Zhice Yang *ShanghaiTech University,*
Qianyi Huang *Hong Kong University of Science and Technology*
Qian Zhang *Hong Kong University of Science and Technology*

Editors: Nic Lane and Xia Zhou

BACKSCATTER AS A COVERT CHANNEL IN MOBILE DEVICES

Excerpted from "NICScatter: Backscatter as a Covert Channel in Mobile Devices" from MobiCom '17, *Proceedings of the 23rd Annual ACM International Conference on Mobile Computing and Networking*, with permission. https://dl.acm.org/citation.cfm?id=3117814 © ACM 2017

Mobile devices, including laptops, smartphones, wearables, etc., have become essential tools in modern life. We rely on them for social activities, document processing, and health status monitoring. As these devices contain sensitive personal information, various security mechanisms, such as firewalls, traffic monitors, and information flow control systems [1], have been developed for mobile devices to prevent unauthorized data leakage.

Nevertheless, our data is not safe enough. Covert channels remain an open threat to data security. A covert channel allows an attacker to leak information from a compromised system even without establishing explicit networked or logical connections. Covert channels exploit media that are usually not treated as data communication channels, and thus they are stealthy and difficult for security systems to detect. For example, electromagnetic radiation (EMR) from a display interface, EMR from a CPU-to-memory bus, the magnetic field from a hard disk, etc. are feasible media to establish covert channels [2].

This article summarizes a covert channel threat on existing mobile systems. Through it, malware can wirelessly leak information without having to make network connections or emit signals, such as sound, EMR, vibration, etc., that we can feel or are aware of, and thus bypass existing security protection systems. The covert channel is built on the discovery of a novel communication method that we call *NICScatter*. NICScatter is a kind of backscatter communication. Specifically, it works with commercial NICs. As shown in Figure 1, it enables mobile devices, such as mobile phones, tablets or laptops, to reflect surrounding RF signals to convey information. We also report flaws in current mobile systems that make operations of NICScatter free of special permissions, and thus the malware is stealthy, hard to detect and brings security risks.

Photo, istockphoto.com

BACKSCATTER COMMUNICATION

Backscatter is a wireless communication method. A backscatter transmitter reflects surrounding RF signals to convey information. It is very power-efficient since the transmitter does not consume power in actively generating RF signals, like conventional RF transmitters do. As a result, it has been widely used in communication scenarios with ultra-low power constraints, such as with RFID tags.

The basic working mechanism of backscatter is shown in Figure 2. When an incident RF signal encounters the antenna of the backscatter transmitter, a fraction of the signal is reflected/scattered off the antenna. When the load impedance Z_c of the antenna is different, say Z_{c1} or Z_{c2}, the amplitude of the reflected signal is also different. This property allows a backscatter transmitter to modulate the reflected signals to convey information by simply switching the load impedance. On the other hand, the receiver decodes the information according to the amplitude changes in the reflected signals.

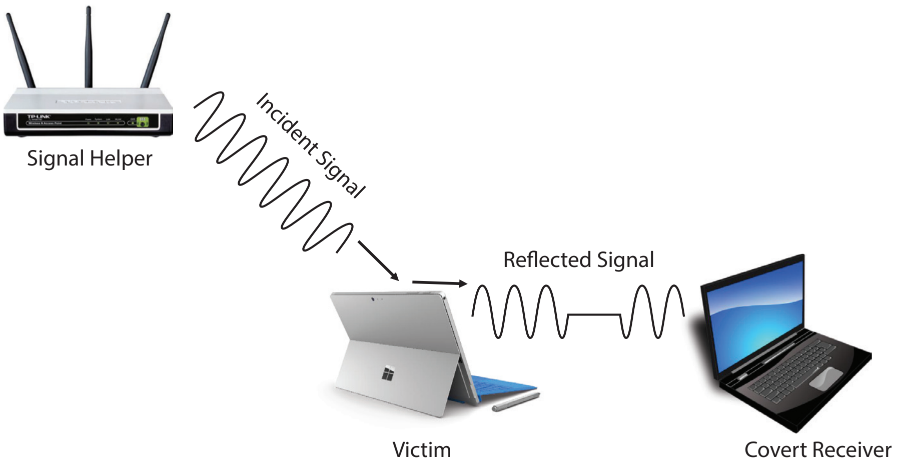

FIGURE 1. Attack Scenario. The malware in the victim leverages NICScatter to reflect the RF signal from the Signal Helper to covertly leak sensitive information, e.g., a password, to the nearby Covert Receiver.

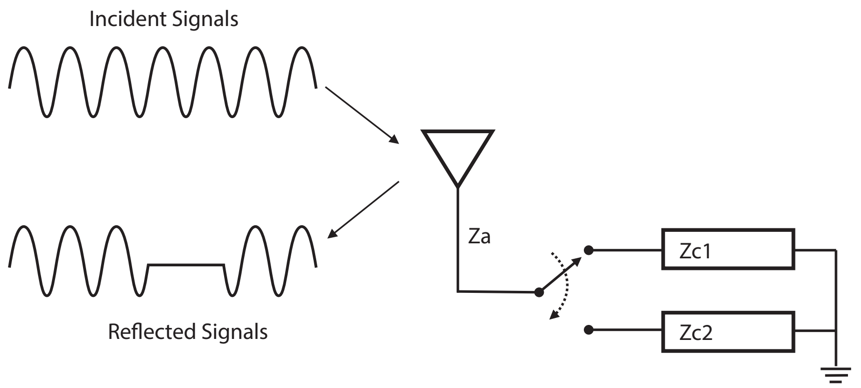

FIGURE 2. Backscatter Communication. The switch of a transmitter toggles its antenna's load to different impedances. As a result, the RF signals reflected by the antenna have different amplitudes, which are used to convey information, e.g., a stream of encoded data, to the receiver.

TABLE 1. NIC Impedance (Ω). Wi-Fi NICs have different impedances in different working states.

Model	Off state	Unstable state	On state
AR9380	22.5 + j9.2	36.3 - j10.0	22.5 + j9.2
BCM1045	12.4 + j14.4	N/A	106.9 – j9.8
Intel5300	119.0 + j137.0	N/A	80.3 + j66.8

BACKSCATTER WITH COMMERCIAL NICS

Backscatter relies on dedicated hardware, i.e., the impedance switching circuits, to manipulate the RF reflections. As consumer mobile devices lack such hardware, backscatter is typically not considered as a possible communication method for them. NICScatter challenges such conventional wisdom. Its design is based on an interesting observation that commercial Wi-Fi NICs have different circuit impedances in different working states.

We measured the impedance of Wi-Fi NICs from some major manufacturers. During the measurement, the NIC was switched between on and off states. The impedance readings are recorded in Table 1. We observe that the impedance of all the NICs changes when they are switched from the off state to the on state.

The design of NICScatter is based on this observation. Similar to the impedance switch in Figure 1, NICScatter uses software instructions to switch the commercial Wi-Fi NICs between on and off states. Different circuit impedances in the two states result in different amplitudes of the reflected RF signals of the mobile device. In this sense, Wi-Fi NICs are also able to modulate reflections like typical backscatter transmitters.

However, when switching the working states of commercial NICs, the detailed process of the impedance change contains complexity and is more than a simple two-value cycling. The study of these complexities is the guideline for the communication design of NICScatter. Although the internal logic of NIC chips is a black box for us, some of the characteristics can be observed from the waveforms reflected by the NICs. In the topology in Figure 1, the captured waveforms at the receiver are shown in Figure 3.

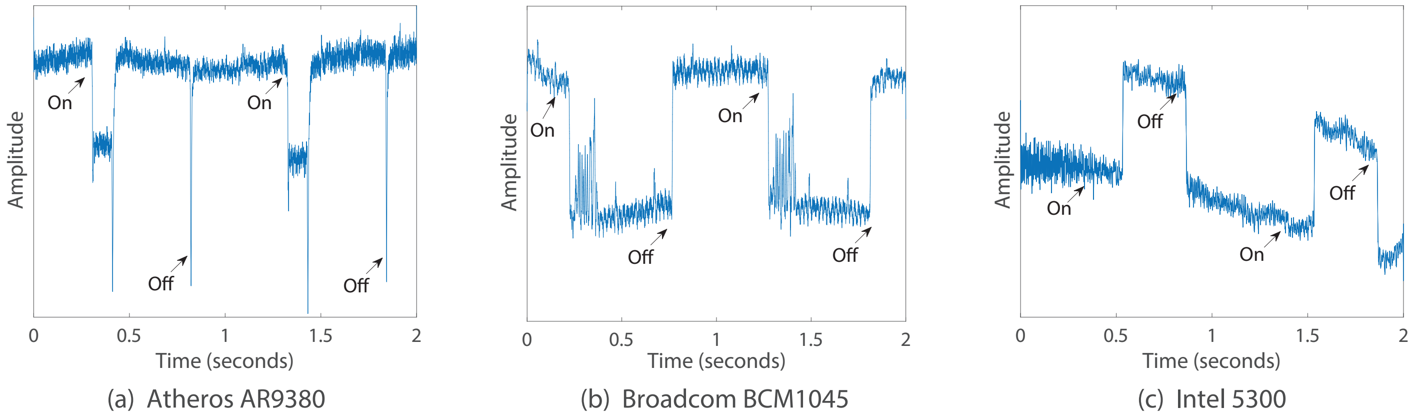

FIGURE 3. Waveforms of the Reflected Signals. A sine signal at 2.4GHz is reflected by commercial NICs when they are turned on and off every 0.5 seconds. The amplitude changes in the reflected RF signals are caused by different NIC impedances in different working states.

We observed that the waveforms (and thus the impedance changes) are quite diverse among different Wi-Fi NICs. The diversity stems from different hardware types, software drivers and locations of devices. We refer the readers to our full paper [3] for a more detailed summary. Here we highlight the properties that classify these NICs into two types.

As shown in Figure 3, the impedance of the first type is almost static in the on or off state. Intel 5300 and BCM1045 belong to this type. We term them as the *static type*. The impedance of the second type only changes in the short period while turning from on to off or off to on. The corresponding waveform in the reflected RF signals is like a pulse or a bump wave. AR9380 belongs to this type. In Figure 3 (a), when the NIC is switched from off to on, it shows a square-shaped bump lasting for 0.1 seconds. When the NIC is switched from on to off, it has a very short pulse. We term NICs of this type as the *pulse type*. The design goal of the modulation scheme of NICScatter is to be compatible for different NIC types. Therefore, according to the above properties of different NICs, especially the pulse type, an intuitive choice is pulse position modulation (PPM). In PPM, time is divided into equal time slots, and a single pulse is transmitted during each time slot. The position of the pulse in each time slot is used to represent information. We consider NICs of different types. For NICs of the pulse type, a single off-to-on or on-to-off instruction is enough to generate a pulse in the reflection. For NICs of the static type, we need an immediate off-to-on and on-to-off pair to generate a pulse. Note that, in both types, we can use the quick off-to-on and on-to-off pair to generate the pulse in the reflection.

Receiving and decoding the pulses in the reflections do not necessarily require a powerful RF receiver. The pulses can be captured by RSSI values, which are provided by most NICs. NICs providing CSI values have a fine-grained receiving resolution, and thus a better decoding ability.

FLAWS IN CURRENT MOBILE SYSTEMS

Manipulating Wi-Fi NIC sounds quite sensitive. However, we show that in current Linux and Android OSs, operations of a NICScatter transmitter do not require special privileges. Thus, communication through NICScatter can be stealthy and hard to detect.

In Linux systems, we leverage the vulnerability of the rfkill subsystem [4]. rfkill is an interface that resides in the Linux kernel to switch radio devices on and off. One typical event handled by rfkill is the radio button/airplane mode button on many laptops. rfkill is controlled by device file at /dev/rfkill. Writing a structured command, e.g., “block wifi”, to /dev/rfkill can switch, e.g., turn off the Wi-Fi radio.

We note that the write permission of the /dev/rfkill does not require root privileges in many Linux desktop distributions. The vulnerability is caused by Gnome-Bluetooth, which is an application shipped with the Gnome desktop environment. For some historical reasons, the application grants the write permission of /dev/rfkill to normal users. Therefore, any program can use the rfkill interface to switch the Wi-Fi NIC without root privileges.

In Android systems, apps using NICScatter only require normal permission, which is CHANGE_WIFI_STATE. Note that Android classifies app permission into two categories [5]. One is the dangerous level, which will explicitly notify users during app installation. The other is the normal level, which is treated as low risk and will not notify the user by default.

ATTACKS THROUGH NICScatter

Leveraging flaws in the current mobile systems, malware can leak information through NICScatter in a covert way. In the attack scenario in Figure 1, we assume that the mobile devices, e.g., smartphones or laptops, have somehow been compromised by an attacker and have had malware installed with NICScatter capability. The malware has access to certain sensitive data of interest but avoids using the network, because network traffic, as a major information flow, is normally under severe monitoring by many security mechanisms [1]. Moreover, if an application with access to sensitive data also requires public network connections, this would raise the serious attention and awareness of users. Leveraging NICScatter instead of network connections to leak information makes the malware inconspicuous and stealthy.

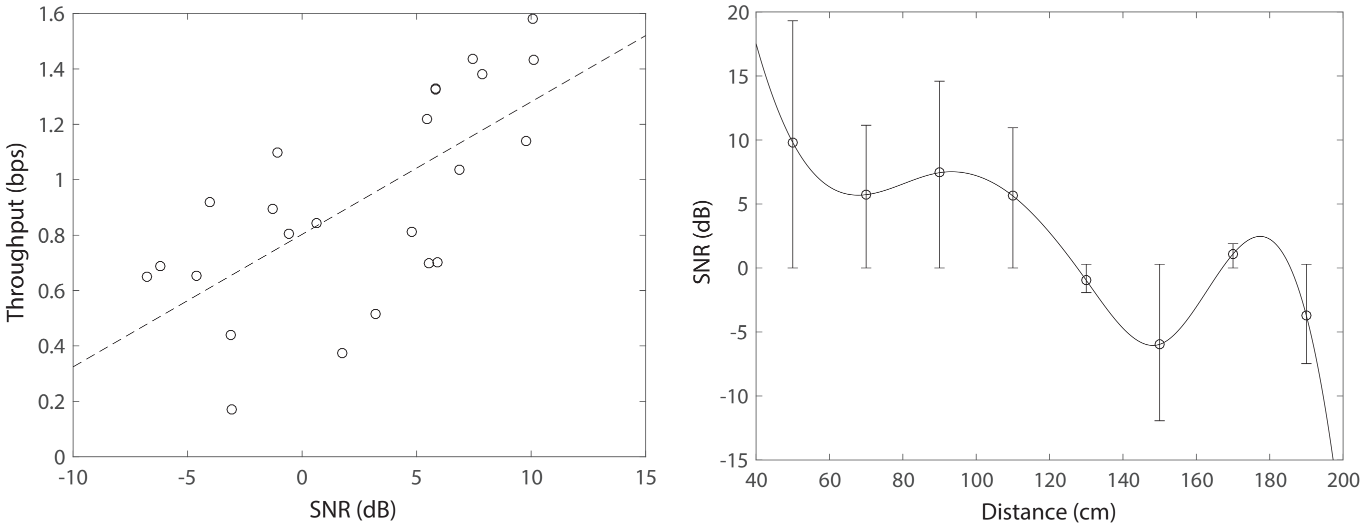

FIGURE 4. NICScatter Performance. The left subfigure shows the throughput under different channel conditions. The right subfigure shows the performance when the receiver and transmitter are separated by different distances.

A covert receiver is positioned close to but isolated from the compromised mobile device, with no physical or logical connections between them. We assume that the covert receiver can exploit the RF signals generated by a signal helper, e.g., a nearby neutral access point, to reveal the transmissions of the NICScatter transmitter.

Attacks are most likely to occur in public places, such as cafés, libraries, transportation stations, etc. The attacker uses his/her laptop/AP as the receiver to gather information from neighboring victims. In another case, as modern mobile machines like UAVs also have the capability to receive and decode covert information, they could be abused to gather information from victims in a dense residential environment.

Evaluation experiments conducted under the above settings demonstrate the attack in practice. Results show that the NICScatter transmitter works for different Wi-Fi NICs and platforms. It can achieve a bitrate of up to 1.6 bps and transmit as far as 2 meters (Figure 4). In a through-the-wall scenario, it can transmit up to 70 cm.

CONCLUSION

This paper introduces NICScatter, a backscatter communication method, which relies on commercial Wi-Fi NICs. NICScatter's design is based on the property that the impedance of a Wi-Fi NIC varies in different working states. The NICScatter transmitter switches NIC hardware between the on and off states to modulate the RF signals reflected by its antenna. The NICScatter receiver extracts information by analyzing the amplitude of the reflected signals of the transmitter. As the communication form of NICScatter hasn't caused enough attention in current mobile systems, we show there are vulnerabilities to exploit NICScatter as a covert way to leak information.

In addition to the covert nature, the insights from NICScatter also imply several interesting points that are worth exploring in future work. NICScatter inherits the beneficial properties in backscatter signals. For example, as we already have knowledge on localizing RFID tags [6], it is possible to reuse the same method to localize conventional wireless devices through NICScatter. Further, although we only verified NICScatter on Wi-Fi NICs in this paper, we believe the method we use, i.e., manipulating the impedance of the RF circuits through changing its working states, may exist in other RF transceivers, such as cellular, Bluetooth, etc., which may bring other interesting communication opportunities and security issues that are publicly unknown. ■

Zhice Yang is an assistant professor at ShanghaiTech University, China. He obtained his PhD from the Hong Kong University of Science and Technology in 2016. His current research is on wireless and mobile systems. yangzhc@shanghaitech.edu.cn

Qianyi Huang is a PhD student in the Department of Computer Science and Engineering at Hong Kong University of Science and Technology, Hong Kong. She received a BS degree in Computer Science from Shanghai Jiao Tong University. Her research interests lie in mobile computing, Internet of things, and digital health. qhuangaa@connect.ust.hk

Dr. Zhang is Tencent Professor of Engineering and Chair Professor of the Department of Computer Science and Engineering at Hong Kong University of Science and Technology. She has published more than 300 refereed papers in the areas of wireless networks, IoT, and mobile systems. She is a Fellow of IEEE. qianzh@cse.ust.hk

REFERENCES

[1] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. "Taintdroid: an information flow tracking system for realtime privacy monitoring on smartphones." *ACM Transactions on Computer Systems* (TOCS), 32(2):5, 2014.

[2] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici. "Gsmem: Data exfiltration from air-gapped computers over gsm frequencies." In *USENIX Security,* pages 849–864, 2015.

[3] Z. Yang, Q. Huang, Q, Zhang. "NICScatter: backscatter as a covert channel in mobile devices." In *ACM MobiCom* 2017.

[4] rfkill subsystem. https://www.kernel.org/doc/Documentation/rfkill.txt.

[5] Android permissions. https://developer.android.com/reference/android/Manifest.permission.html.

[6] J.Wang and D. Katabi. "Dude, where's my card?: RFID positioning that works with multipath and non-line of sight." *ACM special interest group on data communication*, 43(4):51–62, 2013.