EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi

Pragati Shrivastava, Mohd Saalim Jamal, and Kotaro Kataoka

***Abstract*—Spoofing the identity of a WiFi access point (AP) is trivial. Consequently, an adversary can impersonate the legitimate AP (LAP) by mimicking its network name (SSID) and MAC address (BSSID). This fake AP is called the evil twin. An evil twin can perform multiple attacks such as man-in-the-middle (MITM) attack between the LAP and a wireless client as well as service blocking of LAP. Existing solutions rely on the collection and calculation of information with the AP and/or client for finding evidence of evil twins in the WiFi network. Some of them require additional hardware to acquire further information that cannot be provided by the AP/client. In this paper, we propose "EvilScout," an evil twin detection and mitigation framework that utilizes the information of the IP-prefix distribution by the LAP. EvilScout exploits the SDN potential for detection of an evil twin without the need of any additional hardware or modifications at the AP or client. Additionally, the information that becomes available at the SDN controller enables simplified and more accurate evil twin detection. This paper presents the implementation of EvilScout over a real SDN WiFi testbed with an actual evil twin. We verify the successful detection of the evil twin with high accuracy and low processing cost at the SDN WiFi. We perform a rigorous analysis of the evil twin in different WiFi setups and discover a new "AP Service Blocking" attack by the evil twin adversary in the WPA2 protected WiFi for the first time.**

***Index Terms*—Software-defined networks (SDN), WiFi security, evil twin attack, duplicate association.**

I. Introduction

WIFI is the most widely available source of Internet access and it is prone to various security threats, such as the evil twin. An adversary can impersonate a legitimate access point (LAP) as it is trivial to spoof the network name (SSID) and MAC address (BSSID) of a LAP. This fake AP claiming to be a LAP is known as evil twin. The hotspot and software capabilities [1], [2] at client devices (laptop/mobile) are sufficient to launch the evil twin attack. If clients connect to an evil twin, it can invade as a man-in-the-middle (MITM) attack between LAP and clients and can eavesdrop or manipulate client's sensitive data. Besides, multiple simultaneous evil twin attacks are capable of making the WLAN break down by severely affecting Internet services.

Manuscript received June 20, 2019; revised October 26, 2019, January 22, 2020, and February 2, 2020; accepted February 5, 2020. Date of publication February 10, 2020; date of current version March 11, 2020. The associate editor coordinating the review of this article and approving it for publication was Q. Li. *(Corresponding author: Pragati Shrivastava.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Hyderabad 502205, India (e-mail: cs14resch11007@iith.ac.in; cs16mtech11024@iith.ac.in; kotaro@iith.ac.in).

Digital Object Identifier 10.1109/TNSM.2020.2972774

An evil twin attack is easy to perform in weakly secured WiFi networks such as open public WiFi. However, the possibility of this attack cannot be ruled out even for a protected WiFi [3], [4], [5]. Public-WiFi is popular due to its low cost and wide availability [6], particularly in the regions with limited network/Internet connectivity, e.g., "Google Station" [7]. Existing security measures, such as alerts/warnings while connecting to a Public-WiFi may not be adequate. These alerts are not repetitive and may be ignored by users due to various factors such as availability of other options, cost, and security awareness [8]. Therefore, additional measures are required to counter security concerns in such scenarios.

Existing detection mechanisms of the evil twin depend on either client-side modifications or additional measures by the network operator. Client-side modifications are mostly in network interface card (NIC) or firmware [9], [10], [11], [12], making them undesirable as they hinder the basic nature of Bring-Your-Own-Device (BYOD) in WLAN. The mechanisms that require additional measures by the network operator [5], [13], [14], [15], [16] become costly due to specialized hardware, and slow due to requirements of human involvement.

Existing detection capabilities are restricted to a certain attacking scenario: the evil twin and the victim LAP must be on different channels [13], [15], [17]. A robust detection mechanism for an evil twin attack on the same channel as the LAP is not well studied. Furthermore, a completely network-based (no additional hardware) detection mechanism is also not explored. These technical challenges and limitations motivated us to further study evil twin attacks and to develop a network-based detection mechanism of evil twin attacks on the same channel as the LAP.

During our experimental studies, we discovered a fascinating fact, the "Duplicate Association" of the client, as shown in Fig. 1. This discovery, together with monitoring of the communication, provides a highly accurate network-based evil twin detection approach.

Duplicate Association: When an evil twin impersonates a LAP by spoofing its BSSID on the same channel, we found that a WiFi client is associated with both LAP and evil twin regarding the wireless coverage. We named this phenomenon as Duplicate Association. We verified the duplicate association of the client with different hardware APs, as given in Table I. This validates our finding of duplicate association in 802.11 association processes.

During our experiments, we also encountered a major attack on WPA2 protected WiFi AP named as AP Service

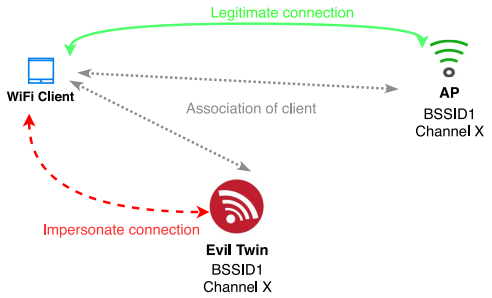


Fig. 1. Duplicate association of client.

TABLE I
TESTED HARDWARE ACCESS POINTS

Wireless (WiFi) APs	Duplicate Association
Cisco AIR-CAP3702I-D-K9	True
TP-Link 1043ND	True
Linksys FBA_ E1200	True
WiFi Adapter Ralink RT5370	True

Blocking (APSB). A similar issue was also mentioned by Jang *et al.* [17].

APSB: We found that if an attacker creates the evil twin of a WPA2 protected LAP, it has the potential to disrupt the authentication handshake between clients and LAP. This successfully blocks the LAP to connect any new client. Hence, an APSB attack becomes possible by interfering with the EAPOL 4-way handshake of WPA2 protected LAP and client, using capabilities of evil twin on the same WiFi channel as LAP.

In this paper, we present "EvilScout," a security framework that monitors and analyzes the client's traffic to find the signature of the evil twin's presence. The detection of the evil twin is enabled by verifying the IP-prefix distributed at LAP. As the evil twin runs its own DHCP server, there can be two possible cases. Either the evil twin distributes the IP addresses in the same IP-prefix as a LAP or another IP-prefix implemented in the NAT. We analyzed both situations and found anomalies to detect the presence of the evil twin of any victim LAP in the network. The EvilScout framework generates minimal burden at LAP, as it does not require additional processing and only the first packet of any communication after the association of the client is sufficient to detect the evil twin.

As a preliminary experiment, this research deployed and tested the evil twin implementation in an SDN enabled WiFi network. The prototype implementation of EvilScout exploits the LAP programmability, global view at the WLAN controller and its capabilities to monitor the LAP client communications. EvilScout is deployed on top of the SDN controller and uses OpenFlow capabilities, such as PacketIn, and writes FlowMod messages.

Contribution: Our approach is cost-effective as it does not require any additional hardware or software capabilities at AP/Network/Client. EvilScout is transparent to the WiFi clients and the attackers. It can also detect the evil twin in either of the cases: when the attacker uses LAP for Internet access or uses any other Internet connectivity, i.e., relay-based or isolated [18]. EvilScout is also capable of fighting against many evil twin attacks performed simultaneously to breakdown the WLAN. Our main contributions are:

- We performed the real-time evil twin analysis with different setups of the WiFi network (Open/WPA2) and explained the vulnerabilities produced by the evil twin attack on the same channel;
- We proposed the EvilScout, the first completely network-based (no additional hardware) evil twin detection mechanism using the novel discovery of "Duplicate Association";
- We proved EvilScout feasibility and analyzed its performance over testbed implementation and mininet-WiFi setup;
- We presented a comprehensive study of the APSB attack in different WiFi setups and novel detection techniques in SDN WiFi.

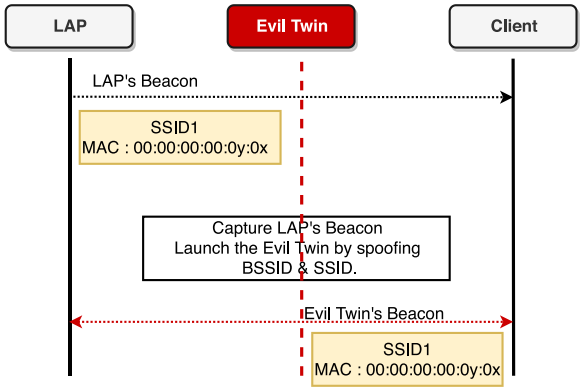


Fig. 2. Evil Twin launched by attacker.

II. THREAT MODEL: EVIL TWIN ATTACK

In a WiFi network, an AP speaks the beacon frames periodically to exhibit its presence in the wireless. A client discovers APs by listening to their beacon frames. These beacon frames cannot be encrypted. Otherwise, the client cannot connect to the AP. Consequently, this vulnerability provides an excellent opportunity to the attackers for invading the WiFi network and launch attacks such as the evil twin.

Evil Twin is a malicious twin of a LAP launched by the attacker, which impersonates the identity of the LAP in the network, as shown in Fig. 2. An adversary captures the beacons of the victim LAP and easily extracts its SSID and BSSID as beacons are not encrypted. Then, the attacker creates a similar AP by spoofing BSSID and SSID of the LAP, as depicted in Fig. 2. The attack becomes successful as the clients listening to the beacons cannot differentiate between the LAP's and evil twin's identity, assuming both are legitimate APs.

A. Attack Performed by Evil Twin

Evil twin attacks can be performed by the following two methods:

- *Passive evil twin attack:* the attacker launches the evil twin and increases the signal strength of the evil twin's hotspot. Thus, whenever any client tries the association, it connects to the evil twin, as shown in Fig. 3;

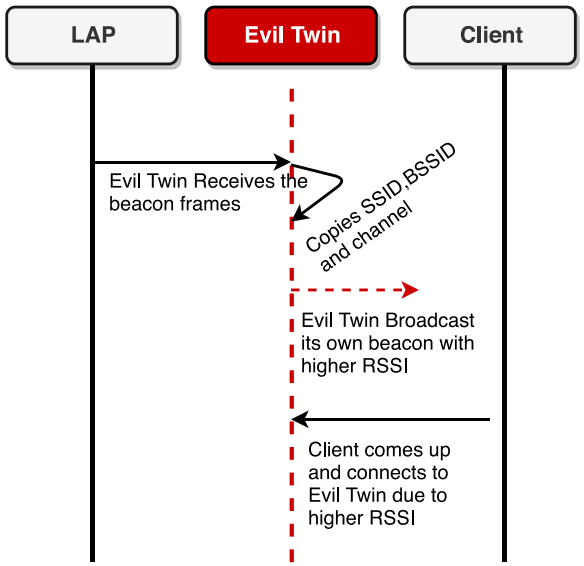


Fig. 3. Evil Twin waits for clients (passive attack).

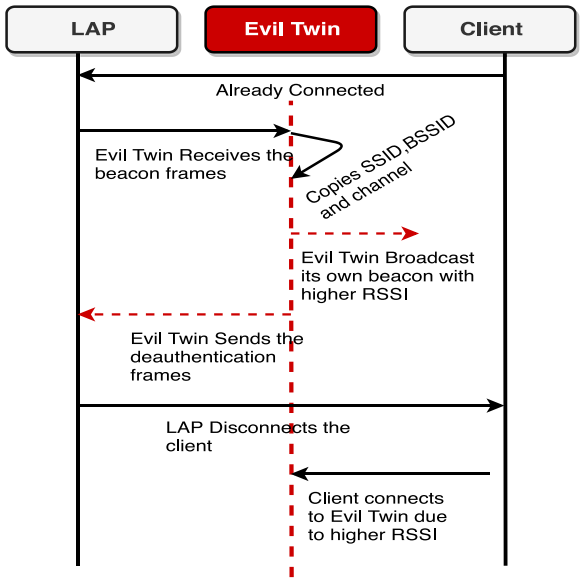


Fig. 4. Evil Twin actively disconnects the clients (active attack).

- *Active evil twin attack:* evil twin targets the clients that are already connected with LAP by performing a de-authentication attack [18] over LAP. Due to that, the client disassociates with LAP and associates with the evil twin, as shown in Fig. 4.

Attack launching strategies: evil twin coexists in the WiFi network by using LAP for Internet connection or a third-party Internet connection: a) the evil twin relays traffic via LAP; b) the evil twin relays traffic through other Internet connection.

B. Severity of the Evil Twin Attack in the WiFi MAC Layer

In public WiFi, some of the well-known attacks are passive monitoring and phishing. Evil twin introduces considerably more severe security threats over passive channel monitoring because: 1) evil Twin is capable of redirecting the communication of WiFi clients; 2) it is capable of causing channel saturation; 3) it can delay client communication. The phishing attacks in WiFi are mostly related to the MAC layer. Phishing attacks performed by WiFi clients can be trivially found. Thus, more severe phishing attacks in WiFi require the exploitation of APs. An attacker can exploit APs to perform phishing as follows: 1) by compromising LAP or 2) by launching an evil twin. Compromising an AP in enterprise WiFi settings is difficult, but launching an evil twin is easy. Consequently, evil twin attack is easy to perform, introducing severe threats, and it is hard to detect.

C. Problem Specification

The evil twin impersonates the identity of LAP. This impersonation prevents clients from differentiating between the evil twin and the LAP and creates an opportunity for the attack. Evil twin detection and mitigation techniques are generally based on physical channel information such as CSI (channel state information), or other wireless parameters such as inter-packet arrival time [19], to fingerprint the attacker devices. Most of the solutions implement wireless monitoring at clients [10], [15], [20], and few [13] utilize an additional wireless device such as a sniffer for detection of the evil twin. These solutions encounter following difficulties:

- Client-based evil twin detection approaches require additional software/hardware capabilities at the client;
- Profiling of APs at wireless clients requires continuous computation on battery-driven client devices;
- Detection using sniffer devices is slow, which gives an opportunity to the attacker for stealing data, blocking network service, etc.;
- Most of the solutions do not specify if their approach detects the evil twin launched on the same channel as LAP or on a different channel;
- Evil twin on the same channel as LAP could not be detected trivially by channel state information as it coincides with the channel state information of LAP.

This paper attempts to find a network-based detection approach that only utilizes the existing information in the WiFi network and overcomes the aforementioned limitations. This demands new insights of the evil twin attacks. Hence, we attempt a thorough analysis of the evil twin in the WiFi network via testbed implementation. To the best of our knowledge, no other study has performed the analysis of the evil twin, AP, and client communication in a real testbed.

III. Experimental Analysis of Evil Twin Attack at a Real Testbed

We performed real-time analysis of the evil twin attack in the WiFi testbed. We attempted a thorough examination of the process, from the association to the start of the communication among a target LAP, an evil twin and a victim client. We also covered it in different WiFi deployment scenarios.

We deployed the evil twin on a Raspberry Pi 3 with two WiFi NICs, one WiFi NIC to launch a hotspot and the other for back-end Internet access via LAP. We deployed a monitoring host to capture the WiFi control and management frames using the Wireshark tool [21], with monitoring mode enabled

TABLE II
EVIL TWIN PRESENCE IN DIFFERENT WIFI DEPLOYMENT

Scenario	Legitimate AP	Evil Twin
Public-WiFi.	Open	Open
Mixed-WiFi.	WPA2	Open
Protected-WiFi.	WPA2	WPA2

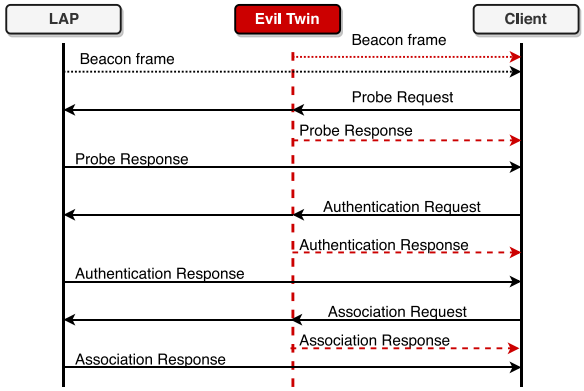
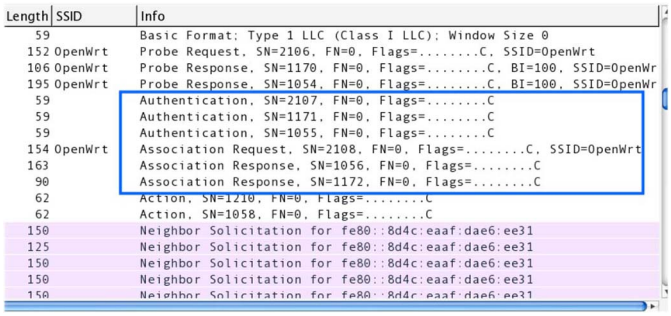


Fig. 5. Public-WiFi: evil twin and LAP configured as Open WiFi.



Length	SSID	Info
59		Basic Format; Type 1 LLC (Class I LLC); Window Size 0
152	OpenWrt	Probe Request, SN=2106, FN=0, Flags=.....C, SSID=OpenWrt
106	OpenWrt	Probe Response, SN=1170, FN=0, Flags=.....C, BI=100, SSID=OpenWr
195	OpenWrt	Probe Response, SN=1054, FN=0, Flags=.....C, BI=100, SSID=OpenWr
59		Authentication, SN=2107, FN=0, Flags=.....C
59		Authentication, SN=1171, FN=0, Flags=.....C
59		Authentication, SN=1055, FN=0, Flags=.....C
154	OpenWrt	Association Request, SN=2108, FN=0, Flags=.....C, SSID=OpenWrt
163		Association Response, SN=1056, FN=0, Flags=.....C
90		Association Response, SN=1172, FN=0, Flags=.....C
62		Action, SN=1210, FN=0, Flags=.....C
62		Action, SN=1058, FN=0, Flags=.....C
150		Neighbor Solicitation for fe80::8d4c:eaaf:dae6:ee31
125		Neighbor Solicitation for fe80::8d4c:eaaf:dae6:ee31
150		Neighbor Solicitation for fe80::8d4c:eaaf:dae6:ee31
150		Neighbor Solicitation for fe80::8d4c:eaaf:dae6:ee31
150		Neighbor Solicitation for fe80::8d4c:eaaf:dae6:ee31

Fig. 6. Wireshark capture of Public-WiFi setup.

on NIC. Table II shows the generated scenarios to extensively examine the effects of the evil twin presence on the same channel as LAP. We discovered an interesting fact named "Duplicate Association," which is useful to detect an evil twin. We also found a vulnerability caused by the evil twin, i.e., *APSB attack*.

A. Study of Public-WiFi Scenario

We launched the evil twin with the same SSID "OpenWrt" and BSSID as the LAP. Both were configured as open WiFi, i.e., the client does not need an access key for connecting to a WiFi AP.

Experiment & Analysis: The WiFi client actively discovers the network by sending probe requests. Whenever an AP receives the probe request, it responds with a probe response message that contains the BSSID and AP capabilities, such as frame control fields and MAC addresses.

In this scenario, the evil twin and LAP both reside in the same WiFi channel and have the same WiFi identity (SSID and BSSID). Therefore, both listen to the association request message of the client and generate the association response back

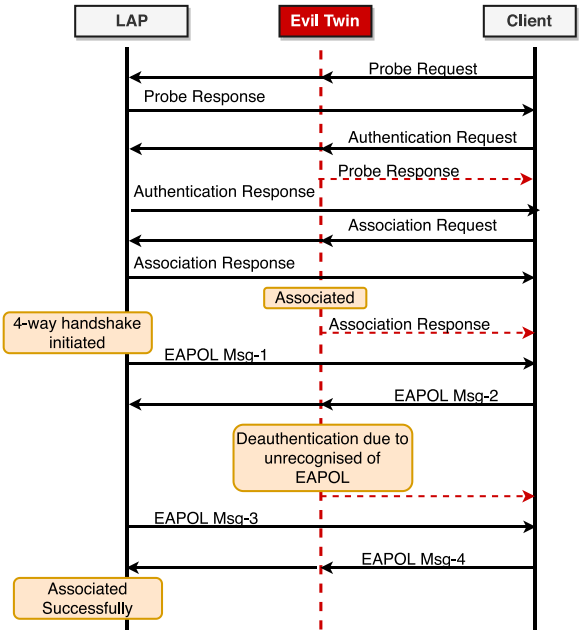


Fig. 7. Mixed-WiFi: Evil Twin Open and legitimate AP is WPA2/PSK(Case 1).

to the client, as shown in Fig. 6. After the completion of association, the client is associated with both LAP and evil twin as no further authentication is required. Now LAP and evil twin can both listen to the client communication as they have completed the association process with the client. We performed the same experiment with different hardware APs (Table I) as LAP and observed the same phenomena with all of them. We named this process as the "Duplicate Association," which allows the detection of the evil twin in a WLAN network. In this scenario, the client cannot distinguish between LAP and evil twin because they have the same BSSID. Therefore, the client is associated with both LAP and evil twin. The duplicate association enables LAP to listen to communication between its victim clients and the evil twin.

B. Study of Mixed-WiFi Scenario

In this scenario, we configured the LAP with WPA2/PSK and the evil twin as open WiFi.

Experiment & Analysis: In this scenario, the client actively discovers the network by sending probe requests. Whenever an AP receives the probe request, it responds with a probe response message. The client receives the probe response from both LAP and evil twin and initiates the 802.11 association process. **Case 1:** if the client listens to the probe response of the evil twin first, it starts an association with the evil twin's open WiFi. The initial messages of the 802.11 association process are received by both evil twin and LAP. The client associates only with the evil twin (Fig. 7) as LAP could not complete the association due to the absence of a WPA2 4-way EAPOL handshake. **Case 2:** if the client listens to the probe response of LAP first, it starts an association with LAP. The first association request messages are received by both evil twin and LAP. The evil twin assumes that it is associated with the client after sending the association response

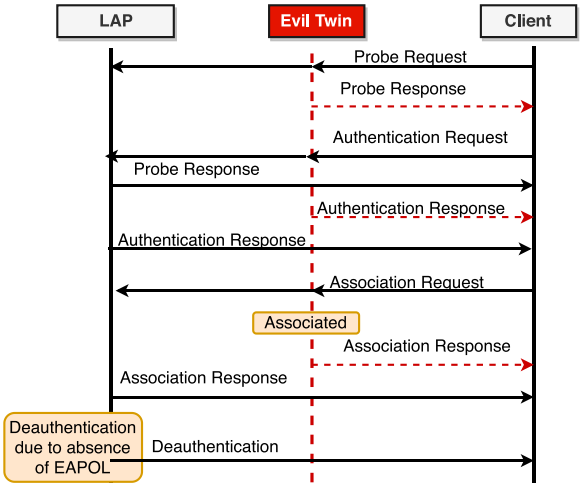


Fig. 8. Mixed-WiFi: Evil Twin Open and Legitimate AP is WPA2/PSK(Case 2).

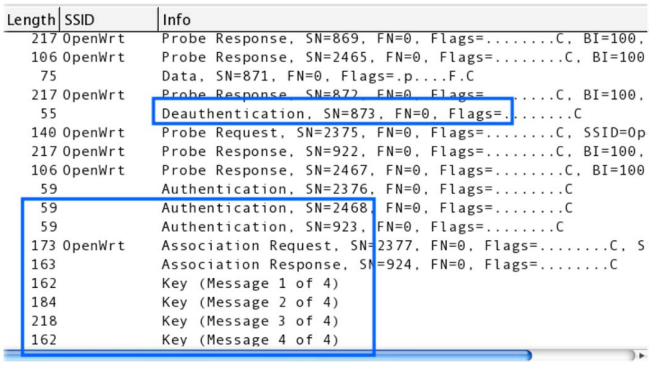


Fig. 9. Wireshark capture of Mixed-WiFi setup(Case2).

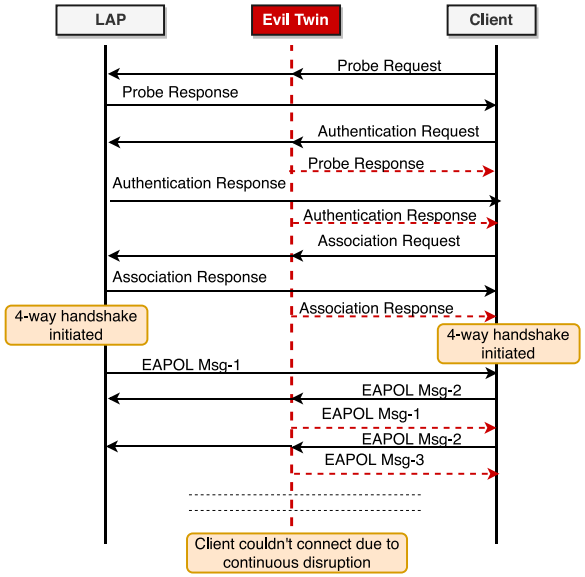


Fig. 10. Protected-WiFi: Evil Twin and Legitimate AP both configured as WPA2/PSK.

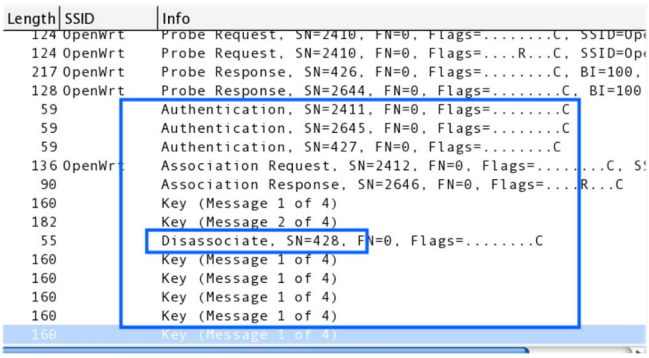


Fig. 11. Wireshark capture: Evil Twin and Legitimate AP both configured as WPA2/PSK.

message. However, the 802.11 association process is carried forward by a 4-way handshake of WPA2 with the LAP, and after the EAPOL exchange is completed, the LAP and the client become associated (Fig. 8). During the authentication process, the evil twin also receives the EAPOL messages from the client, which triggers the disassociation process at the evil twin. Thus, the evil twin sends the disassociation message to the respective client. This event is recorded in the Wireshark, as shown in Fig. 9. Thus, although the evil twin is active in the WiFi network, no attack happens. *In both cases, the connection is not stable for a long time, and service is interrupted due to the reception of the de-authentication frames either from LAP or evil twin, as shown in Fig. 9.*

C. Study of Protected-WiFi Scenario

In this experimental setting, both LAP and evil twin are configured with WPA2/PSK protocol. The process is depicted in Fig. 10.

Experiment & Analysis: Whenever a client tries the active connection to the WiFi network, it sends the probe request and receives the probe response from both the APs, as shown in Fig. 10. After completion of the 802.11 primary association process, the authentication occurs. Initially, both evil twin and LAP generate the message-1 of EAPOL and put their respective generated ANonce (AP nonce: a random sequence generated at AP) for a new session key (pairwise transit key (PTK)) generation at the client and AP. As the client receives message-1 from any of the APs (LAP/evil twin), it responds with the message-2 containing SNonce (Station nonce: a random sequence generated by the client) and the message integrity code (MIC) to the respective AP for the verification of message-1. The client receives remaining message-1 of the evil twin/LAP and gets confused. Hence, due to the presence of the evil twin, the 4-way EAPOL handshake is not completed as both evil twin and LAP receive the EAPOL messages and respond to the client simultaneously.

Consequently, the client cannot be associated due to the continuous disruption of the 4-way handshake, as captured in Wireshark (Fig. 11). Network services at LAP become blocked. Thus, evil twin brings a new vulnerability as it can attack the WPA2 protected network and disable its regular network services. We named this attack as APSB attack. This is an entirely different attack compared to usual MITM attacks performed by the evil twin.

TABLE III
EVIL TWIN'S GENERATED THREATS IN DIFFERENT DEPLOYMENT (ON THE SAME CHANNEL AS THE LAP)

Scenario	Legitimate AP	Evil Twin	Observed attack
Public-WiFi.	Open	Open	MITM
Mixed-WiFi.	WPA	Open	Service Interruption
Protected-WiFi.	WPA	WPA	AP Service Blocking

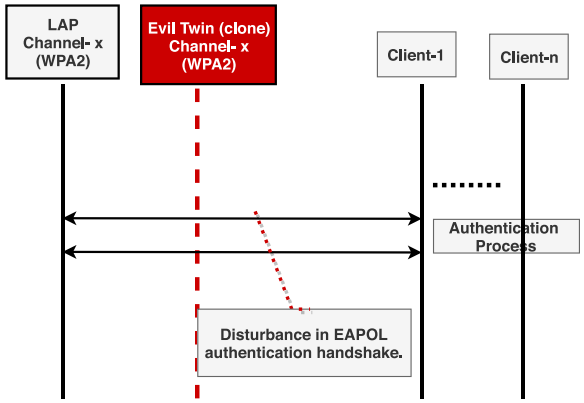


Fig. 12. AP Service Blocking Attack.

D. Summary of Evil Twin Analysis

After analyzing the evil twin in a real testbed with different WiFi setups, we found that it can introduce possible threats, as presented in Table III.

IV. EVIL TWIN TRIGGERED AP SERVICES BLOCKING ATTACK

In the APSB attack, an adversary disrupts the LAP services by launching its clone in a WPA2 protected WiFi network. If LAP's clone is on the same channel, it creates a disturbance in the 4-way EAPOL handshake of LAP and clients, as shown in Fig. 12. This process blocks the LAP to serve clients in the WiFi network. APSB can be considered as a denial-of-service (DoS) on LAP by disrupting the WPA2 EAPOL handshake of LAP and clients. The evil twin is also a clone of the victim LAP and if they work on the same channel, the evil twin can easily launch APSB. This APSB attack is over and above the usual attacks associated with the evil twin.

Assumptions: WiFi network is WPA2 protected. The evil twin is configured with WPA2 and working on the same channel as LAP.

A. Attack Method

The WPA2 security protocol in enterprises uses the EAPOL-based authentication for WiFi clients. EAPOL implements a 4-way handshake for generating the PTK at both wireless client and LAP, as depicted in Fig. 13.

The first EAPOL message is generated by AP, contains the ANonce and the replay counter is set to zero. The replay counter is useful to match each pair of messages between LAP and client and to differentiate the re-transmission of any handshake message. After receiving the message-1, the client sends message-2, which contains the SNonce and the same replay

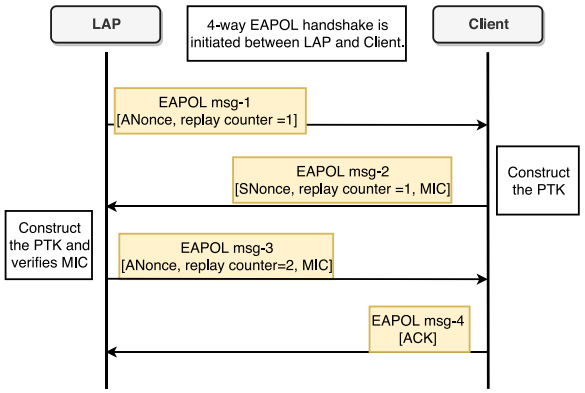


Fig. 13. WiFi standard protocol WPA2: 4-way EAPOL handshake.

counter of message-1. After receiving the ANonce and SNonce at client and LAP, they can generate the PTK, which is used to generate further keys and temporal keys for data encryption/decryption. According to the WPA2 protocol, a LAP can retry the handshake for three times. For every retry, the AP increments the value of replay counter by one. When LAP and evil twin operate on the same channel, the initial association request before the EAPOL authentication handshake is heard by both. Therefore, EAPOL message-1 is generated by both and received at the client. The client generates and sends the EAPOL message-2 for the most recent message-1 received according to the WPA2 protocol standard. Both the LAP and evil twin receive message-2 containing the SNonce of a client and verify that the MIC included in the message-2 is corresponding to their message-1 or not. If the verification fails, the LAP or evil twin discards message-2 and re-transmits message-1 with an incremented replay counter. This creates a race condition among LAP, evil twin and client. Every time during the EAPOL handshake, two scenarios are possible: a) LAP receives the EAPOL messages generated by the client for the reply of evil twin's EAPOL messages, or b) client receives re-transmission of message-1. Moreover, both scenarios force the handshake again and the EAPOL handshake cannot be completed. Therefore, LAP is not able to serve any new client in its vicinity.

Consequently, the attacker blocks the services of LAP in the WPA2 enabled WiFi by launching the evil twin in the LAP's serving channel. Due to a continuous disturbance in the 4-way EAPOL handshake between LAP and client, they are not able to continue with the communication process. Although somehow the handshake is completed between LAP and client, the evil twin still re-transmits EAPOL handshake messages and the client initiates the handshake process again.

B. Severity of AP Service Blocking Attack

There are various ways to launch a DoS attack on an AP. One method is to target LAP's limited resources, such as memory and processing power; however, trivial monitoring solutions can detect them. If we try to victimize a LAP using de-authentication attack, we believe attacking a LAP in a WPA2 protected WiFi requires stealing the client's MAC address. In a WPA2 protected WiFi network, stealing MAC

addresses of clients requires ARP spoofing attack or some other means, which may be costly and alarming to the network operator. Hence, DoS using the de-authentication attack on LAP is difficult and does not guarantee the total service blocking. Moreover, APSB can easily block LAP to associate with new clients. Total service blocking is difficult as the services to current associated clients are not affected. However, we believe that the detection of APSB is difficult. An attacker can perform more severe attacks, i.e., total service blocking if it exploits the de-authentication attack and APSB, both in protected WiFi.

C. Detection/Prevention

Detection of APSB attack is not trivial because the AP discards the authentication message if it is unable to verify the EAPOL message-2/message-4 sent by a client and then retries the handshake. The corruption of messages during transmission is highly probable in wireless, thus the AP is unable to detect the abnormal behavior from the corrupted messages. The possible approaches towards detection/prediction of the APSB attack can be:

a) *AP/Network side prediction:* while observing the scenario of EAPOL handshake disruption, we identified that it could also be used as a basic prediction approach. The EAPOL handshake is initiated after the association process between the AP and the client. Therefore, disruption in EAPOL authentication mechanism is a rare scenario and only occurs due to interference and intermediate packet drop in wireless. However, the APSB attack introduces the continuous disruption of EAPOL handshake for every client and AP pair, which indicates that the attack is happening and can be utilized for the attack prediction;

b) *The client-side detection:* in the presence of WPA2 enabled evil twin in WLAN, the wireless client receives two messages of type EAPOL message-1 from the same AP. Comparisons between ANonces and replay counters of both messages reveal the presence of the evil twin. An AP sends EAPOL message-1 again only if it does not receive the EAPOL message-2 from a client. Therefore, after a timeout, the AP re-transmits message-1 with the incremented replay counter value. An AP can try three re-transmissions before quitting the handshake according to the WPA2 protocol. *Therefore, if the client finds the two messages of type EAPOL message-1 from the same AP with different ANonce values and zero values of replay-counters, it indicates an adversary is impersonating the LAP's identity and interrupting the EAPOL handshake.*

c) *Limitations:* Due to the unavailability of the dataset of the WiFi WPA2 EAPOL handshake, we could not define the prediction-based APSB attack detection. The deployment of client-based detection is out of the scope of this work, but it can be an interesting direction to be further explored.

V. EvilScout System Architecture

This section presents the EvilScout security framework, a mechanism for detecting and mitigating evil twin existing in WiFi networks with the help of the duplicate association phenomenon. EvilScout uses the SDN WiFi deployment to implement the detection mechanism. The reason behind using SDN WiFi framework is that it provides centralized control to all the APs deployed in the WiFi network and it eases the process of monitoring the network. EvilScout monitors the communication packet at the SDN controller and uses the information about IP-prefix distribution for detecting the evil twin. Due to duplicate association, both LAP and evil twin listen to the communication from the victim clients. EvilScout checks the IP address of the packets received with IP-prefix manager present at the SDN WiFi controller. If the IP address of the client is not consistent with the IP-prefix manager information, the presence of the evil twin is detected.

TABLE IV
IP Record to Host Information

IP	Host MAC	Switch-Port	LAP BSSID
10.0.0.12	02:00:00:00:02:10	DPID-a port-x	xx:00:00:00:00:01
10.0.0.10	02:00:00:00:02:11	DPID-b port-y	xx:00:00:00:00:02
10.0.0.13	02:00:00:00:02:12	DPID-a port-z	xx:00:00:00:00:03
10.0.0.11	02:00:00:00:02:13	DPID-d port-w	xx:00:00:00:00:04

A. EvilScout System Design

EvilScout utilizes the SDN controller modules to implement the attack detection technique and communicates through OpenFlow protocol with the data plane devices (i.e., OpenFlow-enabled AP) to monitor the client's packet. EvilScout consists of the following modules to interact with APs and to analyze the network events for the attack detection:

a) *Event Handler:* handles all the OpenFlow events and transfers them to other EvilScout modules accordingly. It forwards PacketIn messages generated due to the absence of corresponding flow rules at the AP to the packet analyzer module. It also redirects all the IP address request packets for further processing to address assignment module (AAM);

b) *Packet Analyzer:* receives the PacketIn messages forwarded by the event handler. It extracts the switch-port information (i.e., the source switch DPID and port number) from the PacketIn header. The switch-port information is forwarded to the IP-prefix manager for host state management. It also provides the packet headers as input to the attack detector module for identification of the evil twin attack;

c) *IP-Prefix Manager:* EvilScout interacts with AAM and manages the IP-prefix information. IP-prefix manager records the leased IP addresses from the IP-prefix domain, such as DHCP. It also maintains a mapping table among the assigned IP and host information. The host information includes MAC address and location (switch-port) of the host, as shown in Table IV.

d) *Attack Detector:* analyzes the IP-prefix distribution and discovers the evil twin presence in SDN WiFi. It implements the detection Algorithm 1 at the control plane. The attack detector receives the packet headers from the packet analyzer and investigates them for detection of the evil twin. The detailed examination of the packet headers is performed by the attack detector to investigate the inconsistency in IP-prefix assignment using queries to the IP-prefix manager.

Algorithm 1 Evil Twin Detection Algorithm

Input: PacketIn, Ingress Legitimate Access Point (LAP)
Output: Evil Twin Detected (True/False)

PacketIn event from access point $LAP1(pk_1)$.
Record switch-port$(pk_1) \rightarrow SP_{pk1}$,
Extract Source IP (ip_s) from IP header of pk_1.
if $(ip_s \notin$ IP-prefix) **then** # Case(1)
  return True
else # If Evil Twin is smart.
  if (ip_s not allotted) **then** # Case(2)
    return True
  else if ip_s is in use **then**
    From table extract switch-port$(ip_s) \rightarrow SP$
    if $SP! = SP_{pk1}$ **then** # Case(3)
      return True
    else # Case(4)
      Extract Source MAC address (mac_s) from Ethernet header of pk_1.
      From table Extract MAC$(ip_s) \rightarrow mac_{ip}$
      if $(mac_{ip}! = mac_s)$ **then**
        return True #Multiple MAC
      else
        return False
      end if
    end if
  end if
end if

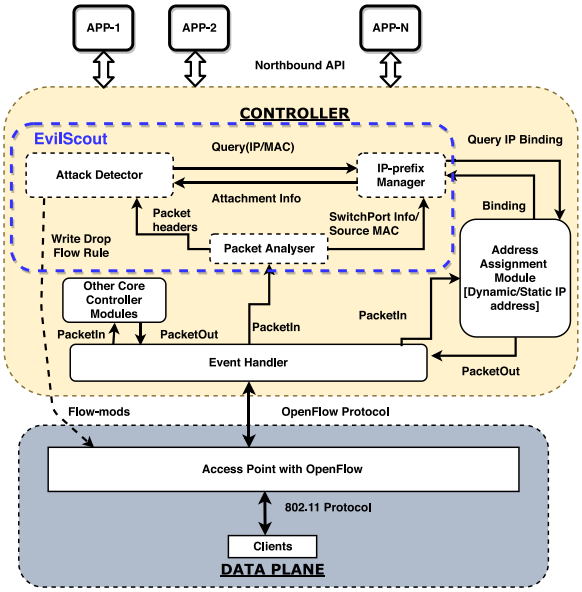


Fig. 14. EvilScout System Architecture.

B. Attack Detection Mechanism

The evil twin can be of two categories: one using LAP for backend Internet connectivity, and the other using the different Internet connections. EvilScout detects the presence of both categories of evil twin in the WiFi network. The basic idea is to find the IP addresses which are not assigned by the LAP but are still in use by the wireless clients.

As we discussed in Section III, although the client is connected to the evil twin, client's communication packet is also received at LAP due to the duplicate association. LAP forwards these packets as PacketIn messages to the controller due to the unavailability of the flow rule. These PacketIn events at the controller trigger the EvilScout's detection mechanism. The attack analyzer finds the IP addresses which are not distributed by the AAM of the controller or allocated to some other LAP (i.e., switch-port). The IP-prefix manager module manages the IP-prefix distribution in SDN WiFi network, as shown in Table IV. The purpose of recording IP-prefix allotments is to identify the signature of the evil twin's presence in the network. IP-prefix manager keeps recording the MAC address of clients, the LAP to which the client is associated and the LAP's location (assigned switch-port). Every IP address can be assigned to a client or unassigned, which helps to identify the inconsistency and indicates the evil twin's presence.

C. EvilScout's Detection Algorithm

There can be four possible cases of IP address distribution when an attacker launches the evil twin in a WiFi network, as discussed in Algorithm 1.

Case(1): if the EvilScout detects a source IP address that does not belong to the predefined IP-prefix of the WiFi network, it means the attacker is distributing different IP-prefixes (i.e., default allocation through the NAT running at the evil twin). However, if the attacker assigns the similar IP-prefix by NAT as LAP, then case(2) happens;

Case(2): if the EvilScout detects a source IP address that is still unassigned by AAM, it indicates that someone else is distributing the IP address on behalf of LAP in the WiFi network;

Case(3): if the IP address is assigned by AAM, but to the client associated with a different LAP than the ingress LAP (PacketIn's switch-port), it indicates someone else distributed the same IP address to another client.

There is a possibility that the IP address is assigned by the ingress LAP itself, then the single IP address assignment to multiple MAC addresses enables the detection. Hence, we have to check the source MAC address of the PacketIn message (i.e., case(4)).

Case(4): if the PacketIn's source MAC address is different from the available binding of client's IP address and MAC address, then multiple MAC addresses are designated to a single IP address. This indicates the IP address allocation anomaly.

All these cases examine the evil twin's signature by monitoring the LAP and client communication. After the evil twin detection, the EvilScout sends a drop FlowMod as PacketOut to the victim LAP, instructing it to drop all the packets of the respective client's MAC address. Thus, the controller does not need to further process any packets of victim clients.

VI. Proof of Concept (PoC) and Implementation

A. Hardware-Based PoC Implementation Details

The PoC implementation is depicted in Fig. 15. We used the Floodlight SDN controller [22] and integrated the EvilScout

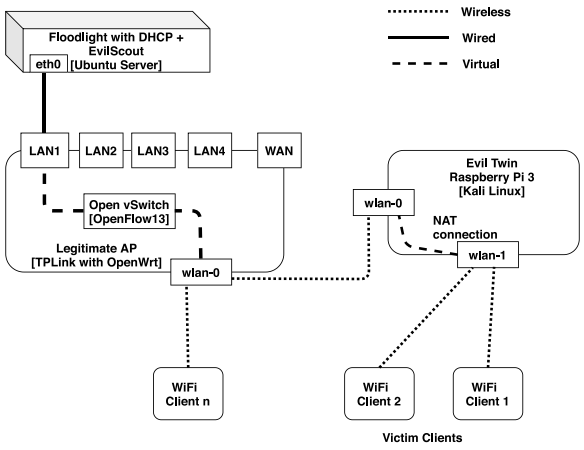


Fig. 15. PoC implementation of EvilScout at SDN WiFi testbed.

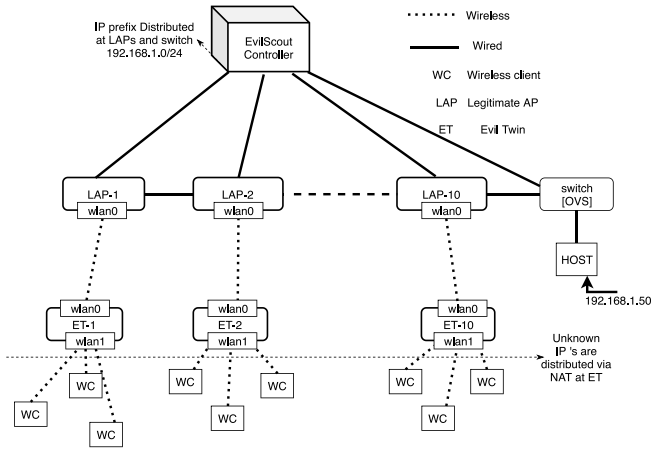


Fig. 16. Mininet-WiFi Topology.

TABLE V
TESTBED IMPLEMENTATION DETAILS

Modules	Software	Hardware
Controller	Floodlight master version [22] JAVA Openjdk 1.8	Intel- Xeon CPU 2.6 Ghz, 12cores, RAM 32Gb
Access Point	OpenWrt[23] Open vSwitch [24] OpenFlow13	TP-Link-1043ND
Evil Twin	Kali Linux [25] Hostapd[1]	Raspberry Pi 3 and USB WiFi adapter Eastech Ralink-RT5370

framework in the Floodlight code. The DHCP server is a module running at the controller and used to deploy the attack detection logic. The LAP is a TP-Link hardware AP flashed with OpenWrt-18.06.1 and Open vSwitch(OVS)-2.8.2 installed. The evil twin is at Raspberry Pi 3 running Kali Linux and has two wireless NICs (wlan0 and wlan1).

Interface wlan0 operates as a WiFi client and connects to LAP. Interface wlan1 runs the hotspot to impersonate the LAP's identity. LAP and evil twin both are tuned into WiFi channel-6.

Implementation details of different entities of the testbed are discussed as follows:

a) *EvilScout deployment:* the EvilScout prototype is deployed over the master version of the Floodlight controller. The packet analyzer implements the IFloodlight module and registers to the IOFMessageListener to receive the PacketIn messages. We utilize the DHCP service of the controller to implement the attack detector. IP-prefix manager obtains the information of the IP addresses issued by DHCP and stores it in the host information, as seen in Table IV. The attack detector analyzes the packet headers with the information provided by the IP-prefix manager, and the controller flags the presence of the evil twin if it exists in the network.

b) *SDN WiFi AP (LAP):* we used the TP-Link AP as the LAP in our experimental setup. The original firmware is replaced by the OpenWrt firmware along with Open vSwitch installation to provide the capabilities of OpenFlow [26] based communication. The wireless interface is added as a port in the OVS bridge. This allows the controller to control the wireless interface of the AP. All the DHCP requests and the communication packets are sent to the controller. The controller can instruct the APs by appropriate flow rules installation using FlowMod messages.

c) *Evil twin deployment:* we implemented the evil twin on a Raspberry Pi 3 with Kali Linux and a USB WiFi adapter (chipset RaLink rt5370). We used hostapd to operate one of the WiFi NIC as AP, which spoofed the SSID and BSSID of the LAP, and launched the evil twin on the same channel as the LAP.

B. Software-Based PoC Implementation Details

Due to limited availability of hardware, experiments on multiple-LAP could not be completed on a hardware-based testbed. To overcome this, we emulated the WiFi topology using mininet-WiFi.

The EvilScout is deployed in FloodLight SDN controller. The experiment uses a linear topology (Fig. 16) of 10 LAPs configured at different channels. Each LAP is targeted by a different evil twin. We created an evil twin by configuring the mininet-WiFi station with two WLAN interfaces (NICs). One interface (wlan1) works as AP to perform the attack and the other (wlan0) connects to LAP as client. All 10 evil twins implemented NAT. Some of the evil twins distribute the same IP-prefix as controller's DHCP and some distribute different IP-prefixes. We also deployed a host in the LAN. This host is used as a gateway to ping in the local network of the mininet. The DHCP running on controller distributes the IP-prefix 192.168.1.0/24 to all the clients connected to the SDN WiFi.

VII. EVALUATION

The Evilscout's detection capabilities and overhead in SDN controller for an attack on a single LAP are evaluated using hardware-based PoC. In the case of evil twin attacks on multiple LAP, questions regarding scalability and feasibility of EvilScout arise. To answer these questions, we performed evaluations on software-based PoC.

```
15:06:26.769 WARN [n.f.l.i.L.s.notification:nioEventLoopGroup-3-1] Link added: Link [src=00:00:70:4f:57:b2:69:4a ou
15:06:26.771 WARN [n.f.l.i.L.s.notification:nioEventLoopGroup-3-2] Link updated: Link [src=00:00:70:4f:57:b2:69:4a
15:06:26.844 INFO [n.f.j.JythonServer:debugserver-main] Starting DebugServer on :6655
15:06:35.705 INFO [n.f.d.DHCPServer:pool-1-thread-11] Cleaning any expired DHCP leases
15:06:37.491 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-1] Evil Twin Dettected for AP MAC:70:4f:57:b2:69:4a
15:06:37.491 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-1] Time to detect: 0.076141 msec
15:06:37.491 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-1] Drop Flow Rule
15:06:37.506 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-2] Evil Twin Dettected for AP MAC:70:4f:57:b2:53:48
15:06:37.508 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-2] Time to detect: 0.059073 msec
15:06:37.508 INFO [n.f.d.DHCPServer:nioEventLoopGroup-3-2] Drop Flow Rule
15:06:40.761 INFO [n.f.l.i.LinkDiscoveryManager:Scheduled-0] Sending LLDP packets out of all the enabled ports
15:06:55.767 INFO [n.f.l.i.LinkDiscoveryManager:Scheduled-4] Sending LLDP packets out of all the enabled ports
15:07:10.771 INFO [n.f.l.i.LinkDiscoveryManager:Scheduled-1] Sending LLDP packets out of all the enabled ports
```

Fig. 17. EvilScout's Floodlight controller log: detection of the Evil Twin.

TABLE VI
OVERHEAD INTRODUCED BY EVILSCOUT AT FLOODLIGHT CONTROLLER

Modules	Average delay overhead
Attack Detector	0.053ms/PacketIN
Write drop Flow rule	0.29ms/PacketIN

A. Experimental Evaluation Using Hardware-Based PoC

1) Detection of Evil Twin in Real Testbed: we integrated the EvilScout's attack detection logic (Algorithm 1) in the DHCP server module of Floodlight controller. The experiment is performed by creating the evil twin of a LAP with a Raspberry Pi. The EvilScout successfully detected the evil twin, as shown in Fig. 17.

2) Impact on SDN Controller's Performance: the overhead introduced by EvilScout on the SDN (Floodlight) controller is mostly due to PacketIn message processing by packet analyzer and attack detector modules. IP address assignment service running on the SDN controller is not considered as overhead as it is a part of the controller's core operations. After detection of the evil twin, EvilScout writes a FlowMod at the LAP to drop the packets match with the source and destination IP addresses contained in the PacketIn header, to block the traffic relayed by the evil twin. Table VI shows the overhead of EvilScout in Floodlight. EvilScout needs to process PaketIn messages before the forwarding module installs the flow rules. Hence, it introduces some time delay in the processing of each PacketIn at the controller, which is considered as overhead.

The average overhead generated by EvilScout from detection to mitigation is 0.343ms, which is 0.021% of the overall controller processing triggered by PacketIn. Therefore, the integration of EvilScout at Floodlight does not introduce considerable overhead.

B. Experimental Evaluation Using Software-Based PoC With Mininet-WiFi

1) Scalability Analysis of EvilScout:

a) Packet processing overhead: EvilScout detects the evil twin's presence by inspecting the packets of victim WiFi clients. To block the evil twin's traffic from the WiFi, EvilScout sends the FlowMod message with drop action (i.e., drop flow rule) to the LAP. The delay introduced by the EvilScout includes examining the PacketIn messages and writing the drop FlowMod at the corresponding LAP. We define T_{es} as the total time of detection and blocking of the evil twin. During T_{es}, the packets generated by victim clients are sent to the controller. These incoming packets occupy the controller channel and computational resources. Thus, these additional PacketIns during T_{es} should be considered as overhead. We

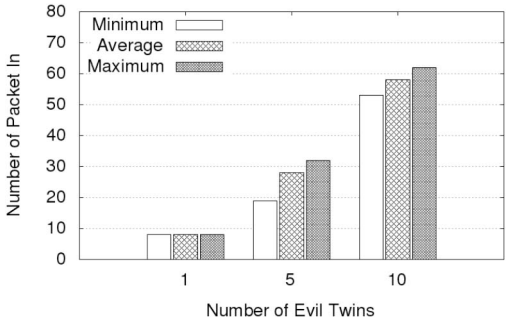


Fig. 18. PacketIn Overhead at EvilScout before blocking the traffic of Evil Twin at WLAN.

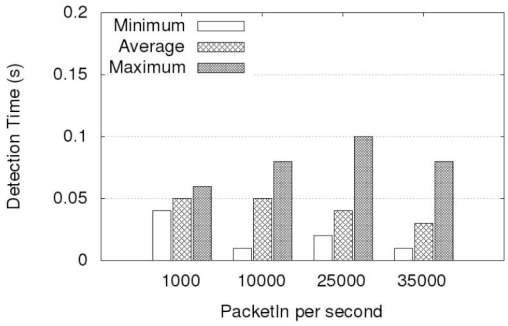


Fig. 19. Evil Twin detection time from receiving the PacketIn to write the flow rule at AP.

reach the worst case when evil twin attacks on multiple LAPs are performed simultaneously, and all the associated victim clients are sending continuous traffic.

We performed the experiment with mininet-WiFi setup by launching 1, 5, and 10 evil twins, each one attacking a different target LAP. All the victim clients connected to evil twins generate traffic using the ping tool. We deployed a host in the mininet topology as a target ping server (Fig. 16). Fig. 18 summarizes the generated PacketIn overhead on SDN controller.

We found EvilScout to be capable of detecting multiple evil twins within some milliseconds and PacketIn overhead to be linear in the number of attacks and victim clients. Hence, EvilScout has the potential to be a scalable and efficient solution against evil twin attacks.

b) Detection accuracy: we test the EvilScout detection accuracy in different scenarios: i) an evil twin shares the same IP-prefix with LAP and ii) the IP-prefix is different. In both cases, the EvilScout successfully detects the evil twin's presence on the same channel as LAP.

c) Stress test on EvilScout: we measure the response time of EvilScout while varying load at the controller to test the system performance in the stressed scenario. We generate 1000 to 35000 PacketIn messages per second by utilizing the DHCP-test tool and launch the evil twin attack at the same time to test the EvilScout's response in a loaded situation. Fig. 19 shows the detection time taken by the EvilScout while varying PacketIn load.

The result of the stress test shows that the detection of evil twin takes only 0.1s in most of the cases, even if a large number of PacketIn messages is handled by EvilScout.

TABLE VII
COMPARISON OF EXISTING EVIL TWIN DETECTION TECHNIQUES AND EVILSCOUT

Technique	**Detection Approach**	**Depends upon (Client and/or Server)**	**Hardware Modification require**	**Software Modification require**	**Issues**
RTT measurement [12] [29]	Measure the RTT to a local DNS server and Time to set up an HTTP connection to the public server.	Both	Yes	Yes	Only evil twins that relay traffic through LAP are detected. Client-side specialized software is required.
Inter-packet arrival time [31] [11]	Calculates the two-hop wireless connection by observing consecutive packet arrival time from one TCP connection.	Both	Yes	Yes	Only evil twins that relay traffic through LAP are detected.
Crowd Sensing [10]	Utilize the RSS measurements of crowd mobile devices.	Client	No	Yes	Data collection through multiple clients is needed; Malicious clients can tamper their measurements.
Channel interference [13]	Intentionally generate interference to all the channels and detect simultaneous disruption if a fake AP relays traffic.	None	Yes	Yes	It cannot effectively detect RAPs, that launched on the same channel as the LAP and did not relay traffic through LAP.
RF signal strength [15]	Measure the distance from AP using received signal strength at the client.	Client	No	Yes	High false positive detection due to the assumption of distant AP must be fake.
Channel state information (CSI) [9] [32]	Device fingerprinting by estimating its Carrier Frequency Offset using CSI values.	Client	No	Yes	Calculation of carrier frequency offset (CFO) over client devices is required.
Time Synchronization Function(TSF) [5] [33]	Analyze the TSF value of beacon frames to fingerprint the h/w and s/w APs.	Client	Yes	No	TSF can be spoofed by device driver modification.
EvilScout (Proposed in this paper)	Detects the inconsistency of assigned IP addresses.	None	No	No	Evil twins on different channels cannot be detected.

C. Complexity Analysis of EvilScout

Following terms are used for complexity analysis:

- n is the number of wireless clients in the SDN WiFi;
- e is the number of evil twin attackers present in SDN WiFi.

Space Complexity: EvilScout only has to manage the IP addresses distributed in the SDN WiFi. Thus, the memory requirement is bounded by $O(n)$. EvilScout only writes one drop flow rule on the target LAP, thus requires only $O(1)$ memory in the LAP.

Time Complexity: EvilScout requires only one PacketIn to be analyzed for one attack. Thus, for a single evil twin attack, the detection time is bounded by $O(1)$, and for multiple attacks, it is bounded by $O(e)$.

VIII. RELATED WORK

Evil twin attack is trivial to perform as many open-source tools [1], [2], [25], [27] are freely available, and almost all mobile devices are equipped with required hardware such as WiFi NIC. Even the equipment needed to perform the attack is inexpensive. Hence, an adversary can steal sensitive information of clients as MITM, and it can perform other attacks at AP or clients. Because of the attack severity, many evil twin detection approaches are studied in a traditional WiFi network.

A. Generalized Approaches

In WiFi network, most of the evil twin detection techniques are based on the fingerprint of any fake AP in the WiFi with parameters such as a device driver [27], [28], clock skew with time synchronization function (TSF) [14], [5], timing analysis [29] and channel statistics. Lanze *et al.* [14], identified the presence of evil twin by the clock skew of the AP combined with temperature information. This idea is an improvement over the concept of software-based evil twins identification based on TSF timer values in the beacon frames [5]. This requires the calculation of differences between the TSF of beacons received and the actual reception time at the client for spotting the software-based evil twins. If the evil twin is launched with hardware-based AP [13], this detection mechanism fails. Some timing analyses are presented to find the evil twin, such as estimation of inter-packet arrival time (IAT) [11], [12] in the same TCP flow between the client and a server or calculation of RTT [29] to a server to predict evil twin's presence in the network. They exploit the two wireless hop delays introduced by evil twin's presence. However, the delay could involve other factors such as a load at AP or backbone network [13]. In addition, these timing analyses could not detect evil twins that do not relay the traffic via LAP. Furthermore, Jang *et al.* [13] proposed a new hardware-based evil twin called "PrAP," which does not introduce a significant delay to communication. They also presented the detection of PrAP using channel overlapping in 802.11n. However, they could not find the evil twin on the same channel as LAP. More schemes further exploit the received signal strength (RSS) [10] collected by many clients to identify the potential adversary in the WiFi network, but some of the clients can intentionally tamper the RSS values, which causes wrong detection. Some approaches utilize the channel state information to find the

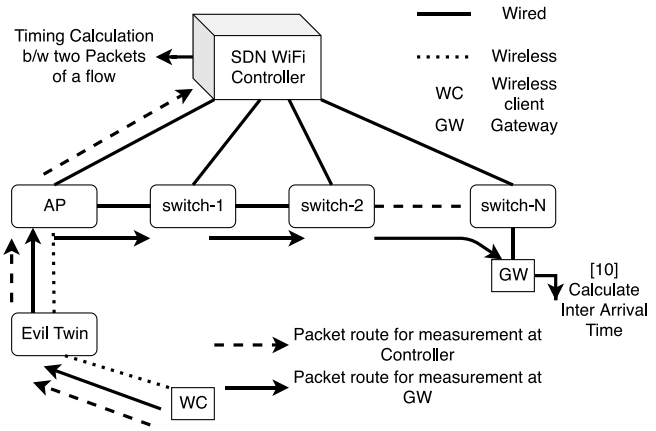


Fig. 20. Timing analysis to detect Evil Twin.

evil twin, by calculating the channel interference [13] as more interference in wireless can be an indication of the evil twin's presence.

B. SDN WiFi Approaches

We are first to explore the SDN WiFi architecture for evil twin detection. Few proposed works investigated the detection techniques for DoS attacks [30] in SDN WiFi, and [16] discussed the unauthorized rogue AP detection by combining the existing approaches in SDN network. However, none of them include evil twin detection. We discovered the duplicate association and used it to exploit the SDN WiFi to spot evil twin's presence in the network.

C. Comparison of Evil Twin Detection Mechanisms and Their Limitations

Recently, proposed evil twin detection mechanisms are compared according to the following measures:

- Client and/or Server: detection is performed only by the client or requires any other server in the network;
- Hardware modification: special hardware is required to implement the detection mechanism;
- Software modification: a software update is needed to deploy the detection approach;
- Issues: discuss the scenario where detection is not possible by its respective approach.

We discussed the various research proposals and compared them on the bases of the mentioned measurement parameters, presented in Table VII. Researchers explored many techniques, but most of them are impractical as they require additional hardware to monitor the WiFi channel or software update at the client-side. Some of these techniques also require changes in the 802.11 protocol. These techniques are not sufficiently generalized to protect the WiFi network from evil twin attacks.

D. Summary

Evil twin attacks are categorized by its launching strategies. Evil twin relays the traffic via LAP or a third-party Internet connection, and it works on the same channel as LAP or a different channel. Existing approaches are capable of detecting evil twin in limited scenarios, and nearly all are prediction-based. However, the duplicate association allows the EvilScout to detect the evil twin on the same channel. All other approaches require the passive collection of parameters either with the help of additional hardware devices to detect the adversary in WiFi network or over the client to differentiate between LAP and evil twin. Most of the detection techniques require additional capabilities over clients, which contradicts the BYOD notion of WiFi. However, EvilScout supports the BYOD by enabling the network side detection with no additional hardware requirement and transparency with clients.

IX. Discussion

A. EvilScout's Limitations

EvilScout does not detect the evil twin on a different WiFi channel from the targeted LAP, because it exploits the duplicate association, which only happens on the same channel. The duplicate association cannot be possible as LAP does not listen to the evil twin's and client's communication when tuning to another channel.

However, the EvilScout framework can help efficiently deploy some existing solutions for evil twin detection in a different channel. For example, the measurement of IAT [11] could be more accurate if calculated between the controller and LAP as it involves the direct one hop dedicated channel between the controller and the LAP, as shown in Fig. 20. The IAT calculation between the packets of a TCP flow is performed at a gateway server [11], which includes other factors that affect IAT, such as varying traffic conditions and the number of hops in between LAP and gateway.

B. EvilScout in WLAN Multi-Channel Scenario

In the enterprise WLAN deployment, LAPs are configured at multiple channels and support periodic channel switching according to channel conditions. In these multi-channel deployments of WLAN, EvilScout detects the evil twin if it is working on the same channel as it's victim LAP. However, the detection of the evil twin attack performed on different channels from the victim LAP is a potential extension.

C. Efficient Designing of Security Framework

Designing an efficient security framework requires the intelligent segregation of tasks managed by AP and controller. Offloading all tasks of AP to the controller is not practical as it requires large computation and introduces the infeasible delay in the processing of WLAN Layer 2 protocol. Security mitigation approaches need more in-depth monitoring of the WLAN conditions such as client's bandwidth usage, number of clients currently served by an AP, AP channel conditions. To design an efficient controller-based security framework, one should consider:

- That the solution should not produce many control traffics between the controller and the AP;
- The controller resource consumption should be minimized;
- Recovery after the attack should be fast.

D. Security Concerns of SDN WiFi Architecture

SDN WiFi architecture allows the developer to integrate new functionalities, but it can introduce new vulnerabilities, such as:

- As the controller is the core of SDN WiFi, there should not be a miss-configuration in its design/implementation. Otherwise, the controller could be compromised;
- SDN WiFi manages a dynamic network topology view using the OpenFlow protocol, which can be tampered by malicious event injections through the data plane. This requires more protection of message exchange between controller and data plane devices.

However, these issues are the inherent concerns of the SDN architecture. Few effective countermeasures are already proposed [34], [35], [36].

X. Conclusion & Future Work

In this paper, we proposed a novel evil twin detection and mitigation technique. Our goal was to provide a lightweight security framework that removes the dependency over wireless clients and operators. We conducted a rigorous analysis of evil twin attacks in the different WiFi setups on a real testbed, enabling a thorough understanding of the behavior of WiFi APs and clients affected by the evil twin attack. We discovered the concept of "Duplicate Association" and utilized it for the detection of evil twin attacks. We encountered a severe attack called APSB in the WPA2 protected WiFi. We utilized the capabilities of SDN to deploy the EvilScout framework. EvilScout was deployed and its performance and scalability were analyzed in a real SDN WiFi testbed. Experimental results showed that EvilScout successfully detects the evil twin on the same channel as LAP. EvilScout is lightweight in terms of overhead at the controller, and the detection is fast as it only needs to verify one packet of any victim client. This study also provides an in-depth understanding of the APSB attack and the possible detection approaches.

In the future, we will extend the EvilScout framework with the integration of APSB attack detection and mitigation, in addition to the detection of evil twin's presence on different channels from LAP. Furthermore, we will explore the EvilScout in the direction of discovering more attacks in WiFi by utilizing the SDN features such as centralized control management of network traffic and monitoring of network state.

References

[1] J. Malinen *et al.* (Feb. 2014). *hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator, Version 2.1*. [Online]. Available: http://hostap.epitest.fi/hostapd/

[2] T. Dótreppe. (2013). *Aircrack-ng*. [Online]. Available: http://www.aircrack-ng.org

[3] A. Bartoli, E. Medvet, and F. Onesti, "Evil twins and WPA2 enterprise: A coming security disaster?" *Comput. Security*, vol. 74, pp. 1–11, May 2018.

[4] B. Antoniewicz *et al.* (2019). *Eap Hamper*. [Online]. Available: https://github.com/s0lst1c3/eaphammer

[5] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Undesired relatives: Protection mechanisms against the evil twin attack in IEEE 802.11," in *Proc. 10th ACM Symp. QoS Security Wireless Mobile Netw.*, 2014, pp. 87–94.

[6] N. Sombatruang, M. A. Sasse, and M. Baddeley, "Why do people use unsecure public Wi-Fi?: An investigation of behaviour and factors driving decisions," in *Proc. 6th Workshop Socio Tech. Aspects Security Trust*, 2016, pp. 61–72.

[7] G. I. Blog. (2015). *Google Station*. [Online]. Available: https://www.blog.google/technology/next-billion-users/400-train-stations-india/

[8] N. Sombatruang, L. Onwuzurike, M. A. Sasse, and M. Baddeley, "Factors influencing users to use unsecured Wi-Fi networks: Evidence in the wild," in *Proc. 12th Conf. Security Privacy Wireless Mobile Netw.*, 2019, pp. 203–213.

[9] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2018, pp. 1700–1708.

[10] T. Zhou, Z. Cai, B. Xiao, Y. Chen, and M. Xu, "Detecting rogue ap with the crowd wisdom," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 2327–2332.

[11] C. Yang, Y. Song, and G. Gu, "Active user-side evil twin access point detection using statistical techniques," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 5, pp. 1638–1651, Oct. 2012.

[12] H. Mustafa and W. Xu, "CETAD: Detecting evil twin access point attacks in wireless hotspots," in *Proc. IEEE Conf. Commun. Netw. Security*, 2014, pp. 238–246.

[13] R. Jang, J. Kang, A. Mohaisen, and D. Nyang, "Rogue access point detector using characteristics of channel overlapping in 802.11n," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 2515–2520.

[14] F. Lanze, A. Panchenko, B. Braatz, and T. Engel, "Letting the puss in boots sweat: Detecting fake access points using dependency of clock skews on temperature," in *Proc. 9th ACM Symp. Inf. Comput. Commun. Security*, 2014, pp. 3–14.

[15] B. Pradeepkumar, K. Talukdar, B. Choudhury, and P. K. Singh, "Predicting external rogue access point in IEEE 802.11b/g WLAN using RF signal strength," in *Proc. Int. Conf. Adv. Comput. Commun. Inf. (ICACCI)*, 2017, pp. 1981–1986.

[16] J. H. Cox, R. Clark, and H. Owen, "Leveraging SDN and WebRTC for rogue access point security," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 756–770, Sep. 2017.

[17] R. Jang, J. Kang, A. Mohaisen, and D. Nyang, "Catch me if you can: Rogue access point detection using intentional channel interference," *IEEE Trans. Mobile Comput.*, early access, doi: 10.1109/TMC.2019.2903052.

[18] B. Alotaibi and K. Elleithy, "Rogue access point detection: Taxonomy, challenges, and future directions," *Wireless Pers. Commun.*, vol. 90, no. 3, pp. 261–1290, 2016.

[19] K. Gao, C. Corbett, and R. Beyah, "A passive approach to wireless device fingerprinting," in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2010, pp. 383–392.

[20] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, 2007, pp. 365–378.

[21] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Amsterdam, The Netherlands: Elsevier, 2006.

[22] R. I. *et al.* (2012). *Floodlight*. [Online]. Available: http://floodlight.openflowhub.org/

[23] F. Fainelli, "The openwrt embedded development framework," in *Proc. Free Open Source Softw. Dev. Eur. Meeting*, 2008.

[24] B. Pfaff *et al.*, "The design and implementation of open vSwitch," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2015, pp. 117–130.

[25] L. Allen, T. Heriyanto, and S. Ali, *Kali Linux–Assuring Security by Penetration Testing*. Birmingham, U.K.: Packt Publ., 2014.

[26] N. McKeown *et al.*, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[27] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Hacker's toolbox: Detecting software-based 802.11 evil twin access points," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, 2015, pp. 225–232.

[28] J. Franklin *et al.*, "Passive data link layer 802.11 wireless device driver fingerprinting." in *Proc. USENIX Security Symp.*, 2006, pp. 16–89.

[29] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, Nov. 2011.

[30] R. Cwalinski and H. Koenig, "SDN-based attack detection in wireless local area networks," in *Proc. 4th IEEE Conf. Netw. Softw. Workshops (NetSoft)*, 2018, pp. 207–211.
[31] Y. Song, C. Yang, and G. Gu, "Who is peeping at your passwords at starbucks?—To catch an evil twin access point," in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, 2010, pp. 323–332.
[32] H. Liu, Y. Wang, J. Liu, J. Yang, and Y. Chen, "Practical user authentication leveraging channel state information (CSI)," in *Proc. 9th ACM Symp. Inf. Comput. Commun. security*, 2014, pp. 389–400.
[33] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mobile Comput.*, vol. 9, no. 3, pp. 449–462, Mar. 2010.
[34] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in *Proc. NDSS Symp.*, vol. 15, 2015, pp. 8–11.
[35] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "Sphinx: Detecting security attacks in software-defined networks," in *Proc. NDSS Symp.*, vol. 15, 2015, pp. 8–11.
[36] S. Gao, Z. Li, B. Xiao, and G. Wei, "Security threats in the data plane of software-defined networks," *IEEE Netw.*, vol. 32, no. 4, pp. 108–113, Jul./Aug. 2018.

Pragati Shrivastava received the B.E. degree in information technology from R.G.P.V. University, Bhopal, India, in 2010, and the M.Tech. degree in computer science from the Indian Institute of Information Technology, Gwalior, in 2013. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad. Her research interests include software-defined networking, network security, and wireless networks.

Mohd Saalim Jamal received the B.Tech. degree in computer science from KNIT, Sultanpur, in 2016. He is currently pursuing the M.Tech. degree with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad. His research interests include network security, software-defined networking, and blockchain technologies.

Kotaro Kataoka received the B.A. degree in environmental information in 2002, and the master's and Ph.D. degrees in media and governance from Keio University in 2004 and 2010, respectively. He is currently an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, a Senior Researcher with Keio Research Institute, SFC, and an Advisor to Tech Japan. His research interest covers Internet architecture, network operation, software-defined networking, network functions virtualization, and blockchain. Since 2001, he has been a member of Asian Internet Interconnection Initiatives project and Widely Integrated Distributed Environment project for research on Satellite Internet and Post-Disaster Networking.