BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 2
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0017

Analysis on Hacking the Secured Air-Gapped Computer and Possible Solution

Vrinda Sati, Raja Muthalagu

Birla Institute of Technology and Science Pilani, Dubai Campus, DIAC, UAE
E-mails: f20170122@dubai.bits-pilani.ac.in raja.m@dubai.bits-pilani.ac.in

***Abstract*:** *The world today runs on data, every minuscule task to the large one requires data. All the data is stored in the various technologies that we use. And to keep data safe, air gaps are introduced. Air gaps are a network security measure where secure computer networks are physically isolated from unsecured networks. Yet, different methods to hack the air gap have come forth. The paper analyzes the problem of hacking an air gap via screen brightness modulations. The proposed solution is a software program used to alert the user of a change in the brightness level of the screen. The concept of Windows Management Instrumentation* (*WMI*) *has been used to put forth the software. Applied to an air-gapped computer, the program displays an alert box immediately, as the screen brightness changes. The solution is an easy and efficient way to counter the attack. The program can be further implemented in different testing environments and the WMI concept can be applied to various other cyber hacks.*

***Keywords*:** *Air gap; malware; WMI; cyber-attack; screen brightness.*

1. Introduction

"Air Gap" is a term used in the IT field to describe a cybersecurity measure taken to isolate a computer physically from any unreliable network. What has been considered as good cyber defense tactics now needs a defense measure to ensure the security of air-gapped computers. However, that has not been always the case. Air gaps have been considered a great measure and have been also implemented by many companies. It was when Stuxnet hit, that the exploited flaws of air gaps have been brought forward.

The most well-known and yet terrifying cyberattack, known as Stuxnet, tarnished the reliability of air gaps. It changed the face of cyber warfare when it destroyed the centrifuges of the Iranian uranium enrichment facility. The Stuxnet codebase is said to be heavily influenced by Flame and Tilded. Flame is malware that has been discovered to be attacking computers with the Windows operating system and gaining every kind of data from the infected computer, from recording audio to monitoring network traffic. On the other hand, Tilde has been designed to allow malware to be transmitted undetected. Using these as its codebase made Stuxnet a

very powerful malware. The malware corrupted the PLC (Programmable Logic Controller) in the facility to work the centrifuges to the point of self-destruction. The SCADA systems were installed in air-gapped computers and yet that did not stop the cyber-attack. This shows how ineffective air gaps are and how the air gap strategy is based on faith rather than security [1]. The major lesson learned from attacks like Stuxnet, was that air gaps are not the best line of defense against cyber-attacks. Air gaps are like fences, they just discourage attack, not stop them in any way. These codes are accessible to anyone and everyone to use, so there is no predicting as to how, when, and where they could be used. Any industrial controller with an exploitable vulnerability could be compromised and could cause far more serious damage than that caused by Stuxnet. Cyber Weapons are quickly replacing nuclear weapons as war tactics, since in today's world everything is technology-driven, which is extremely worrisome and terrifying. The government itself purchases this malware from the black market [2], which is justified by the explanation of patching vulnerabilities to block out Zero Days or for national security. However, what should stop them from spying on the privacy of the citizens?

Breaching the air gap, which has been thought to be impossible, has been achieved by many hackers in different ways. Nevertheless, breaching the air gap is not the only part of the hack. The other important part of the hack is to bridge the gap for as long as possible. Infiltrating the network is the first step of an attack [3]. However, maintaining a communication with the network is what makes the attack successful.

Air gaps are thought to be a reliable security method, so if two nearby computers were to be compromised, it would not be able to exchange data. BitWhisper [4] is a covert channel, which is utilized to exchange data between air-gapped computers. Covert channel is a type of attack that allows information to be exchanged between cmputers, that are not allowed to share information as per security policies. BitWhisper allows exchange of data through thermal manipulations and the sharing of information is bidirectional. The strategy is to infect the computers connected to the public network and then infect the internal network using an infected USB or using an insider. To exchange information, a bridge needs to be created, which is done by sending thermal pings from the air-gapped computer and receiving a thermal ping from a nearby PC. At this point, the attacker can get any information needed from the air-gapped computer. We can use countermeasures such as, by placing computers at distance from each other so that thermal emissions can't be sensed by the nearby computer [4]. Thermal sensors can also be placed between computers containing sensitive data, to detect any irregular thermal emissions.

Sensitive information can be extracted from an air-gapped computer through screen brightness manipulation [5]. These modulations of the screen brightness are invisible to the human eye but can be seen through a video taken by a camera. What makes this even more dangerous is that the data can be stolen while someone is working on the computer. The malware in the infected air-gapped computer converts the data into a stream of bytes and manipulates the brightness to share the data, which needs to be captured by the attacker's camera. The sensitive data is then retrieved using image-processing techniques. The modulation of the brightness is divided into

0’s and 1’s. The above-mentioned method exploits the inability of human vision to differentiate slight contrast changes. There could be prevention and detection methods. The prevention methods include placing computers with sensitive data in an access-restricted area, where any type of camera shouldn’t be allowed. The computer screens should be covered with a polarized film so that CCTV cameras can’t capture the information concerning the screen [5]. The detection methods include investing in camera receivers that can detect changes in brightness when a signal is given.

Similar to how information can be stolen by using screen brightness, acoustic signals can be sent by the optical drive of a speakerless computer, by using CD-Leak (an acoustic covert channel). The malware on the infected computer can extract the data and convert it into a stream of bytes, which is transmitted by the acoustic signals from the mechanical movements of the optical drive. These signals can be picked up by a smartphone or any device with recording capabilities. The signal is then decoded, and the information can be sent to the attacker via the Internet. Nowadays, computers do not have optical drives, but every computer has a chassis fan [6]. Similarly, the Fansmitter malware exploits the fan of the computer to transmit information in the form of acoustic signals. These attacks can be prevented by replacing the optical drives with non-mechanical hardware so that information leaks can be prevented. HIDS (Host Intrusion Detection System) can be installed to detect any irregular or suspicious activity in the optical drive [7].

A novel approach is proposed [8] that can exfiltrate data through an air-gapped computer via its power adapter. This approach utilizes the switched-mode power supply, which exists in all laptops, desktop computers, and servers nowadays. Malware can indirectly control the electromagnetic emission frequency of the power supply by leveraging the CPU utilization demonstrated. An innovative way to design a new air gap bridging covert channel is proposed for remotely communicating with malware already installed on a computer by involving the induced perturbations [9]. An air-gap covert channel for computers based on ARM CPU is proposed, which includes a software algorithm that can effectively cause cache misses [10].

GSMem is a malware that can extract data from an air-gapped computer and transmit it over cellular frequencies using electromagnetic signals. The infected computer acts as a transmitter while a nearby infected cellular device acts as a receiver. When data is transmitted from the CPU to the RAM, radio frequencies are emitted [11]. GSMem uses this concept and generates a continuous stream of bytes to be transmitted over radio frequencies. The binary data modulates the waves by using special CPU instructions. The signal is then captured by an infected cellular device and demodulated by a rootkit in the device. Insulation can be provided around the air-gapped computer, like a Faraday cage, to cancel out any electromagnetic frequencies that are emitted by the computer, which can be used as a countermeasure.

Magneto is a malware that extracts data from an air-gapped computer through magnetic waves generated by CPU cores. It exploits the magnetometer in smartphones for receiving data from the computer. In the experiment, the computer and smartphone are kept shielded using a Faraday casing, which restricts any wireless communication to/from the computer, but as the CPU generates low-frequency

magnetic fields, it bypasses the Faraday cage. As the CPU has the most power consumption, due to the workload, the magnetic waves emitted by it are high. The workload of the CPU can be regulated to get control of the generated magnetic field. The sensitive information is converted into a stream of bytes and transmitted with the help of magnetic waves [12]. Only a small portion of the magnetic waves actually contain the leaked data. The wave can be filtered using the band-pass filter to extract only the region of interest [12].

ODINI is also a malware that works similarly to the Magneto malware. While the Magento malware extracts data at the speed of 2 bits per second, ODINI's speed for data extraction is 40 bits per second, making it more powerful. The latter is more efficient due to the fact it needs to be at a distance of 1 to 1.5 m from the target, while the former needs to be 12 cm away from the target for data extraction. Attacks by both MAGNETO and ODINI can be prevented by using signal jammers, which are used to generate a strong magnetic wave, which will interfere with any other unauthorized magnetic field. They can also be used in a way that a continuous low-frequency magnetic field is produced to jam the malicious magnetic signal [13].

PowerHammer, is a malware used to extract sensitive data by tapping into the power lines of the air-gapped computer. The malware in the compromised computer can modulate the power consumption of the computer and generate data-modulated conduction, in low frequency, on the power lines. The noise generated through the power lines can be measured by the attacker. A transceiver is placed near the power lines, so as to measure the conducted emission, process the modulated signal, decode the signal and send it to the attacker. The process of signal generation is done by changing the workload of the CPU. By changing the CPU's workload, the power consumption can fluctuate to conduct a signal on the power lines. This signal can then be decoded by the transceiver, located near the power lines of the computer [10].

AIR-FI attack [14] exploits the usage of Wi-Fi by devices to extract sensitive data from air-gapped devices. DDR SDRAM buses can be misused to generate electromagnetic waves in the Wi-Fi bandwidth and the sensitive data can be encoded over it. The DDR memory leaks the information through the Wi-Fi signals and a compromised device, which uses Wi-Fi. These devices can receive the information and using a decoding algorithm can intercept the data.

Another peculiar way that air gaps can be compromised is through HVAC systems [15]. These systems are connected to a network that controls the system of an enclosed area. The overlap between the public network and the isolated area usually goes unnoticed and that fact has been taken advantage of. Once the HVAC network is compromised, the HVAC system in the air-gapped area sends out a thermal signal to the air-gapped systems (with the assumption that it is infected). In this attack, the hacker cannot leak any data outside, rather they can cause internal interference in daily work by initiating DDoS attacks or conducting APT attacks.

All methods discussed so far are based on malware extracting the data, but instead, the encryption key can also be extracted to gain sensitive data. The acoustic cryptanalytic side-channel attack exploits the high-pitched noise that is generated by computers due to the vibration of the electrical components, to extract the 4096-bit RSA decryption key from the computer. Interestingly, many of the physical side-

channel countermeasures used in highly sensitive applications, such as air gaps, Faraday cages, and power supply filters, provide no protection against acoustic leakage. There are applications, such as GNU Privacy Guard, whose RSA key can be identified by its acoustic frequency. The key can be extracted within an hour while analyzing the sound generated by the computer during the decryption of cipher text. The attack can be carried out by keeping a compromised phone near the target or a sensitive microphone, to pick up the noise generated by the computer [16, 17].

2. Problem formulation

After analyzing just a few ways in which the air gaps can be exploited, it is safe to say that they are not the best cyber defense measure. Most of the papers taken for the survey also discuss some prevention and detection methods. Nevertheless, those measures require extra hardware, which automatically means more expenditure, in which not many companies will want to invest. Air gaps do discourage hacking to a given extent, but how to stop those that are not discouraged? We would like to propose a solution for the problem of extracting data from the modulation of the screen brightness. Since data can be extracted while someone is working and the change in brightness will not be noticed, a program can be written, that alerts the user as soon as there is a change in the brightness. Through the mentioned method, the attack can be stopped on the spot and the computer can be screened for malware. It could be a good detection method, with higher efficiency and minimal cost for additional equipment. To create a software design, we need first to understand what controls the brightness of the screen. The workings of the computer need to be studied completely before designing software to counter malicious cyber-attacks. It is to be noted that the following solution is proposed with the assumption that the OS is not compromised at the user and kernel levels.

3. Design methodology

To propose a solution for any problem, it is vital to first understand the problem in depth and design a methodology for the solution. For the problem posed in the paper, the main root of the issue is how the hacker has taken control of the brightness of the screen. Therefore, the sub-sections below will explore on WMI and how a tool used to aid the IT department in their day-to-day work can be and is being used to hack into various systems.

3.1. Windows management instrumentation

WMI is a system that acts as an interface to manage and monitor Windows applications and devices in a network, locally and remotely. It is a sub-system of PowerShell and with the combination of the WMI command line interface can perform many functions. It can be used by a manager to find the number of users logged in and to monitor his employees. All actions taken by the WMI are done in the form of objects, which are members of classes, which are members of some namespace, which are derived from a root namespace as shown in Fig. 1.

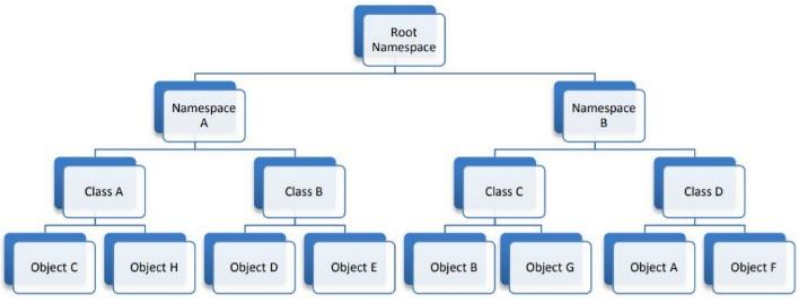

Fig. 1. WMI architecture

There are pre-defined functions and classes provided by WMI, not only to monitor but also to set specifications for the working environment. For example, if details of Excel files of a certain size need to be gathered and the user should be notified of the creation of a new file with the same size, the *Register-WmiEvent* command can be used. It can also respond to queries such as finding the properties of the BIOS of the computer. WMI uses WMI Query Language (WQL), which is a subset of SQL. WQL is specific to WMI events and functions and differs from SQL by minor semantic changes. Not only does WMI allow such queries, but it also allows launching applications on the user's computer or on another computer or a remote one (given that the user has appropriate permissions). A simple query typed in PowerShell, shown in Fig. 2, opens a new Notepad file.

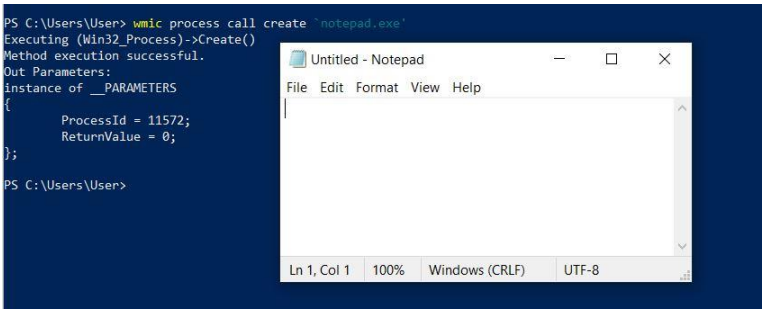

Fig. 2. WQL query to launch a notepad file

Now imagine if WMI can launch a simple file in a different computer within the network, what else could it be used for? The thought of using WMI for cyber-attacks has been first introduced to the world when it was discovered in the Stuxnet attack. Since then many cybercriminals have used WMI for detecting anti-virus and investigating the target before launching the attack. They get in through file-less malware, by lateral movement (moving from one device to another in the network, after gaining control of one). In a file-less malware attack, recognized tools are used to carry out malicious activities. These tools are installed by default and their working does not cause any suspicion. That is the reason that the Stuxnet malware works undetected. It has become important for the defenders to understand WMI and use it to their advantage because the attackers are increasingly using it to theirs.

3.2. WMI: Detection & Response

Just as attackers use WMI for reconnaissance, the defender can use it to survey suspicious activities within the network. It can play the role of an Intrusion Detection System (IDS) for the network. WMI poses a real threat due to its ability to access devices over the network. Protocols such as DCOM TCP Port 135, HTTP, and HTTPS allow malicious activities, as traffic content is not inspected. WQL plays an important role in defending against cyber-attacks. The queries can be used to get information about target devices (instance query) and for object manipulation (event query). WMI event, which is the creation or modification of objects is triggered at almost all OS-related actions. This feature is exploited by attackers and can be used to counter attacks by the defenders. A permanent WMI event can be installed as it runs as the system and remains despite reboots. The installation of a permanent WMI subscription requires three things: event filter, event consumer, and filter-consumer binding as shown in Fig. 3.

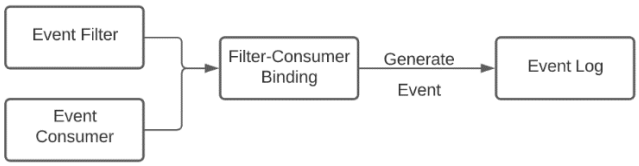

Fig. 3. WMI event subscription

- Event filter. A filter configured to respond to the event of interest. The response can be to alter the user about the event such as creating a file with a specific name, a user logging off, or someone inserting a removable device.
- Event consumer. It is the action taken after the triggering of an event. The action can be sending an email to the administrator, executing a code, or logging the details of the event.
- Filter-Consumer Binding. The association of the type of consumer to the filter has to be created.

Since a WMI event is hard to detect as well as to remove, a malicious actor would choose to use this method. However, it can also be wielded to the advantage of the defenders. Now that the basics of WMI are understood, we can apply the concept to detect the change in screen brightness to extract sensitive data.

4. Proposed solution

Different measures to counter the problem of attacking air gaps and extracting information by modulating the brightness level of the screen have been proposed. These measures have been divided into prevention and detection measures. The preventive measures include keeping air-gapped computers in secure zones, where only authorized employees can enter, without any cameras. For this attack, the CCTV camera could be compromised, so the computer screens can be covered with a polarized film, which darkens the display if viewed from a distance. Keeping camera sensors pointed at the screen to detect changes in brightness is a detection method

applied. However, it is impractical for a large-scale installation due to cost and maintenance.

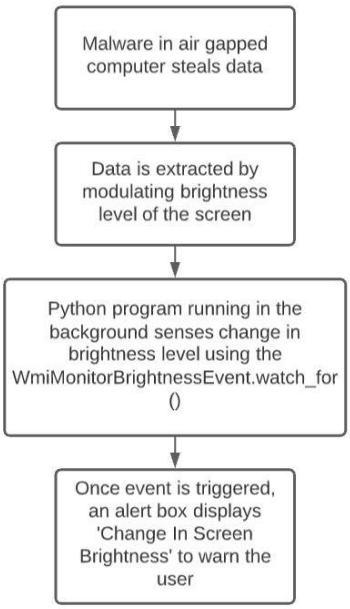

Fig. 4. Flowchart of proposed solution

My proposed solution to the problem is to develop Python software with the use of WMI modules, to alert the user immediately as a change in brightness levels is detected. Since the application of the software is easy, the cost is almost negligible and there is no maintenance that needs to be done, it proves to be a better solution. A more in-depth understanding of the proposed solution is explained below.

4.1. Working behind screen brightness

The monitor driver is a part of the operating system, through which the brightness controls are implemented. A monitor driver provides configuration details on how to communicate with the OS, in the form of an INF file. The brightness levels are controlled using the WMI interface to create applications, such as the brightness control slider, that interact with the brightness levels. The newer Windows OS also gives the option of night light, which allows users to use the screen at a comfortable brightness, during the nighttime only as shown in Fig. 5. This is done through the monitor driver.

The driver also registers with DPPE (Device Power Policy Engine) and ACPI (Advanced Configuration and Power Interface) to associate power policies with brightness levels and to introduce shortcut keys to control brightness levels, respectively. According to the power setting chosen, the screen brightness is adjusted appropriately.

The brightness levels of a screen range from 0 up to 100, with 0 being the darkest and 100 being the brightest. The levels need not be in uniform increments; they can range from any value from 0 up to 100. The change brought by is due to the increase

or decrease of the intensity of the pixels of the screen. Not only does WMI provide an interface for users to interact with the brightness levels, but its commands can also be used to get information about the brightness level as shown in Fig. 6 or set the brightness of the screen as shown in Fig. 7. The *WmiMonitorBrightness* and *WmiMonitorBrightnessMethods* classes support the above-mentioned functions.

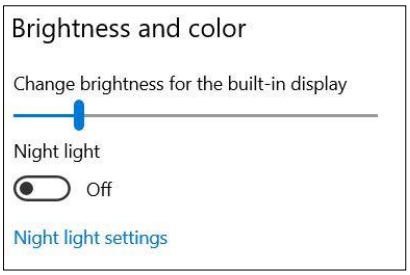

Fig. 5. WMI interface for brightness control and night light

```
PS C:\Users\User> Get-Ciminstance -Namespace root/WMI -ClassName WmiMonitorBrightness


Active            : True
CurrentBrightness : 35
InstanceName      : DISPLAY\BOE06EE\4&b6e5505&0&UID265988_0
Level             : {5, 6, 7, 8...}
Levels            : 67
PSComputerName    :
```

Fig. 6. WMI command to get the brightness level

```
PS C:\Users\User> (Get-WmiObject -Namespace root\wmi -Class WmiMonitorBrightnessMethods).WmiSetBrightness(0,40)
```

Fig. 7. WMI command to set the brightness level

4.2. Detection of change in screen brightness

The paper talks about different ways of extracting data through air-gapped computers and focuses on proposing a solution for data theft through screen brightness modulation. The premise for the cyber-attack is to have the malware in the air-gapped device. Since it is air-gapped, the malware cannot be transmitted through the network or the web. For the malware to enter such devices, an infected USB can be inserted into the target device as shown in Fig. 8. Another way is to compromise the targeted device before it is air-gapped. This can be done by including the malware in the software, which are to be installed into the device. Even anti-virus software can contain malware. This level of planning requires an advanced attacker.

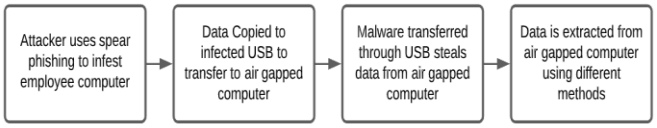

Fig. 8. Transfer of malware to air-gapped computers

Once the malware has entered the target device, it collects the data, converts the data into byte streams and transmits it through the modulation of screen brightness.

The brighter display signifies 1 and darker display signifies 0. The change in brightness is so minimal that the human eye cannot catch it, but the WMI can catch it. As explained in the previous section, WMI classes can be used to get information or manipulate the brightness levels. Since the malware changes the brightness levels, the classes provided by WMI can be used to detect these changes. Integrating WMI classes with Python, a simple, yet effective program was developed to detect any change in brightness and alert the user. The program has been created on Sublime Text and run on PowerShell. The device used was disconnected from Wi-Fi and Bluetooth, in attempts to mimic an air-gapped computer.

The WMI and win32api libraries have to be imported, but before that, the WMI module has to be installed using the pip command. The module is a part of the Python package index (PyPI). The win32api module is by default installed on any Windows device as in Fig. 9.

```
1   import wmi
2   import win32api
3
```

Fig. 9. Importing libraries

As explained before, objects are part of classes, which are a part of namespaces. The below code creates an object for the WMI class. The object also refers to a namespace, WMI, which is the root namespace. Root namespace is not necessary to specify, but can be done if wanted as shown in Fig. 10.

```
5
6   obj = wmi.WMI(namespace='wmi')
7
```

Fig.10. Creating an object

The next snippet of code displays the created object using the WMIMonitorBrightnessEvent class. The class stores the current brightness level whenever there is a change in the intensity. A function, watch_for() is also called. The function watches out for the event and returns the instances of the WMIMonitorBrightnessEvent class. The parameter of the function specifies that it should be notified only when a modification is made as in Fig. 11.

```
7
8  event_trigger = obj.WmiMonitorBrightnessEvent.watch_for (notification_type="Modification", delay_secs=1)
9  event_happened = event_trigger ()
10 print(event_happened)
11
```

Fig. 11. Creating a trigger function

The last part of the code creates a message box, which alerts the user of a change in the level of brightness. The action requires the win32api module, which allows users to use the application interfaces provided by Windows OS. The alert box will display a message “Change in Screen Brightness” with the title “Alert”, only when the function is triggered as in Fig. 12.

```
ctypes.windll.user32.MessageBoxW(0, "CHANGE IN SCREEN BRIGHTNESS.", "Alert", 0)
```

Fig. 12. Creation of alert box

On running the program through PowerShell, it does nothing until there is a change in the brightness level. When the function is triggered, it prints an instance of the WMIMonitorBrightnessEvent class. The instances are the indication of an active monitor, current brightness, instance name, and a time stamp. It also displays the alert message on the screen as shown in Fig. 13.

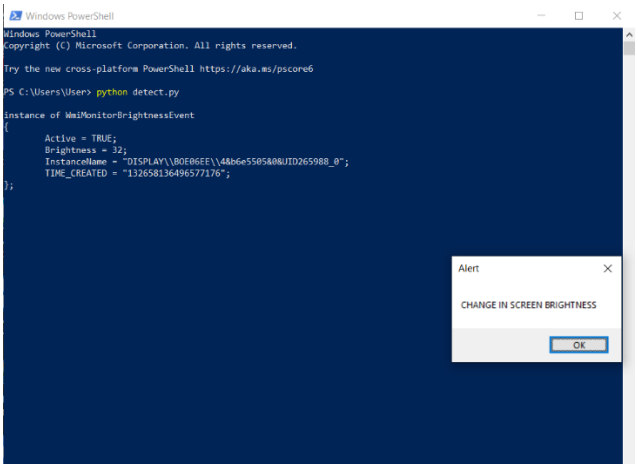

Fig. 13. Result when brightness is changed

5. Summary and conclusion

In the paper, a simple software solution was provided for the detection of cyber-attack, using screen brightness, in an air-gapped device. There are preventative measures such as covering the screen with polarized film, which darkens when seen in pictures or videos. Detection measures such as placing camera sensors to detect changes in brightness levels. Nevertheless, these measures require spending money on these measures in addition to the maintenance costs. With an understanding of the software of the device, a low-cost and efficient solution is developed.

The use of WMI is like turning the tide on the attackers. By applying the concept of WMI, the extraction of sensitive data through brightness modulation can be immediately stopped as the action will trigger the function and alert the user. Once the user is alerted, he/she can take the necessary actions needed. The implementation of this can increase security in air-gapped devices. It can prevent the theft of sensitive data and ensure that the reason behind the detected attack will be tracked down. The early detection of an incoming attack could prevent a lot of destruction and money loss.

Despite that, the proposed model is effective, but the efficiency of the solution may vary according to the size of the organization it is deployed in. A small company may be able to deploy the proposed solution with comparatively more ease than

medium and large companies may. A malware forensic tool can be deployed on systems to detect the malware within a system. Tools like Prodiscover, SANS, etc. can be used or custom forensic tools can also be developed. These tools take in data present in the system and apply constraints to filter out files where the malware could be present [18]. In comparison to the proposed solution, forensic tools are not specific to one attack, whereas the former has been developed keeping only one type of attack in mind. The tools identify where the malware is, whereas the solution proposed only detects the malware during the attack [19].

The need for cyber defense tactics is at an all-time high with the attackers getting more advanced and creative with their method of attack. The paper puts forward different ways, in which a cyber-defense measure, i.e., air-gapped devices, can be hacked. The countermeasure to such attacks needs to be equally advanced as the attack so that there is minimal damage. In addition, as technology advances, there are many cybersecurity measures available. However, that does not mean that we can put our guard down because there will always be someone trying to find a loophole or vulnerability in the system. Thus, the need for cyber security is very important. It also means that creating awareness is extremely important because the more knowledge we have the better chance we have of countering cyber-attacks.

The project has a broad scope for future work. The current program has been developed on a personal air-gapped laptop, which is for a small scale. For implementation at a larger scale, say for an organization, a permanent WMI event would have to be installed. Once a permanent event is installed, it becomes a part of the system and always runs in the background, despite reboots. This need not be installed only in air-gapped devices. It could also work on normal devices as well. The focus of this project is on attacks using screen brightness only, but the research can be extended to other forms of cyber-attacks as well. In addition, the proposed approach can be extended to other environments like Linux, Unix, BSD, etc.

References

1. B e r g h e l, H. A Farewell to Air Gaps. Part 1. – Computer, Vol. **48**, 2015, No 6, pp. 64-68.
2. B e r g h e l, H. A Farewell to Air Gaps. Part 2. – Computer, Vol. **48**, 2015, No 7, pp. 59-63.
3. G u r i, M., Y. E l o v i c i. Bridgeware. – Communications of the ACM, Vol. **61**, 2018, No 4, pp. 74-82. Available: 10.1145/3177230.
4. G u r i, M., M. M o n i t z, Y. M i r s k i, Y. E l o v i c i. BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations. – In: Proc. of 28th IEEE Computer Security Foundations Symposium, 2015.
5. G u r i, M., D. B y k h o v s k y, Y. E l o v i c i. Brightness: Leaking Sensitive Data from Air-Gapped Workstations via Screen Brightness. – In: Proc. of 12th CMI Conference on Cybersecurity and Privacy (CMI'19), 2019.
6. G u r i, M. CD-LEAK: Leaking Secrets from Audioless Air-Gapped Computers Using Covert Acoustic Signals from CD/DVD Drives. – In: Proc. of 44th IEEE Annual Computers, Software, and Applications Conference (COMPSAC'20), 2020.
7. G u r i, M., Y. S o l e w i c z, Y. E l o v i c i. Fansmitter: Acoustic Data Exfiltration from Air-Gapped Computers via Fans Noise. – Computers & Security, Vol. **91**, 2020, 101721.
8. Z h a o, B., M. N i, P. F a n. Powermitter: Data Exfiltration from Air-Gapped Computer through Switching Power Supply. – In: China Communications, Vol. **15**, February 2018, No 2, pp. 170-189. DOI: 10.1109/CC.2018.8300280.

9. K a s m i, C., J. L. E s t e v e s, P. V a l e m b o i s. Air-Gap Limitations and Bypass Techniques: "Command and Control" Using Smart Electromagnetic Interferences. – Le Journal de la Cybercriminalité & Des Investigations Numériques, Vol. **1**, 2015, No 1, pp. 13-19.
10. K e n t a, Y., M. H i r o s e, T. S a i t o. Data Exfiltration from Air-Gapped Computers Based on ARM CPU. – International Journal of Advanced Computer Science and Applications, Vol. **9**, 2018.
11. G u r i, M., A. K a c h l o n, O. H a s s o n, G. K e d m a. GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies. – In: Proc. of USENIX Security Symposium, 2015, pp. 849-864.
12. G u r i, M. MAGNETO: Covert Channel between Air-Gapped Systems and Nearby Smartphones via CPU-Generated Magnetic Fields. – Future Generation Computer Systems, Vol. **115**, 2021, pp. 115-125.
13. G u r i, M. AIR-FI: Generating Covert Wi-Fi Signals from Air-Gapped Computers. – arXiv.org, 2021 (online).
https://arxiv.org/abs/2012.06884.
14. G u r i, M., B. Z a d o v, Y. E l o v i c i. ODINI: Escaping Sensitive Data from Faraday-Caged, Air-Gapped Computers via Magnetic Fields. – IEEE Transactions on Information Forensics and Security, Vol. **15**, 2020, pp. 1190-1203.
15. M i r s k y, Y., M. G u r i, Y. E l o v i c i. HVACKer: Bridging the Air-Gap by Attacking the Air Conditioning System. 2017 (online).
https://www.researchgate.net/publication/315710328_HVACKer_Bridging_the_Air-Gap_by_Attacking_the_Air_Conditioning_System.
16. G e n k i n, D., A. S h a m i r, E. T r o m e r. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. – In: Proc. of Advances in Cryptology (CRYPTO'14), 2014, pp. 444-461.
17. G u r i, M., B. Z a d o v, D. B y k h o v s k y, Y. E l o v i c i. PowerHammer: Exfiltrating Data from Air-Gapped Computers through Power Lines – IEEE Transactions on Information Forensics and Security, Vol. **15**, 2020, pp. 1879-1890.
18. K e i m, Y., A. M o h a p a t r a. Cyber Threat Intelligence Framework Using Advanced Malware Forensics. – International Journal of Information Technology, 2019. Available: 10.1007/s41870-019-00280-3.
19. S h a r m a, P., B. N a g p a l. Regex: An Experimental Approach for Searching in Cyber Forensic. – International Journal of Information Technology, Vol. **12**, 2019, No 2, pp. 339-343. Available: 10.1007/s41870-019-00401-y.

Disclosure of potential conflicts of interest: Author Vrinda Sati declares that he/she has no conflict of interest. Author Raja Muthalagu declares that he/she has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Received: 19.07.2022; Second Version: 17.03.2023; Accepted: 14.04.2023