Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it’s important to you. Thank you.
Cyber Attacks on Smart Farming Infrastructure

Sina Sontowski*, Maanak Gupta†, Sai Sree Laya Chukkapalli‡, Mahmoud Abdelsalam§,
Sudip Mittal¶, Anupam Joshi†∥, Ravi Sandhu∗∗

*†Dept. of Computer Science, Tennessee Technological University, Cookeville, Tennessee, USA
†‡Dept. of Computer Science, University of Maryland, Baltimore County, Baltimore, USA
§¶Dept. of Computer Science, Manhattan College, Bronx, USA
∥∗∗Dept. of Computer Science, University of North Carolina Wilmington, NC, USA

*ssontowsk42@students.tntech.edu, †mgupta@tntech.edu, ‡saisree1@umbc.edu, §mabdelsalam01@manhattan.edu,
¶mittals@uncw.edu, ‖joshi@umbc.edu, **ravi.sandhu@utsa.edu

Abstract—Smart farming also known as precision agriculture is gaining more traction for its promising potential to fulfill increasing global food demand and supply. In a smart farm, technologies and connected devices are used in a variety of ways, from finding the real-time status of crops and soil moisture content to deploying drones to assist with tasks such as applying pesticide spray. However, the use of heterogeneous internet-connected devices has introduced numerous vulnerabilities within the smart farm ecosystem. Attackers can exploit these vulnerabilities to remotely control and disrupt data flowing from/to on-field sensors and autonomous vehicles like smart tractors and drones. This can cause devastating consequences especially during a high-risk time, such as harvesting, where live-monitoring is critical. In this paper, we demonstrate a Denial of Service (DoS) attack that can hinder the functionality of a smart farm by disrupting deployed on-field sensors. In particular, we discuss a Wi-Fi deauthentication attack that exploits IEEE 802.11 vulnerabilities, where the management frames are not encrypted. A MakerFocus ESP8266 Development Board WiFiDeauther Monster is used to detach the connected Raspberry Pi from the network and prevent sensor data from being sent to the remote cloud. Additionally, this attack was expanded to include the entire network, obstructing all devices from connecting to the network. To this end, we urge practitioners to be aware of current vulnerabilities when deploying smart farming ecosystems and encourage the cybersecurity community to further investigate the domain-specific characteristics of smart farming.

Index Terms—Smart Farming, Precision agriculture, Security, Cyber-attack, Internet of Things, Denial of Service

I. INTRODUCTION

In recent years, significant progress has been made in the agricultural sector to develop smart farming and precision agriculture technologies [1] [2]. Agriculture industry accounts for 6.4% of the world’s economic production with a total of $5,084,800 million1. Agriculture, food, and related industries contributed $1.053 trillion to U.S. gross domestic product (GDP) in 20172. Therefore, investing in the smart farming ecosystem and adopting new technologies will have a wider impact on the economy. Further, the rapid growth of population has significantly increased the demand for agriculture and food products. Traditional technologies driving the agriculture sector are incapable of meeting this demand and are becoming obsolete. This has also led the agriculture and food production sector to integrate data driven and Internet of Things (IoT) technologies to increase the quantity and quality of agricultural products. Smart Farming can be a possible solution to boost productivity and maintain product quality. There are numerous smart farming use cases [3]–[5] present globally, e.g., a controlled water supply, recording soil moisture at different levels [6] to increase crop yield. Various sensors allow collection of data and can upload it to the cloud. The collected data provides helpful information about varying environmental conditions and allows for a hands-off approach to smart farm monitoring [2]. Figure 1, shows an end to end interaction among various entities involved in the smart farming ecosystem.

As a result of introducing IoT and connected infrastructure to farms, the agriculture sector will develop a dependency on various information systems to manage and improve operations [7]. However, incorporating IoT systems into the agricultural sector amplifies various cyber risks. These risks are currently not sufficiently addressed because of limited investments in cybersecurity by domain specific companies. In addition, the lack of resources and know-how among members of the farming community will aggravate the issue. Smart farms are a target for foreign competitors and threats, which is a concern to the agricultural sector. Cyber attacks on the smart farming infrastructure enables an attacker to remotely control and exploit on-field sensors and autonomous vehicles (tractors, autonomous vehicles, drones, etc.). Potential agricultural attacks can create an unsafe and unproductive farming environment. For example, exploits that have the ability to destroy an entire field of crops, flood the farmlands, over spray pesticides using smart drones, etc. can cause unsafe consumption as well as economic deterioration. Such attacks in a large coordinated manner, also referred to as Cyber-Agroterrorism [7], [8], also have the potential for disrupting the economy of an agriculture-dependent nation. A report released in 2018 by the US Council of Economic Advisors3.

II presents an overview of cybersecurity and related work in smart farming domain. Section III discusses various types of cyber-attacks that could occur in the networking domain deployed on a smart farm. In Section IV, we introduce our deployed architecture and later describe the experimental setup that demonstrates the Wi-Fi deauthentication attack. In Section V, we provide the details of our process for launching a DOS attack and show the experimental results. In Section VI, we present various use case scenarios of this attack. Finally, Section VII summarizes the work and suggests future work.

II. RELATED WORK

Agricultural companies and farmers are moving towards various smart farming practices that rely on IoT devices for a better crop yield. Interconnecting various sensors deployed on the farm and allowing them to communicate through the Internet provides an attack surface. This has led to a rise in cyber-attacks on agriculture sector such as data breaches, denial of service attacks, website defacement, etc. Recently, Gupta et al. [10] highlighted security and privacy issues in the smart farming ecosystem. They presented a multi-layered architecture and identified potential cybersecurity issues in smart farming. Further, their work also illustrated scenarios of specific cyber attacks categorizing them into data, network, supply chain, and other common attacks.

A popular attack named ‘The Night Dragon’ [11] is an example where the attacker could steal a large amount of information from multiple petrochemical companies. Another example is the damage caused to a German steel mill [12] where attackers used spear phishing to gain access to the mill’s office network and plant production systems.

The exponential rise in number of internet connected devices has raised security concerns especially, in the agriculture sector, as farmers will not be able to bear the potential loss and damage to crops. Therefore, at present, securing various sensors in the smart farm ecosystem is a key task for the agriculture sector. The U.S. Department for Homeland Security released a report [13] which emphasizes the importance of precision agriculture (PA) and associated cybersecurity threat and potential vulnerabilities. The report highlights the confidentiality, integrity, and availability model of information security in farming. It defines different technologies involved in smart farming including, on-farm devices, location and remote sensing technologies, machine learning, etc. It also briefly discusses the groups impacted including farmers, livestock producers, and also industries that support or rely on agriculture. This report also discusses hypothetical threat scenarios. Similarly, the security issues that could arise by deploying IoT sensors in the agriculture sector have been clearly elaborated by Jahn et al. [14] and Lopez et al. [15].

Different types of attacks can be executed by attackers, for instance, a denial of service (DoS) attack on a large scale by utilizing various IoT sensors deployed on the smart farm [16]. The Mirai botnet [17] in 2016 is one such example where multiple DoS attacks were launched by exploiting an army of connected smart home devices. Recently, researchers...
from a security firm named Sucuri [18] discovered that a DoS botnet could deliver 50,000 HTTP requests per second. Here, various websites were attacked by performing DDoS attacks. Similar conditions exist in the smart farming ecosystem. Thus, similar attacks are possible in the context of smart farming. Such attacks cannot only disrupt normal functions of different modules in an individual farm, but also can be leveraged to interrupt legitimate cyber services in other domains.

As many IoT related devices are present in each architectural layer of the smart farm ecosystem [10], these are prone to attacks and can be controlled by a central malicious system called Botnet of Things [19]. An army of infected farm IoT devices [20] can easily be used to infect many other networks through different mediums and hence a smart farm may turn out to be an internet of vulnerabilities for cyber criminals. Smart farms devices are not built with security as a concern and even if they did, users usually neglect the basic step of setting adequate cyber security defense mechanisms [10].

According to the Internet Security Alliance (ISA) [21] attacks on agriculture sector are a relatively low cost ventures that would in turn need a deployment of heavy financial resources to defend this ecosystem. Therefore, it is important for the agriculture sector to understand the consequences of cyber attacks and be conscious of the security challenges that can arise due to massive use of internet connected devices in the farming ecosystem. Artificial Intelligence based security methods have become popular in recent years [22], [23]. Securing smart farms using established security frameworks would also provide a solution to the above, like the Smart Farming Access Control (SFAC) system [24]. Here the goal of the system is to help farmers create and enforce access control rules for their smart farms. The authors also discussed various access control scenarios on a smart farm and how rules written in the Semantic Web Rule Language (SWRL) [25] can be used to determine access.

Next, we discuss in detail various types of network attacks on smart farms.

III. TYPES OF NETWORK ATTACKS ON SMART FARM

In recent years, several security threats [26]–[30] have been observed in IoT domain. Similar attacks can happen on smart farming ecosystem. It is predicted that the attacks on smart farming ecosystem are heavily dependent on the architecture and protocols used in deploying the connected environment. For example, an architecture that uses sensors that work with the Zigbee protocol can have additional attacks such as a replay attack that might be difficult to implement on other protocols. The following network attacks listed below can be orchestrated in smart farms that use IEEE 802.11 protocol:

**Password Cracking:** Hacking the Wi-Fi encrypted protocols is never a complicated task. One of the most popular ways to do that is by cracking the Wi-Fi password that would exploit the user’s network. The requirements needed to complete this attack are very minimal such as laptop or desktop running Kali Linux which utilizes aircrack-ng that has a suite of tools. In addition, a remote card that supports monitor injection mode is required. In order to capture the packets that are transmitted in air, a tool named airodump-ng from the aircrack-ng suite is used. With requirements being satisfied, an attacker can now capture the WiFi Protected Access (WPA) handshake by sending deauthentication packets to the Wi-Fi connected host. Finally, a dictionary attack is performed by testing Wi-Fi passwords present in a previously used word list [31].

**Evil Twin Access Point:** The Evil Twin access point allows an attacker to get credentials by creating a rogue access point. The rogue access point is set up on a reliable network without any permission and tries to persuade a wireless client into associating it with the reliable access point. Also, the rogue access point exploits automatic access point selection techniques. WPA2 is still susceptible to Evil Twin access point attack. This can be a successful approach because BSSID and SSID are simple to retrieve that play an important role in setting up a rogue access point. The attack particularly takes advantage of the auto connect options of the network on the client side [32]. This attack can easily be implemented on smart farms which utilize the 802.11 protocol.

**Key Reinstallation Attacks:** This attack exploits vulnerabilities of the 4-way handshake in WPA2 that secures the modern Wi-Fi. An attacker can trick the victim into reinstalling an already in use key. This is done by manipulating and replaying the cryptographic handshake messages in order to reset the key’s associated parameters to its initial values. This would allow packets to be replayed, decrypted and/or forged. Basically, any information that a victim transmits can be decrypted [33]. Vendor patches have been distributed addressing the vulnerability. In such case, it totally depends on the availability of the patch for a device and the user’s effort to update their devices. Similar to the evil twin access point attack, smart farms that use IEEE 802.11 and have outdated versions without patch are still susceptible to this attack.

**Kr00k - CVE-2019-15126:** This vulnerability affects devices with Wi-Fi chips that belong to Broadcom and Cypress. These Wi-Fi chips are most commonly used in Wi-Fi enabled devices such as smart phones, IoT gadgets, etc. In order to encrypt a part of the communication, an all zero encryption key is used by these vulnerable devices. Therefore, an attacker who wants to launch an attack can decrypt some wireless network packets which are transmitted by these devices. Kr00k also affects Wi-Fi access points and both WPA2-Person, WPA2-Enterprise protocols with AES-CCMP encryption. Patches to fix this vulnerability have been released. However, it is unclear about the number of devices that have been fixed until now [34]. This vulnerability also affects smart farms that include vulnerable devices or access points that use 802.11.
**ARP spoofing attack:** The Address Resolution Protocol (ARP) spoofing attack targets a vulnerability of the ARP protocol. This type of attacks are usually carried out over the local area network (LAN). In this scenario, an attacker fakes the MAC address of the gateway and convinces the victim to send frames to the fake address instead of the destined gateway. In fact, ARP accepts replies without issuing any requests. Also, there is no way to verify a sender since there are no authentication methods in standard ARP. The data traffic can be manipulated and recorded by using ARP spoofing. Therefore, ARP spoofing can be used as a Man-in-the-middle attack to eavesdrop on traffic. Additionally, it can also be used for DoS and session hijacking [35].

**DNS spoofing attack:** In this attack, traffic is directed to a fake website due to the altered Domain Name System (DNS) records. An example is the DNS cache poisoning attack. In this attack, the attacker is used to intercept the traffic between the client and the gateway router. The attacker can now read DNS messages and has two options. In the first option, the attacker can change the IP of the NS (name server) in the DNS response message. In the second option, the attacker can use the same query ID and fake IP to create response messages for the NS. This immensely benefits the attacker because in both the cases, the IP is forged to his benefit [36].

### IV. System Architecture and Experimental Setup

The architecture of deployed single smart farm is based upon Microsoft FarmBeats Student Kit\textsuperscript{11} for precision agriculture. In our setup, we have made some modifications, which include an additional sensor. The Microsoft FarmBeats Student Kit includes Microsoft Azure\textsuperscript{12} cloud services and a Raspberry Pi with soil moisture, light, ambient temperature, and humidity sensors to collect data to improve productivity, increase yield, and save resources, together with data driven\textsuperscript{37} applications. The kit was chosen as the architecture because of its comparable cheap cost, ease of installation, and set-up. In addition, all the data from the Microsoft FarmBeats Student Kit is collected to get a broad picture of precision agriculture deployment and allow researchers to use it as a testbed to deploy proof of concepts smart farming solutions.

The architecture used in this case is used to monitor an indoor plant over an extended period of time. The setup of the architecture which monitors the indoor plant can be seen in Figure 2. The single smart farm multi-layer architecture can be seen in Figure 3, where the Raspberry Pi and its sensors are mounted on an indoor plant to monitor its metrics. As can be seen in Figure 3, the network communication between Raspberry Pi and cloud (Microsoft Azure) will be intercepted and interrupted by a DoS attack, which prevents the Raspberry Pi from connecting to the network. This deployed architecture adapts and extends widely discussed IoT and Cyber Physical System (CPS) multi-layer architectures [38]–[41]. These architectures recognize the use of cloud and edge services, and the infinite capabilities provided by them to fully harness the data generated from smart devices at the physical layer [42]–[47]. The four sensors that were used to monitor indoor plant are listed below (specifications in Table I):

- A **barometer sensor** to detect atmospheric pressure, altitude, temperature, and humidity.
- A **grove light sensor** has light dependent resistor to detect the intensity of the indoor light.
- An **air quality sensor** to detect harmful gases such as carbon monoxide, acetone, and alcohol.
- A **capacitive moisture sensor** measures soil moisture sensor based on capacitance changes.

**TABLE I**

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Interface</th>
<th>Power Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grove-Air Quality sensor</td>
<td>Analog</td>
<td>3.3/5V</td>
</tr>
<tr>
<td>Grove-Light Sensor v1.2</td>
<td>Analog</td>
<td>5V</td>
</tr>
<tr>
<td>Grove - Capacitive Moisture Sensor</td>
<td>Analog</td>
<td>3.3/5V</td>
</tr>
<tr>
<td>Grove - Barometer Sensor (BME 280)</td>
<td>I2C</td>
<td>3.3/5V</td>
</tr>
</tbody>
</table>

\textsuperscript{11}https://farmbeatsstudentkit.com/Student
\textsuperscript{12}https://azure.microsoft.com/en-us/
These sensors were chosen because of their helpful application in monitoring in smart farm. A light sensor is helpful for successfully growing a plant since some plants need more light than others. If there are harmful gases in the air, that might prevent the plant from reaching its full growing potential and therefore an air quality sensor was chosen. Most plants require a specific range of water and the capacitive moisture sensor can display when it is time to water the plant. Different plants require different temperatures, humidity, pressure, and altitude and therefore a barometer sensor was used in this architecture.

These sensors are made by Grove\textsuperscript{13}, and require a Grove Base Hat\textsuperscript{14} for them to be attachable to the Raspberry Pi 3 Model B. The Grove Base Hat provides Digital, Analog, I2C, PWM and UART port. An MCU is build-in which allows for a 12-bit 8 channel ADC \cite{48}. The Grove Base Hat is mounted on a Raspberry Pi 3 Model B. The Raspberry Pi runs Windows 10 IoT Core\textsuperscript{15} which is optimized for smaller devices that have a display or no display. This image is specifically targeted towards embedded IoT devices. IoT Core runs on ARM processors which allows it to be run on the Raspberry Pi \cite{49}. The Raspberry Pi 3 Model B is connected to a personal 2.4 GHz Wi-Fi network. Since a 2.4 GHz network provides coverage at a longer range compared to 5 GHz network, it is applicable to a smart farm environment since the architecture can be farther away from the wireless access point. For this architecture, the transmission time of data was not of critical importance, therefore 2.4 GHz network was used. Alternate protocols that could have been used include Bluetooth and Zigbee. Especially, the application of sensors that use bluetooth or Zigbee to communicate with the Raspberry Pi is another option for a smart-farm architecture. However, in our deployed case only the 802.11 protocol was used to emphasize simplicity and cost-effectiveness. In addition, the Raspberry Pi 3 Model B is connected to Microsoft Azure Cloud Service, more specifically the Azure IoT Central\textsuperscript{16}. The connected sensors send updated data to the cloud as displayed by Figure 4. The cloud allows the sensor data to be manually updated and the Raspberry Pi to be rebooted. If the architecture includes an attached web camera, it would force an update of the image. Data analytic can be accessed by logging into the Azure IoT Central Cloud which provides a template that includes graphs and other visualizations as can be seen in Figure 5. It displays the sensor data as an average over time to visualize changes in the metrics such as changes in temperature, which can be helpful to get a quick overview of the sensor data captured from the field. The telemetry on the left, such as humidity and light, is displayed on the right on the graph in the same color. As an example, temperature stayed constant at about 22 degrees Celsius over five minutes. Any drastic change such as barometric pressure can be explained by the fact that the device rebooted shortly before the first value was read, as seen by the diamond under the graph in Figure 5, and therefore the sensor was still calibrating.

V. METHODOLOGY AND DEMONSTRATION

A denial of service attack was successfully achieved by implementing a Wi-Fi deauthentication attack. In summary, the communication between the Raspberry Pi and the Wi-Fi access point was interrupted by using a Wi-Fi deauther tool. We used the MakerFocus ESP8266 Development Board WiFi Deauther Monster, which allowed us to disconnect the Raspberry Pi off the network which therefore caused no data to be sent to the Azure cloud. In addition, we expanded the attack to include the whole network and therefore disabled any devices to connect to the network. The deauther sends packets that disconnect devices but does not interfere with any frequencies.

A. Overview of the Wi-Fi Deauthentication Attack

A Wi-Fi deauthentication attack is successfully implemented on a smart farm architecture that is connected to a

2.4 GHz network. This attack falls under Denial of Service (DoS) attacks and exploits 802.11 vulnerabilities [51], [52]. An attacker starts by monitoring raw frames that include information such as source and destination Media Access Control (MAC) addresses to find the targeted victim. The adversary sends spoofed deauthentication frames with spoofed source MAC address of access point or victim station once data or association response frame is found [51], [52]. This forces the targeted station to become unauthenticated and therefore is disconnected from the network. The attacked station then tries to reconnect and to prevent that re-connection the attacker continuously keeps sending the deauthentication frames. The sequence of this attack is shown in Figure 6. To be able to reconnect, the attacked client is forced to repeat IEEE 802.11 authentication and association process. The station is unable to connect to the network through prolonged sustaining of the spoofed frames [51]. This repeating transmission of frames is considered a DoS attack against the target MAC address which is then prevented to access the network. This kind of attack is difficult to detect because the frames are sent directly to the client without any detection or logging by the access point (AP) or Intrusion Detection System (IDS). In addition, MAC filtering process is unable to prevent this attack [52]. Often such attacks are used to prevent unauthorized stations from connecting to access points by wireless IDS vendors [51]. A prime reason this attack is possible is due to the fact that management frames are not encrypted in IEEE 802.11 protocol. However, the protocol 802.11w prevents Wi-Fi deauthentication attacks by including cryptographic protection to deauthentication and dissociation frames. Therefore, those frames are very hard to be spoofed in a DoS attack [53]. An important reason for successful demonstration of this attack is because many network providers have not updated to 802.11w. In the following subsection, we will detail how this attack is orchestrated in a single smart farm setup.

B. Steps to a DoS Attack

In order to organize a DoS attack, first, packets were sniffed to ensure the connectivity of Raspberry Pi and to see whether the packets are encrypted. Wireless Diagnostics in Mac OSX was used to sniff the packets, as shown in Figure 7. The built in Wi-Fi stumbler tool was used to identify channels and widths to use for packet sniffing, as illustrated in Figure 8. The channel was found to be 11 for the network. After the channel was identified, the sniffer on Mac OSX was used to trace network traffic on that channel. The packet capture was opened with WireShark\footnote{https://www.wireshark.org/} shown in Figure 9. These displayed packets are filtered by the source. In our case, the source of these packets is attacked Raspberry Pi which is transmitting packets to the router (ARRISGro) and using IP multicast (IPv4mcast) to send packets to multiple sources in one transmission. The device is sending null data to the connected router to establish that it is in active state and that the transmission of frames from the AP to Raspberry Pi should be as expected. After the packets were sniffed, the Wi-Fi deauthentication attack was started. These packets are encrypted in WPA2 which prevents similar attack possibilities.

To successfully implement a Wi-Fi deauthentication attack, the Wi-Fi deauther tool needs to be in range of the network. The MakerFocus ESP8266 Development Board WiFi Deauther Monster comes with an antenna to improve its ability to catch the signal, which makes an adversary located at a Wi-Fi enabled smart farm to perform such attack. Note that this...
The attacked Raspberry Pi is connected to the Wi-Fi network named 'Free Virus Download'. This network is selected in the deauther tool to attack the complete network as shown in Figure 12. The steps for implementing the expanded network attack are similar as for disabling an individual station. First, access points and stations need to be scanned, then the network needs to be selected under APs. And finally, the deauther attack needs to be selected in the main menu. The Wi-Fi deauther attack was done on the whole network which resulted in the disconnection of all devices connected to the network including the Raspberry Pi. Packet capture in Figure 13 displays the router sending deauthentication frames to the stations on the network. This packet capture is filtered by packet info and is therefore not in order. This filtering by packet info was done to show that a large number of deauthentication frames have been sent repeatedly to prevent stations from connecting it from the network. The attacked Raspberry Pi is not connected to the network anymore, and the cloud cannot receive any sensor update. Figure 11 shows that when trying to update the sensors during the attack, no updates were received.

**Attacking the Entire Network:** The attacked Raspberry Pi is connected to the Wi-Fi network named 'Free Virus Download'. This network is selected in the deauther tool to attack the complete network as shown in Figure 12. The steps for implementing the expanded network attack are similar as for disabling an individual station. First, access points and stations need to be scanned, then the network needs to be selected under APs. And finally, the deauther attack needs to be selected in the main menu. The Wi-Fi deauther attack was done on the whole network which resulted in the disconnection of all devices connected to the network including the Raspberry Pi. Packet capture in Figure 13 displays the router sending deauthentication frames to the stations on the network. This packet capture is filtered by packet info and is therefore not in order. This filtering by packet info was done to show that a large number of deauthentication frames have been sent repeatedly to prevent stations from connecting it from the network. The attacked Raspberry Pi is not connected to the network anymore, and the cloud cannot receive any sensor update. Figure 11 shows that when trying to update the sensors during the attack, no updates were received.
to the network. This proves that the Wi-Fi deauthentication attack was a success due to the inability of the Raspberry Pi to connect to the network and sending sensor updates.

Our demonstration of the Wi-Fi deauthentication attack exposes the weakness of the IEEE 802.11 protocol (2.4 GHz), which requires attention and especially relevant not only in smart farming but also other IoT domains. By using 802.11w, management frames are encrypted and will make deauthentication attack much more difficult to implement. However, our deployed Wi-Fi network and numerous other similar networks do not have 802.11w implemented.

VI. IMPLICATIONS OF DEAUTHENTICATION ATTACKS

Wi-Fi Deauthentication attack is one of the major availability attacks [54] which disrupts communication networks and equipment availability, and negatively impacts the smart farms productivity. In our experiments, the Raspberry Pi can be considered an online connected equipment (e.g., smart sensor or drone). Wi-Fi Deauthentication attack targets the Raspberry Pi and detaches it from the network. This attack impacts smart farms in multiple scenarios. A few are discussed below.

Sensor data obstruction: Data acquired from various sensors is the foundation of a smart farm, where most decisions are automated based on the data. For instance, the smart farm’s irrigation system activates and deactivates based on the soil water level measured by the moisture sensors. Typically, it is based on a simple certain threshold; however, modern smart irrigation systems consider more dynamic factors that require real-time data analytics and AI technologies. Real-time AI services can be used to determine how environmental factors influence the crops being irrigated as well as how soil moisture responds to irrigation for different crops, soils, and environmental conditions. As such, Deauthentication attacks, which prevent moisture sensors from connecting to the network, obstruct real-time communication and disrupt the irrigation system’s decision. This leads to crops over or under-watering, and eventually damage crops, negatively affecting a successful harvest. The potential damage of this particular scenario is also valid for livestock, where sensors monitoring their food, water, and health status are unavailable.

Controlling connected devices: As stated in section III, a deauthentication attack can be the basis for a subsequent evil twin access point or a password cracking attack. The attacker fetches the authentication details of the farmer by redirecting the farmer to a similar fake network. After that, the attacker gains access to the entire smart farm where he can control various devices to intentionally cause damage. For example, the attacker can damage the crops by controlling agricultural drones to spray excessive fertilizers over the plants. This would result in damaging crops at an early stage and bring huge loss.

It is important to recover from DoS attacks and communication disruptions quickly before any substantial damage takes place. As such, detection and recovery techniques should be well researched. Such attacks, if launched on a large scale, can cause dramatic economic loss to an entire country.


