Utilizing the «MouseJack» vulnerability in physical assessments

Maksim Iavich [1], Sergei Simonov [1]

[1] *Caucasus University, 1 Paata Saakadze St, Tbilisi, 0102, Georgia*

Abstract

Due to the absence of proper security measures, a huge number of wireless input devices are susceptible to keystroke injection attacks. This makes them an attractive target for attackers. During such an attack, the attacker can mimic a remote keyboard and send any desired text string to the victim machine. This can result in a fast and covert compromise of the system. Antivirus software will not detect the attack, as the keyboard, even if it is remote, is not inherently malicious and is always considered trustworthy. It is very interesting to create the corresponding methodology during the penetration test. The paper illustrates the vulnerability of the wireless input devices, their exploitation methodology, analysis of the possible attacks and payloads. The paper offers the methodology of using the vulnerabilities of wireless input devices during the penetration test.

Keywords

Wireless security, Mousejack, penetration testing, vulnerability

1. Introduction

As the time goes, more and more enterprises and small businesses start utilizing wireless input devices, such as keyboards and mice, which is logical, as these devices require no wires and can be utilized to achieve better user experience. The receiver USB dongle is being connected to the device (PC, Laptop, etc.) and receives the keystrokes from the keyboard. In a lot of cases, the communication between these devices is encrypted, but, often, the receiver dongle can be tricked into reading and executing the unencrypted commands sent by the attacker. Though the eavesdropping is not possible, an attacker can still act as a remote keyboard and send malicious commands to the receiver USB dongle. As a result, this can lead to quick compromise of the victim device, utilizing the vulnerable receiver dongle [1]. Antivirus software is unable to detect the attack, as the payload is being sent in a form of keyboard commands, and no keyboard is being treated like a malicious device. It can be told about the "RubberDucky". The process is illustrated on figure 1.

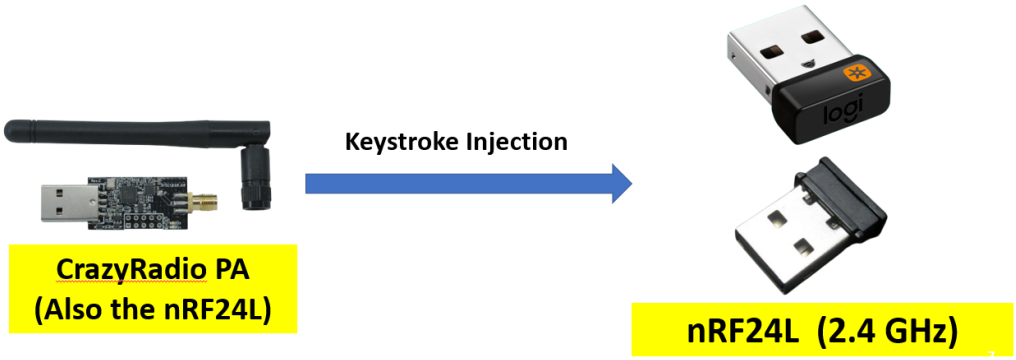

Figure 1. Keystroke injection

INFORMATION SOCIETY AND UNIVERSITY STUDIES (IVUS 2023), 12 May 2023, Kaunas, Lithuania
EMAIL: miavich@cu.edu.ge (M. Iavich); s_simonovi@cu.edu.ge (S. Simonov)
ORCID: 0000-0002-3109-7971 (A. 1)

The goal of the paper is offer the methodology of using keystroke injection attacks during the penetration test. For this, the paper offers the study of the keystroke injection attacks, and the analysis of the hardware vulnerable to these attacks.

2. MouseJack Vulnerability

Announced by Bastille in 2016, Mousejack is a class of vulnerabilities affecting the majority of wireless non-bluetooth mice and keyboards. USB receivers use 2.4 GHz frequency for the communication and are often built with the nRF24L series transceivers. Using the "CrazyRadio PA" device, it is possible to capture the communication between the keyboard and the receiver [2-4]. After that, the data is being collected and fuzzed. As the receiver security mechanisms of each vendor differ, there are a lot of discovered vulnerabilities, but all of them can be divided in the following categories:

2.1. Keystroke injection, spoofing a mouse

When processing the received packets, some receiver dongles don't verify if the type of packet received matches the type of device that transmitted it. Usually, a mouse will only transmit clicks/movement to the dongle, and a keyboard will only transmit keypresses. If the dongle doesn't verify that the packet type and transmitting device type match, there is a possibility for an attacker to spoof the mouse, but transmit a keypress packet. The dongle is not expecting packets coming from a mouse to be encrypted, as a result, it accepts the keypress packet, giving the attacker the opportunity to type arbitrary commands on the victim's computer [5,6]. The process is shown on Figure 2.

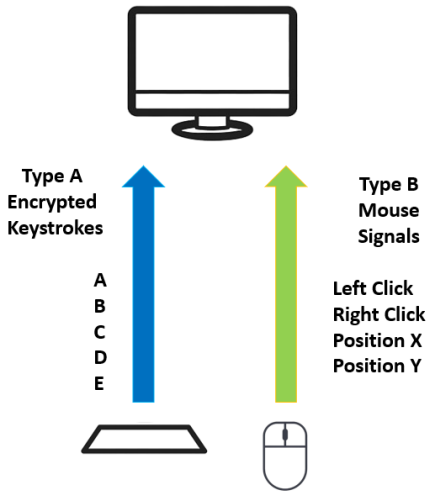

Figure 2. Packet exchange

2.2. Keystroke injection, spoofing a keyboard

The majority of the tested keyboards have to encrypt data before sending it wirelessly to the dongle, but not all of the dongles require encryption to be in place. This creates an opportunity for an attacker to pretend to be a keyboard, and transmit unencrypted keyboard packets to the dongle. This technique bypasses the encryption which is normally used by the keyboard, allowing an attacker to inject arbitrary commands on the victim's dongle [7-10].

2.3. Forced pairing

When the wireless mouse or keyboard is created, it is paired with a dongle. This means that it has the information about the wireless address of the dongle, and in the case of a keyboard, the encryption key to encrypt the transmitted data. Luckily for attackers, some vendors include the ability to pair an

existing keyboard or mouse with a new dongle or pair new devices with a dongle. For example, if a user has the dongle lost, it means that he only needs to buy a new dongle and an entirely new set of keyboard and mouse [11-14].

In order to prevent unauthorized devices from pairing with a dongle, it will only accept new devices when placed into a pairing mode, which has to be performed by the user, which lasts for 30-60 seconds.

There is a possibility to bypass this pairing mode on some of the dongles and pair a new device without user interaction. In the case of a victim only having a mouse, but using a dongle vulnerable to keystroke injection by spoofing a keyboard, an attacker can pair a fake keyboard with the dongle, resulting in using it to inject arbitrary commands on the victim's receiver dongle.

In order to perform the attack, attacker has to have a "CrazyRadio PA" (or any other dongle with the same capabilities) and a software to control the process. Example software can be "JackIt". An attacker can launch the attack within the 100 meters, but, attaching the directed antenna to the "CrazyRadio PA" dongle can increase the distance.

3. The offered methodology

The mousejack can be used in physical assessments as and additional, unexpected attack vector. As the wireless input devices are commonly used and have a wide spread, this vector has a high rate of success and is rather fruitful.

After organizing the survey of 100 organizations 28 of them had the mousejack vulnerability. This number is rather big. Therefore, using this vulnerability could be the good approach for the penetration test.

We think to offer the automatic tool, which will test the devices for mousejack vulnerability. It the vulnerability is found the corresponding attack must be occurred.

After the attack is performed, the tool will provide the following actions:

- Network enumeration
  - Wi-Fi network password extraction
  - ARP scan
  - Routing table information
  - Some manual activities
- Pivoting, giving us a lot of opportunities. The manual intervention can be needed during the process.

Even if enterprise we are assessing utilizes IDS/IPS mechanisms, firewalls and physical security, one single vulnerable dongle would be enough for the "red team" to get the foothold on the network.

As the attacker acts as a keyboard, this attack is stealthy and cannot be detected by the Antivirus/Antimalware/IDS/Firewall software. After gaining the code execution, attacker could open the "cmd" as administrator (UAC bypass is performed by simply sending the "left arrow", "enter" keys sequence), download the "netcat" and launch a reverse shell. This action is not likely to be spotted by the system protecting software. Another approach could be triggering a cloud payload with the following powershell command:

powershell IEX (New-Object Net.WebClient).DownloadString("<URL containing the malicious .ps1 script>")

or the attacker can utilize the reverse shell one-liner:

$client = New-Object System.Net.Sockets.TCPClient("<Attacker IP>",<Attacker PORT>);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + "PS " + (pwd).Path + "> ";$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()

The approach is illustrated on Figure 3.

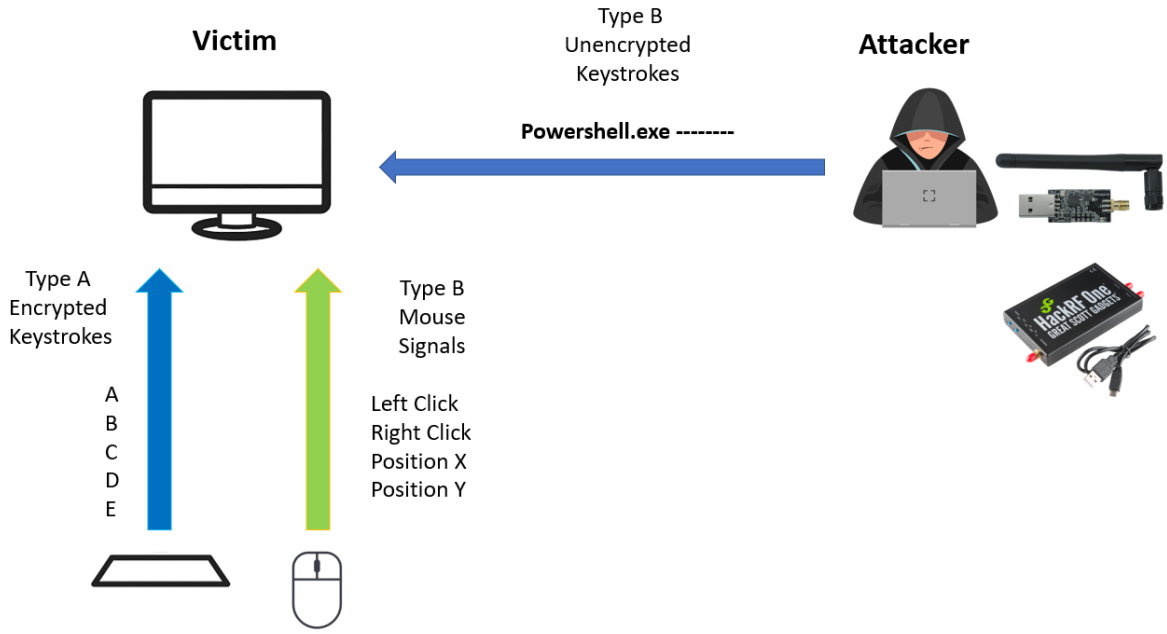

Figure 3. PowerShell attack

Obtaining the reverse shell or extracting information from the victim machine can be performed in a lot of different ways and is only limited by the attacker's knowledge, skill, experience and imagination. Here is a list of possible attacks:

- Keylogger installation
- Backdoor/Trojan Installation
- C2 Beacon Installation
- NTLM Hashes extraction
- Website sessions extraction
- Add a new admin user
- Enable the RDP

All of these attacks vectors should be tried during the penetration test. These attacks are illustrated on Figure 4.

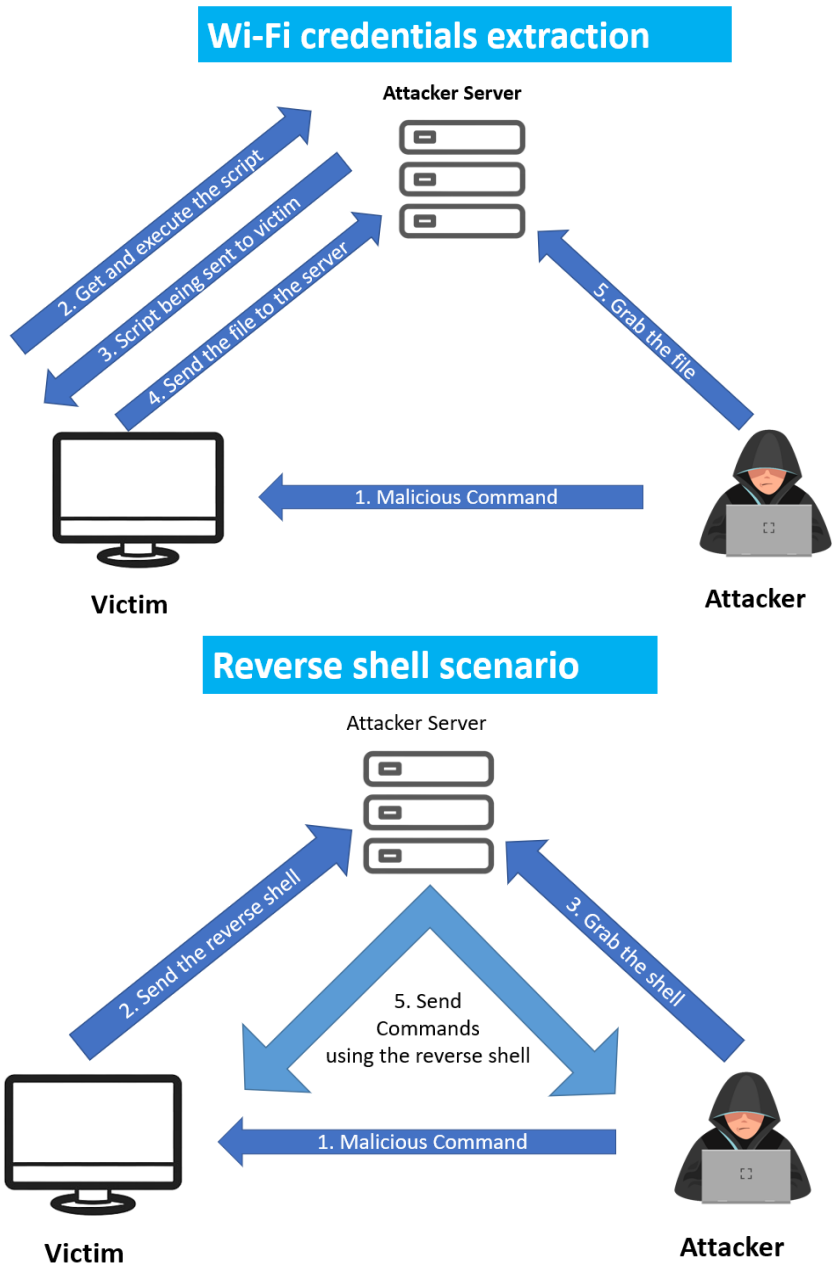

Figure 4. Attacks vectors

4. Impact

The “black hats” could also take a huge advantage of this vulnerability. By default, the reach range of the attack is 10-50 meters, but, if the attacker uses the directed antenna, this range can be amplified to 100 meters and beyond. Let’s imagine the datacenter employee using the vulnerable keyboard dongle and the attacker with the directed antenna. This might result in the massive breach and the compromise of the datacenter. All user’s information might be lost. The attack is illustrated on figure 5.

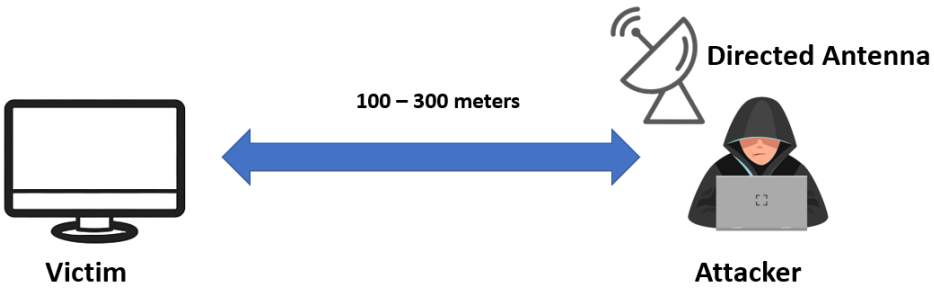

Figure 5. The “black hats” attack

Therefore, the mentioned attack vector must be added to the penetration process.

5. Relevance

The mousejack is not a vulnerability of the single device. It is the class of vulnerabilities and even if it was discovered 6 years ago and major vendors like Logitech patched the flaw, there are still a lot of other popular vendors like trust or defender which may suffer from this bug, which means that mousejack is relevant even today. Also, the amount of already produced vulnerable hardware Is huge and even if there were firmware patches produced, it’s really unlikely that any of users would update the firmware on their keyboard or mice dongle. The integration of the methodology offered by us, can prompt the organizations about such vulnerability. Also our method can help us to check the network for the different attacks vectors.

6. Advantages and disadvantages of the offered methodology

The keystroke injection can never be spotted by the antivirus software. The attacker is spoofing the keyboard and the keyboard is always a trusted device. To use the keystroke injection, the attacker does not have to be on the same local network. The only requirement is a close physical proximity.

Users often run their systems as the high privilege users, and using the keystroke injection we can run any command as the user they are logged in. So, in the most cases we will not need to escalate our privileges.

However, it must be mentioned that the red team has to guess the operating system. When performing the keystroke injection, the only information you see about your victim is the MAC address of their USB dongle. Another disadvantage is the reach range of the attack, which is not great by default, but this can be mitigated by using the directed antennas. Therefore, the direct antennas must be used the penetration testing process.

7. Experiments

We have carried out several experiments, involving the attacker notebook with the “Ubuntu 20.04” operating system and “JackIt” software. The notebook was equipped with the “CrazyRadio PA” dongle. The victim machine was using the Logitech C-U0007 receiver and the K360 keyboard. Attack was successful, and the full system compromise was achieved with no antivirus software being triggered.

We have checked our methodology during 5 penetration tests. The part of the team did not use the offered methodology and another part of the team used. The group who used the offered methodology had much better report.

In 2 reports the penetration testers, which did not use the offered methodology could not get the root access to the system and in other 3 reports the penetration, which did not use the offered methodology did not find 10-15% of the vulnerabilities.

8. Further research goals

As for the future, we plan to investigate less known, but not uncommon vendors (Defender, trust, etc.) for the mousejack vulnerability and assemble the toolkit for their exploitation.

We also plan to create the fully automate tool using the offered methodology.

9. Acknowledgements

This work was supported by Shota Rustaveli National Science Foundation of Georgia (SRNSF) [STEM – 22 -1076].

10. References

[1] MouseJack White Paper, https://github.com/BastilleResearch/mousejack/blob/master/doc/pdf/MouseJack-whitepaper-v1.1.pdf.

[2] Zhao, S., Wang, X.A. (2020). A Survey of Malicious HID Devices. In: Barolli, L., Hellinckx, P., Enokido, T. (eds) Advances on Broad-Band Wireless Computing, Communication and Applications. BWCCA 2019. Lecture Notes in Networks and Systems, vol 97. Springer, Cham. https://doi.org/10.1007/978-3-030-33506-9_71.

[3] Malisa, L., Kostiainen, K., Knell, T., Sommer, D., Capkun, S. (2017). Hacking in the Blind: (Almost) Invisible Runtime User Interface Attacks. In: Fischer, W., Homma, N. (eds) Cryptographic Hardware and Embedded Systems – CHES 2017. CHES 2017. Lecture Notes in Computer Science(), vol 10529. Springer, Cham. https://doi.org/10.1007/978-3-319-66787-4_23.

[4] Rieb, A., Lechner, U. (2017). Towards a Cybersecurity Game: Operation Digital Chameleon. In: Havarneanu, G., Setola, R., Nassopoulos, H., Wolthusen, S. (eds) Critical Information Infrastructures Security. CRITIS 2016. Lecture Notes in Computer Science(), vol 10242. Springer, Cham. https://doi.org/10.1007/978-3-319-71368-7_24.

[5] E. Levy, "Interface illusions," in IEEE Security & Privacy, vol. 2, no. 6, pp. 66-69, Nov.-Dec. 2004, doi: 10.1109/MSP.2004.104.

[6] S. Mondal and P. Bours, "Combining keystroke and mouse dynamics for continuous user authentication and identification," 2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), Sendai, Japan, 2016, pp. 1-8, doi: 10.1109/ISBA.2016.7477228.

[7] A. Negi, S. S. Rathore and D. Sadhya, "USB Keypress Injection Attack Detection via Free-Text Keystroke Dynamics," 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2021, pp. 681-685, doi: 10.1109/SPIN52536.2021.9566083.

[8] L. Arora, N. Thakur and S. K. Yadav, "USB Rubber Ducky Detection by using Heuristic Rules," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 156-160, doi: 10.1109/ICCCIS51004.2021.9397064.

[9] K. Nasaka, T. Takami, T. Yamamoto and M. Nishigaki, "A Keystroke Logger Detection Using Keyboard-Input-Related API Monitoring," 2011 14th International Conference on Network-Based Information Systems, Tirana, Albania, 2011, pp. 651-656, doi: 10.1109/NBiS.2011.109.

[10] K. Xu, H. Xiong, C. Wu, D. Stefan and D. Yao, "Data-Provenance Verification For Secure Hosts," in IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, pp. 173-183, March-April 2012, doi: 10.1109/TDSC.2011.50.

[11] T. Claverie and J. L. Esteves, "BlueMirror: Reflections on Bluetooth Pairing and Provisioning Protocols," 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2021, pp. 339-351, doi: 10.1109/SPW53761.2021.00054.

[12] Baldo, M., Farine, M., Lombardo, U. et al. Surface behaviour of the pairing gap in a slab of nuclear matter. Eur. Phys. J. A 18, 17–23 (2003). https://doi.org/10.1140/epja/i2003-10064-8.

[13] Perez, E., Li, F., Tareste, D. et al. The Surface Force Apparatus to Reveal the Energetics of Biomolecules Assembly. Application to DNA Bases Pairing and SNARE Fusion Proteins Folding. Cel. Mol. Bioeng. 1, 240–246 (2008). https://doi.org/10.1007/s12195-008-0025-7.

[14] Molina, R.A. Pairing and spectral statistics of low energy levels. Eur. Phys. J. A 28, 125–128 (2006). https://doi.org/10.1140/epja/i2005-10282-0.