A privacy-aware federated graph learning for interaction threat tracking in IoT

Presenter: Guangjing Wang
Computer Science, Michigan State University

Supported by MSU Cloud Computing Fellowship

Smart Home System

IoT environment

IoT Devices

eclipse marketplace

Eclipse SmartHome > IoT (52)

IoT app market

IFTTT ifthisthenthat

Microsoft Flow

zapier

Trigger-action app market

https://www.ndss-symposium.org/ndss-paper/iotguard-dynamic-enforcement-of-security-and-safety-policy-in-commodity-iot/

Traditional Threats

ANDY GREENBERG SECURITY 07.21.15 6:00 AM

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

Their code is an automaker's nightmare: software that lets hackers send commands through the Jeep's entertainment system to its dashboard functions, steering, brakes, and transmission, all from a laptop that may be across the country.

SMART HOME

Have a smart lock? Yeah, it can probably be hacked

While wandering in his neighborhood, he noticed a lot of Bluetooth locks popping up and decided to do some sniffing of those "security" gadgets (read: capturing packets being sent between devices). "I discovered plain-text passwords being sent that anybody could read.

Mirai botnet adds three new attacks to target IoT devices

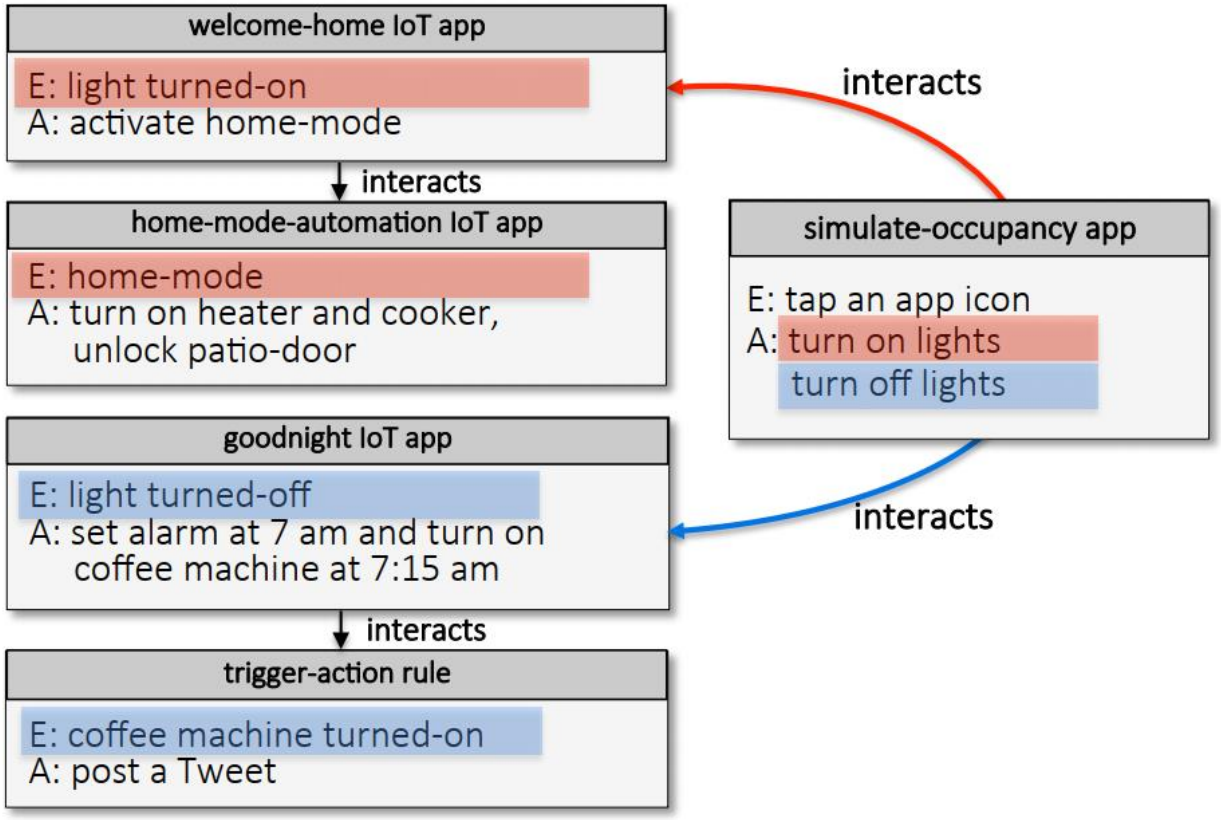
This new version of the botnet uses exploits instead of brute force attacks to gain control of unpatched devices.

By Danny Palmer | May 18, 2018 -- 13:29 GMT (14:29 BST) | Topic: Security

Software bugs, user error, poor configuration, or faulty design

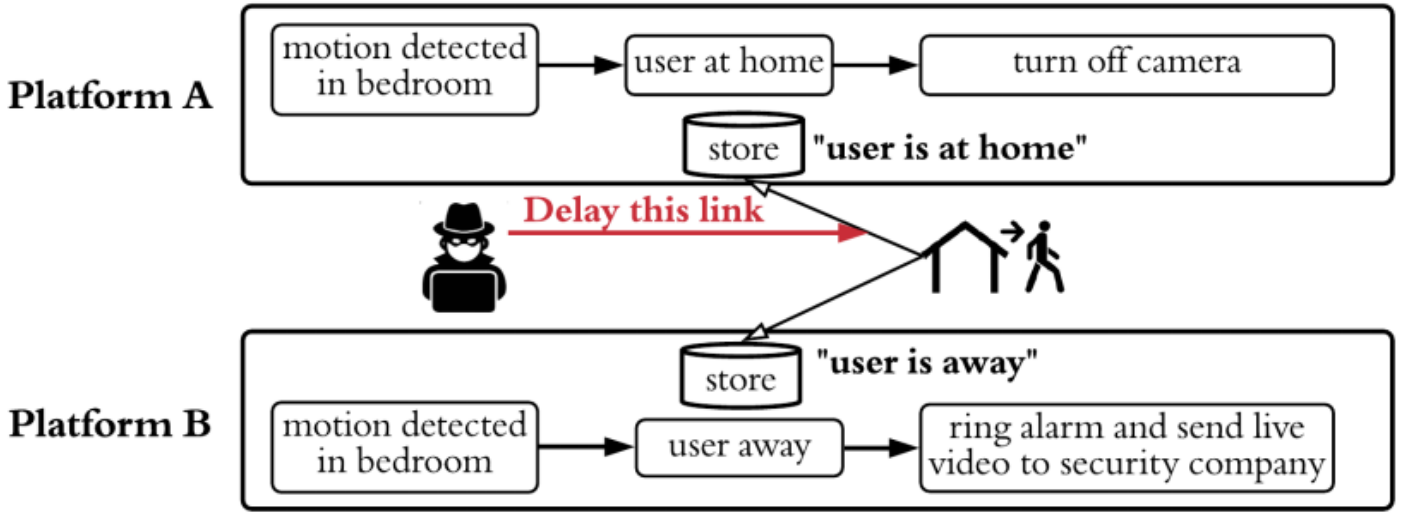
https://www.ndss-symposium.org/ndss-paper/iotguard-dynamic-enforcement-of-security-and-safety-policy-in-commodity-iot/

Device Interactions in IoT



https://www.ndss-symposium.org/ndss-paper/iotguard-dynamic-enforcement-of-security-and-safety-policy-in-commodity-iot/

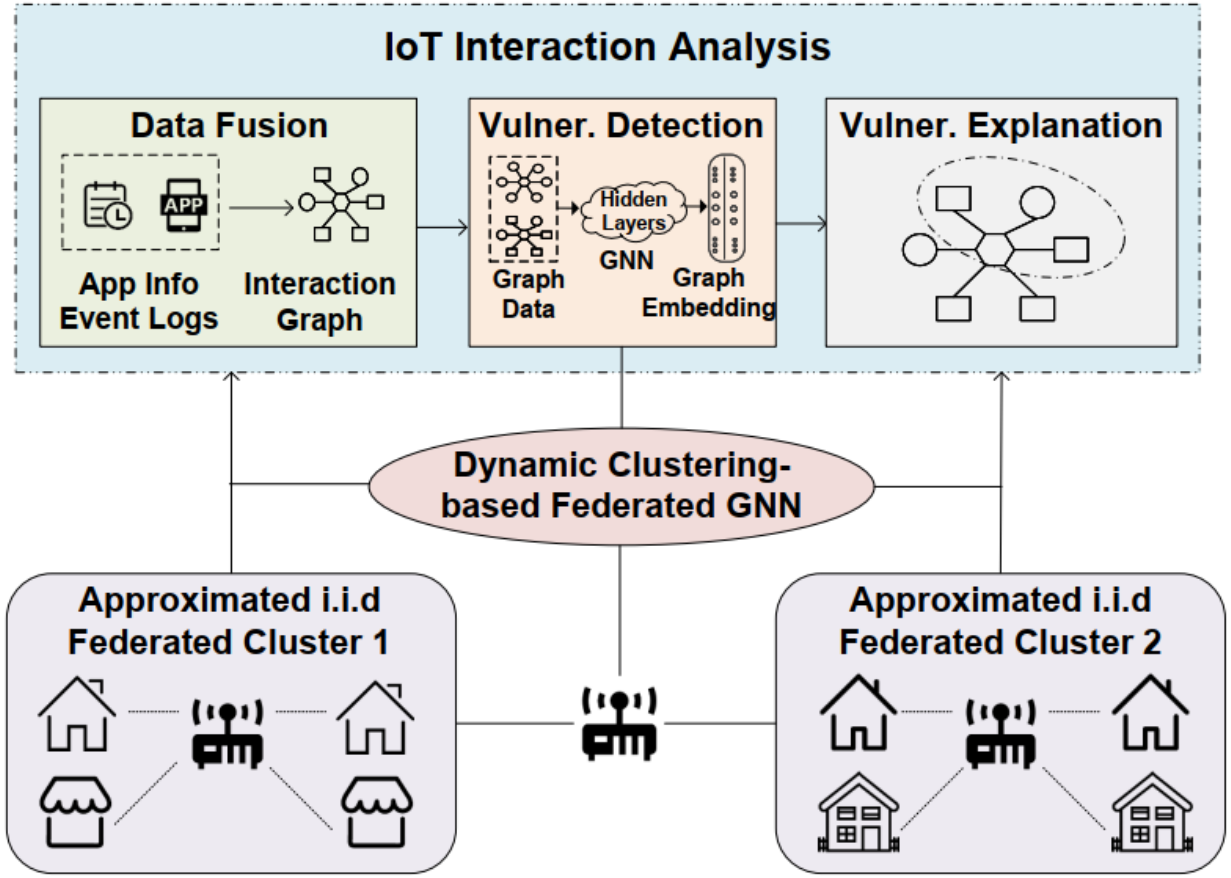
New Type: Interaction Threats



- An app triggers an event that is subscribed by another app
- Multiple apps change the same attribute of some device
- Apps that subscribe to the same event change the attribute in conflicting ways

Haotian Chi, Chenglong Fu, Qiang Zeng, and Xiaojiang Du. 2022. Delay Wreaks Havoc on Your Smart Home: Delay-based: Automation Interference Attacks. In 2022 IEEE Symposium on Security and Privacy (SP)

FexIoT: Federated Interaction Threat Detection

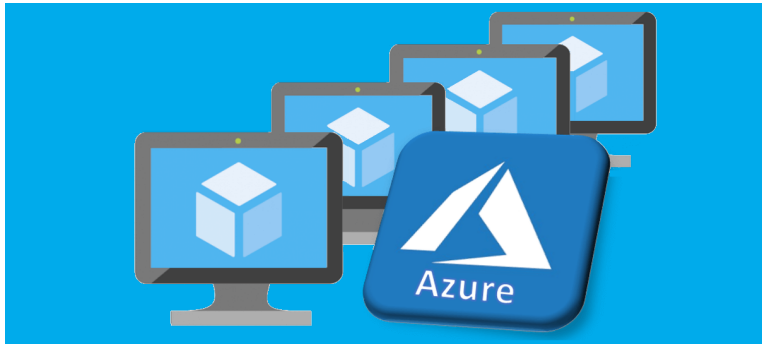


FexIoT: Federated and Explainable Graph Learning for IoT Interaction Analysis

Could Computing Support

Virtual machines:

- One as federated learning server
- Ten as federated learning Clients

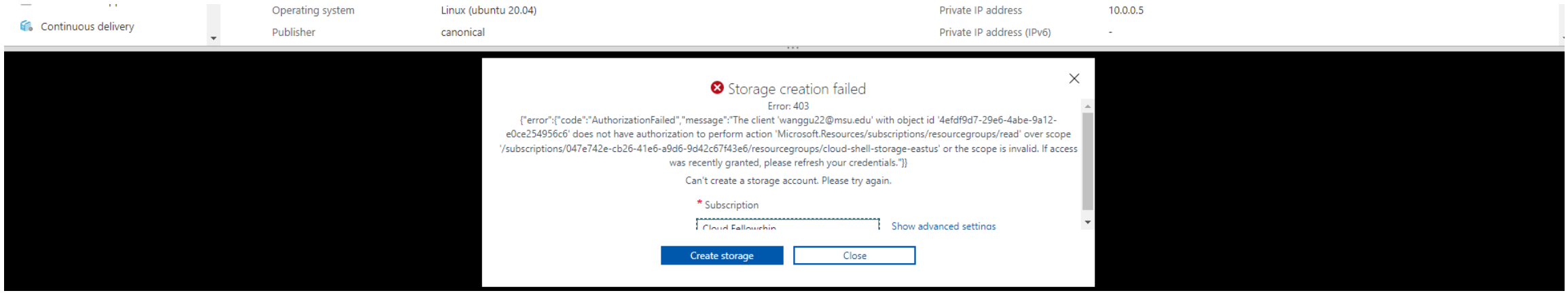


Resource group (move)	: cf21group_wanggu22
Status	: Running
Location	: North Central US
Subscription (move)	: Cloud Fellowship
Subscription ID	: 047e742e-cb26-41e6-a9d6-9d42c67f43e6
Tags (edit)	: fl-server : fl-server

Advantages:

- Stable Communication
- System Homogeneity
- Computing Resource
- Usage Time Freedom
- Flexible Device Simulation

Permission Problem




```
guangjing@seit:~$ az group create --name CloudFellowship --location eastus
(AuthorizationFailed) The client 'wanggu22@msu.edu' with object id '4efdf9d7-29e6-4abe-9a12-e0ce254956c6' does not have authorization to perform action 'Microsoft.Resources/subscriptions/resourcegroups/write' over scope '/subscriptions/047e742e-cb26-41e6-a9d6-9d42c67f43e6/resourcegroups/CloudFellowship' or the scope is invalid. If access was recently granted, please refresh your credentials.
Code: AuthorizationFailed
Message: The client 'wanggu22@msu.edu' with object id '4efdf9d7-29e6-4abe-9a12-e0ce254956c6' does not have authorization to perform action 'Microsoft.Resources/subscriptions/resourcegroups/write' over scope '/subscriptions/047e742e-cb26-41e6-a9d6-9d42c67f43e6/resourcegroups/CloudFellowship' or the scope is invalid. If access was recently granted, please refresh your credentials.
guangjing@seit:~$ az vmss create -n MyVmss -g cf21group_wanggu22 --instance-count 5 --image UbuntuLTS --data-disk-sizes-gb 2 --os-disk-size-gb 10An RSA key file or key value must be supplied to SSH Key Value. You can use --generate-ssh-keys to let CLI generate one for you
guangjing@seit:~$ az vmss create -n MyVmss -g cf21group_wanggu22 --instance-count 5 --image UbuntuLTS --data-disk-sizes-gb 2 --os-disk-size-gb 10 --generate-ssh-keys
SSH key files '/home/guangjing/.ssh/id_rsa' and '/home/guangjing/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If using machines without permanent storage, back up your keys to a safe location.
{"status":"Failed","error":{"code":"DeploymentFailed","message":"At least one resource deployment operation failed. Please list deployment operations for details. Please see https://aka.ms/DeployOperations for usage details.","details":[{"code":"Conflict","message":"{\r\n  \"error\": {\r\n    \"code\": \"OperationNotAllowed\",\r\n    \"message\": \"The specified disk size 10 GB is smaller than the size of the corresponding disk in the VM image: 30 GB. This is not allowed. Please choose equal or greater size or do not specify an explicit size.\",\r\n    \"target\": \"osDisk.diskSizeGB\"\r\n  }\r\n}"}]}}
guangjing@seit:~$ az vmss create -n MyVmss -g cf21group_wanggu22 --instance-count 2 --image UbuntuLTS --data-disk-sizes-gb 2 --os-disk-size-gb 30 --generate-ssh-keys
{
```

- Use our own resource group: cf21group_netid
- Disk size should be at least the size of disk in the VM image

Outcome

1. Design the dynamic clustering-based federated graph learning algorithm for interaction threat analysis

2. Plan to submit a paper to CIKM Conference

FexIoT: Federated and Explainable Graph Learning for IoT Interaction Analysis

Anonymous Author(s)
Submission Id: 4089

ABSTRACT

Heterogeneous smart home platforms provide users with great convenience. Yet, the increasingly complex interdependent behaviors across different devices and platforms may yield unexpected interaction behaviors, which could lead to serious security and privacy issues. Prior studies have attempted to discover interaction vulnerabilities via code instrumentation and event log analysis. However, the closed-source nature of smart home apps limits the effectiveness of code analysis, while the interaction logic cannot be fully mined from event logs. In light of this, we propose a novel system *FexIoT* for IoT interaction analysis. Facing the closed-source platforms, *FexIoT* builds an interaction graph to model devices' interactions based on the descriptions of apps and real-time event logs. The interaction graph representation is encoded by graph neural networks (GNNs). To avoid exposing users' private smart home usage data and cope with the statistical heterogeneity of graph data, we design a dynamic clustering-based federated GNN model to collaboratively learn intrinsic clustering relationships among users' model weights. In addition, we propose to apply the Monte Carlo beam search with SHAP values to search and measure the risk of subgraphs. Thus, we can interpret the GNN prediction result. We evaluate our prototype on 5 platforms with a real-world testbed. The results show that *FexIoT* outperforms state-of-the-art methods, which achieves 97.18% average accuracy for interaction vulnerability detection of clients in federated learning. Besides, it achieves significantly improved IoT interaction vulnerability analysis.

1 INTRODUCTION

The Internet of Things (IoT) has developed significantly in recent years. In particular, a growing number of smart devices are deployed in modern homes to achieve home automation. Although home automation brings great benefits to our daily lives, the unexpected interactions among different devices, users, and the environment could lead to *interaction vulnerabilities*, which result in severe security and privacy risks. Specifically, many factors such as software flaws [21, 55], user mistakes [27, 52, 68], and various attacks [1, 78, 85] can cause vulnerable interaction during the operation of smart home systems. Moreover, the smart devices can interact with each other via user channels, device channels, and smart app channels [22]. These complex correlations could expose severe vulnerabilities that can be exploited by attackers.

Most existing work relies on source code analysis or code instrumentation [2, 8, 13, 17, 18, 22, 67] to detect vulnerable interactions. However, many platforms such as IFTTT [30], Google Assistant [24], and Amazon Alexa [3] are closed-source, which impedes the existing methods in proprietary platforms. The data mining-based methods [14, 19, 25, 31, 38, 63, 79, 89] for threat detection usually take as input event logs to profile the behaviors of systems. However, they cannot fully consider the interaction correlations between different events, which need to be mined from apps and devices. The limitations are threefold. First, the interactions among smart apps are hard to be mined accurately considering the complex logic between two rules. For example, *"Close the water valve when water leak is detected in kitchen"* and *"If smoke is detected, open the water valve and trigger alarm beeping"* will conflict with each other if the smoke is detected and then water leak is detected in the kitchen. As a result, a vulnerable interaction "the water valve fails to turn on" may arise. Second, the privacy concerns of smart home data prevent sharing data with a third party, while dataset from a single house can hardly train a GNN model with high generalization ability. Third, the mining results are hard to interpret. When a threat alarming is generated, users can hardly know which part of device interactions causing the problem.

To fill the gap, we propose *FexIoT*, a Federated and explainable GNN-based approach for IoT interaction analysis in closed-source smart home systems. The smart home rules follow the general "trigger-action" paradigm, and the interactions among rules can constitute an interaction graph. The nodes are rules from different platforms, and edges are "trigger-action" correlations among different rules. The node features are the semantic-aware word or sentence embeddings. The interaction graphs that contain abnormal behaviors have different graph patterns compared with the benign interaction graphs. Meanwhile, as the event logs belong to different smart home platforms and households, by leveraging federated learning, we can collaboratively develop a more generalized model while retaining users' data locally. Moreover, within the graph, we can search and extract a special execution flow path that is related to abnormal behaviors, so that we can explain the model's predictions and identify the causes of vulnerable interactions.

However, there are three core technical challenges for interaction vulnerability analysis across the closed-source platforms.

(i) How to extract and represent real-time interaction causality information from heterogeneous closed-source platforms? Existing data mining methods take as inputs event logs [19, 25, 42, 65, 79, 89], which ignore the semantic information such as automation logic, and device relations. Moreover, event logs from different apps have different granularity, and most event logs are coarse-grained as they only contain a timestamp, object, and attribute status. To address the challenge, instead of purely using the sequence data, we fuse information from app descriptions and event logs into interaction graphs using natural language processing (NLP) techniques. App descriptions contain essential trigger-action causality information, while event logs contain real-time device states, events, and control commands. Thus, we can encode interaction graphs with GNN models for vulnerability analysis in a unified format.

(ii) How to build the vulnerability detection model with significant heterogeneity in datasets while preserving users' privacy? On

CIKM'22, October 17-22, 2022, Atlanta, USA Anon. Submission Id: 4089

calculate dynamic time warping [59] distance of word embeddings as the similarity score. (ii) We calculate the causal relation features. Specifically, we check whether two sentences have synonym, hypernym, meronym, or holonym relations. We use one-hot vectors to represent these causal relation features. (iii) We compute sentence-level features by using sentence encoder [9] to generate sentence embeddings, which express word sequences in sentence-level embedding space. The extracted features are used for "action-trigger" correlation discovery.

Interaction Graph Construction. The app descriptions contain the "trigger-action" information. To obtain the "action-trigger" correlations, we model the "action-trigger" correlation discovery as a binary classification problem. If two sentences have an "action-trigger" connection, we labeled it as true, otherwise, it is false. We use the manually labeled IFTTT interaction flow dataset from [72] and our new crawled IFTTT and SmartThings interaction flow dataset as the training dataset. There are more non-correlation pairs than correlation pairs in the dataset. To alleviate the class imbalance problem, we use the class weights in classifiers' loss function. Finally, we apply the well-trained model to classify the unlabeled sentence pairs. We also manually check the classification results to guarantee the correctness of correlations. Finally, we chain the "trigger-action" and "action-trigger" correlations into interaction graphs. In this way, we build interaction graph datasets from application descriptions.

Then, with event logs, we can construct real-time interaction graphs. Devices have different types of "trigger-action" interactions, which can compose different interaction graphs. Thus, we need to match the current device status in event logs to the possible interaction graphs. To achieve that, we extract the device name, status, and event time from the event logs. With the "trigger-action" logic in existing interaction graphs, we can match device type and current status with explicit correlations in interaction graphs, while the event time helps determine the event sequences. In this way, we can integrate event logs and app descriptions into a fine-grained real-time interaction graph.

Algorithm 1: Dynamic clustering-based federated GNN

Input: Graph dataset of each client G_c, ϵ_1, ϵ_2, L layers of GNN model, clients C, global update round T

```
for c in C do
    for l ∈ L do
        Local training in each client c;
        Send W_c^l to the server;
        Receive W_s from the server;
        W_c = [W_s, W_c[len(W_s) : end]];
    end
end
for t < T do
    Receive W_c^l from each client c;
    for each cluster do
        while ε1 > || Σ_{i∈[n]} |G_i|/|G| ΔW_i^l || &
          ε2 < max(||ΔW_c^l||) do
            a_{i,j} ← CosineSimilarity(W_i, W_j) for i, j ∈ C
            cluster_1, cluster_2 ← ClusteringModel(a_{i,j})
            θ_c^* ← Agg(θ^*, c_1) for c ∈ cluster_1
            θ_c^* ← Agg(θ^*, c_2) for c ∈ cluster_2
        end
        else
            θ_c^* ← Agg(θ_S^*, c) for c ∈ C
        end
    end
end
```

4.2 Federated GNN for Vulnerability Detection

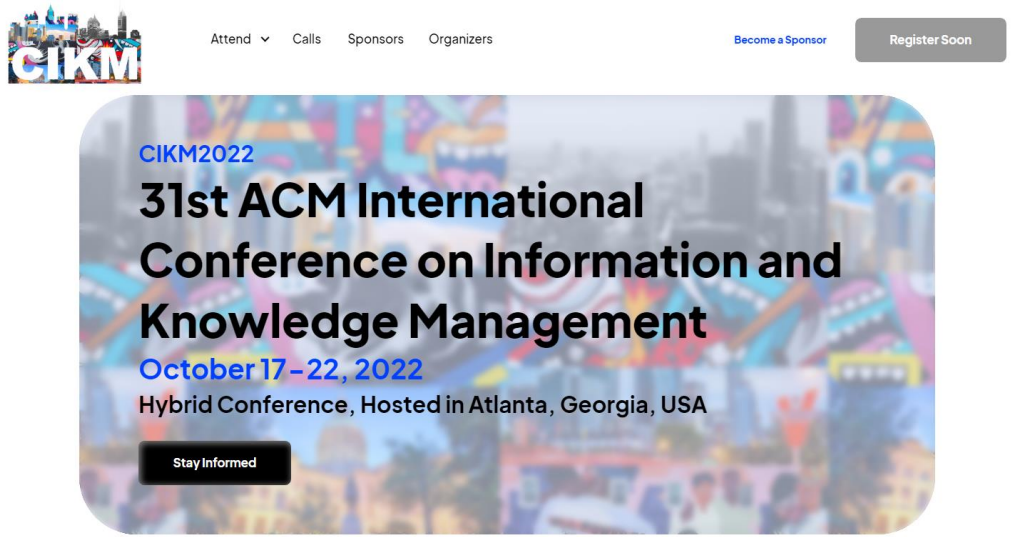
To avoid the leakage of raw data, we propose a clustering-based federated GNN algorithm to collaboratively train the vulnerable interaction detection model while keeping users' data locally.

FL can be regarded as transferring knowledge among different users, but there will be a domain shift when knowledge transfers across *non-i.i.d.* datasets [15]. Different households deploy heterogeneous devices and different users have their own usage habitats, which cause graph heterogeneity. The heterogeneous graph data will cause negative transfer among users and decrease the model performance. The *non-i.i.d.* graph datasets from different households will cause biased stochastic gradients, which will impede the convergence of FL models. Therefore, directly using FL is impractical, given the heterogeneous data distributions in smart homes.

One key observation is that despite the heterogeneity, there exist similar data distributions in smart homes because of the limited types of smart devices and functionalities. We assume that there exist several clusters of households, where the graph datasets from every cluster satisfy the *i.i.d.* property following [60]. We argue that the assumption is reasonable since even though there are heterogeneous smart devices, from the semantic perspective, the functionalities of devices are indeed limited. There will be some graph datasets that share common feature information, which could be grouped into several clusters. As a result, a new challenge ensues on how to effectively aggregate clients into different clusters.

The generated interaction graphs can be regarded as data samples for graph classification. Each household dynamically generates interaction graphs with different device status. Consider the clustering-based FL setting with a central server and a set of n clients $\{C_1, C_2, \cdots, C_n\}$, which can be dynamically clustered in to different cluster set $\{S_1, S_2, \cdots\}$. Each client C_i have a set of interaction graphs $G = \{G_1, G_2, \cdots\}$, and conduct the graph classification $y = h_k^*(G_i)$, where h_k^* is the optimal graph classification model for cluster set S_k. The graph feature information can be reflected by the model parameters and their gradients [60]. Existing federated graph classification over *non-i.i.d.* graphs [76] is coarse-grained since they only consider the similarity of parameters of a whole model. However, from the bottom up, the degree of similarity among deep models decreases [45, 50, 82]. Therefore, to learn the fine-grained clustering structure, we design the bottom-up layer-wise dynamic clustering algorithm to obtain the similarity among weights of clients as shown in Algorithm 1.

Specifically, suppose n is the number of clients in FL training, each client c performs local GNN training, and send the server n local models W_c^l (line 3-6 in Algorithm 1). The dynamic clustering



Acknowledgement

Institute for Cyber-Enabled Research and IT Services Data Management and Analytics, Michigan State University

MSU Cloud Fellowship

Home

Contact

About

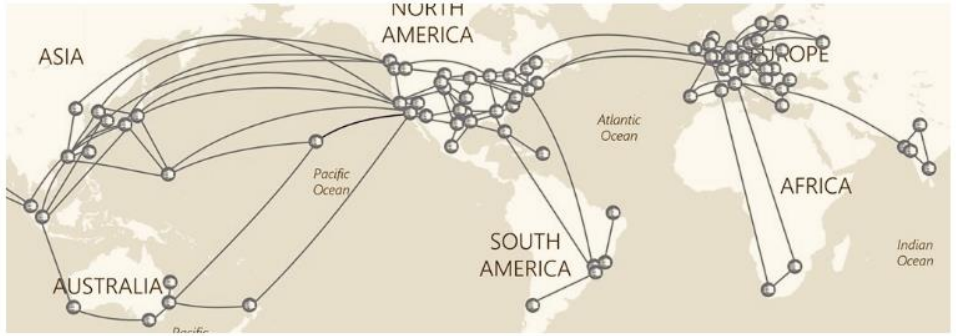
Sessions

1. Introduction
2. How does the cloud work?
3. Cloud Storage
4. Moving Data
5. Big Data and the cloud
6. Cloud Data Systems
7. Serverless, Containers, and FaaS

References

Cloud Glossary

Contacting Us



If you are a cloud fellowship participant this year (or past participant!), please contact the instructors Pat Bills or Mahmoud Parvizi with any issues or questions related to the material or activities.

ICER

SEIT