	Huge financial loss
Watch monitor, close monitor	Moderate	Privacy leaks, life threatening scenarios
Open front door, open oven	High	Harm to individuals, major financial losses

Based on the aforementioned categories of impact levels, we proposed a baseline impact model in Table II which includes some of the SHS behaviors, so that other SHS behaviors can be mapped into the model accordingly. Through assigning each operation an impact level, threat-aware action management (see Section IV.*F*) can be implemented to handle suspicious behaviors. For instance, the failure of detecting main door break-in is unforgiveable; hence, we should pay more attention to those critical behaviors even when such operations are only slightly suspicious. Another example is that, the detection of one suspicious minor operation (e.g. turn off a light) is usually trivial; hence such detection should be selectively suppressed.

III. SMART HOME CONTEXT DATA STRUCTURE (SHCDS)

The development of SHCDS is based on the concept of context modeling which is to represent the information

acquired from the sensors and resources during execution of the system into context which can explicitly describe the system's behavior [24]. In general, the information acquired from IoT devices exhibit two inherent characteristics: redundancy and heterogeneity. By utilizing our data structure, the redundancy of the IoT information can be mitigated by grouping the useful information into corresponding classes based on their categories. On the other hand, the heterogeneity of information can be eliminated by mapping the relevant information into higher level context which is more precise and more accurate.

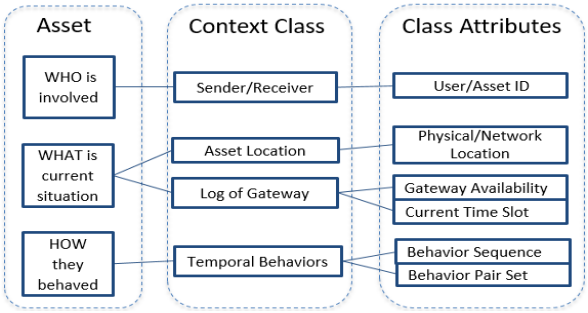

Fig. 1 SHCDS

The hierarchical structure of the SHCDS is shown in Fig. 1. The highest hierarchy defines the sentence format of our context with tuples ordered as: 1) Who: which assets are involved; 2) How: how they behave; and 3) What: what is the current situation [25]. In the second hierarchy of SHCDS, a set of context classes are created to perform grouping and mapping operations on the IoT information. During the grouping operation, each context class selects the information they are interested in as input and discard the other information. For instance, the context class named Sender is configured to select the following information as input: 1) Sender's IP address, sender's authentication information and sender's HTTP cookies, which are all extracted from the packet stream. 2) User's account information which is acquired from the log of the configuration. 3) Sender's physical location which is acquired from a GPS signal. During the mapping operation, the grouped features are aggregated and mapped into higher-level information based on predefined methods. For instance, we can obtain the higher-level information named "User ID" by aggregating the HTTP cookies and credential information provided by the sender with pre-defined account information. The higher-level information generated will be stored as attributes into classes as shown in the lowest hierarchy. Note that class attributes listed in Fig. 1 contain high-level information that can be generated in different ways based on different applications. An implementation of the SHCDS is introduced in detail in Section IV.*D*.

Note that our SHCDS aims at describing the control events which occur between users/gateway (as sender) and end devices (as receiver). However, some end devices such as sensors may act as senders to update local data to the gateway. Such events can be protected by the Cyber DNA techniques we proposed previously in [5], [26].

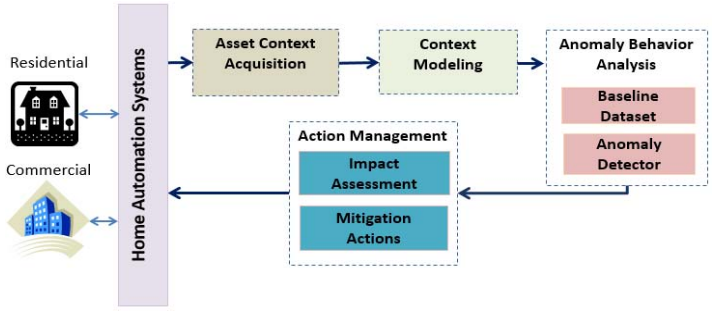

Fig. 2 CA-ABAS Architecture

IV. Context Aware IDS

A. Architecture

Our IDS framework consists of four modules that perform Context Aware ABA for protecting SHS networks, as shown in Fig. 2. The asset context acquisition module is in charge of continuously monitoring SHS networks and providing the required low-level context information for further processing. By utilizing our SHCDS, the context modeling module aggregates and maps the context information into higher-level context format named contextual array. The anomaly behavior analyzer determines whether or not the contextual array is normal based on our novel anomaly characterization function. The output of the analyzer is then taken by the action management module to perform impact analysis, so that appropriate protective actions that can stop or mitigate the impact of the detected attacks can be determined.

B. SHS Testbed

The experimental testbed (see Fig. 4) used to evaluate our approach includes a secure gateway which can provide services for automation and networking purposes, and end devices which include all the home appliances and sensors. The goal of our testbed is to simulate the operations of a home automation system with the functionality of safety control, energy saving, and the improvement of occupants' comfort and convenience. The home appliances in our testbed include: front door, garage door, oven, faucet, light, window, cooking bench, and water heater.

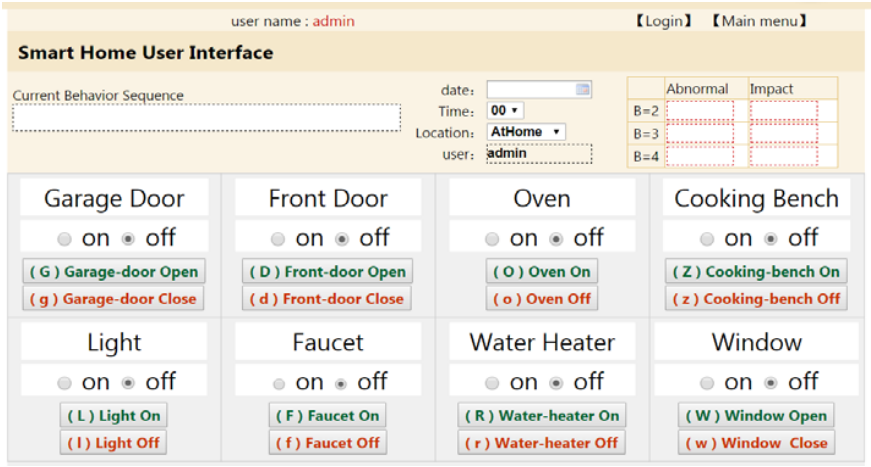

Fig. 3 Smart home website user interface

In our system, a gateway is designed to provide HTTP-based web services with website interface (see Fig. 3), so that the home owners can use mobile devices or computers to remotely monitor and control the appliances. In the meantime, the gateway can act as a central controller to trigger commands to end devices based on user-defined control scenarios. To verify the users' credential, the gateway utilizes a password authentication whenever an operator becomes active. If an operator's usage pattern is found suspicious by our behavior analysis approach, a further authentication mechanism, which can be either face recognition or text message validation, will be triggered to the suspicious operator.

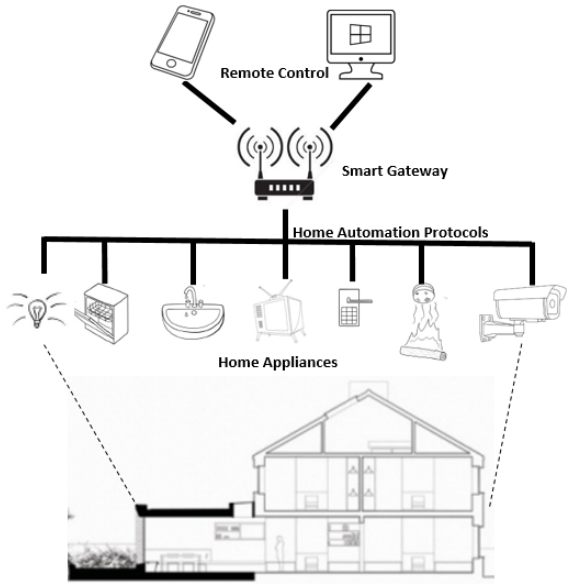

Fig. 4 Home Automation System Testbed

C.Asset Context Acquisition

In our approach, all the information which might be used to describe the situation of SHSs will be collected. Table III shows in detail the context information we collected about the SHS.

Based on the types of acquisition process, the collected information is classified as: 1) features extracted from packet streams, 2) manually provided features, and 3) locally provided features. In what follows, we describe the characteristics of each feature.

1) Features Extracted from Packet Streams

The packet streams in our ABA testbed consist of both network packet traffic and serial port traffic. The packet stream (such as HTTP stream) carries the interaction between users' mobile device and secure gateway; hence, it is the major resource of features regarding users' remote control usages. Moreover, some of the SHS assets (such as Nest products) may interact with users' mobile devices through IP router instead of SHS gateway, these interaction can be captured either through sniffing the router traffic or through integrating the routing function into the secure gateway so that the IP router is no longer needed. As to the users' physical control usages, which are invoked through physically accessing to the end devices, the end devices can be configured to update these usages to the gateway through sending either serial port frames or smart home protocol packets. To implement the data collection in an integrated SHS system with communication traffic encrypted, refined APIs from asset providers are needed so that the data collection can be performed through their gateway assets [27]-[29].

2) Manually Provided Features

Home occupants can manually provide information to the gateway through the user interface. Such information may include preference, asset configuration, account information, and etc. For instance, whenever a new home appliance is put into use, owners should register it in the secure gateway by providing its device type and weight of importance.

3) Locally Provided Features

These features can be acquired locally from the system log of our secure gateway. An example of such features includes

current time, available memory, current CPU utilization, and current buffer utilization of the gateway.

D.Context Modeling

Based on the utilization of the SHCDS, the heterogeneous features are dynamically mapped and aggregated into high-level context information called Context Attributes. Those attributes are then formed as a well-organized feature space called Contextual Array which will be used for further analysis. Our modeling process consists of two steps: 1) Unified Model Language (UML) model development, and 2) Contextual Array formation. In what follows, we describe each step in further detail.

TABLE III
SHS ASSET CONTEXT INFORMATION

Feature identifiers	Resource of the context features	Description
User's primary credential	Manually provided	Each user's pre-registered account name, password, and authorization level.
User's Secondary credential	Manually provided	Each user's pre-registered phone number, face recognition pattern.
Device ID	Manually provided	The identity of each registered appliance assigned based on its usage (e.g. an oven can be tagged as 'oven#1').
Impact Levels	Manually provided	The level of security impact (low/moderate/high) for each control behavior (see Section II.*C*).
Current behavior	Packet streams	The user's remote control behavior is carried by a communication packet. The physical control behavior is carried by a change-of-state notification sent by the home assets.
Status of home appliances	Serial port traffic	The latest operation status of a home appliance (e.g. on/off).
Behavior Invoke Time	Locally provided	The time when a remote control behavior or a physical control behavior is captured by the gateway.
Available Memory	Locally provided	The current available memory of the gateway by taking current operations into consideration.
CPU usage	Locally provided	The amount of CPU used by the current operations.
Buffer Utilization	Locally provided	The amount of buffer of gateway required by current operations.

1) UML Model Development

In the first step, a set of context classes are created as a class diagram by utilizing UML modeling technique [30]. As defined in our Context Aware Data Structure, the class diagram consists of four context classes: 1) sender class to maintain the credential information of each user, 2) asset location class to derive the location information of users, 3) log of gateway class to maintain gateway system log, and 4) temporal behavior class to describe the usage of home assets in the time domain.

Once the features for an asset usage are obtained via context acquisition, the context classes perform data modeling on those features based on two operations: grouping and mapping. During the grouping operation, each class selects its relevant context features as their input. For instances, HTTP cookies and credential information of the users are grouped into the Sender class, and all the information which is locally provided by the secure gateway is grouped into Log of Gateway class.

After the grouping operation, context classes perform pre-defined mapping operations to map the grouped features into context attributes as follows:

User ID (U) is a string with one to one correspondence to each valid user. A user is regarded as valid if only if he/she provides a valid primary credential (e.g. enter the right password).

Behavior Sequence (B) is a string that contains a sequence of captured behaviors which are consecutive to each other. A captured behavior is represented as a string indicating a state change of the home appliance which is caused by the captured behavior. For instance, a Nest thermostat has four operational models as "heat", "cool, "off" and "eco"; and its temperature mode can be switched within a predefined range. To map its modes into finite states, we can use temperature slots to map every certain range of temperatures into specific certain slot. Specifically, if we set a range of 10℃ as one time slot and name the time slot as its temperature up-bound, a thermostat "T1" with operational model as "eco" and temperature mode as 27℃ (belongs to the temperature slot named "30") can be mapped into a state as "T1E30". Similarly, if a captured behavior causes the main door "D1" to switch to state "O" (open/on), we can represent this behavior as "D1O". These captured behaviors are considered as consecutive to each other if 1) they are invoked by the same user, and 2) the time interval between two behaviors is below a pre-defined threshold (e.g. 10 seconds). We defined the maximum length of behavior sequence as $|B|$ and use sliding window to discard the earliest behaviors if the number of consecutive behaviors exceeds the value of $|B|$. For instance, given $|B| = 3$, if a user invoked the following five consecutive behaviors: open garage door (G1O), turn on light No. 2 (L2O), close garage door (G1C), close light No. 2 (L2C), and open main door (D1O), we can represent those behaviors in turn as the following three Behavior Sequences:{"G1C", "L2C", "D1O"}, {"G1O", "L2O", "G1C"}, { "L2O", "G1C", "L2C"}, and {"G1C", "L2C", "D1O"}. On the other hand, if a user invoked a behavior G1O which is not consecutive to any of the other existing behaviors which means no behavior is invoked during 10 seconds before and after the behavior G1O, the corresponding Behavior sequence would be formed as {"G1O", "N/A", "N/A"}.

Behavior Pair Set (S) is a multiset that consists of the pairs of captured behaviors which are belonging to the same Behavior Sequence. The Behavior Pair Set can explicitly describe the one-to-one correlation of behaviors within each Behavior Sequence. The formation of the Behavior Pair Set for a Behavior Sequence where $|B| = 3$ is as follows: Given a Behavior Sequence {"G1C", "L2C", "D1O"}, the corresponding Behavior Pair Set is formed as {"G1C,D1O", "G1C,L2C", "L2C,D1O"}. Given a Behavior Sequence {"G1O", "L2C", "N/A"} which includes two consecutive

behaviors, the corresponding Behavior Pair Set is formed as {"G1C,L2C", "G1C,N/A", "L2C,N/A"}. Given a Behavior Sequence {"G1C", "N/A", "N/A"} which includes only one behavior, the corresponding Behavior Pair Set is formed as {"G1C,N/A", "G1C,N/A", "G1C,N/A"}. We defined the number of behavior pairs included in the Behavior Pair Set as $|S|$. As shown from the previous examples, $|S| = 3$ when $|B| = 3$. Similarly, if $|B|$ of Behavior Sequence is set to either 2 or 4, $|S|$ will be equal to either 1 or 6.

Time Slot (T) is an integer indicating which slot is the current time belongs to. In our research, we set the duration of one time slot as two hours so that a whole day is divided into 12 slots. For instance, if a behavior is invoked at 8:25 a.m., its corresponding time slot is 5. By setting the duration of a time slot as two hours instead of one or half an hour, more behavior patterns can be included into each time slot.

Physical Location (L) is a string indicating if the captured behavior invoked from either inside or outside of the home. At-home behaviors are those commands invoked through manual access of home appliances or through remote commands carried by the local Wi-Fi network traffic. Outside-of-home behaviors are invoked through remote commands that are carried by external IP network traffic.

Gateway Availability (G) is an integer that represents the current availability of the gateway, and is used to identify possible threats targeting the gateway availability. Since the gateway is used to communicate sensors and actuators with the internet and, at the same time provide local access to appliances, it is important to constantly verify its availability [15]. In order to achieve this task, our secure gateway performs continuous monitoring for three parameters [15]: 1) available memory (AM), 2) CPU utilization (CU), and 3) buffer utilization (BU). We assume that under no attack and no monitoring activity, all the parameters are at full capacity; hence, we assign 100% to gateway availability. For example N_{AM}=*76,1832* is the nominal available memory of the system, which means that 76,1832 equals 100% of memory availability. From here, any memory consumption (C_{AM}) is compared against N_{AM}, giving a percentage of available memory. Equation (1) shows how to compute the impact *I* of a given attack, where C_{AM}, C_{BU}, and C_{CU} are the current values of AM, BU, and CU, and N_{AM}, N_{BU}, and N_{CU} are the nominal values for each feature.

$$I = 1 - [\frac{C_{AM}}{N_{AM}} \cdot 0.4 + \frac{C_{BU}}{N_{BU}} \cdot 0.3 + \frac{C_{CU}}{N_{CU}} \cdot 0.3] \quad (1)$$

From (1) we notice that when the current values are nominal and the impact is zero. The output of (1) cannot be directly used in our context array due to the continuous nature of its values. We use (2) to obtain four levels of availability, where level 3 is the desired stage under normal operation and level 0 is the worst case scenario.

$$G = \begin{cases} 0 & if\ I \geq 0.75 \\ 1 & if\ 0.5 \leq I < 0.75 \\ 2 & if\ 0.25 \leq I < 0.5 \\ 3 & if\ I < 0.25 \end{cases} \quad (2)$$

IV. Contextual Array Formation

In the second step, context attributes are retrieved from classes to form a contextual array which is a data array describing the current context. The contextual array is then stored into the SQL database for further analysis. All the contextual arrays are stored as a matrix with six dimensions $C = [U, T, B, S, L, G]$, where the capital letters within the brace are the abbreviations of attribute dimensions. For instance if user Bob remotely turns on the light "L1", air conditioner "A1", and water heater "H1" at 6:20 p.m. before arriving home, the contextual array to describe such context is shown in Fig. 5.

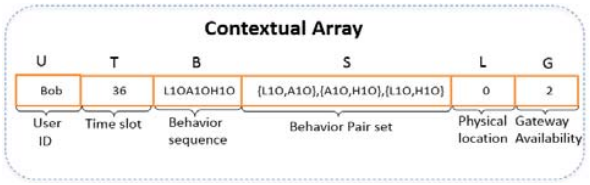

Fig. 5 Contextual array for a sample context

E. Anomaly Behavior Analysis

Our anomaly behavior analysis (ABA) models a user's usage pattern of domestic devices by statistically analyzing their consecutive behaviors, so that the potential threat of the captured behaviors can be estimated. The analyzer for such analysis is defined as a system $A = (f, N)$ with two components, N and f. The component N is a baseline dataset $N = [U^N, T^N, B^N, S^N, L^N, G^N]$ which includes all the historical normal behaviors that have been captured. Since a user's recent behaviors can reflect their usage pattern better than older behaviors, the analyzer keeps removing the outdated arrays (e.g. arrays stored for more than two months) from the database to avoid data drift. In the proposed system, the baseline dataset N consists of a number of sub-dataset $N^{u,l}$, where $\{u, l\}$ is the index of sub-dataset. In this way, the contextual arrays triggered by different users u at different locations l are stored separately into the corresponding sub-dataset.

The component f is an anomaly detector defined as $f(E, N)$, where E is the contextual array to be analyzed. The output of $f(E, N)$ is the detection decision indicating whether the contextual array being analyzed is normal or not. The design of $f(E, N)$ is inspired by works presented in [31]-[33], where the normal consecutive transitions of WI-FI and DNS protocols are modeled through an anomaly characterization function. The functions proposed in these works apply statistical techniques to quantify the n-transition patterns in the protocols and then characterize the normal score of these transitions [31], [32]. Specifically, if the frequency of a sequence of n protocol transitions over an observation time window is far different from the frequency of the same transition that is stored in the baseline model, the normal score of this transition would be low. Such score characterization is based on an assumption that the transition patterns follow a well-defined protocol state machine with no context information involved. However, in the SHS applications, users are likely to perform the same set of

consecutive behaviors with different orders or to skip some of the behaviors when performing the same set of consecutive behaviors, we design $f(E,N)$ as a combination of detection decisions for both Behavior Sequence and Behavior Pair Set. That is, given an array input $E=\{u,t,b,s,l,g\}$, where $u \in U^N, t \in T^N, b \in B^N, t \in T^N, b \in B^N, s \in S^N, l \in L^N, g \in G^N$. The output of $f(E,N)$ is the array's normal score which can be represented as:

$$f(E,N) = Decision^b(u,t,b,l,g) + Decision^s(u,t,s,l,g) \quad (3)$$

where $Decision^b(u,t,b,l,g)$ is the detection decision for behavior sequence b and $Decision^s(u,t,s,l,g)$ is the detection decision for the behavior pair set S.

The value of $Decision^b(u,t,b,l,g)$ can be either 1 as normal or 0 as abnormal, and it can be determined based on the following equation:

$$Decision^b = \begin{cases} 1 \; if \; score^b(u,t,b,l,g) \geq score^b(N^{u,l}) \\ 0 \; if \; score^b(u,t,b,l,g) < score^b(N^{u,l}) \end{cases} \quad (4)$$

where $score^b(u,t,b,l,g)$ is the normal score of the behavior sequence being analyzed, $score^b(N^{u,l})$ is the baseline score corresponding to a dataset of a specific user ID and physical location. The value of $score^b(u,t,b,l,g)$ can be obtained as:

$$score^b(u,t,b,l,g) = \frac{count(u,t,b,l,g)}{count(u,t,l,g)/kindofb(u,t,l,g)} \quad (5)$$

where $count(u,t,b,l,g)$ is a database operation that counts the number of contextual arrays in the baseline dataset $N^{u,l}$ whose values of $\{U,T,B,L,G\}$ equal to $\{u,t,b,l,g\}$. In other words, it counts how many behavior sequence b the user u has invoked at time t and location l with gateway availability as g during normal operations. The $count(u,t,l,g)$ operation counts the number of contextual arrays that user u has invoked at time t and location l with availability g during normal operations. $kindofb(u,t,l,g)$ is a database operation that counts how many kinds of behavior sequence that user u has ever invoked at time t and location l with availability g during normal operation. Hence, $count(u,t,l,g)/kindofb(u,t,l,g)$ indicate the average number of behavior sequences that user u has ever invoked at time t and location l with availability g during normal operation. The baseline score $score^b(N^{u,l})$ can be obtained by calculating $score^b$ for every contextual array stored within baseline dataset $N^{u,l}$ and then selecting the $score^b$ with the smallest value as our baseline score for $N^{u,l}$. Similarly, the value of $decision^s(u,t,s,l,g)$ can be determined as:

$$Decision^s = \begin{cases} 1 \; if \; score^s(u,t,s,l,g) \geq score^s(N^{u,l}) \\ 0 \; if \; score^s(u,t,s,l,g) < score^s(N^{u,l}) \end{cases} \quad (6)$$

where $score^s(u,t,s,l,g)$ is the normal score of the Behavior Pair Set b being analyzed. The value of $score^s(u,t,s,l,g)$ can be obtained as:

$$Score^s(u,t,s,l,g) = \frac{\sum_{i=0}^{|s|} count(u,t,s_i,l,g)}{count(u,t,l,g)/kindofs(u,t,l,g)} \quad (7)$$

where $kindofs(u,t,l,g)$ is a database operation that counts how many kinds of behavior pair set s that user u has ever invoked at time t and location l with gateway availability as g during normal operation and $count(u,t,s,l,g)$ is an operation counting how many times user u invokes behavior pair s_i at time t and location l with availability g during normal operation. The baseline score $score^s(N^{u,l})$ can be obtained by calculating the $score^s$ for every contextual array stored within baseline dataset $N^{u,l}$ and then selecting the $score^s$ with the smallest value.

Since the normal score $f(E,N)$ of a context array is the sum of $Decision^b(u,t,b,l,g)$and $Decision^s(u,t,s,l,g)$, where values of the two decisions can be either 0 or 1, our anomaly detector f is a ternary classifier, which classifies the contextual arrays as either normal ($f(E,N)=2$), suspicious ($f(E,N)=1$), or abnormal ($f(E,N)=0$).

F. Threat-Aware Action Management

In this phase, threat-aware action management is performed to handle suspicious behaviors by further classifying them as either normal or abnormal and handle anomaly behaviors by triggering mitigation actions. The meaning of threat-aware is that more efforts should be put to detect and mitigate the anomalous behaviors with higher potential impact. To achieve such goal, we need to make the detection rate of high-threat anomaly behaviors as high as possible, and in the meantime, reduce the overall false positive rate by ignoring the suspicious trivial behaviors. Based on time window T (we named it the threat–aware window), which slides along the current set of consecutive behaviors, the consecutive behaviors within window T are considered as high-impact if consist only of at least one behavior whose impact level is moderate or high. Otherwise, it is considered as a low-impact. The impact level of each behavior is determined based on the impact model shown in Table I. In this way, the detection rates of high-impact contextual arrays are increased by detecting a high-impact array as anomaly if its normal score $f(E,N)$ is either 0 or 1. Similarly, the false positive errors are minimized by considering a low-impact contextual array as an anomaly only if its normal score $f(E,N)$ is 0.

Since the length of behavior sequence ($|B|$) and the length of threat-aware window ($|T|$) are the two parameters which affect the detection performance of our proposed approach, the process for determining $|B|$ and $|T|$ can be considered as a single parameter optimization problem. Given that the maximum length of consecutive behaviors observed during training data collection is C_{max}, it is predictable that the detection performance of the proposed function is constant for any $|B|$ and $|T|$ whose value are greater than or equal to C_{max}. This is because if the length of sliding window is set as greater than or equal to C_{max}, we only need one window (no need to slide) to describe the consecutive behaviors whose length is smaller than or equal to C_{max}. Consequently, we proposed a data training process to determine the proper value of $|B|$ and $|T|$ from the

value range of $|B| \in [2,\ C_{max}]$ and $|T| \in [1,\ C_{max}]$. The data training process can be divided into two steps. In the first step, the value of $|B|$ is determined by setting $|T|$ as $(C_{max}+1)/2$ if C_{max} is odd or as $C_{max}/2$ if C_{max} is even. After fixing the value of $|T|$, we perform training $C_{max}-1$ times to generate baseline models based on every possible value of $|B|$ and select the value of $|B|$ which can achieve the highest detection accuracy as the optimal value. In the second step, given the optimal value of $|B|$, we perform data training C_{max} times to generate baseline models based on every possible value of $|T|$ and select the value of $|T|$ which can achieve the highest detection accuracy as the optimal value. The detection accuracy can be calculated as:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

where TP, FP, TN, FN is true positive rate, true negative rate, true negative rate, false negative rate, respectively.

Algorithm 1. Mitigation Action Management

ActionManagement(E, u, $f(E,N)$):
1: **if** u.ip_address is changed **or** u.cookie is changed **then**
2: u.valid← False
3: **if** E.impact = Low **and** f(E,N) = 0 **and** u.valid = False **then**
4: E.abnormal← True
5: **else if** E.impact ≠ Low **and** $f(E,N) \leq 1$ **and** u.valid = False **then**
6: E.abnormal← True
7: **end if**
8: **else**
9: store E into baseline dataset N
10: **end if**
11: **if** E.abnormal = True **then**
12: Send notification to home owners
13: Suspend authorization of user u
14: Send identity challenge to user u and waiting for feedback
15: **if** u passed the identity challenge **then**
16: update u.ip_address and u.cookie
17: u.valid← True
18: **else if** feedback from home owners not received **or** Identity challenge of user u is failed **then**
19: Trigger security alarms and terminate the authorization of user u
20: **end if**
21: **end if**

Fig. 6 Mitigation Action Management Algorithm

Whenever the behavior of a user is detected as an anomaly, the action management module will consider the user's identity as suspicious and trigger secondary authentication by either sending a text message with a verification code to the user's mobile phone or by conducting facial recognition. In the meantime, the home owner will be notified about the anomaly behaviors through a text massage. If the anomaly user passes the secondary authentication, it means that the current alarm is very likely to be a false alarm. Consequently, our action handler will consider the user's identity as valid for a predefined time period (e.g. one hour). During such time period, a valid user will not be challenged with a secondary authentication again and all the behaviors triggered by the user will be considered as normal. The time period for considering a user as valid will be terminated early if the credential (e.g. HTTP Cookie) for the valid user has expired or the network address of the valid user has changed. Otherwise, if the suspicious user cannot pass the secondary authentication, the user will be identified as malicious and the user privilege will be terminated. The pseudocode for our mitigation action management scenario is shown in Fig. 6.

V. Experimental Results

A. Training Stage

In order to generate the daily usage data of a Smart Home environment, we invited one of our research group members at the University of Arizona Center for Cloud and Autonomic Computing to continuously log his/her daily usage of home appliances through a web interface (see Fig. 3). These daily usages include both physical usages and remote usages. Once the user inputs control commands via the interface, a packet will be sent from the web interface to the smart gateway as a normal record. Some of the SHS assets can be configured to automatically trigger a set of behaviors based on sensor inputs. Since the normal patterns of these automatically-triggered behaviors follow some pre-determined control logics, these normal patterns are much simpler than normal patterns of the behaviors being manually triggered. Hence, we evaluate the performance of our approach by only analyzing the behaviors which are triggered manually without following any pre-determined logic. A total of 8,110 normal records are collected in a 90-day period.

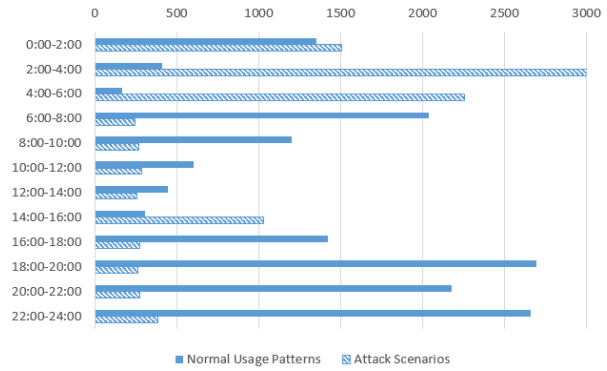

Fig. 7 The temporal distribution of collected dataset

The objective of our data training stage aims at generating baseline models for real-time detection. A baseline model is built using a baseline dataset $N^{u,l}$ and its corresponding baseline scores $score^b(N^{u,l})$ and $score^s(N^{u,l})$. Since there is one user involved in our dataset ($u = "Bob"$) and the user's physical location can be recognized as either at home ($l = 0$) or out of home ($l = 1$), we generated two baseline models which describe the user's at-home usages and out-of-home usages separately as: 1) $N^{Bob,1}$, $score^b(N^{Bob,1})$ and $score^s(N^{Bob,1})$; 2) $N^{Bob,0}$, $score^b(N^{Bob,0})$ and $score^s(N^{Bob,0})$. In order to evaluate the detection performance over time, we perform data training four times for four time intervals by taking four different datasets N (including contextual arrays collected during the first 15 days, 30 days, 45 days, and 60 days) as the training dataset. It is noted that this data training is a one-class ABA process since no abnormal behaviors are involved.

B. Detection Stage

During the detection phase, a set of cyberattack scenarios were simulated to generate anomalous behavior records for performance evaluation purpose. Those scenarios consist of single behavior or multiple malicious behaviors, and can be launched either by a user's account being spoofed or by the smart gateway being compromised. The attack scenarios we simulated are as follows:

1) **Status Switching attacks** are launched by continuously switching the operational status of home assets. We launched such an attack based on two scenarios: 1) sending about 100 control commands to a home asset every second for denial-of-service purpose; 2) sending three control commands to a non-critical home asset (e.g. lights) every second for prank purpose.
2) **Asset Manipulation attacks** are launched by invoking control behaviors to critical home assets when home occupants are not aware.
3) **Break-in attacks** are launched by opening the door, garage door or windows and then physically invoke some malicious behaviors. We invoke such attack by opening a door or window when the home occupants are more likely to be sleep or out of the house, for example at work. About 65% of the behaviors which are invoked at night were done so without turning on the lights. About 80% of the attack scenarios involve at least one asset control behavior after the break-in. These asset control behaviors may include: 1) switching light on and off, 2) closing doors or windows, 3) manipulating critical appliances such as oven, faucets and monitors to cause financial loss and privacy leakage. The rest (about 20%) of the attack scenarios involve no asset control behavior after the break-in, which means intruders were trying to sneak in for stealing possessions.

Approximately 10,520 attack records were generated during the online detection test. The temporal distribution of both normal and malicious records based on time slots is illustrated in Fig. 7. As shown in the figure, the majority of attacks were launched during the night when the home occupants were likely to be sleeping. Note that, since the data training are performed without a dataset which describes attack behaviors (namely anomaly dataset), the patterns of all the simulated attack scenarios are unknown by our baseline model.

The detection rates of our approach were evaluated by counting the percentage of attack scenarios that have been detected against all attack scenarios launched. The false positive rate for analyzing normal behaviors was evaluated by taking the normal behaviors collected in a 30-day period as input for real-time detection. For instance, to evaluate the false positive rate for the baseline model generated based on the normal behaviors collected during the first 60 days, the normal behaviors collected between day 61 and day 90 were tested.

C. Detection Results

In the first experiment, we determine the effectiveness of the proposed approach by evaluating the detection performance variation of the approach in the time domain. After determining the optimized value of $|B|$ and $|T|$ through the data training process, detection was performed with $|B|=3$ and $|T|=2$. In Fig. 8, the points in the detection performance curve from left to right indicate the detection performance of the baseline models that are generated based on the dataset collected in a 15-day, 30-day, 45-day, and 60-day period, respectively. As the period increases, the false positive error rate of each baseline model is reduced; however, the detection rate also declined. The reduction of the false positive rate in Fig. 8 is caused by the fact that more usage patterns were gathered when the collection period is extended. On the other hand, since the normal usage dataset was generated by recording the asset usage behaviors of a user who has an unfixed life style (e.g. sometimes staying up late, sometimes staying at home during workday), some of his/her usage patterns were somehow similar to the attack scenarios. Consequently, the detection rate declined slightly when more patterns were collected.

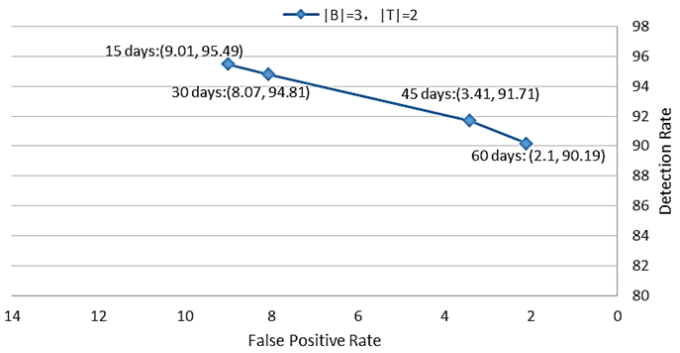

Fig. 8 Detection performance of baseline models generated based on datasets collected in different lengths of time

In the second experiment, we determine the effectiveness of our threat-aware detection mechanism (introduced in Section IV.*F*). Our threat-aware mechanism can further classify the suspicious behaviors into either normal or abnormal based on the impact level of behaviors within sliding window T. In this way, the proposed approach can put more effort on detecting anomaly behaviors with higher potential impacts. Moreover, since normal users are likely to invoke more low-impact behaviors than attackers, this approach can avoid frequent false alarms by putting less effort on detecting low-impact anomaly

behaviors. The detection accuracies of the proposed approach with and without the threat-aware mechanism are shown in Table IV. The accuracy evaluation is made based on the training dataset which is collected in a 60-day period. The detection performance of our approach with the threat-aware mechanism is shown as the left column of the table. The middle column indicates the detection performance of the approach with a detection mechanism, which simply classifies all the suspicious behaviors as abnormal behaviors. The right column indicates the detection performance of the approach with a mechanism which simply regards all the suspicious behaviors as normal behaviors. Results in the table show that the proposed threat-aware mechanism can achieve the highest detection rate for detecting high-impact anomaly behaviors and trigger less false alarms (lower false positive rate) when analyzing low-impact normal behaviors. This indicates that our mechanism can put more effort on detecting attacks with higher potential impacts and put less effort on triggering trivial alarms. Moreover, the proposed approach with a threat-aware mechanism achieves the highest detection accuracy. This indicates that the threat-aware mechanism can achieve a better tradeoff between false positive rates and detection rates.

TABLE IV
PERFORMANCE EVALUATION OF THREAT AWARE MECHANISM

	Proposed mechanism	All suspicious as Abnormal	All suspicious as Normal
Detection Accuracy	**93.59%**	92.14%	87.4%
Detection rate of high impact behaviors	**96.85%**	96.85%	83.61%
FP rate of high impact behaviors	10.27%	10.27%	4.3%
Detection rate of low impact behaviors	93.13%	94.36%	82.31%
FP rate of low impact behaviors	**5.73%**	8.63%	5.17%

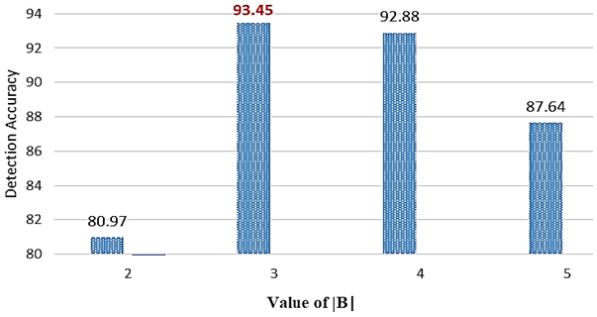

Fig. 9 Detection performance of baseline models generated based on different values of $|B|$

In the third experiment, we evaluate how the value of $|T|$ and $|B|$ affect the detection performance of the proposed approach. The evaluation for $|B|$ was made by generating multiple baseline models based on different values of $|B|$ and comparing the detection performances of these baseline models. Similarly, the evaluation for $|T|$ was made by comparing the detection performances of baseline models generated based on different values of $|T|$. The baseline models are generated based on the dataset in a 60-day period. Since the maximum length of consecutive behaviors observed within the training dataset is $5(C_{max} = 5)$, the value ranges of $|T|$ and $|B|$ are $|T| \in [1,5]$ and $|B| \in [2,5]$. By setting the value of $|T|$ as 3, which is the median of [1,5], the detection accuracies (calculated based on (8)) of baseline models for $|B| \in [2,5]$ are illustrated in Fig. 9. It is shown that the baseline model generated with $|B|$=3 achieves the highest detection accuracy (93.45%) and its detection accuracy increases 0.57% and 12.49% compared with the second best model ($|B|$=4) and the worst model ($|B|$=2). By setting the value of $|B|$ as 3, which has been proved to be the optimal value for our training dataset, the detection accuracies of baseline models for $|T| \in [1,5]$ are illustrated in Fig. 10. It is shown that the baseline model generated with $|T|$=2 achieves the best detection performance and its detection accuracy increases 0.14% and 2.79% compared with the second best model ($|T|$=3) and the worst model ($|T|$=1). The result of this experiment proves that tuning the value of $|B|$ and $|T|$ is necessary for achieving optimal detection performance. The baseline model generated based on the optimal value of $|B|$ and $|T|$ can obtain a better tradeoff between the detection rate and the false positive rate.

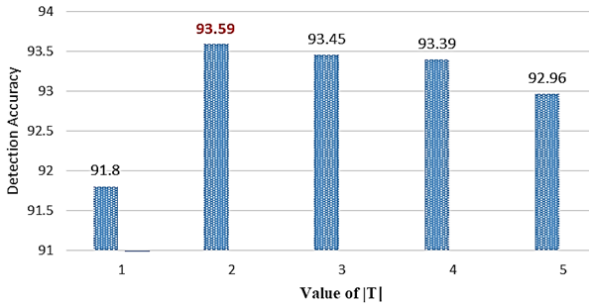

Fig. 10 Detection performance of baseline models generated based on different values of $|T|$

In the fourth experiment, the detection performance of the baseline models for the user's at-home usages and out-of-home usages are tested separately against the same types of attacks. The detection performance curves for the two types of baseline models are depicted in Fig. 11. As shown in this experiment, the baseline models for at-home usages obtained a lower false positive rate and lower detection rate. The reason for its lower false positive rate is that users invoke out-of-home behaviors (remote control behaviors) more casually by sometimes invoking behaviors without following any historical patterns. Those casual normal usages are more likely to be mistakenly detected as anomalous behaviors. The main reason for the lower detection rate is that there are more patterns in the at-home usage database which are similar to the attackers'

usage patterns.

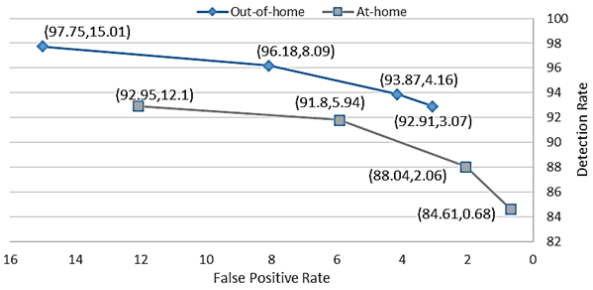

Fig. 11 Detection performance of baseline models generated with N^{out} and N^{in}

In the fifth experiment, we determine the effectiveness of our context modeling approach by comparing the performance of our contextual arrays with the three other types of contextual arrays. We removed the context attributes named Time Slot T from the first type of contextual array, so that the same behavior sequences triggered in a different time slot in a day are considered as identical. We removed the context attributes named Gateway Availability G from the second type of contextual array, so that the current availability of the smart gateway will not be taken into consideration when analyzing SHS behaviors. We removed the context attributes named Physical Location L from the third type of contextual array, and behaviors triggered within the home and out of the home are stored in the same baseline dataset N, instead of being stored separately in N^{out} and N^{in}. As shown in Fig. 12, it turns out that the proposed contextual array achieves better tradeoff than the other three types of arrays. The result in this experiment proves that all three attributes including Time Slot, Gateway Availability, and Physical Location can help the baseline model is describing the normal usage pattern more explicitly. Through this experiment, we validated that the context modeling method being proposed is effective in generating more accurate baseline models.

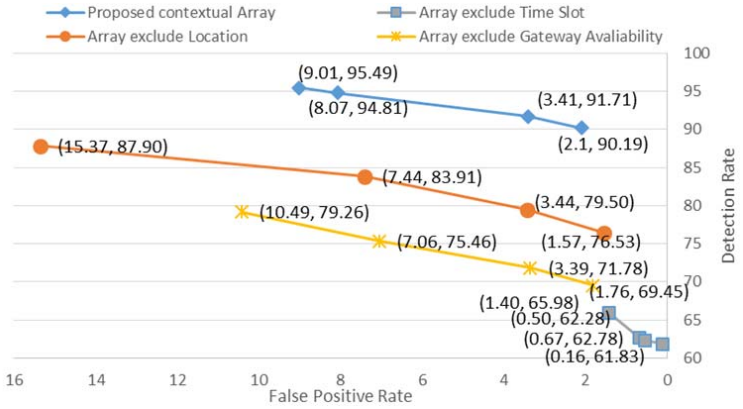

Fig. 12 Detection performance of different types of contextual arrays

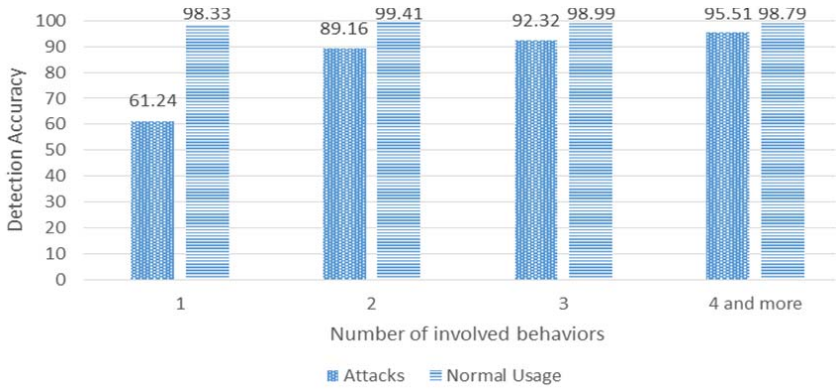

Fig. 13 Detection accuracy for behavior sets with different numbers of involved behaviors

In the sixth experiment, we evaluated the detection accuracy of the proposed approach for asset usages with different numbers of behaviors involved. For instance, a sample behavior sequence for describing a usage with one behavior is {“A1O”, “N/A”, “N/A”}. Moreover, an example of behavior sequences for describing a usage with four involved behaviors can be {“A1O”, “A1C”, “A1O”} and {“A1C”, “A1O”, “A1C”}. Fig. 13 illustrates the detection accuracy of attack scenarios and normal usages with different numbers of involved behaviors. The baseline model used in this experiment

is generated based on the dataset collected in a 60-day period. As shown in the figure, the detection accuracies for normal usages with different numbers of involved behaviors are all above 98%. The detection accuracy of attacks carried by a single behavior is as low as 61%, and the detection accuracy is increased to more than 94% when dealing with attacks with more than two involved behaviors. The result in this experiment indicates that our approach cannot detect single-behavior attacks efficiently, since the correlation of behaviors cannot be reflected if there is only one behavior included in a Behavior Sequence.

VI. Conclusion

In this paper, we have introduced a context-aware anomaly behavior analysis methodology which can detect attacks targeting SHS with low false positive rate. Based on a novel SHCDS, the heterogeneous information acquired from the SHS can be dynamically modeled into contextual arrays which are more precise and accurate. We have shown that the time and location attributes in the contextual array can help the behavior analyzer to achieve better detection performance. Since our behavior analysis is based on one-class learning, which means the baseline model is generated with users' normal behaviors, it is possible for the behavior analyzer to detect diverse types of anomaly behaviors. We have shown that by using the contextual array with Context Sequence of length 3, we achieve a 2.1% false positive rate. Since our behavior analysis is based on finding the correlations among consecutive behaviors, it can achieve more than 94% detection rate for attacks carried by more than two behaviors.

As to the future works, the proposed approach should be improved to achieve detection accuracy for attacks carried by only one behavior. Moreover, our proposed method cannot protect the assets which are operated without depending on any context information. Such independent assets may include vacuum cleaner, voice assistant, etc. Even though the operational state transitions of those assets are independent on context information, the on-off switching of some of these assets may still be related to context information. For instance, the on-off switching of voice assistant may be related to the presence of home occupants and has correlations with the state transition of other home appliance. As the second future work, we will integrate these independent assets into our Smart Home testbed and make security evaluations. Our third future work is to integrate home assets which can perform sophisticated state transitions (e.g. the mode changes of Nest thermostat) so that, the detection performance of the proposed approach on assets with sophisticated state transitions can be evaluated. Our fourth future work is to add more occupants to our Smart Home testbed. In this way, the model sets that can describe usage patterns of multiple users can be generated and the model drift that occurs with a change of the occupant can be evaluated.

This work is proposed based on the outlook that more and more assets and services will be integrated into home systems with remote monitoring and control enabled. Such an outlook is becoming clear since many companies have already built SHS platforms (either open or private) for home asset venders [26]-[28]. This work is a first step in providing fine-grained intrusion detection for the SHS based on comprehensive context information. To implement such intrusion detection functionality on an existing integrated SHS, refined APIs and the privileges for functionality development will be needed.

Acknowledgment

This work is supported by CCF-NSFOCUS KunPeng Research Fund (2018013). We would like to thank the University of Arizona NSF Cloud an Autonomic Computer Center for giving access to their testbeds.

References

[1] Verizon. *Create intelligent, more meaningful business connections*. Retrieved from http://www.verizonenterprise.com/solutions/connected-machines/, 2016.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, Internet of Things for Smart Cities, in IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22-32, Feb. 2014.doi: 10.1109/JIOT.2014.2306328.

[3] J. Pacheco and S. Hariri, IoT Security Framework for Smart Cyber Infrastructures, 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, 2016, pp. 242-247.doi: 10.1109/FAS-W.2016.58.

[4] Jesus Pacheco, Autonomic Cyber Security (ACS): A paradigm shift in cyber security, http://nsfcac.arizona.edu/research/iot-security-framework.html.

[5] Pacheco J, Hariri S. Anomaly behavior analysis for IoT sensors. Trans Emerging Tel Tech. 2018; 29: e3188. https://doi.org/10.1002/ett.3188.

[6] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo and L. Zhou, On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age, in IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 3, pp. 237-248, 1 May-June 2017.doi: 10.1109/TDSC.2016.2577022.

[7] J. Y. Kim, W. Hu, H. Shafagh and S. Jha, "SEDA: Secure Over-The-Air Code Dissemination Protocol for the Internet of Things," in IEEE Transactions on Dependable and Secure Computing. doi: 10.1109/TDSC.2016.2639503.

[8] Migault D, Guggemos T, Killian S, et al. Diet-ESP: IP layer security for IoT. Journal of Computer Security, 2017, 25(2):1-31. DOI: 10.3233/JCS-16857.

[9] P. P. Gaikwad, J. P. Gabhane and S. S. Golait, A survey based on Smart Homes system using Internet-of-Things, 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Chennai, 2015, pp. 0330-0335.

[10] Dan Goodin.: Is your refrigerator really part of a massive spam-sending botnet?, http://arstechnica.com/security/2014/01/is-your-refrigerator-really-part-of-a-massive-spam-sending-botnet/.

[11] P. B. Nassar, Y. Badr, K. Barbar and F. Biennier, Risk management and security in service-based architectures, 2009 International Conference on Advances in Computational Tools for Engineering Applications, Zouk Mosbeh, 2009, pp. 214-218.doi: 10.1109/ACTEA.2009.5227927.

[12] H. Suo, J. Wan, C. Zou, J. Liu, Security in the Internet of Things: A Review, International Conference on Computer Science and Electronics Engineering (ICCSEE), vol. 3.2012.

[13] I. Butun, S. D. Morgera and R. Sankar, A Survey of Intrusion Detection Systems in Wireless Sensor Networks, in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 266-282, First Quarter 2014. doi: 10.1109/SURV.2013.050113.00191.

[14] D. P. Duarte et al., Substation-based self-healing system with advanced features for control and monitoring of distribution systems, 2016 17th International Conference on Harmonics and Quality of Power (ICHQP), Belo Horizonte, 2016, pp. 301-305.

[15] Nobakht, Mehdi, Vijay Sivaraman and Roksana Boreli. "A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow." 2016 11th International Conference on Availability, Reliability and Security (ARES) (2016): 147-156.

[16] Jonathan Roux, Eric Alata, Guillaume Auriol, Vincent Nicomette, Mohamed Kaâniche. Toward an Intrusion Detection Approach for IoT based on Radio Communications Profiling. 13th European ependable

Computing Conference, Sep 2017, Geneva, Switzerland. 4p., 2017. <hal-01561710>.
[17] Gajewski, Mariusz, Jordi Mongay Batalla, George Mastorakis and Constandinos X. Mavromoustakis. "A distributed IDS architecture model for Smart Home systems." Cluster Computing (2017): 1-11.
[18] Abowd G. D., Dey A. K., Brown P. J., Davies N., Smith M., Steggles P. Towards a Better Understanding of Context and Context-Awareness. In: Gellersen HW. (eds) Handheld and Ubiquitous Computing. HUC 1999. Lecture Notes in Computer Science, vol 1707. Springer, Berlin, Heidelberg.1999.
[19] Smart Home Energy. What is a "smart home"? Web page, date: 2013-09-25. (Online). Available: http://smarthomeenergy.co.uk/whatsmart-home.
[20] Xiaoyan Jia, Xiaoyong Li, and Yali Gao. 2017. A Novel Semi-Automatic Vulnerability Detection System for Smart Home. In Proceedings of the International Conference on Big Data and Internet of Thing (BDIOT2017). ACM, New York, NY, USA, 195-199.
[21] C. Lee, L. Zappaterra, Kwanghee Choi and Hyeong-Ah Choi, Securing smart home: Technologies, security challenges, and security requirements, 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, 2014, pp. 67-72. doi: 10.1109/CNS.2014.6997467.
[22] FIPS PUB 199. Standards for Security Categorization of Federal Information and Information Systems. Technical report, National Institute of Standards and Technology. Federal Information Processing Standards Publication.
[23] E. Fernandes, J. Jung and A. Prakash, Security Analysis of Emerging Smart Home Applications, 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 636-654.doi: 10.1109/SP.2016.44.
[24] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, Context Aware Computing for The Internet of Things: A Survey, in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 414-454, First Quarter 2014.
[25] G. Castelli, M. Mamei, A. Rosi, and F. Zambonelli, Extracting high-level information from location data: the w4 diary example, Mob. Netw. Appl. 2009, vol. 14, no. 1, pp. 107–119, (Online). Available: http://dx.doi.org/10.1007/s11036-008-0104-y.
[26] Z. Pan, S. Hariri and Y. Al-Nashif, Anomaly based intrusion detection for Building Automation and Control networks, 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, 2014, pp. 72-77.doi: 10.1109/AICCSA.2014.7073181.
[27] Works with Nest, https://nest.com/works-with-nest/, 2018.
[28] Apple Developer: HomeKit, https://developer.apple.com/homekit/, 2018.
[29] Xiaomi Gateway (Aqara), https://wiki.domoticz.com/wiki/Xiaomi _Gateway_(Aqara), 2018.
[30] uml.org, Unified modeling language (uml), 2012, http://www.uml.org/ (Addressed on: 2012-05-29).
[31] H. Alipour, Y. B. Al-Nashif, P. Satam and S. Hariri, Wireless Anomaly Detection Based on IEEE 802.11 Behavior Analysis, in IEEE Transactions on Information Forensics and Security, vol. 10, no. 10, pp. 2158-2170, Oct. 2015.doi: 10.1109/TIFS.2015.2433898.
[32] P Satam, H Alipour, Y Al-Nashif, S Hariri, Anomaly behavior analysis of DNS protocol., in Internet Serv. Inf. Secur, 2015, (4), 85–97.
[33] H. Alipour, Y. B. Al-Nashif and S. Hariri, IEEE 802.11 anomaly-based behavior analysis, 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA, 2013, pp. 369-373.

Zhiwen Pan received the B.S. degree from the Purdue University Calumet, in 2012, and the M.S. and Ph.D. degrees from the University of Arizona, in 2014 and 2017. He is currently an Assistant Research Fellow in Research Center for Ubiquitous Computing System, Institute of Computing Technology, Chinese Academy of Science. His current research focuses on Outlier Detection, Internet of Things, and Context Computing.

Jesus Pacheco. PhD. from the Electrical and Computer Engineering department of The University of Arizona. He is currently working as full professor at the University of Sonora in Sonora, Mexico. His research interest includes cyber security for critical infrastructures and cyber-physical systems.
Salim Hariri received the M.Sc. degree from The Ohio State University, in 1982, and the Ph.D. degree in computer engineering from the University of Southern California, in 1986. He is currently a Professor with the Department of Electrical and Computer Engineering, The University of Ari-zona, and the Director of the NSF Center for Cloud and Autonomic Computing. His current re-search focuses on autonomic computing, cyber-security, cyber resilience, secure critical infra-structures, and cloud security.

Yiqiang Chen received the BSc and MS degrees from the University of Xiangtan, Xiangtan, China in 1996 and 1999, respectively, and the PhD degree from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2002. In 2004, he was a Visiting Scholar Researcher with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Hong Kong. He is currently a Professor and Director with the Pervasive Computing Research Center, ICT, CAS. His research interests include artificial intelligence, pervasive computing and human-computer interface.

Bozhi Liu is a Ph.D. student in the Department of Electrical and Computer Engineering at the University of Arizona. He received his B.S. degree in Engineering Mechanics from Shandong University, Jinan, China in 2013 and his M.S. degree in Computer Engineering from New York University in 2015. From 2012 to 2013, he was a researcher in the Micro-computing Laboratory in GPU design. From 2014 to 2015, he worked as a Graduate Research Assistant on hardware security and he joined the Embedded Systems Laboratory in 2015. His research interests include hardware security, high performance computing with GPU and FPGAs, IoT systems, and machine learning with embedded systems including voice synthesizing and object detection.