GOVERNMENT POLYTECHNIC COLLEGE
CHELAKKARA, THONOORKARA PIN: 680586

SEMINAR REPORT
ON
WIFI CALLING

SUBMITTED BY
IRSHAD
Reg.no : 20150945

DEPARTMENT OF
COMPUTER HARDWARE ENGINEERING
2022-2023

GOVERNMENT POLYTECHNIC COLLEGE
CHELAKKARA
DEPARTMENT OF
COMPUTER HARDWARE ENGINEERING

This is to certify that the seminar report titled “WIFI CALLING” was presented by IRSHAD, Reg.no: 20150945 of the final year Computer Hardware Engineering in partial fulfillment of the requirements for the award of Diploma in Computer Hardware Engineering under the Directorate Of Technical Education, Kerala State During the year 2022-2023.

Staff in charge **Head of section**

Internal examiner **External examiner**

ACKNOWLEDGEMENT

It is a pleasure to recollect the faces that passes through the way of completing my effort successfully. I am not sure, if these words are enough to express my liability to pay my thanks.

I would like to thank **Dr. Ahamed Seyd P T**, principal of our polytechnic for providing a pleasant atmosphere to complete my seminar.

I express thanks to **Mrs.Yamini KP**, Head of Department who gave permission to do the seminar.

At last but not the least, I would like to thank my seminar guide **Mrs.Binitha.K.S** and finally a remembrance to the **God Almighty**, without whose blessing, the thoughts about the seminar would not go fourth.

With pleasure,

IRSHAD

ABSTRACT

Many enterprise campuses have poor signal coverage indoors from one or more mobile operators, and thus are increasingly embracing carrier Wi-Fi calling services, allowing their users to make and receive mobile phone calls over the enterprise Wi-Fi connection. Mobile carriers employ IPSec tunnels to secure user calls and messages that traverse untrusted enterprise networks and possibly the public Internet. These encrypted connections from user handsets are seen as potential security threats in enterprise networks. In this paper, we develop a machine learning-based system for monitoring encrypted traffic of IPSec tunnels on the network to distinguish Wi-Fi calling traffic from anomalies. Our contributions are as follows: (1) We analyse traffic traces consisting of carrier Wi-Fi calls made over four mobile networks to highlight network behavioural characteristics of this enterprise application. We develop a set of models using one-class and multi-class classification algorithms to determine if Wi-Fi calling application is present on the IPSec tunnel (if so, to classify its state), otherwise generate a notification to block the non-Wi-Fi calling flow, and (2) We evaluate the efficacy of our system in detecting real calls and their states (initiation, heartbeat, and actual call) as well as raising true alarms in case of anomalous traffic.

CONTENTS

CHAPTER 1

INTRODUCTION

Carrier Wi-Fi calling, one of the voice-over-Wi-Fi applications, was initially launched by a handful of mobile operators in the US and Europe and is being offered by a growing number of Tier-1 operators around the world. According to Cisco, it will take 53% of mobile IP voice service usage by 2020, a more than three-fold increase from 2015. This technology was primarily designed for consumers to compensate poor cellular coverage inside their homes but also benefits enterprises with wireless-shielded buildings. It is a standard protocol that allows users to make regular mobile phone calls (and also send texts and other media) over Wi-Fi networks with no need for an additional application – just from phones native dialer. For mobile network operators, offload of mobile voice traffic onto Wi-Fi access networks is beneficial, since they can offer high-quality voice calls and SMS at a lower price, thereby combating over-the-top applications like Skype and WhatsApp.

Mobile carriers do not trust third-party-owned Wi-Fi access networks (and the Internet) to carry important voice calls, and therefore they establish secure tunnels (IPSec) between user devices and their core packet gateway to protect the Wi-Fi-calling traffic. However, the use of IPSec tunnels creates a security concern for enterprises who want to allow the Wi-Fi- calling service within their organization, since their firewall policy needs to unblock IPSec flows with no ability to detect the application carrying the encrypted traffic.

This paper describes our solution for modelling network behaviour of Wi-Fi calling traffic that enables enterprise net-work operators to automatically monitor encrypted UDP flows and detect anomalies in real-time. We begin by analysing real traffic traces of Wi-Fi calls from four mobile network operators and highlight the key characteristics of Wi-Fi calling flows on the network. We then train one-class classifiers as well as a multi-class classifier that are collectively able to distinguish Wi-Fi-calling flows from anomalous ones. Finally, we prototype our system and validate it with real traffic

CHAPTER 2

RELATED WORK

Mobile data offload onto Wi-Fi access networks is well understood and practiced by industry and academia. However, mobile voice offload has become prevalent only in recent years when phone manufacturers started to natively support Wi-Fi calling. Analysis of network traffic to identify applications has been an active research area for decades. Specifically, voice over IPSec has been studied in where the authors quantified the performance of the application in terms of bandwidth usage and transmission delay. Similar to our work, authors of aim to detect VoIP packets over IPSec tunnels to forward them with the highest priority (i.e., for quality of service). Their proposed method is quite simple and only checks the packet size within a fixed range. We, instead, compute 8 attributes from time-series profile of each IPSec flow and develop four ML models to capture the normal behaviour of Wi-Fi calling application

In terms of security, work in surveys threats and vulnerabilities of VoIP technology. However, security of Wi-Fi calling application has not been well studied until recently. Authors of demonstrate various attacks on end-user devices where IPSec keys can be extracted from SIM cards. Work in conducts a comprehensive study on vulnerabilities of the Wi-Fi calling service over major mobile operators in the US, and highlight several privacy risks (e.g., inferring user identity, call statistics, and device information) for users of this application. Our work primarily aims to automatically ensure that only Wi-Fi calling traffic is exchanged with trusted mobile operators over IPSec tunnels from enterprise networks – anomalous flows are blocked in real-time.

CHAPTER 3

MODELLING WIFI CALLING APPLICATION

A. *Network Behaviour of Wi-Fi Calling*

A Wi-Fi calling session starts with IPSec tunnel establishment, followed by initiation phase wherein it exchanges device details like phone number, and then stays in a heartbeat keep-alive) phase until a call is made. Let us now look into a real trace of a real Wi-Fi calling session and its various phases in Fig. 1. The Wi-Fi call was made over a major US-based carrier network using an LG Android device – we observed similar network profiles on iOS devices across various carrier networks.

IPSec Tunnel Establishment. When a mobile device (with Wi-Fi calling feature enabled) connects to a Wi-Fi network, it first fetches a server IP corresponding to carrier's Wi-Fi calling endpoint by sending a DNS query. The device next establishes a secure IPSec tunnel with the server using the IKE and ISAKMP protocols and encapsulates the data in the tunnel using ESP protocol over UDP. Note that we cannot solely rely on DNS information to ensure that the application is Wi-Fi calling or not. For example, in case of a DNS spoofing attack, the man-in-the-middle attacker can reply to the legitimate DNS query, causing the device to establish the tunnel to a malicious server. Therefore, it is needed to detect the application using the profile of traffic exchanged over the tunnel which is a non-trivial exercise.

Initiation. Following tunnel establishment, the Wi-Fi Call-ing application exchanges device specific information (e.g., phone number) with the carrier's server. This typically takes about 7 to 10 seconds and results in the first peak highlighted in red (Fig. 1). This phase consists of ESP packets (typically over 1000 bytes each) transferred at a rate of 200-400 Kbps.

Keep-Alive. Following the initiation, the application enters into a Keep-Alive phase wherein one packet (of 60 bytes) is sent every 20 seconds from the server to the device only for keeping the NAT mappings alive (as described in detail in). Also, every couple of minutes, a pair of ISAKMP request/response (Informational type) packets of size 122 bytes are exchanged. The application continues to be in this phase, until a call is made.

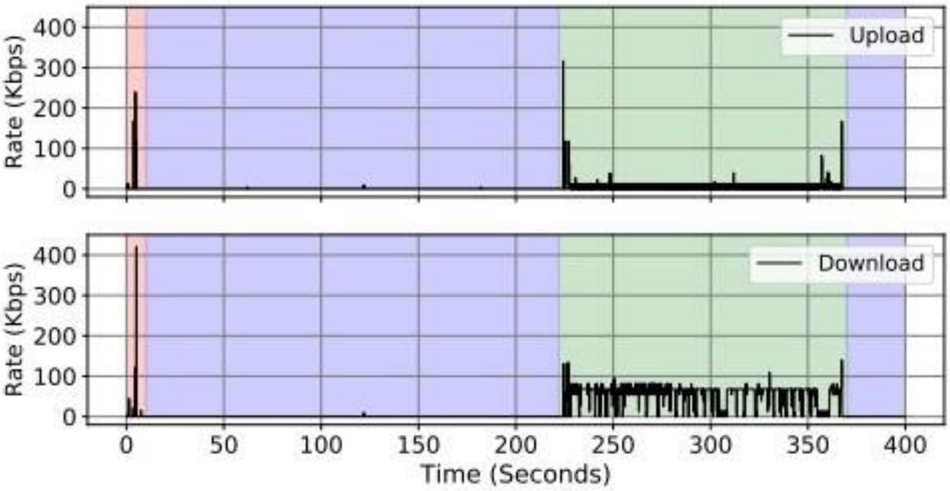


Fig. 2.1 Network profile of Wi-Fi calling traffic.

Call. When a user makes a call, the device send/receives call data using the IPSec tunnel established over Wi-Fi. It can be seen in Fig. 1 that data packets flow bidirectionally. The amount of data transferred depends on the conversation, i.e., in this example, the user who made the call does not seem to talk much and hence less upload traffic. Further, in comparison to the initiation, the traffic rates is lower and seems to have an upper cap of around 100 Kbps. The packets exchanged are typically 174 bytes. Note that upon termination of the call, the application switches back to the Keep-Alive phase.

Although a representative Wi-Fi calling session occurs this way, we have also observed certain minor differences. For example, the IPSec connection might get terminated and re-established periodically (commonly observed in iOS devices).

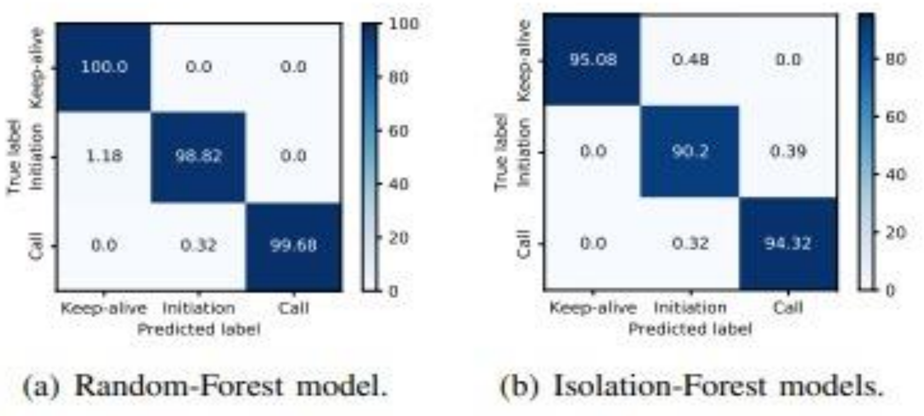
We think it is probably because the phone disconnects the session to save battery during idle periods (i.e., keep-alive phase). We note that only one IPSec connection is active at any point in time. Further, we observed that sometimes during tunnel re-establishment the server IP might change (but it can be captured by an updated DNS response) due to dynamics of cloud-based services. Additionally, we have observed that the UDP port used on the client is selected at random in Android devices, but is set to 4500 (identical to the server port) in iOS devices. Nonetheless, across the 4 major providers and the two operating systems, we have observed the traffic profile of each phase to be almost identical and thus helping us build a general model to detect and monitor the Wi-Fi calling.

B. *Phase Classification and Anomaly Detection*

We now explain our method to detect Wi-Fi calling sessions using the network activity data. As explained in previous section a Wi-Fi calling flow can be in one of three phases: Initiation, Keep-Alive or Call. Thus, given an IPSec flow established for Wi-Fi calling, we need to detect and monitor these phases in real-time. We also need to identify IPSec flows which are not established for Wi-Fi calling, considered as anomalies in our use-case. To perform these tasks, we first break the traffic profile (i.e., time-series signal) into fixed-length windows and extract appropriate features from its activity. These features are used to train models which perform two tasks: (a) classify phases of a legitimate Wi-Fi calling session, and (b) detect anomalous behaviour by a non-Wi-Fi calling session.

Feature Extraction. We have observed that Wi-Fi calling application exhibits certain characteristics: (a) transfers content at rate less than 500 Kbps, (b) is generally idle (i.e., mostly in Keep-Alive phase), and (c) has a distinct pattern of packet sizes. Using these observations, we extracted a set of attributes for a 10-second window of each IPSec flow. We have chosen 10 seconds for our window size since the initiation phase typically takes 7-10 seconds to finish and to classify this phase, we need a minimum of 10-second worth of data. Each 10-second window consists of byte counts and packet counts computed every 100ms, a total of 100 data points.

For each window, we compute the following set of attributes in both upload and download directions: (1) average packet size for highlighting the initiation phase; (2) zero fraction (fraction of time no data is exchanged) for highlighting the Keep-Alive phase, (3) average, and (4) max transfer rate for highlighting the call phase. These attributes help us classify phases and detect anomalies as explained below.



(a) Random-Forest model.

(b) Isolation-Forest models.

Fig. 2.2 Performance of: (a) Random-Forest, and (b) Isolation-Forest, models.

Training Dataset. We collected 20 PCAP traces of Wi-Fi calling sessions from Cisco labs. We then built a dataset consisting of a total of 4,162 labelled instances, each corresponding to 10-second worth of traffic trace for phases Initiation (255 instances), Keep-Alive (3,574 instances), and Call (317 instances). Unsurprisingly, most of instances are keep-alive since the application tends to spend more time in this phase. Each labelled contains 8 attributes mentioned above.

Multi-Class Classification. We trained a Random Forest classifier (decision tree-based learning) on our dataset using the sickie-learn library in Python – this model would generate a pair of outputs: a label (Initiation, Keep-Alive, or Call) and a confidence-level. We used 80% of the data to train our model and the remaining 20% for testing it. We have achieved an accuracy of 99.7% in classifying phases. Fig. 1.2(a) shows the confusion matrix of our Random-Forest model. We can see that this model performs well in learning patterns to distinguish among the expected phases. We also need more precise and possibly sensitive models (one-class classifier is explained next) for individual phases which together with the Random-Forest model enable us to detect anomalous flows**.**

One-Class Classification. We built three models using Isolation Forest algorithm, each is specialized in expected behaviour of one phase in a Wi-Fi calling flow – each model would generate a binary output (positive if expected profile is detected in the instance, otherwise negative). Consequently, an anomaly (i.e., a non-Wi-Fi calling flow) would be detected if none of these models generate a positive output. To train the models, we passed 80% of the data corresponding to each phase and set the contamination rate (of Isolation forest algorithm) to 0.05. We tested each model by two datasets: (a) remaining 20% of instances from its corresponding phase, and (b) all instances from other two phases. For example, the initiation model was tested by 20% of Initiation instances (unseen by the model during training) and all instances of Call and Keep-Alive. Fig. 1.2(b) shows the confusion matrix of testing Isolation-forest models. For example, 95.08% of KeepAlive instances are correctly detected by its intended model (top left cell), while less than 1% incorrectly detected by the Initiation model and none by the Call model. We also note that 4% of Keep-Alive instances are not detected by any of the three models, this measure is 9% and 5% for the Initiation and Call instances, respectively.

Combination of One-Class and Multi-Class. Note that Random Forest performs well in capturing decision boundaries and develops rules to differentiate the behaviour of various classes. However, even if the input does not belong to any of trained classes (anomalies), the model still predicts one of leaned classes – such predictions are typically accompanied by lower confidence values since there will be a disagreement among decision trees within the forest. This task of identifying anomalous data can be better done using Isolation Forest. They form tight bounds on attribute values of a benign class, and report anomalies if a small non-conformance is observed. However, these models tend to be very sensitive and may miss to output positive signal for benign instances. do not generalize well to differentiate among classes and perform the task of classification. Thus, in our system, we use both types of models to accurately monitor the carrier Wi-Fi calling application, i.e., tracking its intended phases and also reporting non-Wi-Fi calling IPSec traffic.

CHAPTER 4

PROTOTYPE AND EXPERIMENTATION

We prototyped our scheme in a small testbed, depicted in Fig 3.1, which can be readily deployed in enterprise networks. In this architecture, “Device Agent” (running on a network switch) extracts flow level information from the raw packets passing through the switch. In our prototype, we have used Cisco’s Joy [13] open-source software to perform this task. Joy extracts a time-trace of byte and packet counts from each flow and aggregates them over a time window, say 10 seconds, and sends an IPFIX packet to “Data Processor”. The data processor decodes the IPFIX packet and performs the following tasks: (a) extracts DNS query name and the corresponding server IP (used to identify carrier Wi-Fi calling endpoints), (b) filters IPSec flows (UDP 4500), and (b) aggregates raw bytecount and packet-count at resolution of 100ms to generate 100 data points. These inputs are passed on to “Arbiter” which performs multiple functions and finally outputs whether the application in the IPSec tunnel is Wi-Fi calling with a confidence value.

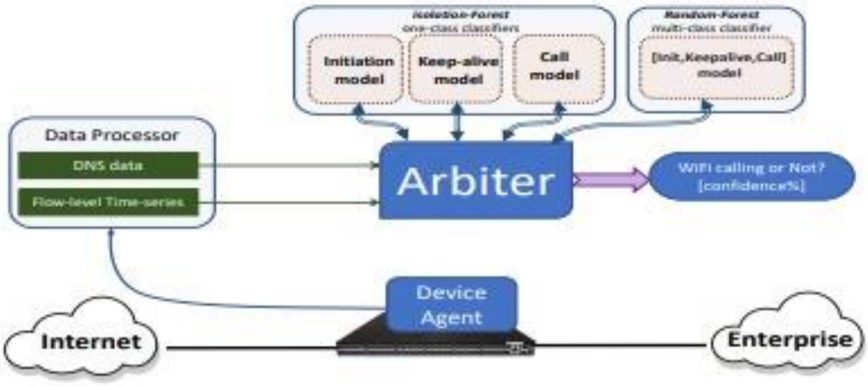
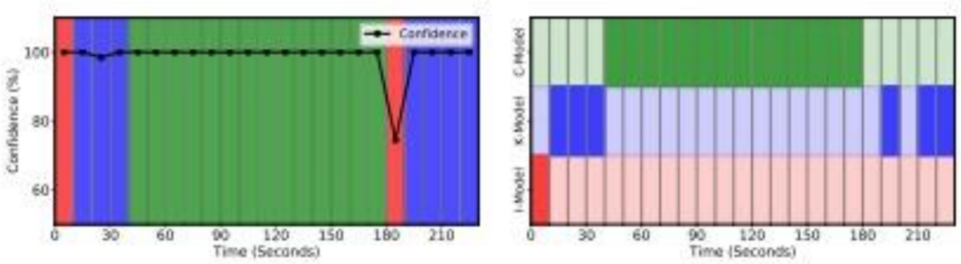


Fig. 3.1 System architecture of prototype for Wi-Fi calling monitoring.

Before understanding the arbiter module and its functions, let us walk through how our models perform for real Wi-Fi calling and non-Wi-Fi calling flows as shown in Figures 4 and 5. For the Random-Forest model (Fig. 4(a) and Fig. 5(a)), each column represents a time window of 10 seconds and its colour shows the classified phase – red, blue, and green respectively correspond to Initiation, Keep-Alive and Call. For the Isolation-Forest models (Fig. 4(b) and Fig. 5(b)) colour codes are identical to of Random-Forest and each row represents the output of each Isolation-Forest mode – a darker colour indicates the positive signal from the model.



(a) Random-Forest model.

(b) Isolation-Forest models.

Fig. 3.2 Performance of models with a Wi-Fi calling flow.

Considering a legitimate Wi-Fi call, the Random-Forest model, as shown in Fig. 4(a), consistently classifies with high confidence (i.e., >= 97%), the first instance as initiation, followed by three Keep-Alive instances, and 14 call instances until the second 180. The Isolation-Forest models also accurately detect the same phases in the first 18 epochs, as shown in Fig. 4(b). The next instance (i.e., 180-190 second), however, gets classified as initiation by random forest with lower confidence and isolation forest models do not detect any phase. Right after that, the last four instances are accurately classified by random forest while isolation forest misses one instance (200-210 second). We investigated the misclassified instance between 180s and 190s and found that the call was active for the first 3 seconds followed by 7 seconds of Keep-Alive. This caused the models to be confused as they were trained on epochs containing just one phase. We acknowledge that such false alarms might occur for a single instance but instances surrounding this miss-classified instance should still be accurately classified (given that the traffic corresponds to a Wi-Fi call). For a non-Wi-Fi calling application, we can see that random forest detects Keep-Alive first (which never occurs in Wi-Fi calls) and subsequently mostly classifies instances as initiation phase with a highly varying confidence (< 85%). Further, the isolation forest models, shown in Fig. 5(b), do not detect any of the phases for most of the time except detecting keep-alive in the beginning and one initiation between seconds 30 and 40.

These observations from performance of our models on real traffic helped us design the functions of the Arbiter to accurately predict the phase of Wi-Fi calling flows, monitor them, and detect anomalies. The arbiter needs to perform the following functions: (a) to maintain a window of models' outputs to discount one-off miss-classifications and accurately report anomalies, (b) to combine outputs of multiple models to take a decision, and (c) to report events such as IPSec session detected, presence of and current phase of Wi-Fi calling application, and anomalous non-WIFI calling IPSec flows. The arbiter keeps track of each bidirectional IPSec flow and the corresponding model outputs for the last "k" epochs. Then it looks at these k epochs to decide whether the application is indeed Wi-Fi calling. For example, with k=3 (over last three epochs), if the isolation forests did not detect any of the phases and average confidence of random forest classifier is lower than a configurable threshold, say 80%, it deems the flow to be anomalous and sends an anomaly event notification. Otherwise, it is a normal Wi-Fi calling session. We additionally use the DNS information captured by the arbiter to decide the parameter k. If the domain queried is indeed legitimate, we set k to be a larger value say 5 to tolerate miss-classifications, if any, as DNS information suggests a legitimate Wi-Fi call. The phases are reported by using the classification output and confidence of random forest classifier as it specializes in that task. Whenever the arbiter detects a flow that is not present in its state, it sends a new IPSec flow event notification. It ages out flows based on inactivity using a timer and sends a notification for the IPSec flow terminated. With this system design, we were able to accurately identify Wi-Fi calls and raise anomalies within the first window of observation for non-Wi-Fi calling IPsec sessions.

CHAPTER 5

SECURITY VULNERABILITIES OF WI-FI CALLING

In this section, we first introduce three security vulnerabilities discovered from operational Wi-Fi calling services in the U.S., and then present a study on non-U.S. operators and a feedback from the industry

V1:WLAN selection mechanisms for Wi-Fi calling devices merely consider radio/connectivity capabilities of available Wi-Fi networks

The first vulnerability is that all studied Wi-Fi calling devices cannot exclude an insecure Wi-Fi network while enabling Wi-Fi calling services. According to Wi-Fi calling standards ,there are two Wi-Fi network selection modes: manual and automatic modes. In the manual mode, devices maintain a prioritized list of selected Wi-Fi net-works, the implementation of which is vendor-specific. In the automatic mode, devices select their connected Wi-Fi networks by following the guidance from the network infrastructure based on the ANDSF (Access Network Discovery and Selection Function) procedure described in .However, both modes do not consider security risks of available Wi-Fi networks but radio quality (e.g., Thresh Beacon RSSIWLAN Low)and connectivity capabilities, such as Maximum BSS Load (i.e., the loading of Wi-Fi AP), Minimum Backhaul Threshold (e.g., 2 Mbps in the downlink).

Validation: We deploy two Wi-Fi routers of the same model to test the Wi-Fi network selection of the Wi-Fi calling devices. The experiment is conducted with four steps as follows. First, those two routers are deployed 5 and 10 meters, respectively, away from the tested devices. All test Wi-Fi calling devices are preinstalled with the required credentials to access these two Wi-Fi routers. Second, the security mechanism against the ARP (Address Resolution Protocol) spoofing attack, which is the prerequisite of various MitM (Man-in-the-Middle) attacks, is enabled on the far router, but it is disabled on the near router. Third, we launch an ARP spoofing attack from a computer that connects to the near router, to perform an MitM attack against all the other devices connecting to the router. Fourth, we enable the Wi-Fi calling service on all the tested devices, and then make a Wi-Fi calling call on each device whenever the device successfully has a Wi-Fi network connected. We have three observations from the experiment. First, all the test Wi-Fi calling devices connect to the near Wi-Fi router. Second, all the Wi-Fi calling packets from the tested devices are intercepted by the computer based on the ARP spoofing attack. Third, none of the tested devices disconnects from the near router or terminates their Wi-Fi calling services; not any alerts or warnings are observed from the tested devices. This validation experiment confirms that current WLAN selection mechanisms do not prevent the Wi-Fi calling devices from connecting to an insecure Wi-Fi network, thereby causing them to suffer from the MitM attack. Note that the MitM attack does not need to compromise or control the near router.

Security implications: It is not without reasons that the WLAN selection mechanisms do not take security issues into consideration but consider only the radio quality or/and WLAN performance, since the Wi-Fi calling sessions have been protected by the IPSec tunnels with the end-to-end confidentiality and integrity protection. Although the security protection can prevent the Wi-Fi calling packets from being decrypted or altered, intercepting or discarding those packets for further attacks is still possible. We believe that 3GPP and GSMA shall revisit the Wi-Fi network selection mechanisms for the Wi-Fi calling service in terms of security; otherwise, the Wi-Fi calling users are being exposed to potential security threats

V2: Potential Side-channel Inference

Given the security mechanisms of untrusted access, the packets of the cellular services under untrusted Wi-Fi networks can be securely delivered through the IPSec channel between the UE and the ePDG. However, we discover that for all the test operators, the Wi-Fi calling service is the only service carried by the IPSec channel. This monotonous operation may allow the adversary to monitor the channel and then launch a side-channel attack to infer user privacy from the Wi-Fi calling events (e.g., call and text messaging statuses) and call statistics. **Validation:** We examine whether any information can be inferred based on the intercepted Wi-Fi calling packets, which are encrypted by IPSec. After analysing their patterns, we discover that for all the three operators, there are six service events of the Wi-Fi calling service, namely dialling/receiving a call, sending/receiving a text message, and activating/deactivating the service. IPSec packets captured on a Wi-Fi AP when the above six events are triggered on a test phone connecting to the AP. It is observed that all the events differ from each other in terms of traffic patterns, which are composed of packet direction (uplink or downlink), packet size, and packet interval. In order to automatically identify them based on the encrypted Wi-Fi traffic, we apply a decision tree method, the C4.5 algorithm .To prepare a set of training data, we trigger those six events on the test phone with 50 runs each while collecting all the IPSec packets on the Wi-Fi AP. Based on the training data, a classification model can be generated by the C4.5 algorithm. We assess the classification accuracy of the model using 50 tests by comparing the model’s output with the test phone’s packet trace as shown in Figure 5. The result shows that the model can give 100% accuracy. Note that the test phone is Nexus 6P with the Wi-Fi calling service of US-I. We next examine whether the classification model works for cross-phone and cross-carrier cases. We consider various devices with the Wi-Fi calling services of the three carriers.

Table 2 summarizes the result. It is observed that those six events in all the test cases can be identified accurately. Accordingly, the model that is derived based on the training data collected from Nexus 6P with the US-I's Wi-Fi calling service can be applied to the other devices and carriers, which include the Samsung Galaxy J7/S6/S7/S8 and iPhone 6/7/8 devices with the US-II/US-III networks. **Security implications:** The IPSec channel can prevent man in-the-middle attackers from decrypting or altering the Wi-Fi calling packets, but does not block the side-channel inference attack. Its monotonous operation allows the adversary to collect 'clean' Wi-Fi calling traffic, which simplifies the side-channel inference.

V3: The Inter-system Service Continuity Mechanism of Wi-Fi Calling can be Bypassed

The inter-system service continuity mechanism can seamlessly switch the voice service of Wi-Fi calling on a device back to the cellular-based voice service (e.g., VoLTE), when the device disconnects from its connected Wi-Fi network or it cannot be reached through the Wi-Fi network (e.g., no response from the device in the Wi-Fi calling service). The mechanism can be triggered by the device or the cellular network infrastructure, and mainly consists of two steps, namely an inter-system handover between Wi-Fi and the cellular network, and a procedure of the IMS service continuity. Its operation can inherently protect the device against a DoS attack on the Wi-Fi calling service. For example, when all the Wi-Fi calling packets are maliciously dropped, the device is unreachable. However, the operation is not bullet-proof and may be bypassed with a sophisticated attack. Validation: We conduct experiments to examine whether the mechanism can be bypassed in any scenarios. We test a Wi-Fi calling device with the following four scenarios, together with their corresponding results. First, the device with an established voice call of Wi-Fi calling moves out of its connected Wi-Fi network. We observe that the ongoing voice call can successfully migrate from Wi-Fi calling to VoLTE without any call interruption. Second, the device is dialling a Wi-Fi calling call while all its Wi-Fi calling packets are discarded from the Wi-Fi AP. We find that the device successively sends a packet of SIP INVITE to the Wi-Fi calling server; after six attempts, it switches to initiating a VoLTE call. Third, while the device is having an incoming call, all the Wi-Fi calling packets are discarded. It is observed that the device switches to VoLTE for the incoming call. Fourth, the packets of a Wi-Fi calling call on the device are discarded right after the call is established. We observe that no voice can be heard from two call ends, but the inter-system switch is not triggered and the device keeps the connection of the Wi-Fi network. In summary, the inter-system service continuity mechanism is triggered only when the radio quality of the connected Wi-Fi network becomes bad, or the device and the network infrastructure cannot reach each other in the Wi-Fi calling service. As in the above fourth case, where the device and the network can reach each other but some packets are dropped, the adversary can attack a device’s Wi-Fi calling call while keeping the device using the Wi-Fi calling service by preventing the inter-system switch from being triggered. Security implications: Although the Wi-Fi calling standard provides the inter-system switch mechanism for the Wi-Fi calling service continuity, it may suffer from some sophisticated attacks where the Wi-Fi calling packets can be intercepted. The interception is possible since the Wi-Fi calling traffic needs to traverse untrusted Wi-Fi networks and the Internet. To prevent the attacks, the service continuity mechanism should also take security concerns into consideration.

A Vulnerability Study on Non-U.S. Operators

We conduct a study of the Wi-Fi calling vulnerabilities on two Taiwan operators to examine whether they are limited to only U.S. operators or not. We summarize the result of the test phone, Samsung Galaxy S8, for each vulnerability as follows. V1: We repeat the validation experiment of V1 on the phone with the Taiwan operators, and observe the same result that the WLAN selection mechanism does not prevent the device from connecting to an insecure Wi-Fi network, where an ARP spoofing attack is launched. V2: For both Taiwan operators, we observe that the Wi-Fi calling service is also the only one service carried by the IPSec channel, and then apply the same classification method described in Section. into classifying the aforementioned six events. The result confirms that the method can give 100% accuracy for the event inference. V3: We test the device with the Wi-Fi calling services of the Taiwan operators for the inter-system service continuity mechanism. It is also observed that the mechanism is deployed and can be bypassed in the fourth scenario described in Section.

Industry Feedback

We have reported the vulnerabilities to the U.S. operators that are studied in this work and several device manufacturers including Google, Samsung, and Apple. In particular, the Google Android security team gives a positive feedback that the team has confirmed our findings after a security analysis of the vulnerabilities, and will address them in an upcoming security patch. We thus received a Google Security Reward in Jan. 2020. On the other hand, we are awaiting hearing from the other operators and manufacturers.

CHAPTER 6

TELEPHONY HARASSMENT/DENIAL OF VOICE SERVICE (THDOS) ATTACK

We next devise the THDoS attack, which can cause tele phony harassment or denial of voice service on the Wi-Fi calling users. In the following, we describe the attack design, evaluation and real-world impact.

Attack Design

In this attack, the adversary manages to discard particular signalling or/and voice packets of Wi-Fi calling from the victim device, while preventing the inter-system service continuity mechanism from being triggered. The attack can cause damage on the device's voice service supported by Wi-Fi calling, and let the damage last by getting the device stuck with the Wi-Fi calling service. To discard particular packets between the device and the network infrastructure, the adversary needs to identify encrypted IPSec packets. We next start with an illustrative example of the Wi-Fi calling call, and then analyse the traffic patterns of the Wi-Fi calling messages and events based on the encrypted packets.

Traffic Pattern Analysis: We have five observations on the traffic patterns of the Wi-Fi calling messages and events.

1. The sizes of the voice packets in IPSec are smaller than 200 bytes (e.g., 176 bytes).
2. The sizes of the SIP packets that contain signalling messages, including INVITE, 180 RINGING, 200 OK, and BYE), in IPSec are much larger than the voice packets (e.g., 800-1360 bytes).
3. The callee starts to receive the voice packets from the Wi-Fi calling server after the 180 RINGING message is sent.
4. No voice packets are sent out by the callee before the call conversation starts.

5. The callee keeps receiving more than 10 voice packets every two seconds from the Wi-Fi calling server after the call conversation starts.

These patterns allow us to identify call events, e.g., an outgoing call is initiated, an incoming call attempt arrives, and an ongoing call ends. Moreover, by correlating them with the call flow of Wi-Fi calling, the signalling messages of Wi-Fi calling can be identified purely based on the encrypted IPSec packets. Note that the third observation is made only from US-I and US-II; the others can be observed from all the test operators

Attack Evaluation

We launch attacks by discarding different patterns of the signalling and voice packets for an outgoing call of Wi-Fi calling. We exploit the results to devise four attack variants as follows. Note that the damage that is caused to mobile phones may not be applied to other SIP phones (e.g., Cisco SPA 525G2).

Annoying-Incoming-Call Attack: The callee as the victim would receive multiple incoming calls from the caller. There are two approaches. First, the adversary drops the 183 Session Progress message sent by the callee, and then the caller’s Wi-Fi calling device would initiate another VoLTE call towards the callee. Second, the adversary discards the 180 Ringing message sent by the callee, and then it would cause the caller’s Wi-Fi calling device to get stuck in the dialling screen. The caller does not hear any alerting tone, but the callee’s device would ring. The caller may thus keep redialling.

Zombie-Call Attack: The caller’s device can be forced to get stuck in the dialling screen, when the adversary discards the 200 OK messages sent by the callee. The message indicates that the call has been answered, so without receiving the message, the caller’s device gets stuck in the dialling screen and keeps hearing the alerting tone. The call conversation is thus never started.

Intermittent Mute Call Attack: Two parties of a Wi-Fi calling call are both victims. This attack does not aim to terminate the call but only mute the victims’ voice for a certain time. Our result shows that the adversary can mute the call up to 8 seconds by dropping voice packets. If the voice suspension time is longer than 8 seconds, the call would be terminated by the network. To prolong the attack period, the adversary can launch a cyclical attack that drops voice packets for 7 s and skip the packets for the next 1 s to mute the call intermittently.

Telephony Denial-of-Voice-Service Attack: Both the caller and the callee are victims. This attack downgrades the voice quality of a Wi-Fi calling call so that the conversation is hard to be continued; meanwhile, the inter-system service continuality mechanism is not triggered. It is achieved by controlling the drop rate of the intercepted Wi-Fi calling packets to/from the victim. Table 5 shows the negative impact on the voice quality with different drop rates. There are four findings. First, when the drop rate is below 20%, the caller/callee users do not complain about any voice quality downgrade. Second, when the drop rate increases to 40%- 60%, some of the users may notice some noises. Third, when the drop rate becomes 70%-90%, the voice call is hardly continued. Fourth, when the drop rate is 100%, the call is terminated within 8 seconds. Note that when the drop rate is below 90%, the call termination is never triggered.

Real-world Impact

The impact of the THDoS attack can be significant in practice. Our studies show that the campus Wi-Fi networks, which most U.S. universities have deployed, are the best attack surfaces for the adversary. For example, the campus Wi-Fi (MSUNet) in Michigan State University provides students, the faculty, and the staff with free Wi-Fi access. In a 2-min experiment, we discover that more than 700 devices including smartphones, tablets, and computers, connect to MSUNet. All the devices are served by the same gateway which is vulnerable to an ARP spoofing attack, so their Wi-Fi calling packets can be intercepted if there are any. Therefore, it allows the adversary to launch the THDoS attack against the Wi-Fi calling devices under the gateway. Note that MSUNet is not the only Wi-Fi infrastructure that suffers from the ARP spoofing and THDoS attacks. We find that such vulnerability also exists in the campus Wi-Fi of many other universities, such as New York University, University of California Berkeley, North-eastern University, etc

CHAPTER 7

CONCLUTION

In this paper we have developed a solution for enterprise networks who want to allow Wi-Fi calling over encrypted IPSec tunnels. We have analysed real traces of Wi-Fi calling traffic and identified key behavioural network patterns. We then developed ML-based models to classify three phases of a Wi-Fi calling flow and detect anomalous traffic exchanged inside an IPSec tunnel. Lastly, we prototyped our scheme in a testbed to show how benign Wi-Fi calling traffic can be automatically distinguished from anomalous IPSec flows.

CHAPTER 8

REFERENCES

[1] Apple. (2019) Wireless carrier support and features for iPhone. [Online]. Available: https://support.apple.com/en-au/HT203982

[2] Cisco. (2016) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020. [Online]. Available: https://bit.ly/2UvEB73

[3] S. Dimatteo et al., “Cellular Traffic Offloading Through WiFi Networks,” in Proc. IEEE MASS, 2011.

[4] F. Rebecchi et al., “Data Offloading Techniques in Cellular Networks: A Survey,” IEEE Communications Surveys Tutorials, vol. 17, no. 2, pp. 580–603, Secondquarter 2015.

[5] B. Pularikkal et al., “Carrier Wi-Fi Calling Deployment Considerations,” Working Draft, IETF Secretariat, Internet-Draft, January 2017. [Online]. Available:https://www.ietf.org/archive/id/draft-pularikkal-opsawg-wificalling-03.txt

[6] R. Barbieri, D. Bruschi, and E. Rosti, “Voice over IPsec: Analysis and Solutions,” in Proc. ACSAC, Washington, DC, USA, Dec 2002.

[7] T. Yildirim and P. Radcliffe, “VoIP traffic classification in IPSec tunnels,” in Proc. ICEIE, Kyoto, Japan, Aug 2010.

[8] A. D. Keromytis, “A Comprehensive Survey of Voice over IP Security Research,” IEEE Communications Surveys Tutorials, vol. 14, no. 2, pp. 514–537, Second 2012.

[9] S. Chalakkal et al., “White paper: Practical Attacks on VoLTE and VoWiFi,” ERNW Enno Rey Netzwerke, Tech. Rep., July 2017. [Online]. Available: https://bit.ly/2XZk882

[10] T. Xie et al., “The Dark Side of Operational Wi-Fi Calling Services,” in Proc. IEEE CNS, Beijing, China, May 2018

[11] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, “Internet key exchange protocol version 2 (ikev2),” Tech. Rep., 2014.

[12] A. Huttunen et al., “UDP encapsulation of IPsec ESP packets,” Tech. Rep., 2004.

[13] Cisco. (2019) Joy Tool. [Online]. Available: https://github.com/cisco/joy