Bridging the Air Gap: Inaudible Data Exfiltration by Insiders

Completed Research Paper

Samuel Joseph O'Malley
University of South Australia
omalsa04@gmail.com

Kim-Kwang Raymond Choo
University of South Australia
raymond.choo@unisa.edu.au

Abstract

As critical systems are increasingly dependent on software and are connected to the Internet, insider threats will be of ongoing concern. For example, corrupt insiders could deliberately introduce malicious software into the organisation's system to surreptitiously gain control, and launch online attacks via and against compromised systems. In this paper, we present a method that an insider can use to facilitate data exfiltration from an air-gapped system without using any modified hardware. The method presented here uses inaudible sounds transmitted from the target machine's speakers, which can transfer data to a nearby computer equipped with a microphone. We demonstrate how inaudible communication bridge air-gapped systems without any additional hardware. Our system is low-risk for an insider as it only requires one-off access to a system, and can be erased leaving little-to-no trace once it is no longer required. Finally we provide some recommendations for organisations to avoid similar data exfiltration techniques.

Keywords

Cybercrime, Data exfiltration, Insider threats, Inaudible sound

Introduction

Employees have intimate knowledge of the systems within an organisation, and they have the potential to be a great threat to security (Choo, Smith & McCusker 2007). Insiders can unwittingly weaken security by responding to phishing attacks, accidentally installing viruses, or being fooled through social engineering, they can also take malicious actions against their organisation for monetary gains or for revenge (Choo 2008; Choo & Smith 2008). In this paper, we present a method that an insider can use to facilitate data exfiltration from an air-gapped system (e.g. in a classified network), using no modified or customised hardware or cables. The method presented here uses inaudible sounds transmitted from the target machine's speakers, which can transfer data to a nearby computer equipped with a microphone. The target environment used as an example in this paper is a classified work environment where there are classified, air-gapped computers in close proximity to unclassified Internet-enabled computers. In this paper, we will demonstrate how an insider can gain continued access to an air-gapped system through a simple one-off software install (e.g. via a USB), even once they no longer have physical access to the system.

Related Work

A malware dubbed BadBios was reportedly uncovered by security consultant Dragos Ruiu in 2010 and has recently gained publicity (Goodin 2013). BadBios is a controversial topic among security professionals with many believing it is not real and is either a hoax or an artefact of Ruiu's paranoia. However, one of the notable features of BadBios is its reported ability to communicate between infected, air-gapped machines that are not otherwise connected via a traditional network connection. It is rumoured that BadBios uses high-pitched sounds inaudible to the human ear in order to communicate. This technique is ideal for a classified environment where traditional communication mediums are disabled or highly controlled. Flame, one of the largest and one of the most sophisticated malware ever discovered (Munro 2012), uses Bluetooth to communicate between air-gapped machines. Bluetooth is a very fast and common communication medium. However, it is highly likely that it would be disabled in a secure environment and would not be suitable for data exfiltration in this case.

Many forms of covert data exfiltration exist. Yali et al. (2009) present a system called Sensitive Information Dissemination Detection (SIID), which is designed to detect sensitive data exfiltration by insiders. Yali et al. (2009) utilise signal processing and statistical methods to detect suspicious traffic communicating with the external network. The methods presented in this paper would not be detected by systems like SIID because it bypasses the traditional networking mediums. Other covert data exfiltration methods such as printing sensitive documents, copying data to a USB key, or burning a data disc, can all be monitored and detected easily (Giani et al. 2006). Methods such as these can also be avoided by disabling external devices on the air-gapped computers.

Transient Electromagnetic Pulse Emanation Standard (TEMPEST) is the National Security Agency (NSA) codename for spying on information systems through leaking emanations of devices, mainly through electromagnetic signals. Many of the TEMPEST standards are classified but some recommendations have been published on avoiding accidental data leakage (McConnell 1995). TEMPEST describes a Red/Black concept of physically separating classified plaintext information (Red) from encrypted or unclassified information (Black). This separation is done through maintaining a strict separation distance and appropriate shielding. According to the document NSTISSAM TEMPEST/2-95 (McConnell 1995), Red and Black computers should be separated by at least one meter (39 inches) in distance. While this distance might be large enough to stop electromagnetic signals being detected, we will show in this paper that our technique can easily communicate over this distance. Attacks based on the TEMPEST principles require hardware to be installed on-site for the duration of the attack. If this hardware is discovered, then it can be used as evidence that a breach has occurred and could likely lead to the malicious insider being caught. TEMPEST standards also apply to acoustic data leakage, Genkin et al. (2013) demonstrated the ability to use a high-powered microphone to extract encryption keys from a target computer. Genkin et al. (2003) managed to perform this extraction at a distance of up to 4 meters, simply by recording the sounds a computer's power supply makes while the CPU is decrypting data.

A recent article by Sanger et al. (2014) in the New York Times alleges that the NSA implants tiny radio transmitters into target machines in order to exfiltrate data, even when that device is not connected to the Internet. The article states that the hardware might be inserted by a spy, a manufacturer or even by a trusted user (Sanger et al. 2014). If a manufacturer can surreptitiously add hardware to a computer in order to exfiltrate data, then it is surely possible for a manufacturer or software vendor to embed some code on a computer. A large downside to this technique is that there is physical evidence left behind; whereas a software-only attack can be erased after it is no longer needed, leaving little-to-no trace. Software can also be unwittingly installed through many manipulative techniques; whereas it is very difficult to accidentally install hardware.

Contributions

To the best of our knowledge, this is the first open source published work on using inaudible sound waves to exfiltrate data using consumer grade equipment. The methods presented in this paper require no physical modification or addition of hardware, and leaves no physical trace that a breach has occurred. Attacks using this method can be put in place by a malicious or careless employee, which would allow sensitive data to be exfiltrated over a long period of time.

The methods presented in this paper will demonstrate that even an air-gapped system is not totally secure. We also hope to reinforce that the biggest threat to an organisation's security is its own employees.

Outline of Paper

In the next section, we describe the system design and explain the design decisions made in the development of this system. In the following section, we break down the experiment set-up so that others can reproduce it accurately. This section describes the hardware, software, and environmental setup used in our experiments. It also describes how the evaluation was performed and the data used. Next in this paper, we present the findings and give a discussion of what impact this research has on secure environments today. Finally in the last section, we conclude the paper and outline some possible directions for future work.

System Design

The technique presented in this paper uses commodity hardware to transmit data using inaudible high-frequency sounds. Humans can typically hear sounds between 100Hz and 20kHz, although the effective range decreases with age and exposure to loud sounds (Lee et al. 2012). Most commodity sound-cards operate at a sample rate of 48kHz, which gives a maximum possible transmitted frequency of 24kHz. In practice, however, we found that commodity speakers would not function when approaching 23kHz. This gives an effective transmission frequency range of 20kHz to 23kHz that we can use without human detection. We can also assume that there are no children or animals in a secure environment that might be able to detect these high frequencies.

In order to encode data as audio we need to modulate it to convert the digital signal into an analogue signal. Phase-Shift-Keying (PSK) is very efficient for transmitting data via radio waves but we found that in practice it is not very resistant to a noisy medium such as audio. Therefore, it was decided to use the simpler Frequency-Shift-Keying (FSK) technique, used in FM Radio and dial-up modems. A very simple method to implement FSK is to output a frequency for binary-one, and for binary-zero (see Figure 1). While this implementation works very well at audible frequencies, at higher frequencies it causes the cheap speakers to crackle and pop, making the sounds detectable by human ears and destroying the data quality. By performing a simple Gaussian smoothing on the resultant signal, we can eliminate these detectable artefacts and improve the transmission quality.

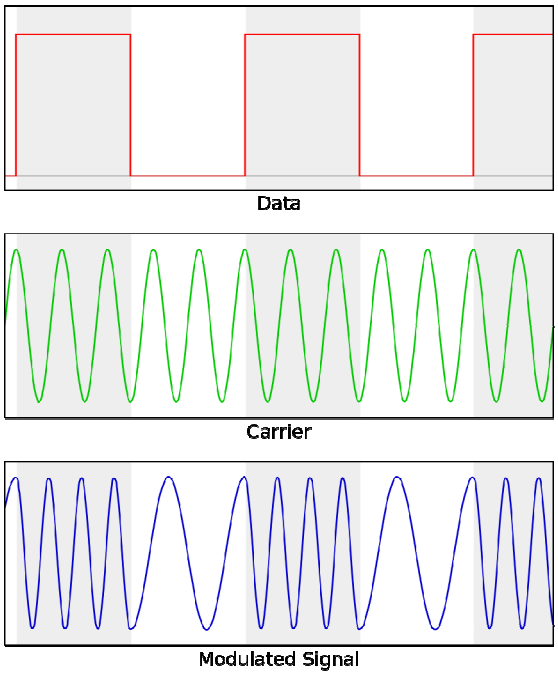

Figure 1: Example of Binary FSK (Ktims 2006)

Error detection methods were employed in order to ensure that the data transmitted is the same that is received. Data to be modulated is broken into separate packets (see Table 1) containing a preamble, to signify that the packet has begun, the length of the data, and a cyclic redundancy check (CRC) to validate that the data is free of errors.

Table 1: Data framing

PREAMBLE	DATA LENGTH	PAYLOAD	CYCLIC REDUNDANCY CHECK (CRC)

Different types of data require different error mitigation techniques. Data such as passwords and secret keys need to be transmitted with bit-for-bit accuracy. However, data such as plain-text files can potentially lose characters and still be human-readable. Data is often lost in chunks, so if a packet fails to transmit whole words can be lost. This is a problem for text because missing "key" words can make a sentence hard to understand. To reduce this effect, the text can be permuted, effectively scrambling the characters, so that if a portion of the data is lost then it will result in only some characters missing over multiple words (Berrou et al. 2003).

Two-way communication provides the opportunity to obtain a much higher transmission success rate. The TCP protocol requires receipt of a packet to be acknowledged, lest it be re-sent. We implemented a simple method following the same principles of TCP. Each packet included a sequence number (see Table 2), and the acknowledgements (see Table 3) were only transmitted from the Internet enabled host to the target machine. Separate bands of frequencies were used so that the signals would not interfere with each other. The return path for acknowledgements was given a smaller frequency bandwidth because it did not need to transmit as much data.

Table 2: Two-way communication data framing - Data

PREAMBLE	DATA LENGTH	SEQ #	PAYLOAD	CRC

Table 3: Two-way communication data framing - Acknowledgement

PREAMBLE	DATA LENGTH	SEQ #	ACKNOWLEDGEMENT	CRC

Experiment Setup

The experiment was performed using a MacBook Pro as the target air-gapped computer and a Lenovo X220 ThinkPad Tablet as the Internet enabled computer. Both devices have 16GB of RAM and an Intel Core i7 processor. The machines can both transmit and receive high frequency sounds up to approximately 23kHz. A cheap pair of external speakers was used rather than the internal speakers to control the directionality of the output sounds. Laptop speakers generally point straight up and are not ideal for transmitting sound over a horizontal distance. See Figure 2 for a diagram of the hardware setup.

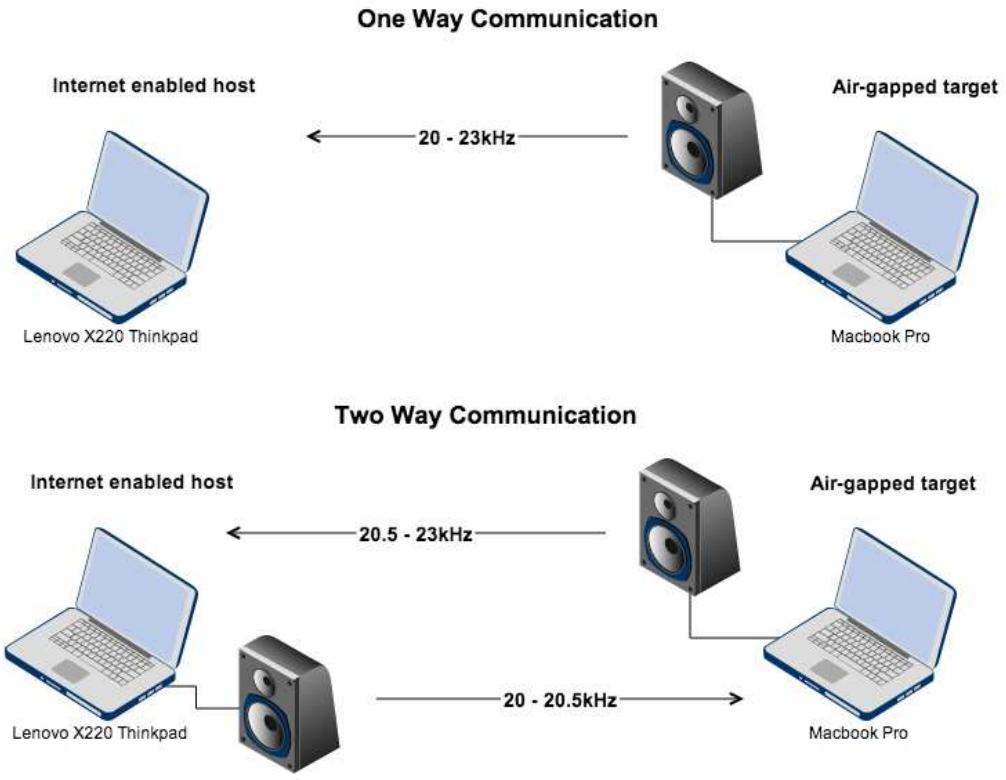

Figure 2: Hardware Setup

Interestingly, the data transmission worked even when headphones were plugged into the audio jack of the computer, albeit at a reduced effective distance. A standard pair of iPhone headphones was used and demonstrated the ability to produce high frequency sounds. A comparison between the speakers and headphone's power at different distances was performed in order to show that data transmission is possible even when headphones are used. Figure 3 shows us that although the power of the headphone's output is lower than the speakers, it is still distinguishable from the background noise floor of -80dB (at 20kHz).

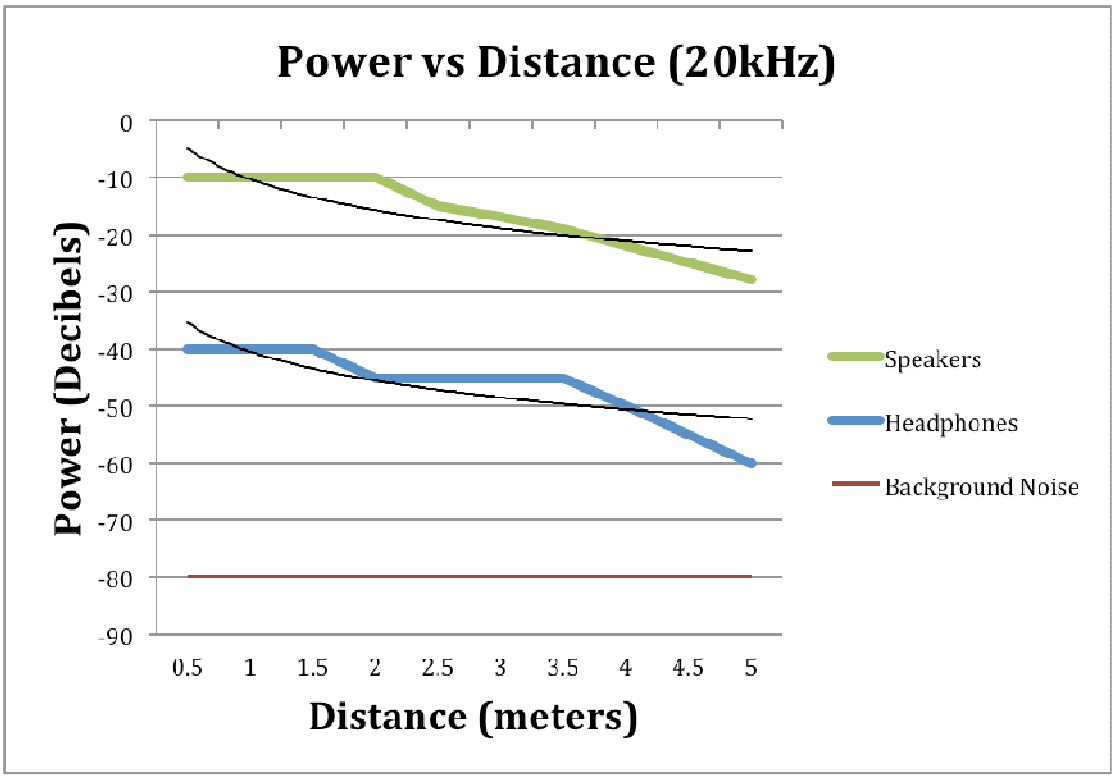

Figure 3: Comparison of Power vs. Distance for Speakers and Headphones

Each device was loaded with Ubuntu 13.10 so that the software environment was identical. GNU Radio[1] was used to develop the prototype communication software, which was then rewritten into pure Python code. GNU Radio is an open-source software development toolkit that allows prototyping of software-defined radios by providing signal-processing blocks that can be linked together to form a full radio. GNU Radio generates python code that implements the software radio, this code can then be modified and extended to implement any specific features needed.

The experimental environment was designed to match a typical open-plan workplace. Open plan workplaces are free of loud noises, especially high-pitched noises, so the experiment was performed in an environment with a low level of background noise. The human voice, at such a low frequency, doesn't interfere with the data transmission unless it is so loud that it overpowers the microphone. Each laptop was placed on tables of equal height with no obstructions between them, and no other special measure was taken to reduce echoes between the two devices.

In order to set up the communication between two devices, one-off access is needed to install a small client application on the target machine. An insider could maliciously install the client software in order to retain access even once they have left the company, or they could be tricked into installing the software so that an attacker could access the air-gapped computer remotely. The client application was tested against 40 popular virus scanners using the Virus Total online service and it was not detected[2]. This indicates that the communication software could easily be loaded onto the system without detection, and a more malicious payload could be transmitted later via audio.

[1] http://gnuradio.org/redmine/projects/gnuradio/wiki

[2] http://www.virustotal.com/

Evaluation

In order to evaluate the performance of our system, many different types of data was transmitted over various distances and the speed and accuracy were recorded.

Two different methods were tested in this experiment, and each one was evaluated based on transfer rate, transmission distance, and exfiltration error rate. Method 1 evaluates the performance of one-way communication with a single speaker and microphone pair. Method 2 uses two-way communication using two pairs of microphones and speakers to allow for two-way communication. This method implements a protocol similar to TCP/IP that can signal data re-transmission if it does not receive data correctly.

One thousand username and password pairs were used initially to evaluate both methods. Each username was a randomly generated 10-character string, and each password was a randomly generated 30-character string. Transmission rate is calculated by comparing the total time taken to transmit the data, with the number of correctly received data items.

Next we transmitted 100 different 1024-bit and 2048-bit RSA secret keys. The payload size of the transmission packet was adjusted so that an entire key fit in a single packet. This way the whole key would be discarded if any data corruption occurred. Each received key was compared against the transmitted version to verify that no errors had occurred.

Adobe PDF and Microsoft Word documents were also transmitted in order to test binary data transmission of large files. Binary data must be transmitted accurately or it will become corrupted. Due to the large size of the files, there is a higher chance that some data will not be transmitted and the file will be unreadable. In many cases, it would probably be smarter to transmit text from the document rather than the whole document.

Findings

The following table (Table 4) shows the results of the one-way communication setup, using headphones as the audio-source. This test is only included to provide context and to show that the power or loudness of the transmitter does not affect signal quality, but does correlate with maximum transmission distance.

Table 4: Comparison results of transferring 1,000 usernames and passwords using headphones

Types of Data	Transmission Distance	Transfer Rate	Success Rate
Username (10 characters) and Password (30 characters)	1 meter (~3.3 feet)	6.2 usernames and passwords per second	100%
	2 meters (~6.5 feet)	6.2 usernames and passwords per second	100%
	3 meters (~10 feet)	5.9 usernames and passwords per second	95%
	4 meters (~13 feet)	1.7 usernames and passwords per second	28%
	5 meters (~16.4 feet)	N/A	0%

Table 5 outlines our results for the full set of test data using speakers as the audio-source. Some of the data failed to transmit but we have included the results to provide a comparison against Method 3, using two-way communication.

Table 5: Results of Method 1 - one-way communication

Types of Data	Transmission Distance	Transfer Rate	Success Rate
Username (10 characters) and Password (30 characters) – 1,000 username and password pairs	1 meter (~3.3 feet)	6.2 usernames and passwords per second	100%
	2 meters (~6.5 feet)	6.2 usernames and passwords per second	100%
	3 meters (~10 feet)	5.3 usernames and passwords per second	87%
	4 meters (~13 feet)	2.8 usernames and passwords per second	45%
	5 meters (~16.4 feet)	1.24 usernames and passwords per second	5%
1024-bit RSA Key – 100 keys	1 meter (~3.3 feet)	2 keys per second	100%
	2 meters (~6.5 feet)	1.9 keys per second	98%
	3 meters (~10 feet)	1.7 keys per second	89%
	4 meters (~13 feet)	0.8 keys per second	43%
	5 meters (~16.4 feet)	N/A	0%
2048-bit RSA Key – 100 keys	1 meter (~3.3 feet)	1 key per second	100%
	2 meters (~6.5 feet)	0.9 keys per second	88%
	3 meters (~10 feet)	0.6 keys per second	65%
	4 meters (~13 feet)	0.3 keys per second	32%
	5 meters (~16.4 feet)	N/A	0%
50 Text Documents (1,000 characters) Note: Only 10 documents were transferred at 5 meters due to time constraints	1 meter (~3.3 feet)	4 seconds per document	100% bit-success 100% readable
	2 meters (~6.5 feet)	4 seconds per document	100% bit-success 100% readable
	3 meters (~10 feet)	4.4 seconds per document	90% bit-success 100% readable
	4 meters (~13 feet)	8 seconds per document	50% bit-success 80% readable
	5 meters (~16.4 feet)	50 seconds per document	8% bit-success 0% readable
PDF Document (140 KB)	1 meter (~3.3 feet)	~10 minutes per document	Not successful – Corrupted file
Word Document (240 KB)	1 meter (~3.3 feet)	~20 minutes per document	Not successful – Corrupted file

Table 6 shows the results for the same dataset and distances as in Method 1, but uses two-way communication (Method 2). Two-way communication should increase the success rate by sacrificing transfer rate due to the need for data re-transmission.

Table 6: Results of Method 2 - two-way communication

Types of Data	Transmission Distance	Transfer Rate	Success Rate
Username (10 characters) and Password (30 characters) – 1,000 username and password pairs	1 meter (~3.3 feet)	5 usernames and passwords per second	100%
	2 meters (~6.5 feet)	5 usernames and passwords per second	100%
	3 meters (~10 feet)	3.8 usernames and passwords per second	100%
	4 meters (~13 feet)	1 username and password per second	100%
	5 meters (~16.4 feet)	80 seconds per username and password	100%
1024-bit RSA Key – 100 keys	1 meter (~3.3 feet)	1.6 keys per second	100%
	2 meters (~6.5 feet)	1.5 keys per second	100%
	3 meters (~10 feet)	1.2 keys per second	100%
	4 meters (~13 feet)	3.5 seconds per key	100%
	5 meters (~16.4 feet)	N/A	0%
2048-bit RSA Key – 100 keys	1 meter (~3.3 feet)	1.3 seconds per key	100%
	2 meters (~6.5 feet)	1.6 seconds per key	100%
	3 meters (~10 feet)	3 seconds per key	100%
	4 meters (~13 feet)	12.5 seconds per key	100%
	5 meters (~16.4 feet)	N/A	0%
50 Text Documents (1,000 characters) Note: Only 2 documents were transferred at 5 meters due to time constraints	1 meter (~3.3 feet)	5 seconds per document	100%
	2 meters (~6.5 feet)	5 seconds per document	100%
	3 meters (~10 feet)	6 seconds per document	100%
	4 meters (~13 feet)	20 seconds per document	100%
	5 meters (~16.4 feet)	13 minutes per document	100%
PDF Document (140 KB)	1 meter (~3.3 feet)	11.6 minutes per document	Successful
	2 meters (~6.5 feet)	14.4 minutes per document	Successful
	3 meters (~10 feet)	18.2 minutes per document	Successful
	4 meters (~13 feet)	N/A	Not successful - Cancelled after an hour
	5 meters (~16.4 feet)	N/A	Not successful - Cancelled after an hour

Word Document (240 KB)	1 meter (~3.3 feet)	20 minutes per document	Successful
	2 meters (~6.5 feet)	27.7 minutes per document	Successful
	3 meters (~10 feet)	40.8 minutes per document	Successful
	4 meters (~13 feet)	N/A	Not successful - Cancelled after an hour
	5 meters (~16.4 feet)	N/A	Not successful - Cancelled after an hour

Discussion

The results show that data transmission is reliable up to two or three meters away, and using two-way communication can extend the limit up to four or five meters. Two-way communication improves the success rate but sacrifices transfer speed due to the number of re-transmissions required. However, this is not an issue because data exfiltration can be undertaken overnight. By comparing the results of the two audio-sources, speakers and headphones, an interesting trend is can be seen. Headphones were *more* effective at transferring data at three meters than the speakers were. This could indicate that the headphones are better at producing high-frequency sounds than the cheap speakers, or it could show that the power of the speakers is too high and we are getting interference from echoes and reflected signals.

The results indicate that our method is a viable threat to secure environments, and it could be used by insiders in a variety of ways to exfiltrate data from secure air-gapped systems. The advantage of our approach is that it can be implemented purely in software and does not require any special hardware modifications. This means that an insider could install a small piece of software, and an attacker could then gain long-term access to the infected system. Our results show that the transmission rates are quite slow compared to traditional communication methods, however it can be made quite reliable and it allows two-way communication with an infected machine over a long period of time. It is not unreasonable to think that an attacker could transmit some very sensitive data using this method. The methods presented here aren't just applicable to staff with a valid login, a cleaner or contractor could potentially infect a secure machine with one-off access using a USB key.

Covert data exfiltration methods are quite often very high risk to the insider. For example, if an employee is caught smuggling a stack of printed documents from the building then they have no defence if they are caught. The methods presented in this paper, however, are low risk. A simple software install is performed once, and then further access to the system can be performed covertly. Once the access is no longer required, the software can be signalled to erase itself and leave no physical trace, whereas some of the attacks presented in the Related Research section of this paper would leave some hardware behind. This hardware would either need to be recovered, increasing the risk of being caught, or abandoned, which would show evidence of the breach if it was ever discovered.

Employees can unwittingly or maliciously install the communication software presented here, which can then be used to gain remote access to an air-gapped, secure system. In contrast to methods that require hardware modifications, software can be installed by accident or bundled into legitimate software. Many hardware modifications cannot be accidentally performed, and require the attacker to directly manipulate the system hardware. A software-only solution has the potential to be installed without any physical access to the site or hardware, through a variety of difficult techniques used to trick the employees into facilitating its install. A disgruntled employee could potentially use this system to retain access to an air-gapped secure network even once they are fired or lose physical access.

Conclusion and Future Work

In summary, we have demonstrated the ability to exfiltrate data from an air-gapped system covertly, without any hardware modifications. The method we have presented in this paper, however, has many challenges to face. Commodity hardware is not designed to handle high frequencies outside of the human audible spectrum. High frequency signals cannot travel very far in air, and lose power on every reflection or echo. Yet even with these challenges, our system was still able to demonstrate reliable data transmission.

In the future, more complicated signal processing can be performed in order to correctly handle reflected signals. This would allow communication in a broader range of situations. Further transmission distances could be achieved by using intermediate devices as relay stations. These relay stations could repeat the signal so that the attacker could receive the data at a further distance. Transmission bandwidth can be improved by utilising more frequencies within the 20-23kHz range, or even by using multiple infected machines to transmit parts of the same data independently.

Currently our system can only communicate between two machines. However, we could potentially implement similar techniques used in Wi-Fi communication and produce a one-to-many communication protocol. This would allow a single command server to receive data from multiple infected targets. Infected machines might also be able to communicate with each other in order to roughly map out the physical layout of the environment.

Many of the limitations faced by our system are caused by the commodity hardware. If the attacker was able to install high-quality directional microphones in the target environment, then the maximum transmission distance might be significantly improved.

In conclusion, we would like to recommend that air-gapped computers should not be fitted with audio output devices as most air-gapped computers do not have a legitimate use for audio output devices. Headphones may be used but must be unplugged when not needed because we have shown that they can also effectively transmit data. Employees must remain vigilant that their computers do not make unusual sounds because it could potentially be data transmissions. Care must also be taken when adding any peripheral device to a computer to ensure that it cannot be used for covert data communication.

REFERENCES

Berrou, C., and Glavieux, A. 2003. "Turbo Codes," in *Wiley Encyclopedia of Telecommunications*, John Wiley & Sons, Inc.

Choo, K.-K. R. 2008. “Organised crime groups in cyberspace: a typology,” Trends in Organized Crime 11(3), pp. 270–295.

Choo, K.-K. R., and Smith, R. G. 2008. “Criminal exploitation of online systems by organised crime groups,” Asian Journal of Criminology 3(1) , pp. 37–59.

Choo, K.-K. R., Smith, R. G., and McCusker, R. 2007. “Future directions in technology-enabled crime : 2007-2009,” Research and public policy No 78, Canberra: Australian Institute of Criminology

Genkin, D., Shamir, A., and Tromer, E. 2013. "RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis," International Association for Cryptologic Research ePrint Archive.

Giani, A., Berk, V. H., and Cybenko, G. V. Year. "Data exfiltration and covert channels," SPIE2006, pp. 1-11.

Goodin, D. 2013. "Meet “badBIOS,” the mysterious Mac and PC malware that jumps airgaps," Ars Technica.

Ktims 2006. "An example demonstrating binary FSK," in *Wikipedia Commons*, Wikipedia.

Lee, J., Dhar, S., Abel, R., Banakis, R., Grolley, E., Lee, J., Zecker, S., and Siegel, J. 2012. "Behavioral hearing thresholds between 0.125 and 20 kHz using depth-compensated ear simulator calibration," *Ear Hear* (33:3) May-Jun, pp 315-329.

McConnell, J. M. 1995. "Red/Black Installation Guidance," in *NSTISSAM TEMPEST 2-95*, National Security Telecommunications and Information Systems Security.

Munro, K. 2012. "Deconstructing Flame: the limitations of traditional defences," *Computer Fraud & Security* (2012:10), pp 8-11.

Sanger, D. E., and Shanker, T. 2014. "N.S.A. Devises Radio Pathway Into Computers," New York Times, January 15, 2014.

Yali, L., Corbett, C., Ken, C., Archibald, R., Mukherjee, B., and Ghosal, D. Year. "SIDD: A Framework for Detecting Sensitive Data Exfiltration by an Insider Attack," Proceedings of the 42nd Hawaii International Conference on System Sciences 2009, pp. 1-10.