AIR-FI: Leaking Data From Air-Gapped Computers Using Wi-Fi Frequencies

Mordechai Guri

Abstract—This article presents a new attack allowing attackers to exfiltrate data from isolated, air-gapped computers via Wi-Fi frequencies. We show that malware in a compromised air-gapped computer can generate signals in the Wi-Fi frequency bands. The signals are generated through the memory buses - no special hardware is required. Sensitive data can be modulated and secretly exfiltrated on top of the signals. We show that nearby Wi-Fi-capable devices (e.g., smartphones, laptops, and IoT devices) can intercept these signals, decode them, and send them to the attacker over the Internet. We utilized the physical layer information exposed by the Wi-Fi chips to extract the signals. We further implemented the transmitter and receiver and discussed design considerations and implementation details. We evaluated this covert channel in terms of bandwidth and distance and presented a set of countermeasures. Our evaluation shows that data can be exfiltrated from air-gapped computers to nearby Wi-Fi receivers located meters away at bit rates of 16 bit/sec.

Index Terms—Network-level security and protection, covert channels, air-gap, Wi-Fi, exfiltration, electromagnetic

1 Introduction

One of the initial phases in the kill chain of advanced persistent threats (APTs) is infiltrating the network of the target organization. To achieve this goal, adversaries may use attack vectors such as phishing emails, compromised websites, malicious documents, exploit kits, and other types of online attacks [1]. Having a foothold in the organization network, the attacker can move to the exfiltration phase, gathering and silently leaking sensitive information outward.

1.1 Isolated, Air-Gapped Networks

When highly sensitive or confidential information is involved, an organization may resort to air-gapped networks. Such networks are disconnected from the Internet logically and physically, where any wired or wireless connection to the Internet is strictly prohibited [2]. Certain industries maintain their data within air-gapped networks, including governmental, financial, defense, and critical infrastructure sectors. In many cases, operational technology (OT) networks are also kept isolated from the Internet to protect the physical processes and the machinery they manage [3]. Classified military networks such as the Top-Secret Joint Worldwide Intelligence Communications System (JWICS) and the Secret Internet Protocol Router Network (SIPRNet) are also known to be air-gapped [4].

• *The author is with the Department of Software and Information Systems Engineering, Cyber-Security Research Center, Ben-Gurion University of the Negev, Beersheba 84105, Israel. E-mail: gurim@bgu.ac.il.*

Manuscript received 23 March 2022; revised 20 June 2022; accepted 22 June 2022. Date of publication 27 June 2022; date of current version 13 May 2023. (Corresponding author: Mordechai Guri.)
Recommended for acceptance by H. Hu.
Digital Object Identifier no. 10.1109/TDSC.2022.3186627

1.2 Infecting Air-Gapped Networks

Despite the high degree of isolation, air-gapped networks are not immune to cyber-attacks. To penetrate highly secure networks, motivated adversaries may employ complex attack vectors, such as sabotaging the supply chain, compromising a third-party software, using malicious insiders, and exploiting deceived insiders [3], [5]. These techniques allow the attackers to insert targeted malware into systems within the isolated environment.

One of the first famous incidents in which air-gap facility was breached involved the Stuxnet worm. Stuxnet was considered a revolutionary in the industry because it was the first targeted cyber-attack against isolated cyber-physical control systems [6]. In 2018, the Wall Street Journal reported that Russian hackers who worked for a state-sponsored group compromised supposedly secured, air-gapped networks of the control rooms of U.S. electric utility [7]. In 2019, the media reported that India's largest nuclear power plant was the target of a successful cyber-attack [8]. In 2020 Trend Micro researchers identified an APT called USBFerry, the targets the air-gapped networks of the Taiwanese and Philippine military organizations. The reports said hackers used USB devices to have an initial foothold and to jump the air gap to adjacent networks, even across government organizations [7]. Other sophisticated malware capable of compromising air-gapped networks were reported in recent years [9], [10], [11].

1.3 Air-Gap Exfiltration

Once the attacker has taken his/her initial step into the air-gapped network, he/she moves on to the next phases of the kill chain. In these subsequent phases, sensitive data is collected, including documents, images, keylogging, credentials, and other information of interest. In the case of Internet-connected networks, the gathered data is exfiltrated through the Internet, usually, in covert channels

within common protocols such as HTTPS, FTP, SSH, and SMTP [12]. However, in the case of isolated, air-gapped networks, the attacker must find non-conventional communication techniques to leak the data out - methods which are referred to as in the literature air-gap covert channels [13], [14]. For example, malware may exploit electromagnetic radiation from various computer components to transmit data [15], [16], [17]. Acoustic [18], [19], optical [20], [21], [22], thermal [23], magnetic [24], and electric [25] air-gap covert channels have also been demonstrated over the past twenty years.

1.4 Our Contribution

This paper introduces a new type of covert channel that exploits Wi-Fi frequencies to leak data from air-gapped networks. The AIR-FI attack presented in this paper does not require Wi-Fi-capable hardware in the air-gapped computers. Instead, we show that an attacker can exploit the DDR SDRAM buses to generate electromagnetic emissions in the 2.4 GHz Wi-Fi bands and encode binary data on top of it. The AIR-FI attack generates interference in the Wi-Fi channels and does not construct standard Wi-Fi packets. We show that nearby Wi-Fi-capable devices such as smartphones, laptops, and Internet of Things (IoT) devices, can receive the transmissions, decode the information and then send it to the attacker via the Internet.

The AIR-FI covert channel has the following characteristics:

- *Requires no Wi-Fi transmitter.* The method doesn't require any Wi-Fi hardware in the air-gapped computer. Instead, it uses computer memory hardware (DDR SDRAM) to generate the signals.
- *Has many potential receivers.* Modern IT environments are equipped with many types of Wi-Fi devices: smartphones, laptops, IoT devices, environmental sensors, embedded systems, and wearables devices. The attacker can compromise such equipment to receive the AIR-FI transmissions from air-gapped computers.
- *Requires no special privileges.* The transmitting code does not require special privileges (e.g., root), kernel drivers, or access to hardware resources. Furthermore, it can be initiated from an ordinary user-space process. Note that the attacker might have to modify the DDR SDRAM clock frequency setting in some cases (e.g., overclocking and underclocking). In these cases, root-level privileges are required at the preliminary step of the attack.
- *Works in virtual machines (VMs).* The covert channel works effectively, even from within isolated virtual machines.

The rest of this paper is organized as follows: Related work is presented in Section 2. The attack model is discussed in Section 3. Technical background on DDR SDRAM and Wi-Fi is provided in Section 4. Sections 5 and 6, respectively, contain details on signal generation and modulation, and data transmission and reception. In Section 7 we present the evaluation and measurement results. A set of countermeasures is discussed in Section 8, and we conclude in Section 9.

2 Related Work

Air-gap covert channels are classified into different categories: physical media, electromagnetic and magnetic, electric, optical, acoustic, and thermal.

Physical media can be used to maintain a covert channel from/to air-gapped networks. Several APTs discovered in the past (e.g., Stuxnet [6], USBCulprit [26], Ramsay [27], and others [27]) used this techniques, collecting documents with particular extensions, encrypting and passing them on to USB devices when they are connected to the system. However, this method required a USB device to be attached to the air-gapped system, which is commonly forbidden due to security policies [28].

Researchers showed that it is possible to exploit the electromagnetic emissions from the computer display and graphical units for data exfiltration [16]. AirHopper, presented in 2014, is a malware designed for leaking data from air-gapped computers to a nearby smartphone via FM radio waves emitted from the screen cable [15]. In 2015, Guri *et al.* presented GSMem, malware that transmits data from air-gapped computers to nearby mobile phones using cellular frequencies [29]. USBee is malware that uses the USB data buses to generate electromagnetic signals in certain frequencies around 250 MHz [30]. Sehatbakhsh *et al.* presented a side-channel that exploits the electromagnetic emanations from the power management unit [31]. LANTENNA attack, introduced in 2021, uses the electromagnetic radiation from Ethernet cables to encode data and transfer it to a nearby receiver [32]. BitJabber [33] and LoRa EMR [34] are electromagnetic covert channels used to transmit data from air-gap devices to nearby dedicated radio frequencies (RF) receivers. In order to prevent electromagnetic leakage, Faraday cages can be used to shield sensitive systems. Guri *et al.* presented ODINI [35] and MAGNETO [24], two types of malware that can exfiltrate data from Faraday-caged air-gapped computers via magnetic fields generated by the computer's CPU. Researchers also introduced MagView, a magnetic covert channel that use video encoding and decoding process for signal generation from isolated devices [36]. Matyunin *et al.* used the Hard-Disk Drive (HDD) to generate magnetic signals from air-gapped computers [37]. Note that all the magnetic channels reside at a range of very low frequencies, mostly less than 200 Hz. In 2019, researchers showed how to leak data from air-gapped computers by modulating binary information on the power lines [25]. The data is modulated, conducted to the power lines, and received by an adversary tapping the wires. Previous work proposed to use optical emanations from computerized systems for covert communication. Loughry introduced the use of network devices and keyboard LEDs for data leakage [20]. Researchers also manipulated the hard drive indicator LED [21], USB keyboard LEDs [38], router and switch LEDs [39], and security cameras and their infrared LEDs [40], in order to exfiltrate data from air-gapped environments. At the acoustic category, several studies used sound to maintain a covert channel between disconnected PCs or laptops in a room, using their speakers and microphones [41]. Guri *et al.* introduced Fansmitter [19], Diskfiltration [42], and CD-LEAK [43], attacks which enabled the leakage of information from an air-gapped computer via noise intentionally

TABLE 1
Electromagnetic Covert Channels

Method	Year	Receiver	Band
LCD (soft tempest) [16]	1998	AM Radio	80 - 120 MHz
AirHopper [15]	2014	FM Radio	88 - 104 MHz
GSMem [29]	2015	Feature-phone/SDR	850 MHz
USBee [30]	2016	SDR+antenna	250 MHz
BitJabber [33]	2020	SDR+antenna	800 MHz
Power Unit [31]	2020	SDR+antenna	250 - 2000 kHz
EMR LoRa [34]	2021	SDR+antenna	799.9 MHz
LANTENNA [32]	2021	SDR+antenna	250/500/750 MHz
AIR-FI	-	**Wi-Fi receivers/SDR**	**2.4 GHz (Wi-Fi)**

emitted from the PC fans, hard disk drives [42], and CD/DVD drives [43], respectively. With these techniques, the transmitting computer does not need to be equipped with audio hardware or internal/external speakers. Researchers also showed that the computer fans generate mechanical vibrations that can be sensed by a nearby smartphone using the accelerometer sensor [44]. Recently, researchers demonstrated how malware could turn the computer power supply into a secondary speaker in order to exfiltrate information [45]. BitWhisper is a unique thermal-based covert channel enabling inbound and outbound communication between air-gapped computers by encoding information on temperature fluctuations [46].

2.1 Novelty

Our work is the first to propose using Wi-Fi in the attack model on air-gapped computers. We exploit the internal DRAM buses to generate emissions in the 2.4 GHz frequency band and receive them by nearby Wi-Fi receivers. Our work contributes to the transmitter side by introducing the generation of 2.4 GHz frequencies in all 11 Wi-Fi channels. Our work also contributes to the receiver side by demonstrating that nearby Wi-Fi devices can receive the transmitted data.

2.2 Previous Work

In addition to the contribution mentioned above, previous work that used the RAM, such as GSMem [29], BitJabber [33], and EMR LoRa [34], were capable of generating emissions around 800 MHz. We tuned the emitted frequencies from DDR4 modules into the Wi-Fi bands and received them by nearby receivers rather than dedicated SDR receivers. Note that dedicated hardware receivers and specialized antennas allow previous work to achieve greater bandwidth of 1 - 100 kbps, but they require the installation of dedicated hardware in the target site. Our work uses Wi-Fi devices for the attack with the cost of low bandwidth of 1 - 16 bit/sec.

Table 1 shows the existing electromagnetic covert channels and AIR-FI.

3 Attack Model

We present an attack model which enables motivated adversaries to leak data from air-gapped environments. Our attack model consists of a transmitter and receiver. The transmitter is an air-gapped workstation and is usually part of a larger air-gapped network. The receiver is a device with Wi-Fi capabilities. It could be a Wi-Fi-capable device located in the area or a Wi-Fi device implanted by the attacker. Note that a bifurcated attack model consists of compromised transmitter and receiver is common, and shared by many works in a field of covert-channels, e.g., [16], [31], [33], [34], [35], [36], [41], [47], [48], [49].

3.1 Infecting the Air-Gapped Network

In a preliminary stage, the air-gapped network is infected with an APT. Although the breach of secure, air-gapped networks was considered a sensational anecdote in the past, it is shown feasible in the modern era of cyber-security. A report from ESET revisited 17 malicious frameworks between 2009 and 2021 used to attack air-gapped networks [28].

In a typical APT kill chain, the attackers research their targets and carefully plan the attacks [1]. After defining the initial target, attackers might install malware on the network via various infection vectors: supply chain attacks, contaminated USB drives, social engineering techniques, or by using malicious insiders or deceived employees. Note that infecting air-gapped networks can be accomplished, as demonstrated by the attacks involving Stuxnet [50], Agent. Btz [51], and other malware [52], [53], [54]. Other cyberattacks on secured facilities such as governmental agencies and nuclear plants are known in recent years [55]. It is interesting to note that a report attributed to NIST suggests that when they investigated industrial control systems (ICS) that claimed to be air-gapped, they found that many of them are connected to less-secure networks [56]. In 2017, malware known as Copperfield infected critical infrastructure computers in the Middle East [31]. This malware could infect USB drives and spread to other devices in the network. Other air-gap targeting APTs from recent years include USBCulprit [26] and Ramsay [27]. After the initial infection, the APT might exploit vulnerabilities to spread in the target network to strengthen its foothold.

3.2 Infecting Wi-Fi Devices

The attacker must infect Wi-Fi-capable devices in the area of the air-gapped network. Such devices might be smartphones of visitors or employees, desktop and laptop computers with wireless networking, or IoT devices with Wi-Fi transceivers. Since the devices use wireless networking, they can be infected through Wi-Fi. Compromising the

TABLE 2
Attack Vectors Demonstrated on Air-Gap Networks and Wi-Fi Devices

Attack phase	Attack vectors	Attack examples
Air-gap network	Supply-chain attacks, malicious insiders, deceived insiders	Stuxnet [50], Agent.BTZ [51], Fanny [66], Tick [66], Turla [52], ProjectSauron [67], SymonLoders [28], Ramsay [10], USBCulprit [26], Emotional Simian [68]
Wi-Fi devices	OS & application vulnerabilities, wireless interfaces, social engineering, malicious apps, messaging, etc.	Smartphones & tablets [63], [64], [69], laptops [57], [70], wearable devices [71] IoT devices [59], [60], [62], [72]

devices can be done by exploiting Wi-Fi hardware/software vulnerabilities or via flaws in the network protocols, OS, and applications. Such attacks were demonstrated on smartphones [57], laptops with Wi-Fi network interface cards (NICs) [58], and a wide range of IoT devices such as smart bulbs [59], smart locks [60], and others [61], [62]. In particular, there are many popular attack surfaces on mobile devices that attackers may use, including email attachments, messaging applications, and malicious websites & advertisements [63], [64], [65]. The compromised Wi-Fi devices are installed with the receiver side of the malware. Another option for the attacker is to intentionally hide or implant a dedicated Wi-Fi transceiver in the area of compromised computers. Table 2 summarizes the attack vectors and relevant examples.

3.3 Data Exfiltration

As a part of the exfiltration phase, the attacker might collect data from compromised computers. The data could be documents and images, keystrokes logging, credential tokens, encryption keys, etc. Once the data is collected, the malware initiates the AIR-FI covert channel. It encodes the data and transmits it to the air (in the Wi-Fi band at 2.4 GHz) by exploiting the electromagnetic emissions generated from the DDR SDRAM buses. The compromised devices in the area collect Wi-Fi signals, detect the covert AIR-FI transmission, decode the information, and sends it to the attacker over the Internet.

The chain of attack described above is presented in Fig. 1, and the attack is illustrated in Fig. 2. Malware in the air-gapped computer (A) uses the memory to generate signals in the 2.4 GHz Wi-Fi frequency band. Binary information is modulated on top of the signals and received by nearby Wi-Fi receivers (e.g., a laptop (B) and a smartphone (C)).

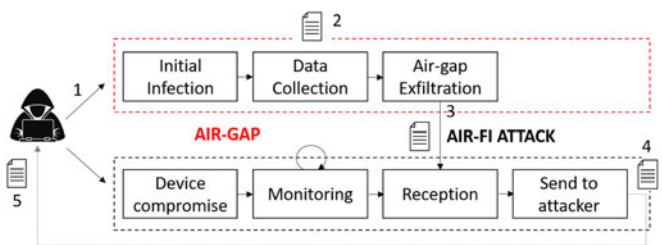


Fig. 1. The chain of AIR-FI attack.

4 Technical Background

This section provides the fundamental background necessary for understanding the AIR-FI covert channel.

4.1 DDR SDRAM

The double data rate (DDR) synchronous dynamic random-access memory (SDRAM) are the type of memory modules integrated into modern motherboards. The DDR technology doubles the bus bandwidth by transferring data on both the rising and falling edges of the memory bus clock. In DDR SDRAM, the bus bandwidth is referred to in megabits per second. The bandwidth B is calculated by the formula $B = (f * 2 * l)/8$, where f is the memory bus clock rate and l is the width of the line transfer. Another important parameter of memory modules is the Column Address Strobe (CAS) latency, also known as the CL. This is the time delay between when the read command is delivered to the memory and the beginning of the data response.

4.2 DDR Memory Bus

Data is exchanged between the CPU and the memory over dedicated buses (Fig. 3). The memory buses maintain two types of signals: (1) the address bus, which transfers addresses and commands, and (2) the data bus (DQ bus), which transfers the actual data. The address bus sends commands and instructions from the controller to the SDRAM. The bus is synchronized to the clock (CLK) signals, with the signals on the address bus being sampled by the SDRAMs on the rising edge of the CLK signal.

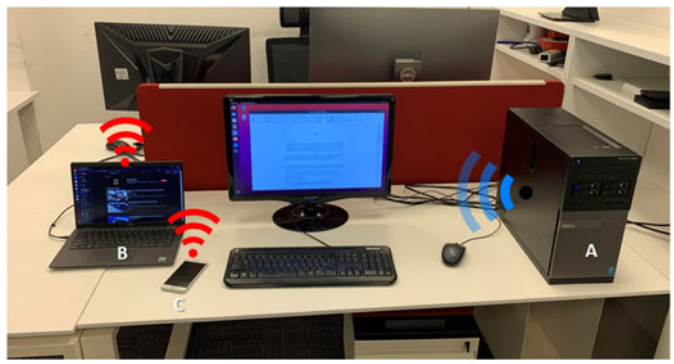


Fig. 2. Illustration of the AIR-FI attack. Malware in the air-gapped computer (A) uses the DDR memory to generate signals in the 2.4 GHz Wi-Fi frequency band. Binary information is modulated on top of the signals and received by nearby Wi-Fi receivers (e.g., a laptop (B) and a smartphone (C)).

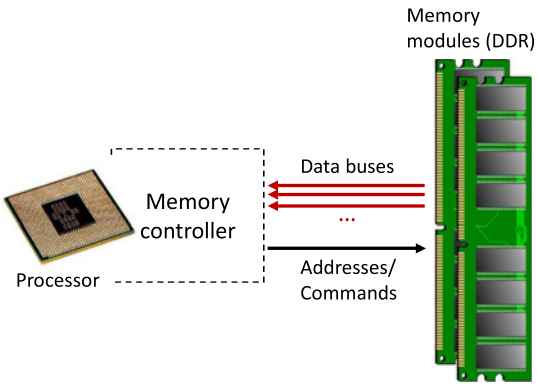


Fig. 3. DDR SDRAM memory buses.

The memory buses generate electromagnetic radiation at a frequency correlated to its clock frequency and harmonics. For example, DDR4-2400 emits electromagnetic radiation at around 2400 MHz.

4.3 Overclocking/Underclocking

The memory modules provide the BIOS/UEFI (Unified Extensible Firmware Interface) a set of frequencies that it can operate at. This information is defined according to the JEDEC (Joint Electron Device Engineering Council) specification, and it is passed during the boot through a mechanism called Serial Presence Detect (SPD). Intel allows the standard timing parameters of the installed memory to be changed via a specification called Extreme Memory Profile (XMP). With XMP, the user can modify the memory parameters, such as the frequency and CAS latency. Changing the operating frequency of the memory modules is referred to as overclocking (for increasing the frequency) and underclocking/downclocking (for decreasing the frequency).

4.4 Wi-Fi Frequency Bands

The IEEE 802.11 standard defines the frequency ranges in the electromagnetic spectrum allowed for Wi-Fi communications. There are several versions of the 802.11 standards. These standards define factors such as frequency ranges, bandwidths, and distances. Today, most Wi-Fi chips support the 802.11b/g/n standards. The 802.11b/g/n standards are often referred to as the 2.4 GHz band. 2.400 - 2.490 GHz range is the most widely used and certified range available for Wi-Fi. The standards define 14 channels in the 2.4 GHz band, but only 11 of these channels are allowed in all countries. The first 11 channels have a space of 5 MHz between them and 12 MHz between channels 13 and 14. A common bandwidth of a Wi-Fi channel is 20 MHz which means that signals of adjacent channels may interfere with each other. Table 3 contains a list of the regulated Wi-Fi channels supported by the 802.11b/g/n standards.

TABLE 3
List of the Regulated Wi-Fi Channels (802.11b/g/n)

Channel	Center (MHz)	Range (MHz)	North America	Japan	Others
1	2412	2401-2423	Yes	Yes	Yes
2	2417	2406-2428	Yes	Yes	Yes
3	2422	2411-2433	Yes	Yes	Yes
4	2427	2416-2438	Yes	Yes	Yes
5	2432	2421-2443	Yes	Yes	Yes
6	2437	2426-2448	Yes	Yes	Yes
7	2442	2431-2453	Yes	Yes	Yes
8	2447	2436-2458	Yes	Yes	Yes
9	2452	2441-2463	Yes	Yes	Yes
10	2457	2446-2468	Yes	Yes	Yes
11	2462	2451-2473	Yes	Yes	Yes
12	2467	2456-2478	Canada only	Yes	Yes
13	2472	2461-2483	No	Yes	Yes
14	2484	2473-2495	No	11b only	No

5 Transmission

This section presents the signal generation technique, data modulation, and data transmission protocol.

5.1 Electromagnetic Emission

We explored two types of electromagnetic emissions that emanate from memory buses.

- *Persistent Emission.* An electromagnetic emission that is continuously generated by the memory controller, regardless of the activity in the address/data buses. This radiation spans the entire spectrum of the DDR SDRAM frequency when the computer is turned on.
- *Triggered Emission.* An electromagnetic emission that is generated from the electronic activities (current flow) in the data bus. This emission is correlated with the memory read/write operations executed by processes currently running in the system.

5.2 Signal Generation

We used two techniques to generate Wi-Fi signals from air-gapped computers based on the above observations.

- Memory operations. We transfer data in the data bus to generate an electromagnetic emission at the frequency of the memory modules. Since the clock speed of memory modules is around the frequency of 2.4 GHz or its harmonics, the memory operations generate electromagnetic emissions around the IEEE 802.11b/g/n Wi-Fi frequency bands.
- Memory operations + clocking. When the operational frequency of the memory modules is not near the 2.4 GHz frequency or its harmonics, we initially overclock/downclock the memory speed to the frequency of Wi-Fi bands or its harmonics. The overlocking/downclocking operation can be done programmatically or at the BIOS/UEFI configuration level. We perform the memory operation schemes described above to generate emissions at the Wi-Fi frequency band following the frequency adjustments. Note that malware that are capable of reconfiguring BIOS/UEFI were found in the wild [73], [74].

5.3 Channel Interference

The generated emission from the data bus interferes with the Wi-Fi channels. The interferences in the corresponding channel can be measured at the PHY layer of the 802.11 protocol stack. The operation is illustrated in Fig. 4. In this case, the AIR-FI signals are generated at 2.44000 GHz. The signal

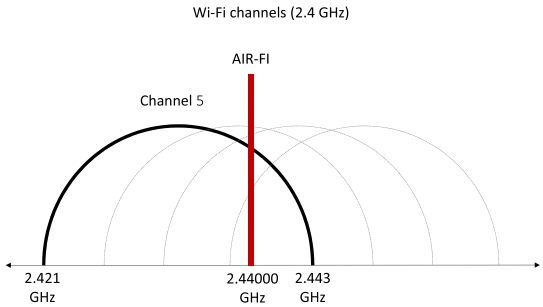


Fig. 4. AIR-FI channel interference.

is interfering with channels 5-8. Using different timing parameters, we could adjust the signals in each of the 11 802.11 b/g/n Wi-Fi channels.

5.4 Modulation

Algorithm 1 shows the signal modulation process using the memory operation technique using on-off keying (OOK) modulation. The `modulateRAM` function receives the array of bits to transmit (`bits`) and the bit time in milliseconds (`bitTimeMillis`). This function iterates over the bits, and according to the current bit, the algorithm determines the operation to perform during a bit time period. If the bit is '1' (line 4), it performs a series of memory write operations which consists of sequential memory copying between two arrays, with a size of 1 MB each (lines 6-7). This memory transfer effectively generates the emission from the data bus. If the bit is '0', the algorithm sleeps for a bit time period, which stops the emission from the RAM bus. To adjust the frequency, use maintained a pulse-width modulation (PWM) technique to generate a square-wave on the given carrier. We synchronize the transmission and sleep times with the current pulse time using a POSIX per-process timer (`timer_create`). To eliminate the effect of caching, we randomize the content of the array every few iterations. That way, we assure the generation of the signal and its timing in every iteration since the actual data is transferred in the DRAM bus. Another possible strategy is to use a number of predefined arrays and choose random arrays in each iteration.

Algorithm 1. ModulateRAM (Freq, Bits, bitTimeMillis)

1: $bitEndTime \leftarrow getCurrentTimeMillis()$
2: **for** bit *in* $bits$ **do**
3: $bitEndTime \leftarrow bitEndTime + bitTimeMillis$
4: **if** $bit == 1$ **then**
5: **while** $getCurrentTimeMillis() < bitEndTime$ **do**
6: $memcopy(array1, array2)$
7: $memcopy(array2, array1)$
8: $n = n + 1$
9: **if** $n == 5$ **then**
10: $RandomizeArrays()$
11: $n = 0$
12: **end if**
13: **end while**
14: **else**
15: $sych(bitTimeMillis, PWM(freq))$
16: **end if**
17: **end for**

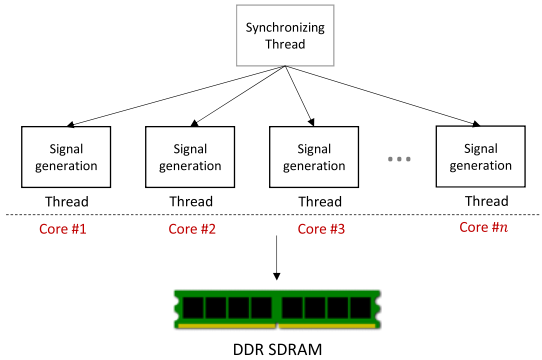


Fig. 5. Signal generation with concurrent threads.

5.4.1 Multi Cores

The signal generation algorithm shown above runs on a single CPU core. In order to amplify the signal, we execute the signal generation in several concurrent threads, where each thread is bound to a specific core. The memory operations of the threads are synchronized by a governor thread using the POSIX thread functions such as `thread_barrier_wait`. Signal generation with concurrent threads is depicted in Fig. 5.

5.5 Packets

The data is transmitted in packets that consist of a preamble, payload, and error-detecting code.

- Preamble. The packet begins with a `0xAA` hex value. This sequence of `10101010` in binary allows the receiver to synchronize with the beginning of each packet and determine the carrier amplitude and one/zero thresholds.
- Payload. The payload is the raw binary data transmitted within the packet. It consists of 32 bits.
- Error detection. For error detection, we use the CRC-8 (a cyclic redundancy check) error detection algorithm. The CRC is calculated on the payload data and added at the end of each packet. If the received CRC and the calculated CRC differ, the packet is omitted on the receiver side.

Fig. 6 shows an AIR-FI packet transmitted from a workstation with a DDR4 (2400 MHz) memory module. In this case, the transmission around 2.42 GHz overlaps Wi-Fi channels 3, 4, and 5.

6 Reception

As shown in Section 5, the electromagnetic emissions generated by the data bus are around the 2.4 GHz frequency range and overlap the Wi-Fi channels. In Wi-Fi transceiver chips, the baseband processor handles the radio, PHY, and MAC layers. The software protocol stack processes the Internet, transport, and application layers, usually in the kernel drivers. In order to measure the interference generated, the attacker has to access the low-level radio measurement information from the PHY layer. This can be done by using user-level interfaces provided by the manufacturer or by compromising the driver or firmware of the Wi-Fi chips and passing the required radio measurements to the

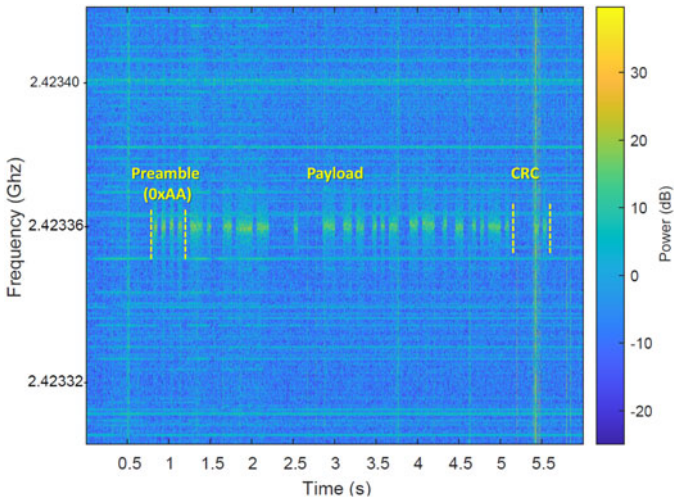


Fig. 6. AIR-FI packet as transmitted from a workstation with a DDR4 (2400 MHz) memory module. The transmission overlaps channels 3,4, and 5.

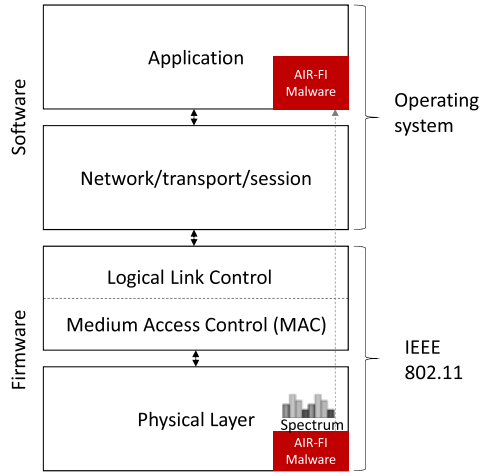


Fig. 7. The receiver side of AIR-FI malware.

software stack. The architecture of AIR-FI malware is illustrated in Fig. 7. The driver/firmware level code invokes the radio frequency (RF) information, usually maintained through the Rx chain, to reach the baseband processing. The data is passed to the AIR-FI at the application layer through the operating system (e.g., via kernel module).

6.1 Accessing Wi-Fi PHY Layer

As presented in Section 3, the attacker must compromise Wi-Fi-capable devices in the area of the air-gapped network. There are many potential Wi-Fi receivers in the modern IT environment, including desktops and laptops, mobile phones, wearable devices, IoT, and office equipment such as printers, scanners, TVs, and others.

To receive the AIR-FI transmissions, the attacker must access the low-level radio information from the PHY layer in the compromised devices. There are three different layers within the Wi-Fi stack from which the PHY information might be accessed from:

- *Application layer (user-level).* Some chipsets expose the physical layer information such as the I/Q phase of the wireless signal and FFT information. The information is delivered from the kernel to the application layer via command-line tools and userspace APIs, which use a virtual file system (Linux OS) and device input/output control (Windows OS). For example, some Qualcomm Atheros 802.11n chipsets include documented spectral features exposed to the user-level applications [75].

  In the attack context, the attacker is required to gain access to the targeted device, usually via remote exploitation, without required root access. Once the attacker has a foothold in the device, he can launch a process or thread to monitor the Wi-Fi physical layer information. To evade security products, the attacker may inject the userspace code into other legitimate processes, using techniques such as shellcode injection or thread injection [76].
- *Device-drivers layer (kernel-level).* Many chipsets expose the physical layer information to the device drivers in the kernel. The kernel drivers receive this information from the hardware via a common bus, usually by issuing ioctl commands. For example, NICs chipsets such as Intel IWL5300, Intel AX200, Intel AX210, Qualcomm QCA9379, Broadcom bcm4330, and others support retrieving the physical layer information in the device driver. Some projects patched the Wi-Fi device drivers in off-the-shelf smartphones to access the low-level MAC and PHY layers information [77]. In the attack's context, the attacker must gain access to the targeted device via remote exploitation and inject itself into the kernel space, using privileges escalation techniques. The malicious code at the kernel level can directly query the information from the Wi-Fi hardware or modify/replace the existing Wi-Fi drivers. Notably, kernel-level rootkits capable of installing or modifying device drivers have been found in the wild in the past for all major OS including Linux [78], Windows [79], Android [80], and iOS [81].
- *Firmware-level.* Embedded devices such as IoTs, have firmware-controlled Wi-Fi chips. In many cases, the device hardware is driven by a specific version of Linux or other embedded OS stored in its firmware. The Wi-Fi protocol stack is managed by a firmware level and stored, loaded, and fully executed within the Wi-Fi chip. In the attack's context, the attacker must gain access to the targeted device via remote exploitation and then compromise its Wi-Fi firmware. From a technical perspective, compromising device firmware requires root privileges that allow running the kernel space and modifying the firmware file system. However, the attacker also required advanced skills involving reverse engineering of firmware, working with non-standard processors, dealing with the boot process, and diversity of versions between different devices. Despite its complexity, attacks on the devices' firmware are not beyond the capabilities of motivated attackers, as demonstrated in the past [82], [83], [84], [85]. Moreover, previous work shows that firmware suffers from a wide

TABLE 4
Different Attack Layers

Attack	Level	Req. root	Potential devices	Attack phases	Examples	Attacker skills
Application	User-level	No	Wireless NICs with supporting chipsets	Remote exploitation	[89], [90], [91], [92], [93]	Intermediate
Device-drivers	Kernel-level	Yes	Laptops, Smartphones, Wireless NICs, Smart TVs, Etc,	Remote exploitation, kernel modules/ shellcodes	[78], [79], [80], [81], [94]	Intermediate-high
Firmware	Firmware	Yes	Virtually any Wi-Fi capable device (E.g, IoT)	Remote exploitation, firmware exploits	[85], [95], [96], [97], [98]	High

range of security vulnerabilities, mainly due to their old code-base or usage of many vulnerable libraries [86], [87]. It is interesting to note that a report by Microsoft in March 2021 finds that, in the past two years, 83% of enterprises have experienced at least one firmware level attack [88].

Table 4 summarizes the three types of attacks, their required privileges, relevant devices, exploration types, and examples from past attacks. Note that although the application-layer attack requires the least privileges and no root access, most vendors keep the support of this feature undocumented. Thus, operating at the kernel and firmware levels would be the attackers' most straightforward and reliable strategy. As noted above, although these attacks are considered advanced, they are not beyond skilled adversaries' capabilities.

6.2 Atheros Chipset

We used the spectral analysis feature within Atheros 802.11n Wi-Fi chipsets to access the radio and PHY layer data. The Atheros chips (AR92xx and AR93xx) can report the data of the raw FFT measurement data from the baseband processor to the software stack. The data consists of a vector of FFT bins for 56 subcarriers of the 20 MHz bandwidth channels. The data includes the absolute magnitude $(abs(i) + abs(q))$ for each bin, an index for the strongest FFT bin, and the maximum signal magnitude.

Figs. 8 and 9 show Wi-Fi channel 3 with and without AIR-FI transmission, respectively. The 56 bins of FFT are measured by the Atheros Wi-Fi chipset and delivered to the application layer. As can be seen, with the AIR-FI transmission, the amount of energy in the 2.424 GHz frequency bin is significantly higher than other bins in this channel, with an SNR value of 9 dB.

6.3 Reception Modes

The Atheros chips support two main modes of reception: (1) scanning mode, and (2) triggering mode.

6.3.1 Scanning Mode

In this mode, the FFT information is returned for every Wi-Fi channel when a channel scan is performed. This can stop the Wi-Fi reception from the several hundred milliseconds it takes to scan the whole spectrum. This mode is maintained by setting the `chanscan` value to the `spectral_scan_ctl` control device. The attacker can use this mode to search for a covert transmission if the channel is unknown in advance.

6.3.2 Triggering Mode

In this mode, the FFT information is returned for a specific Wi-Fi channel when the Wi-Fi is operating. This mode is maintained by setting the value of the `spectral_scan_ctl` control device to `manual` and then initiating

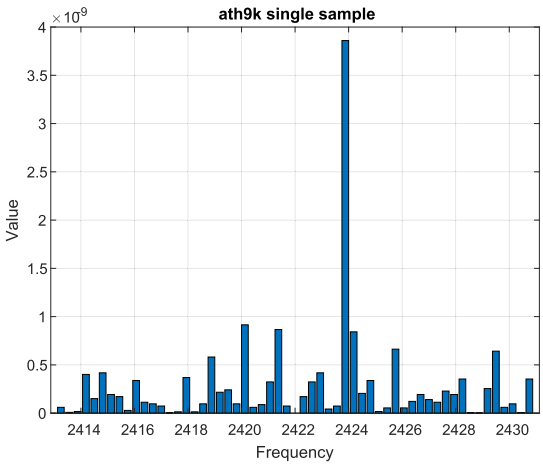


Fig. 8. The FFT measurements of Wi-Fi channel 3 as measured by the Atheros Wi-Fi receiver, with a transmission from the air-gapped computer. The signal can be seen in the 2424 MHz bin.

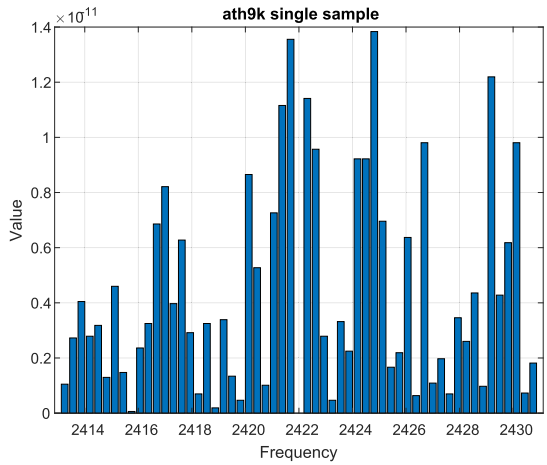


Fig. 9. The FFT measurements of Wi-Fi channel 3 as measured by the Atheros Wi-Fi receiver, without a transmission from the air-gapped computer.

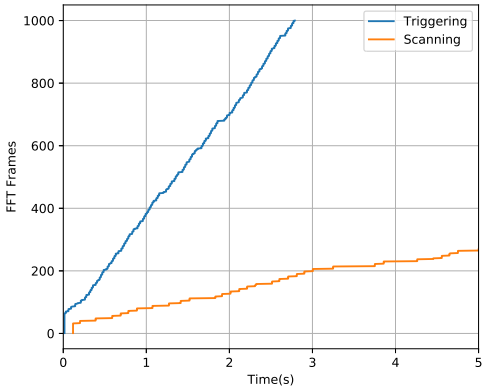


Fig. 10. Number of FFT frames received in the scanning and triggering modes.

`trigger` commands. The scan samples are returned continuously from the channel currently configured.

As seen in Fig. 10, the triggering mode is considerably faster than the scanning mode. The graph shows the number of FFT frames received in the scanning and triggering modes over five seconds. The scanning mode can be used by malware to search the AIR-FI transmissions if the operational frequency is unknown in advance. After detecting a transmission, the malware can begin to operate in the triggering mode to receive the actual data (Fig. 11).

Algorithm 2. Demodulate(deviceAddress, Freq, sampleRate, bufferSize, bitTime, windowSize)

1: $enabled \leftarrow False$
2: $ctx \leftarrow setupContext(deviceAddress)$
3: $rxbuf \leftarrow setupRxBuf(ctx, freq, sampleRate, \leftarrow .., bufferSize)$
4: 5: **while** $True$ **do**
6: $rxbuf.refill()$
7: $buffer = rxbuf.read()$
8: $windows = splitToWindows(buffer, windowSize)$
9: **for** $window$ *in* $windows$ **do**
10: $spectrum = welch(window)$
11: $sampleValue \leftarrow spectrum[0]$
12: $sample \leftarrow [getCurrentTime(), sampleValue]$
13: $samples.append(sample)$
14: **end for**
15: **if** *not enabled* **then**
16: $thresh, enabled \leftarrow detectEnable(samples, bitTime)$
17: **end if**
18: **while** $enabled$ *and* $enoughSamplesForBit \leftarrow (samples, bitTime)$ **do**
19: $bit \leftarrow samplesToBit(samples, bitTime, thresh)$
20: $output(bit)$
21: **end while**
22: **end while**

6.4 Demodulation

The pseudo-code of the demodulator is presented in Algorithm 2. We provide the implementation for software-defined radio (SDR) receivers.

a) Atheros Wi-Fi Chip: Note that the implementation for the Atheros Wi-Fi receiver is based on the same concepts of the SDR code shown in Algorithm 2. However, the Atheros

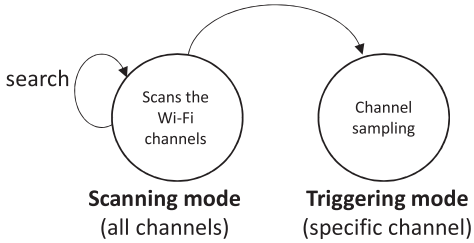


Fig. 11. The transition between the scanning and triggering mode.

implementation includes the extra steps of triggering the `spectral_scan_ctl` device, and receiving, buffing, decoding, and parsing the FFT frames exposed by the Atheros chip. To simplify the discussion and since we are not considering this as the main contribution of our work, we omitted the chip-specific details from the demodulation algorithm.

Algorithm 3. DetectEnable(samples, bitTime)

1: $enableSequence \leftarrow [1, 0, 1, 0, 1, 0, 1, 0]$
2: $samplesDuration \leftarrow getSamplesDuration(samples)$
3: $bitsInSamples \leftarrow samplesDuration/bitTime$
4: **if** $bitsInSamples < 2 * len(enableSequence)$ **then**
5: $return\ 0, False$
6: **end if**
7:
8: $calculatedCorr \leftarrow calculateSampleCorrelationToBits \leftarrow (samples, enableSequence, bitTime)$
9: **if** $calculatedCorr < CORR\_THRESH$ **then**
10: $samples[:] \leftarrow samples[1:]$
11: $return\ 0, False$
12: **end if**
13: 14: $maxCorr \leftarrow calculatedCorr$
15: $maxCorrIndex \leftarrow 0$
16: **for** $index$ *in* $range(1, len(samples))$ **do**
17: $calculatedCorr \leftarrow calculateSampleCorrelationToBits (samples[index:], enableSequence, bitTime)$
18: **if** $calculatedCorr > maxCorr$ **then**
19: $maxCorr \leftarrow calculatedCorr$
20: $maxCorrIndex \leftarrow index$
21: **end if**
22: **end for**
23: 24: $samples[:] = samples[maxCorrIndex:]$
25: $enableSamples = extractEnableSamples(samples)$
26: $thresh = calculateThresh(enableSamples)$
27: $return\ thresh, True$

The OOK demodulator is based on sampling and processing the FFT information for the specific Wi-Fi channel. In lines 2-3, the SDR device is initialized, and the receiving buffer is configured with the frequency (in MHz) of the channel to monitor the sampling rate and the buffer size. The demodulator continuously samples the data in the required frequency and splits it into windows of `windowSize` size. The algorithm estimates the power spectral density for each window using Welch's method (lines 9-14). It then detects the *enable sequence* (`10101010`) using the `detectEnable` routine (Algorithm 3), and determines the thresholds (amplitudes) for '1' and '0' bits (lines 15-18). Finally, the bits are demodulated and added to the output vector (lines 18-21).

TABLE 5
Receivers Used in the Evaluation

Receiver #	Device	Specs
SDR	ADALM-PLUTO	Frequency range from 325 MHz to 3.8 GHz, based on AD9363 transceiver
Wi-Fi	TL-WN722N V1.10	Frequency range from 2.4 GHz to 2.4835 GHz, 4 dBi detachable omni directional antenna

TABLE 6
The Workstations Used for the Evaluation

PC	Hardware	RAM	OS
WORKSTATION-1	ASRock ATX DDR4 X99 Extreme4 CPU-Intel Core i7-6900K @ 3.2Ghz- 16 cores	Crucial 4 * 4GB DDR4 SRAM 2.4GHz RAM clock	Ubuntu 18.04.1 4.15.0-72-generic
WORKSTATION-2	ASRock ATX DDR4 X99 Extreme4 CPU-Intel Core i7-6900K @ 3.2Ghz- 16 cores	SK Hynix 4 * 4GB DDR4 SRAM 2.4GHz RAM clock	Ubuntu 18.04.1 4.15.0-72-generic
WORKSTATION-3 (overclocked)	X99-UD4-CF Intel Core i5-5820K	4 * 4GB DIMM DDR3 2133MHz Micron	Ubuntu 18.04.2 5.0.0-36-generic
WORKSTATION-4 (overclocked)	H97M-D3H Intel Core i7-4790	4 * 4GB DIMM DDR3 1600MHz Hynix	Ubuntu 18.04.1 4.15.0-72-generic

7 Evaluation

This section presents the analysis and evaluation of the AIR-FI covert channel. We describe the experimental setup and test the different reception modes used to maintain the covert channel. We also evaluate the efficacy of the covert channel in virtualized environments.

7.1 Experimental Setup

7.1.1 Receivers

We used two types of receivers:

- A software-defined radio (SDR) receiver, for the SNR measurements.
- A USB Wi-Fi network adapter.

Table 5 contains the specs of the receiver devices. The ADALM-PLUTO SDR is capable of sampling the Wi-Fi frequency band and has RF coverage from 325 MHz to 3.8 GHz. The TL-WN722N Wi-Fi USB wireless network adapter is equipped with the Atheros AR9271 chipset, supporting spectral scan capabilities. During the evaluation, we connected the receivers to a Lenovo ThinkCentre M93p workstation with an Intel Core i7-4785T and Ubuntu 16.04.1 4.4.0 OS.

7.1.2 Transmitters

For the transmission, we used the four types of off-the-shelf workstations listed in Table 6. WORKSTATION-1 and WORKSTATION-2 were installed with two standard DDR4 2400 MHz modules. WORKSTATION-3 and WORKSTATION-4 were equipped with DDR3 modules (2133 MHz and 1600 MHz, respectively). WORKSTATION-3 and WORKSTATION-4 were used to evaluate the attack scenario in which the memory is maliciously overclocked to reach the Wi-Fi frequency band.

The following subsections present the results obtained for the four workstations. During the experiments, we transmitted sequences of frame packets. We tested three receiver modes: (1) the SDR (raw SNR), (2) the Wi-Fi adapter operating in the scanning mode, and (3) the Wi-Fi adapter operating in the triggering mode. We measured the raw SNR values using the SDR receiver, and the BER values were calculated using the Wi-Fi receiver.

7.2 WORKSTATION-1,2 (2.4 GHz)

With WORKSTATION-1 we evaluate the attack on a workstation with 2.4 GHz SK Hynix (Hyundai) memory modules. With WORKSTATION-2 we evaluate the attack on a workstation with 2.4 GHz Crucial memory modules. These DDR4 workstations don't require overclocking, and the whole attack could operate from user mode.

Fig. 12 presents the signal generated from WORKSTATION-1 with all cores participating in the transmission. A signal with a bandwidth of 1 kHz exists in the 2423.804 - 2423.805 MHz range. In this case, the preamble sequence (10101010) can be seen at the beginning of the transmission. The signal generated by WORKSTATION-1 interferes with channels 3, 4, and 5. Table 7 presents the signal-to-noise ratio (SNR) and BER results with WORKSTATION-1. Note that due to the local ramifications and interference, the

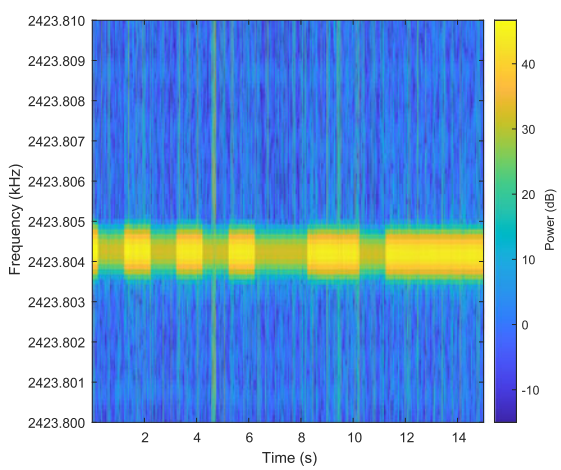


Fig. 12. A transmission from WORKSTATION-1.

TABLE 7
The SNR/BER Measurements of WORKSTATION-1

Distance (cm)	0	30	60	90	120	150	180	210
SNR	14 dB	10 dB	13 dB	5 dB	18 dB	13 dB	20 dB	3 dB
BER (scanning)	0%	0%	0%	0%	0%	0%	0%	16.67%
BER (triggering)	0%	0%	0%	8.33%	0%	4.16%	0%	0%

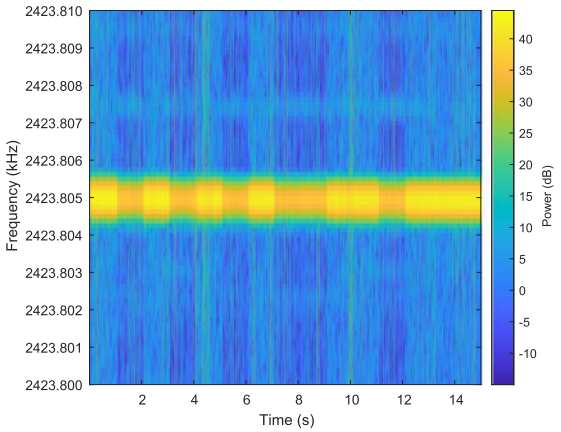


Fig. 13. A transmission from WORKSTATION-2.

signal quality may vary with the distance and location of the receiver. In scanning mode, we were able to maintain BER of 0% for the entire distance range of 0 - 180 cm, with a bit rate of 1 bit/sec. In triggering mode, we maintained a BER of 0 - 8.33% for the range of 0 - 210 cm, with a bit rate of 10 bit/sec.

Fig. 13 presents the signal generated from WORKSTATION-2 with all cores participating in the transmission. A signal with a bandwidth of 1 kHz exists in the 2423.8045 - 2423.8055 MHz frequency range. In this case, the preamble sequence (10101010) can be seen at the beginning of the transmission. The signal generated by WORKSTATION-2 overlaps with channels 3, 4, and 5. Table 8 presents the signal-to-noise ratio (SNR) and BER results with WORKSTATION-2. In scanning mode, we maintained a BER of 0% for the entire range of 0 - 270 cm, with a bit rate of 1 bit/sec. In triggering mode, we maintained a BER of 0 - 4.16% for the range of 0 - 210 cm, with a bit rate of 16 bit/sec.

7.3 WORKSTATION-3,4 (Overclocked)

With WORKSTATION-3 we evaluate the attack on an overclocked workstation with 2133 MHz memory modules. With WORKSTATION-4 we evaluate the attack on an overclocked workstation with 1600 MHz memory modules.

Table 9 presents the signal-to-noise ratio (SNR) and BER results with WORKSTATION-3. The workstation DRAM was overclocked to 2.4 GHz to target the Wi-Fi frequency bands. In scanning mode, we maintained a BER of 3% for the entire range of 0 - 300 cm, with a bit rate of 1 bit/sec. In triggering mode, we maintained a BER of 4.0 - 12.7 % for the range of 0 - 300 cm, with a bit rate of 16 bit/sec.

Table 10 presents the signal-to-noise ratio (SNR) and BER results with WORKSTATION-4. In scanning mode, we were able to maintain a BER of 0% for the entire range of 0 - 800 cm, with a bit rate of 1 bit/sec. In triggering mode, we were able to maintain a BER of 0 - 0.17% for the range of 0 - 800 cm, with a bit rate of 16 bit/sec.

7.4 Channels

AIR-FI can be adjusted to transmit in each of 11 Wi-Fi channels. We measured the SNR values of AIR-FI transmission in 2.4 GHz Wi-Fi channels 1 - 11 in our lab. We configured the timing parameters of the AIR-FI transmitter to target each of the standard 802.11 channels. Fig. 15 shows the FFT measurements of channels 1-11 as measured by the Atheros Wi-Fi receiver, with a transmission from WORKSTATION-1. The AIR-FI signals can be seen in different frequencies of the channel. Table 12 summarizes the SNR values measured in each case. The SNR values ranged from 4.5 dB in channel 5 to 13 dB in channel 6. Note that the SNR of Wi-Fi channels is highly dependent on the specific environment, the receiver location, and the Wi-Fi channels in use. In our lab, channels 1, 5, and 7 were mostly used by the local Wi-Fi routers, yielding lower SNR of 5 dB, 4.5 dB, 8 dB, and 10 dB, respectively.

7.5 Non-Standards Channels

The 2.4 GHz Wi-Fi channels 1-11 evaluated above are the standard and most common channels used worldwide [99], [100], [101]. The 2.4 GHz Wi-Fi channels 12 and 13 are permitted only in low-power mode (low-power transmission) to avoid interference with satellite phones and other low-speed data communication devices [100], [101]. Channel 14 is completely banned and allowed only in Japan [100], [101]. The limitations on these extra channels are specified in Table 11. Note that due to the FCC regulations that ban using channel 14 even in low-power mode, we didn't conduct in-lab experiments with channel 14. We tested AIR-FI in permitted channels 12-13 and measured the corresponding SNR

TABLE 8
The SNR/BER Measurements of WORKSTATION-2

Distance (cm)	0	30	60	90	120	150	180	210	240	270
SNR	13 dB	14 dB	6 dB	5dB	11dB	8 dB	10 dB	4 dB	4 dB	3 dB
BER (scanning)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
BER (triggering)	0%	4.16%	4.16%	0%	4.16%	0%	0%	0%	-	-

TABLE 9
The SNR/BER Measurements of WORKSTATION-3

Distance (cm)	0	50	100	150	200	250	300
SNR	15 dB	14 dB	10 dB	10 dB	9 dB	8 dB	8 dB
BER (scanning)	0 %	0%	0.17%	1.0%	2.0%	2.0%	3.1%
BER (triggering)	6.8%	12.7%	14.7%	5.6%	4%	11.9%	10.2%

TABLE 10
The SNR/BER Measurements of WORKSTATION-4

Distance (cm)	0	100	200	300	400	500	600	700	800
SNR	20 dB	18 dB	17 dB	17 dB	16 dB	15 dB	13 dB	9 dB	7 dB
BER (scanning)	0%	0%	0%	0%	0%	0%	0%	0%	0%
BER (triggering)	0.04%	0.09%	0.02%	0.06%	0.17%	0%	0.1%	0.08%	0%

TABLE 11
Restrictions of the Non-Standard Channels 12,13,14

Channel	Range	Restrictions
12	2456-2478	Low-power mode only
13	2461-2483	Low-power mode only
14	2473-2495	Illegal worldwide (except in Japan)

values. Our evaluation shows that channels 12 and 13 yielded SNR of 6.5 dB and 9 dB, respectively, as shown in Fig. 14.

Note that channels 12-13 are rarely used by home and public Wi-Fi routers, so attackers may choose them to maintain the covert channel due to the lack of potential interferences. However, in the context of covertness and stealth, the transmission activities in these non-standard channels are anomalous (relatively to channels 1-11). Therefore, one might easily detect them by monitoring the Wi-Fi spectrum in the area.

7.6 Threat Radius

Table 13 summaries the bitrate and max distances measured. The results indicate that the covert channel can be used to transfer data with bit rates of 16 bit/sec at distances of 0 - 200 cm with off-the-shelf workstations. The bit rates values depend on the compromised Wi-Fi devices in the area and the reception mode in use (e.g., scanning or triggering). Such bit rates could be used to exfiltrate information such as binary data, keystrokes logging, and text files. The covert channel could be effective for greater distances (up to 800 cm) when using overclocked workstations. The main difference in the effective distance between DDR4 and DDR3-overclocked is rooted in their operating voltage; DDR3 operates in 1.5V while DDR4 operates in 1.2V. The

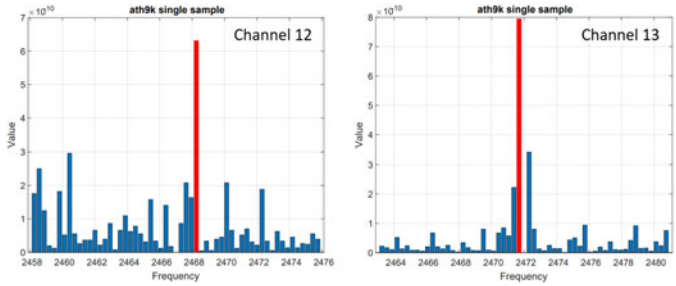


Fig. 14. AIR-FI transmissions in channels 12,13.

25% higher operating voltage of DDR3 generates more current flow in the RAM bus and which results in more electromagnetic radiant energy which propagates farther. Note that previous work [29], [33], [34] mainly used DDR3 in their evaluation.

7.7 Virtual Machines (VMs)

Virtualization technologies are commonly used in modern IT environments. One of their advantages is the isolation of hardware resources they enforce. Hypervisors/virtual machine monitors (VMMs) provide a layer of abstraction between the virtual machine and the physical hardware (CPU and peripherals). Since the covert channel is closely related to the memory access timing, we examined whether the virtualization layer caused interruptions and delays, which may affect the signal quality. Generally speaking, in Intel VT-x, mapping the guest's physical addresses and the host physical address is done through the extended page table (EPT). With the EPT, for each memory access operation, the MMU maps the guest's linear address to the host physical address (Fig. 16). Note that the measurements show that

TABLE 12
SNR Values of AIR-FI Transmission in Channels 1-11

Channel	1	2	3	4	5	6	7	8	9	10	11
AIR-FI frequency (GHz)	2.411	2.414	2.421	2.428	2.432	2.436	2.442	2.446	2.452	2.454	2.461
SNR	5 dB	6 dB	11.5 dB	10 dB	4.5 dB	13 dB	8 dB	10 dB	8 dB	10 dB	10 dB

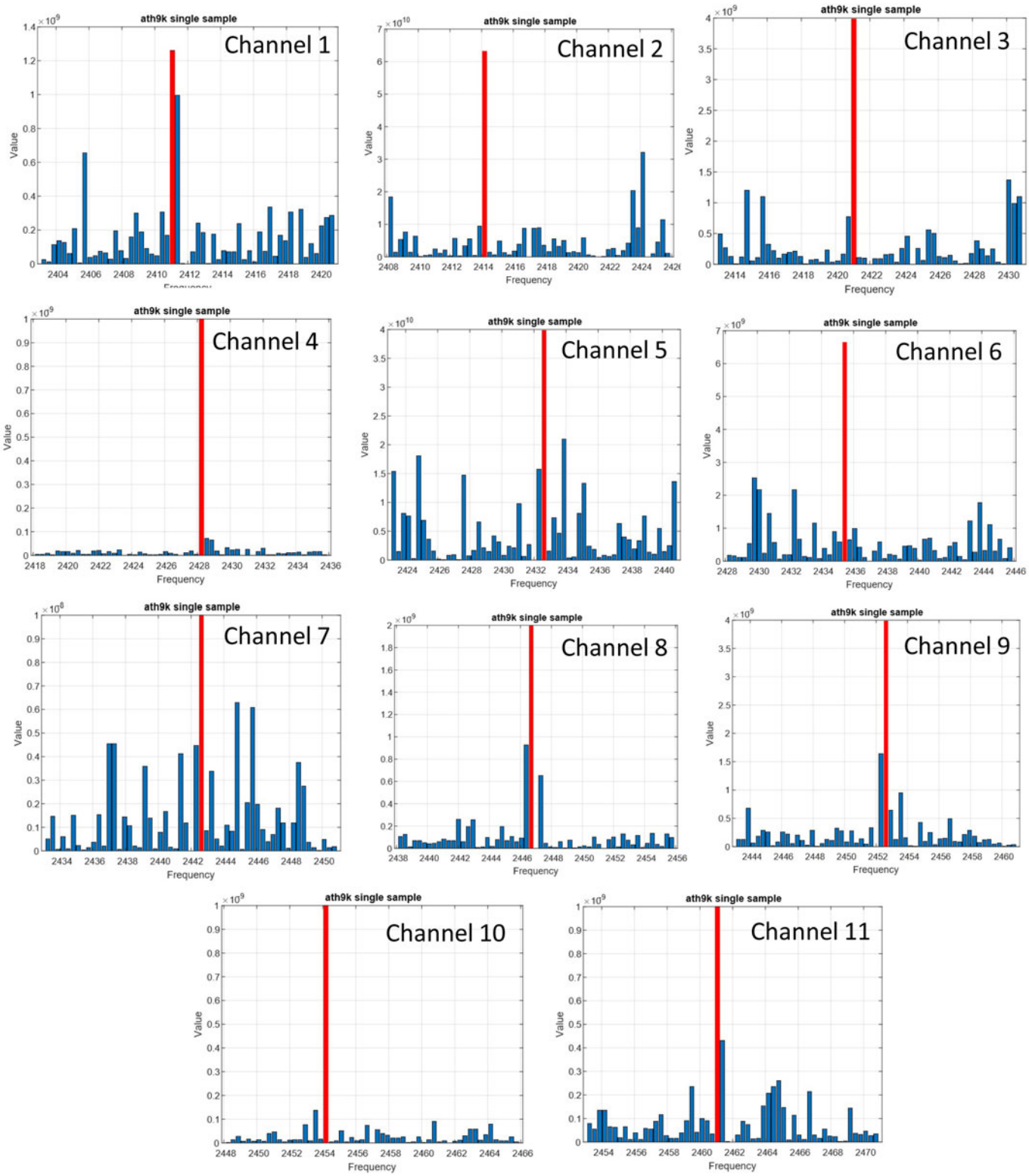


Fig. 15. AIR-FI transmissions in various Wi-Fi channels. The FFT measurements of channels 1 - 11 as measured by the Atheros Wi-Fi receiver, with AIR-FI transmissions from WORKSTATION-1. The relevant bins are marked in red.

this level of indirection may increase memory access latencies for some workloads [5]. In addition to translating a guest's physical address to a host physical address, EPT might cause extra delay by triggering VM-exit transitions (transitions between the VM and the hypervisor). We examined a sequence of transmissions from WORKSTATION-1, WORKSTATION-2, and WORKSTATION-3 using three setups: a bare-metal machine, a VMware VMM, and a VirtualBox VMM. Table 14 contains details on the systems examined. Our experiments show that the covert channel can be maintained from within virtual machines. In a term of signal strength (amplitude) we measured a difference of at

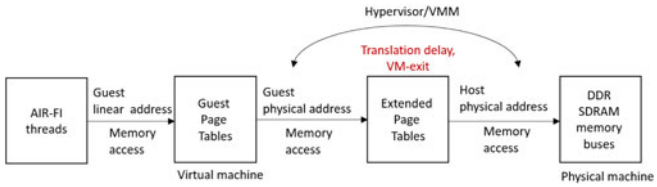


Fig. 16. AIR-FI and the Extended Page Table (EPT) memory translation.

TABLE 13
Bitrate and Distances Summary

#	Transmitter	Scanning	Triggering	Max distance
1	WORKSTATION-1	1 bit/sec	10 bit/sec	210 cm
2	WORKSTATION-2	1 bit/sec	16 bit/sec	270/210 cm
3	WORKSTATION-3	1 bit/sec	16 bit/sec	300 cm
4	WORKSTATION-4	1 bit/sec	16 bit/sec	800 cm

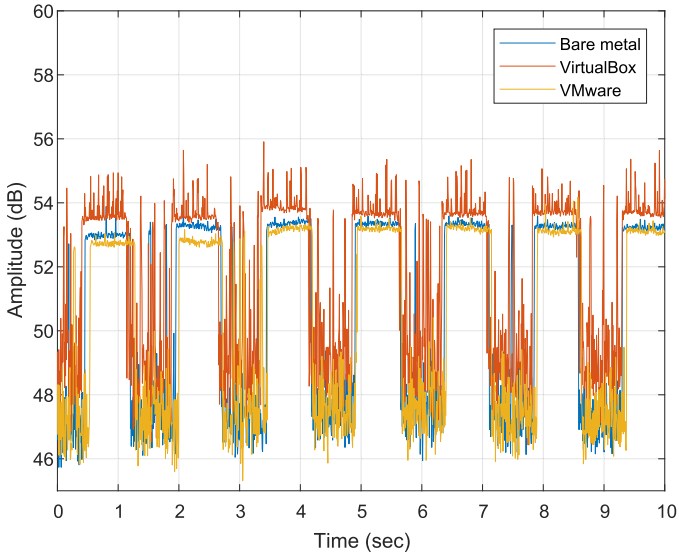


Fig. 17. AIR-FI signal generated on bare-metal, VMware and VirtualBox.

most 1 dB between the bare metal, VMware, and VirtualBox transmissions. Moreover, we measured negligible timing differences between the VM and bare-metal signals with less than 1% difference in each signal's beginning and/or ending. The results are depicted in Fig. 17 which shows a sequence of alternating bits transmitted from within bare-metal, VMware, and VirtualBox environments.

7.8 Multiple Covert Channels

It is possible to increase the bandwidth of the covert channel by using multiple transmitters and receivers for the data exfiltration. Our experiments show that there are two practical ways to maintain multiple covert channels;

- *Multiple transmitters, single receiver.* In this method, the attacker uses several workstations in the area to transmit the data to a single receiver. Note that since the receiver is locked on a specific channel, each transmitter in the area has to use a different, predefined sub-carrier within a specific channel to avoid collision with other transmitters. The transmitting workstations must be located near the receiver, on the same desk, or in the same room. Theoretically, by using multiple transmitters and a single receiver, the attacker can increase the bandwidth with factor N, where N is the number of transmitters participating in the communication. The upper bound is the number of potential sub-carriers in each channel, which is 52 in our case. However, it is important to note that real-life scenarios may involve only a few transmitting workstations close enough to a single receiver due to practical constraints of space. For example, with four concurrent transmitters ($N = 4$) exfiltrating at 16 bit/sec, we increase the bandwidth to 64 bit/sec.
- *Multiple transmitters, multiple receivers.* In this method, the attacker can use many workstations to transmit the data to different receivers, e.g., employees' smartphones. The transmitters and receivers can be located anywhere in the organization and are not limited to a specific channel. The main limitation of this approach is distributing the information to all computers participating and synchronizing between them. Theoretically, by using multiple transmitters and receivers, the attacker can increase the bandwidth with factor R, where R is the number of receivers participating in the communication. For example, with ten concurrent transmitters and receivers ($R = 10$) at 16 bit/sec, we increase the bandwidth to 160 bit/sec.

Table 15 list the differences between the two methods in terms of limitations and bandwidth.

8 Countermeasures

Several defensive approaches can be used against the proposed covert channel.

8.1 Physical Separation

TEMPEST (Telecommunications Electronics Materials Protected from Emanating Spurious Transmissions) threats

TABLE 14
AIR-FI in Virtualization

#	Host	VMM/Hypervisor	Guest	SNR (dB)
Bare metal	Ubuntu 18.04.1 5.3.0-53-generic	N/A	N/A	5.09
Virtualbox	Ubuntu 18.04.1 5.3.0-53-generic	Virtualbox: 6.0.22 r137980	Ubuntu 18.04.1 5.3.0-28-generic	4.36
VMware	Ubuntu 18.04.1 5.3.0-53-generic	VMware Player: 15.5.2 build-15785246	Ubuntu 18.04.1 5.3.0-28-generic	5.32

TABLE 15
Multiple Transmitters

Method	Limitations	Channels	Speedup
Multiple transmitters, single receiver	Transmitters close to a receiver	Same channel, different sub-carriers	$min\{N, 52\}$
Multiple transmitters, multiple receivers	Synchronization and distribution of data	Different channels	R

deals with radiated and conducted electromagnetic waves of equipment. The U.S and NATO telecommunication security standards (e.g., NATO SDIP-27, NATO AMSG, NATO Zones, NSTISSAM TEMPEST/2-95 [102]) propose zone separation to protect against TEMPEST threats and other types of radiated energy attacks. In this approach, Wi-Fi transceivers are not allowed in certain classified areas. The NATO zoning procedure defines measures in which areas within a secured perimeter are classified as zone 0 to zone 3, depending on the safety requirements of the specific asset. In our case, Wi-Fi-capable devices, such as smartphones, smartwatches, and laptops, should be banned from the area of air-gapped systems.

8.2 Runtime Detection

The signal generation algorithm is based on memory operations that trigger the DDR SDRAM emissions. Host-based intrusion detection systems can monitor the activity of the processes in the OS. In our case, a process that abnormally performs memory transfer operations would be reported and inspected. A challenge to the runtime detection approach is that the signal generation algorithm (presented in Section 5) involves bare memory operations such as `memcpy()`.

8.3 Memory Access Monitor

Monitoring the memory access instructions and system calls at runtime necessitates sandboxing or debugging of the process, which severely degrades performance [29], and can easily be bypassed by malware using rootkit techniques [103]. In our case, the malware may inject a shellcode with a signal generation code into a legitimate, trusted process to bypass the security products or use other bypass techniques such as process hollowing [104]. To overcome these evasion techniques, it is possible to employ solutions such as MemoryMonRWX, which is a bare-metal hypervisor that can track and trap all types of memory access: read, write, and execute [105]. However, all these detection techniques would likely suffer from high rates of false alarms since many processes intensively use the memory for legitimate needs (e.g., image processing, matrix calculations, etc.).

8.4 Wi-Fi Monitoring

Another approach is to use Wi-Fi monitoring hardware equipment in order to identify anomalies in the PHY layer of the Wi-Fi channels in the 802.11 bands [104], [106]. For example, researchers suggest using Machine Learning, and RF inspection techniques for covert channels detection [107]. However, the Wi-Fi monitoring approach is less effective in environments where Wi-Fi devices are allowed. Due to the legitimate activities of local access points and devices operating on the Wi-Fi channels, such a detection approach will lead to many false positives.

8.5 Signal Jamming (Hardware)

It is possible to block the covert channel by jamming the Wi-Fi frequency bands in the area. Modern Wi-Fi jammers are signal blocking devices with radio frequency (RF) hardware that transmits radio waves in the entire range of Wi-Fi frequency bands (2.4 / 5.0 GHz). A typical Wi-Fi jammer generates high power, constant radio transmissions which span the channels and mask any legitimate or non-legitimate Wi-Fi transmissions [108].

8.6 Signal Jamming (Software)

In this approach, a background activity that performs random memory operations is launched. The random workloads interfere with the execution of the malicious process and hence, interrupt the generation of the electromagnetic wave emanated from the memory buses. Fig. 18 shows the noise generated by WORKSTATION-1 when intensive prime number calculations were executed on one to eight cores using the `matho-primes` Linux command. Our measurements show that processes bound to six and eight cores can significantly reduce the SNR of the original signal to SNR levels of 4.8 dB 3.1 dB, respectively.

8.7 Faraday Shielding

Faraday shielding is a particular type of container used to block or limit the electromagnetic fields from interfering with or emanating from the shielded system. Faraday shielding copes with the threat presented in this paper by preventing the leakage of Wi-Fi signals from the shielded case. Generally, the computer shielding involves encompassing the computer in a Faraday cage that does not permit stray

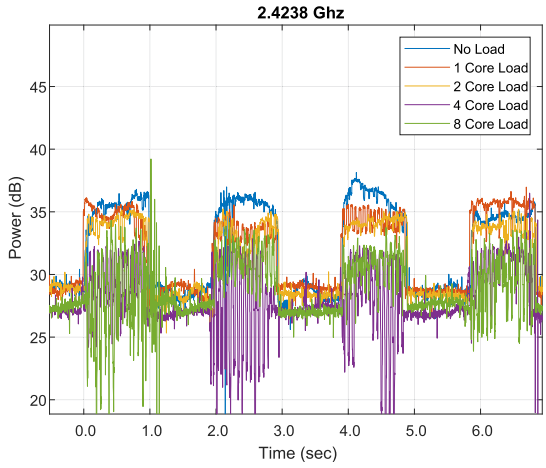


Fig. 18. Signal jamming using intensive CPU operations.

TABLE 16
Countermeasures

#	Method	Type
1	Separation & NATO zoning	Procedural
2	Runtime process monitoring, OS-level (user/kernel)	Software
3	Memory access monitoring, hypervisor-level	Software/hardware
4	Wi-Fi PHY & RF monitoring	Hardware
5	Signal jamming (electromagnetic)	Hardware
6	Memory operations jamming	Software
7	Faraday shielding & Faraday rooms	Physical

electromagnetic emanations. Physical isolation in which the whole room functions as an integral Faraday cage is also an option [109]. While this solution can be used in certain cases, it is impractical as a large-scale solution [108]. The defensive countermeasures are listed in Table 16.

9 Conclusion

This paper demonstrated a new attack that enables the exfiltration of information from air-gapped computers to nearby Wi-Fi receivers. The AIR-FI malware invokes signals in the 2.4 GHz Wi-Fi frequency bands. The signals are generated through DDR SDRAM buses and do not require special Wi-Fi hardware. Binary data can be modulated and encoded on top of the signals. We showed that a compromised nearby Wi-Fi device (e.g., smartphones, laptops, and IoT devices) could intercept these signals and decode the data. We utilized the low-level physical layer information that the Wi-Fi chips expose to the application layers to extract the signals. We implemented transmitters and receivers in different reception modes and discussed design considerations and implementation details. We evaluated this covert channel in terms of bandwidth and distance and presented a set of countermeasures. Our results show that the covert channel can be effective at distances up to several meters from air-gapped computers. We achieved effective bit rates ranging from 1 to 16 bit/sec.

References

[1] P. N. Bahrami, A. Dehghantanha, T. Dargahi, R. M. Parizi, K.-K. R. Choo, and H. H. Javadi, "Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures," *J. Inf. Process. Syst.*, vol. 15, no. 4, pp. 865–889, 2019.

[2] May-2018_government-security-classifications-2.pdf. Accessed: Sep. 05, 2022. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/715778/May-2018_Government-Security-Classifications-2.pdf

[3] Post | dreamlab technologies. Accessed: Sep. 05, 2022. [Online]. Available: https://dreamlab.net/en/blog/post/bypassing-air-gaps-in-ics-systems/

[4] Storefront - Top secret/sensitive compartmented information data. Accessed: Sep. 05, 2022. [Online]. Available: https://storefront.disa.mil/kinetic/disa/service-catalog#/forms/top-secretsensitive-compartmented-information-data

[5] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Inf. Sharing Anal. Center*, vol. 388, 2016.

[6] D. Kushner, "The real story of stuxnet," *IEEE Spectr.*, vol. 3, no. 50, pp. 48–53, Mar. 2013.

[7] Hackers target the air-gapped networks of the taiwanese and philippine military | zdnet. Accessed: Sep. 05, 2022. [Online]. Available: https://www.zdnet.com/article/hackers-target-the-air-gapped-networks-of-the-taiwanese-and-philippine-military/

[8] Russian hackers reach U.S. utility control rooms, homeland security officials say - wsj. Accessed: Sep. 05, 2022. [Online]. Available: https://www.wsj.com/articles/russian-hackers-reach-u-s-utility-control-rooms-homeland-security-officials-say-1532388110

[9] Tick espionage group is likely trying to hop air gaps, researchers say. Accessed: Sep. 05, 2022. [Online]. Available: https://www.cyberscoop.com/tick-espionage-usb-air-gaps-palo-alto-networks/

[10] Eset research discovers cyber espionage framework ramsay | eset. Accessed: Sep. 05, 2022. [Online]. Available: https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-discovers-cyber-espionage-framework-ramsay/

[11] India's largest nuclear power plant and the truth about air-gapping. Accessed: Sep. 05, 2022. [Online]. Available: https://www.missionsecure.com/blog/cyber-attack-india-largest-nuclear-plant-truth-air-gapping

[12] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 3, pp. 44–57, Third Quarter 2007.

[13] M. Guri and Y. Elovici, "Bridgeware: The air-gap malware," *Commun. ACM*, vol. 61, pp. 74–82, Mar. 2018.

[14] B. Carrara, "Air-gap covert channels," PhD thesis, Université d'Ottawa/University of Ottawa, Ottawa, Canada, 2016.

[15] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," in *Proc. 9th Int. Conf. Malicious Unwanted Softw.: The Americas*, 2014, pp. 58–67.

[16] M. G. Kuhn and R. J. Anderson, "Soft tempest: Hidden data transmission using electromagnetic emanations," in *Proc. Int. Workshop Inf. Hiding*, 1998, vol. 1525, pp. 124–142.

[17] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proc. 18th Conf. USENIX Secur. Symp.*, 2009, pp. 1–16.

[18] B. Carrara and C. Adams, "On acoustic covert channels between air-gapped systems," in *Proc. Int. Symp. Found. Pract. Secur.*, 2014, pp. 3–16.

[19] M. Guri, Y. Solewicz, and Y. Elovici, "Fansmitter: Acoustic data exfiltration from air-gapped computers via fans noise," *Comput. Secur.*, vol. 91, 2020, Art. no. 101721.

[20] J. Loughry and D. A. Umphress, "Information leakage from optical emanations," *ACM Trans. Inf. Syst. Secur.*, vol. 5, no. 3, pp. 262–289, 2002.

[21] M. Guri, B. Zadov, and Y. Elovici, *LED-It-GO: Leaking (A Lot of) Data From Air-Gapped Computers via the (Small) Hard Drive LED*. Cham, Switzerland: Springer International Publishing, 2017, pp. 161–184.

[22] M. Guri, B. Zadov, and Y. Elovici, *LED-It-GO: Leaking (A Lot of) Data From Air-Gapped Computers via the (Small) Hard Drive LED*. Cham, Switzerland: Springer International Publishing, 2017, pp. 161–184.

[23] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations," in *Proc. IEEE 28th Comput. Secur. Found. Symp.*, 2015, pp. 276–289.

[24] M. Guri, "MAGNETO: Covert channel between air-gapped systems and nearby smartphones via CPU-generated magnetic fields," *Future Gener. Comput. Syst.*, vol. 115, pp. 115–125, 2021.

[25] M. Guri, B. Zadov, D. Bykhovsky, and Y. Elovici, "PowerHammer: Exfiltrating data from air-gapped computers through power lines," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1879–1890, Nov. 2019.

[26] M. Guri, "USBCulprit: USB-borne air-gap malware," in *Proc. Eur. Interdiscipl. Cybersecurity Conf.*, 2021, pp. 7–13.

[27] New ramsay malware can steal sensitive documents from air-gapped networks | zdnet. Accessed: Sep. 05, 2022. [Online]. Available: https://www.zdnet.com/article/new-ramsay-malware-can-steal-sensitive-documents-from-air-gapped-networks/

[28] A. Dorais-Joncas and F. Munõz, "Jumping the air gap," 2021.

[29] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici, "GSMem: Data exfiltration from air-gapped computers over GSM frequencies," in *Proc. 24th USENIX Conf. Secur. Symp.*, 2015, pp. 849–864.

[30] M. Guri, M. Monitz, and Y. Elovici, "USBee: Air-gap covert-channel via electromagnetic emission from USB," in *Proc. 14th Annu. Conf. Privacy Secur. Trust*, 2016, pp. 264–268.

[31] N. Sehatbakhsh, B. B. Yilmaz, A. Zajic, and M. Prvulovic, "A new side-channel vulnerability on modern computers by exploiting electromagnetic emanations from the power management unit," in *Proc. IEEE Int. Symp. High Perform. Comput. Architecture*, 2020, pp. 123–138.

[32] M. Guri, "LANTENNA: Exfiltrating data from air-gapped networks via ethernet cables emission," in *Proc. IEEE 45th Annu. Comput. Softw. Appl. Conf.*, 2021, pp. 745–754.

[33] Z. Zhan, Z. Zhang, and X. Koutsoukos, "BitJabber: The world's fastest electromagnetic covert channel," in *Proc. IEEE Int. Symp. Hardware Oriented Secur. Trust*, 2020, pp. 35–45.

[34] C. Shen, T. Liu, J. Huang, and R. Tan, "When LoRA meets EMR: Electromagnetic covert channels can be super resilient," in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 1304–1317.

[35] M. Guri, B. Zadov, and Y. Elovici, "ODINI: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1190–1203, Aug. 2019.

[36] J. Zhang, X. Ji, W. Xu, Y.-C. Chen, Y. Tang, and G. Qu, "MagView: A distributed magnetic covert channel via video encoding and decoding," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 357–366.

[37] N. Matyunin, J. Szefer, S. Biedermann, and S. Katzenbeisser, "Covert channels using mobile device's magnetic field sensors," in *Proc. 21st Asia South Pacific Des. Autom. Conf.*, 2016, pp. 525–532.

[38] M. Guri, B. Zadov, D. Bykhovsky, and Y. Elovici, "CTRL-ALT-LED: Leaking data from air-gapped computers via keyboard LEDs," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf.*, 2019, vol. 1, pp. 801–810.

[39] M. Guri, B. Zadov, A. Daidakulov, and Y. Elovici, "xLED: Covert data exfiltration from air-gapped networks via switch and router LEDs," in *Proc. 16th Annu. Conf. Privacy Secur. Trust*, 2018, pp. 1–12.

[40] M. Guri and D. Bykhovsky, "aIR-Jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (IR)," *Comput. Secur.*, vol. 82, pp. 15–29, 2019.

[41] M. Hanspach and M. Goetz, "On covert acoustical mesh networks in air," 2014, *arXiv:1406.1213*.

[42] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici, "Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise (diskfiltration)," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 98–115.

[43] M. Guri, "CD-LEAK: Leaking secrets from audioless air-gapped computers using covert acoustic signals from CD/DVD drives," in *Proc. IEEE 44th Annu. Comput. Softw. Appl. Conf.*, 2020, pp. 808–816.

[44] M. Guri, "Exfiltrating data from air-gapped computers via vibrations," *Future Gener. Comput. Syst.*, vol. 122, pp. 69–81, 2021.

[45] M. Guri, "POWER-SUPPLaY: Leaking sensitive data from air-gapped, audio-gapped systems by turning the power supplies into speakers," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2021.3133406.

[46] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations," in *Proc. IEEE 28th Comput. Secur. Found. Symp.*, 2015, pp. 276–289.

[47] L. Deshotels, "Inaudible sound as a covert channel in mobile devices," in *Proc. 8th USENIX Conf. Offensive Technol.*, 2014, Art. no. 16.

[48] N. Hou and Y. Zheng, "CloakLoRa: A covert channel over LoRa phy," in *Proc. IEEE 28th Int. Conf. Netw. Protoc.*, 2020, pp. 1–11.

[49] Z. Yang, Q. Huang, and Q. Zhang, "NICScatter: Backscatter as a covert channel in mobile devices," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 356–367.

[50] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Secur. Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.

[51] R. Grant, "The cyber menace," *Air Force Mag.*, vol. 92, no. 3, 2009.

[52] The epic turla (snake/uroburos) attacks | virus definition | kaspersky lab. Accessed: Sep. 05, 2022. [Online]. Available: https://www.kaspersky.com/resource-center/threats/epic-turla-snake-malware-attacks, 2018.

[53] 'red october' diplomatic cyber attacks investigation | securelist. Accessed: Sep. 05, 2022. [Online]. Available: https://securelist.com/red-october-diplomatic-cyber-attacks-investigation/36740/

[54] A fanny equation: "i am your father, stuxnet" - securelist. 2018. Accessed: Sep. 05, 2022. [Online]. Available: https://securelist.com/a-fanny-equation-i-am-your-father-stuxnet/68787/

[55] Russian state hackers appear to have breached a federal agency. Accessed: Sep. 05, 2022. [Online]. Available: https://finance.yahoo.com/news/russia-fancy-bear-reportedly-hacked-us-agency-213304085.html?guccounter=1

[56] More on air gaps | nexor. Accessed: Sep. 05, 2022. [Online]. Available: https://www.nexor.com/more-on-air-gaps/

[57] M. Vanhoef and F. Piessens, "Advanced Wi-Fi attacks using commodity hardware," in *Proc. 30th Annu. Comput. Secur. Appl. Conf.*, 2014, pp. 256–265.

[58] H. Berghel and J. Uecker, "WiFi attack vectors," *Commun. ACM*, vol. 48, no. 8, pp. 21–28, 2005.

[59] E. Ronen and A. Shamir, "Extended functionality attacks on IoT devices: The case of smart lights," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 3–12.

[60] G. Ho, D. Leung, P. Mishra, A. Hosseini, D. Song, and D. Wagner, "Smart locks: Lessons for securing commodity Internet of Things devices," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 461–472.

[61] M. Stute *et al.*, "A billion open interfaces for eve and mallory: MitM, DoS, and tracking attacks on iOS and macOS through apple wireless direct link," in *Proc. 28th USENIX Secur. Symp.*, 2019, pp. 37–54.

[62] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *Proc. 3rd Int. Conf. Electron. Des.*, 2016, pp. 321–326.

[63] O. Shwartz, G. Shitrit, A. Shabtai, and Y. Oren, "From smashed screens to smashed stacks: Attacking mobile phones using malicious aftermarket parts," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, 2017, pp. 94–98.

[64] Update now – Whatsapp flaw gave attackers access to local files – Naked security. Accessed: Sep. 05, 2022. [Online]. Available: https://nakedsecurity.sophos.com/2020/02/06/update-now-whatsapp-flaw-gave-attackers-access-to-local-files/

[65] G. Delac, M. Silic, and J. Krolo, "Emerging security threats for mobile platforms," in *Proc. 34th Int. Conv. MIPRO*, 2011, pp. 1468–1473.

[66] Equationdrug, the hacking platform of equation group aptsecurity affairs. Accessed: Sep. 05, 2022. [Online]. Available: http://securityaffairs.co/wordpress/34769/intelligence/equationdrug-platform-equation-group.html

[67] Projectsauron: Top level cyber-espionage platform covertly extracts encrypted government comms | securelist. Accessed: Sep. 05, 2022. [Online]. Available: https://securelist.com/faq-the-projectsauron-apt/75533/

[68] Wikileaks - emotional simian v2.1 - tdr. Accessed: Sep. 05, 2022. [Online]. Available: https://wikileaks.org/vault7/document/Emotional_Simian-v2_1-TDR/

[69] K. Sharma and B. Gupta, "Attack in smartphone Wi-Fi access channel: State of the art, current issues, and challenges," in *Proc. Next-Gener. Netw.*, 2018, pp. 555–561.

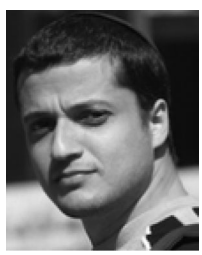
[70] Eclypsium discovers multiple vulnerabilities in dell biosconnect. Accessed: Mar. 23, 2022. [Online]. Available: https://eclypsium.com/2021/06/24/biosdisconnect/

[71] Y. Shah and S. Sengupta, "A survey on classification of cyber-attacks on IoT and IIoT devices," in *Proc. 11th IEEE Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf.*, 2020, pp. 0406–0413.

[72] M. Yu, J. Zhuge, M. Cao, Z. Shi, and L. Jiang, "A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices," *Future Internet*, vol. 12, no. 2, 2020, Art. no. 27.

[73] Uefi as a malware delivery mechanism | kaspersky official blog. Accessed: Mar. 23, 2022. [Online]. Available: https://www.kaspersky.com/blog/mosaicregressor-uefi-malware/37252/

[74] Trickbot now offers 'trickboot': Persist, brick, profit - eclypsium. Accessed: Mar. 23, 2022. [Online]. Available: https://eclypsium.com/2020/12/03/trickbot-now-offers-trickboot-persist-brick-profit/
[75] *en:users:drivers:ath9k:spectral_scan [linux wireless]*. Accessed: Jun. 19, 2022. [Online]. Available: https://wireless.wiki.kernel.org/en/users/drivers/ath9k/spectral_scan
[76] T. Barabosch and E. Gerhards-Padilla, "Host-based code injection attacks: A popular technique used by malware," in *Proc. 9th Int. Conf. Malicious Unwanted Softw.: The Americas*, 2014, pp. 8–17.
[77] M. hias Schulz, D. Wegemer, and M. hias Hollick, "Nexmon: Build your own Wi-Fi testbeds with low-level MAC and PHY-access using firmware patches on o-the-shelf mobile devices," 2017.
[78] Linux threat hunting: 'syslogk' a kernel rootkit found under development in the wild - avast threat labs. Accessed: Jun. 19, 2022. [Online]. Available: https://decoded.avast.io/davidalvarez/linux-threat-hunting-syslogk-a-kernel-rootkit-found-under-development-in-the-wild/
[79] Ghostemperor: From proxylogon to kernel mode | securelist. Accessed: Jun. 19, 2022. [Online]. Available: https://securelist.com/ghostemperor-from-proxylogon-to-kernel-mode/104407/
[80] Android trojan xhelper uses persistent re-infection tactics: Here's how to remove | malwarebytes labs. Accessed: Jun. 19, 2022. [Online]. Available: https://blog.malwarebytes.com/android/2020/02/new-variant-of-android-trojan-xhelper-reinfects-with-help-from-google-play/
[81] pegasus-exploits-technical-details.pdf. Accessed: Jun. 19, 2022. [Online]. Available: https://info.lookout.com/rs/051-ESQ-475/images/pegasus-exploits-technical-details.pdf
[82] April firmware threat report - eclypsium. Accessed: Jun. 19, 2022. [Online]. Available: https://eclypsium.com/2021/04/26/april-firmware-threat-report-2021/
[83] M. Bettayeb, Q. Nasir, and M. A. Talib, "Firmware update attacks and security for IoT devices: Survey," in *Proc. ArabWIC 6th Annu. Int. Conf. Res. Track*, 2019, pp. 1–6.
[84] A. Cui, M. Costello, and S. Stolfo, "When firmware modifications attack: A case study of embedded exploitation," 2013.
[85] Mothership unlocked: The equation apt | kaspersky official blog. Accessed: Jun. 19, 2022. [Online]. Available: https://www.kaspersky.com/blog/mothership-unlocked-the-equation-apt/15052/
[86] A. Qasem, P. Shirani, M. Debbabi, L. Wang, B. Lebel, and B. L. Agba, "Automatic vulnerability detection in embedded devices and firmware: Survey and layered taxonomies," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–42, 2021.
[87] Y. David, N. Partush, and E. Yahav, "FirmUp: Precise static detection of common vulnerabilities in firmware," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 392–404, 2018.
[88] New security signals study shows firmware attacks on the rise; here's how microsoft is working to help eliminate this entire class of threats - microsoft security blog. Accessed: Jun. 19, 2022. [Online]. Available: https://www.microsoft.com/security/blog/2021/03/30/new-security-signals-study-shows-firmware-attacks-on-the-rise-heres-how-microsoft-is-working-to-help-eliminate-this-entire-class-of-threats/
[89] Meet wifidemon: IoS WiFi rce 0-day vulnerability & a 'zero-click' vulnerability that was silently patched. Accessed: Jun. 19, 2022. [Online]. Available: https://blog.zecops.com/research/meet-wifidemon-ios-wifi-rce-0-day-vulnerability-and-a-zero-click-vulnerability-that-was-silently-patched/
[90] Set a record for new linux malware families - intezer. Accessed: Jun. 19, 2022. [Online]. Available: https://www.intezer.com/blog/cloud-security/2020-set-record-for-new-linux-malware-families/
[91] F5 labs investigates malibot | f5 labs. Accessed: Jun. 19, 2022. [Online]. Available: https://www.f5.com/labs/articles/threat-intelligence/f5-labs-investigates-malibot
[92] iOS malware, xcodeghost, infects millions of apple store customers. Accessed: Jun. 19, 2022. [Online]. Available: https://us.norton.com/internetsecurity-emerging-threats-ios-malware-xcodeghost-infects-millions-of-apple-store-customers.html
[93] J. A. Pendergrass *et al.*, "Runtime detection of userspace implants," in *Proc. IEEE Mil. Commun. Conf.*, 2019, pp. 1–6.
[94] Analysis and exploitation of the iOS kernel vulnerability. Accessed: Jun. 19, 2022. [Online]. Available: https://www.synacktiv.com/en/publications/analysis-and-exploitation-of-the-ios-kernel-vulnerability-cve-2021–1782.html
[95] Moonbounce: The dark side of uefi firmware | securelist. Accessed: Jun. 19, 2022. [Online]. Available: https://securelist.com/moonbounce-the-dark-side-of-uefi-firmware/105468/
[96] Hackers infect 500,000 consumer routers all over the world with malware | ars technica. Accessed: Jun. 19, 2022. [Online]. Available: https://arstechnica.com/information-technology/2018/05/hackers-infect-500000-consumer-routers-all-over-the-world-with-malware/
[97] The-top-5-firmware-attack-vectors.pdf. Accessed: Jun. 19, 2022. [Online]. Available: https://eclypsium.com/wp-content/uploads/2020/12/The-Top-5-Firmware-Attack-Vectors.pdf
[98] Eset-lojax.pdf. Accessed: Jun. 19, 2022. [Online]. Available: https://www.welivesecurity.com/wp-content/uploads/2018/09/ESET-LoJax.pdf
[99] J. Lansford, A. Stephens, and R. Nevo, "Wi-Fi (802.11 b) and bluetooth: Enabling coexistence," *IEEE Netw.*, vol. 15, no. 5, pp. 20–27, Sep./Oct. 2001.
[100] I. C. S. L. S. Committee *et al.*, "IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 2007.
[101] List of WLAN channels - wikipedia. Accessed: Jun. 20, 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_WLAN_channels#2.4_GHz_(802.11b/g/n/ax)
[102] https://cryptome.org, "Nstissam tempest/2–95. Accessed: Sep. 05, 2022. [Online]. Available: https://cryptome.org/tempest-2–95.htm, 2000.
[103] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. 6th ACM Symp. Inf. Comput. Commun. Secur.*, 2011, pp. 355–366.
[104] A. Mohanta and A. Saldanha, "Code injection, process hollowing, and API hooking," in *Proc. Malware Anal. Detection Eng.*, 2020, pp. 267–329.
[105] I. Korkin and S. Tanda, "Detect kernel-mode rootkits via real time logging & controlling memory access," 2017, *arXiv: 1705.06784*.
[106] Wi-fi_overview_of_the_802.11_physical_layer.pdf. Accessed: Mar. 23, 2022. [Online]. Available: https://www.cnrood.com/en/media/solutions/Wi-Fi_Overview_of_the_802.11_Physical_Layer.pdf
[107] P. Nowakowski, P. Zórawski, K. Cabaj, and W. Mazurczyk, "Detecting network covert channels using machine learning, data mining and hierarchical organisation of frequent sets," *J. Wirel. Mobile Netw. Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 1, pp. 20–43, 2021.
[108] WiFi signal jammer bluetooth internet blocker anti wireless jamming. Accessed: Mar. 23, 2022. [Online]. Available: https://www.perfectjammer.com/wireless-wifi-bluetooth-jammers.html
[109] Faraday cages, emi/rfi-shielded rooms and faraday tents. Accessed: Mar. 23, 2022. [Online]. Available: https://hollandshielding.com/Faraday-cages-EMI-RFI-shielded-tents-rooms-and-shielded-enclosures



Mordechai Guri received the BSc and MSc degrees in computer science from the Hebrew University of Jerusalem, and the PhD degree from BGU in the Department of Information Systems Engineering. He is the head of R&D of the cyber-security research center with Ben-Gurion University of the Negev, Israel. He is a faculty member in the Software and Information System Engineering Department. He manages a research team in various topics of cyber security. His research interests include advanced malware, rootkits, embedded systems, and mobile security. He was selected to receive the prestigious PhD Fellowship Award. He is a known researcher in covert channels and air-gap security (https://www.wired.com/story/air-gap-researcher-mordechai-guri/).