KRACK and the Ethics of Disclosure

Adon Shapiro

December 13, 2017

Abstract

Vulnerabilities and exploits in software are discovered every day. This is a well-known fact in the cyber-security industry. What is not so universally agreed-upon is the proper way for an ethical hacker or security researcher to disclose a vulnerability they have discovered, and the argument re-surfaces with the disclosure of each new high-profile bug. The most recent such bug is KRACK. In October 2017, researchers at the University of Leuven published details of a vulnerability they had discovered in the WPA2 protocol that secures Wi-Fi connections. They called it KRACK (Key Reinstallation AttaCK) because of the way it takes advantage of a bug in WPA2 that allows the re-use of cryptographic keys. KRACK is significant because it is a fundamental weakness not in a specific piece of software, but rather in the Wi-Fi standard itself, and so everyone who uses Wi-Fi was susceptible to the attack. Because of its recentness and its far-reaching impact, KRACK is a perfect lens through which to view the problem of ethical disclosure.

Introduction

Disclosure, as it pertains to cyber-security, refers to the manner in which people are made aware of vulnerabilities in their software. This can mean many different things, however. To formalize different approaches to disclosure, it is helpful to define three roles in the process: Vendors, Customers, and Reporters. As defined by Christey and Wysopal, a Vendor is "an individual or organization who provides, develops, or maintains software, hardware, or services," a customer is an "end user of the software, hardware, or service that may be affected by the vulnerability," and a Reporter is an "individual or organization that informs (or attempts to inform) the Vendor of the vulnerability."[1] The crux of the issue is how

[1] Christey & Wysopal, *Responsible Vulnerability Disclosure Process.*

different approaches to disclosure affect Customers, and the "different methods" refer to the actions and reactions of Reporters and Vendors. At the extreme ends of the disclosure spectrum we have full disclosure and non-disclosure. Full disclosure is when a Reporter makes their discovery completely public with little to no consideration given to the consequences of publicizing an exploit. The reasoning behind this method is generally that the best way to get a Vendor to fix a vulnerability is to have it be as noticeable as possible, and that the consequences of malicious individuals learning of the attack are negligible, as it may well have already been known by less ethical hackers.[2] Non-disclosure involves keeping the veulnerability as secret as possible. If a Reporter even alerts a Vendor, it is unlikely either will alert the public, sometimes even after the vulnerability has been taken care of. The rationale being that Customers will be safest if the exploit is kept completely secret, since no one with questionable motives will be aware either.[2] Of course, these are both extremes and in reality, most vulnerabilities are disclosed in a manner somewhere between these two approaches. Most bugs, KRACK included, are disclosed in manner known as "responsible disclosure," which attempts to find a happy medium, but what this actually means is also a matter of contention.

KRACK is an inherent weakness in the WPA2 protocol that secures protected Wi-Fi networks. Essentially, the vulnerability allows an attacker to authenticate themselves by forcing the network to reinstall an already in-use cryptographic key (thus the name KRACK, for Key Reinstallation AttaCK). The exploit was discovered by Belgian researchers Mathy Vanhoef and Frank Piessens at KU Leuven in 2016 and published a paper with the results of their findings in October 2017, however some affected Vendors were notified earlier than this.[3] On the spectrum of disclosure, this is what most people would label responsible disclosure with perhaps a bit of a bent towards full disclosure. Though some vendors were alerted before the bug was fully disclosed, ultimately the vulnerability was made fully known to the public, and soon after, most Vendors released patches to their implementations of WPA2.

[2]Stephen A. Shepherd, *Vulnerability Disclosure: How do we define Responsible Disclosure.*
[3]Catalin Cimpanu, *New KRACK Attack Breaks WPA2 WiFi Protocol.*

To the Community

So far, it may seem that KRACK is unremarkable. It was discovered by academic researchers, disclosed in a relatively uncontroversial manner, and quickly patched. But it is easy to overlook the severity of the exploit. For one thing, almost anyone who uses the internet is susceptible to it. I use the present-tense because the fact that a patch has been released has absolutely no bearing on whether a user has a properly updated system, especially considering how recently the bug was disclosed. WPA2 is the standard for secure wireless access and is relied upon for protecting almost every modern Wi-Fi network, so the only way to be immune from KRACK without a patched WPA2 implementation is to use a wired connection.

In terms of disclosure, KRACK's fresh discovery allows us to view the ramifications of its disclosure in real time, and its academic source gives it an unusual amount of legitimacy. All told, it is an interesting opportunity to gain new perspective on the old debate.

Technical Details of the Exploit

Since KRACK is a fundamental vulnerability in the WPA2 protocol, some understanding of the protocol is necessary to comprehend the specifics of a KRACK attack. Wi-Fi Protected Access II (WPA2) is the name given to implementations of the standard for secure Wi-Fi outlined in IEEE 802.11i-2004. The key elements of the protocol which allow encrypted traffic to flow between a client and an access-point without transmitting the encryption key itself are the four-way handshake and group key handshake. First, a user must provide a pre-shared key for initial authentication (the Wi-Fi password). A secret Pairwise Master Key (PMK) is then generated for the session. Next, the four-way handshake verifies that both the access point and the client have the proper PMK without needing to disclose the key. Once this is established, two more keys are generated: the Pairwise Transient Key

(PTK) and the Group Temporal Key (GTK).[4]

KRACK exploits a vulnerability in the four-way handshake process of the protocol. The GTK is installed after the third message of the four-way handshake, but to account for dropped or interrupted connections, WPA2 allows the re-transmission of this message. An attacker can intercept the encrypted traffic of the handshake between another (legitimate) user and the access point and need only resend the third handshake to reset the WPA2 encryption key. This is the "nonce reuse" described in the title of Vanhoef's paper. Every time the key is reset the data sent over the network will be encrypted with the same key, allowing traffic to be gradually decrypted until the entire key is decrypted and the network no longer secure.[5] The attack is particularly devastating against wpa_supplicant, the open-source Wi-Fi client typically used on Linux operating systems. This implementation, which is also found in many Android phones, allows the attacker to simply install an all-zero encryption key with no need at all for the real key.[5]

Defense

Unfortunately, without a patched implementation of WPA2, there is no way to defend against a KRACK attack. Fortunately, many vendors have released patches that completely fix the issue. Mathy Vanhoef, the researcher who discovered the vulnerability, provides scripts to test your machine for the vulnerability on the website he created to provide information about the KRACK attack.

Disclosure as it Pertains to KRACK

I said earlier that most people would consider Vanhoef's disclosure "responsible." Let us now define what responsible disclosure entails. As defined by Christey & Wysopal, responsible disclosure aims to:

[4]IEEE802.11i-2004.

[5]Mathy Vanhoef, *Key Reinstallation Attacks Breaking WPA2 by forcing nonce reuse.*

1. Ensure that vulnerabilities can be identified and eliminated effectively and efficiently for all parties.
2. Minimize the risk to customers from vulnerabilities that could allow damage to their systems.
3. Provide customers with sufficient information for them to evaluate the level of security in vendors' products.
4. Provide the security community with the information necessary to develop tools and methods for identifying, managing, and reducing the risks of vulnerabilities in information technology.
5. Minimize the amount of time and resources required to manage vulnerability information.
6. Facilitate long-term research and development of techniques, products, and processes for avoiding or mitigating vulnerabilities.[6]

To these ends Christey & Wysopal also outline the responsibilities of both Vendors and Reporters, which the KRACK researchers have followed perfectly.[6] The Reporters initially contacted Vendors, alerting them to the vulnerability[7], so that they might protect their Customers. It is not known if the Vendors in this scenario also fulfilled their responsibilities, but it is interesting to note that no patches were released until after the exploit was fully and publicly disclosed.

KRACK is somewhat unique in that it is a product of academic research, but is otherwise unremarkable in terms of its disclosure. It was disclosed in a "responsible" manner that has spawned little controversy. So what can we learn from this? First of all, it is worth mentioning that despite its severity, the overall impact of KRACK has been low. There have been no documented, high-profile cases of serious data being compromised by a KRACK attack even though some Vendors took almost a whole month to roll out patches.

KRACK was disclosed "responsibly," and this would usually be cited as a reason for the minimal impact of the vulnerability, but the timeline of disclosure doesn't support this hypothesis. Vendors were notified months before the paper was published, but in many cases

[6]Christey & Wysopal
[7]Cimpanu

patches were not released until a month after the disclosure date, and some products still have yet to be patched. The best response to the bug only occurred after it was fully disclosed to the public. Bruce Schneier provides some insight into this phenomenon while arguing against non-disclosure: "To a software company, vulnerabilities are largely an externality. That is, they affect you – the user – much more than thy affect it."[8] He argues that to most Vendors, vulnerabilities are more of a public relations problem than a software problem, and that Vendors usually rely on secrecy for security rather than actually fixing their code. It is usually only after an exploit is made public knowledge that Vendors will be motivated to fix their broken software to avoid bad press.[8] This is objectively bad policy. If an ethical hacker can discover a vulnerability, so can an unethical one, and secrecy does nothing to protect Customers. However, the Vendors involved in the disclosure of KRACK seem to have precisely followed this predictive model. Though some of them were notified months before the Reporters research was disclosed to the public, no fixes for KRACK attacks were released until after public disclosure, and some Vendors still have not fixed the problem months after disclosure.

Essentially, the "responsible" aspects of KRACK's disclosure did nothing to mitigate its effect and its impact was minimal after full disclosure despite it being a quite severe vulnerability, thus discounting the argument that disclosing vulnerabilities makes Customers more at risk. The problem is that a Reporter can act perfectly responsibly but the whole process can be ruined by Vendors who cannot be held accountable. With these facts taken into account, the only truly responsible disclosure that satisfies Christey and Wysopal's goals is full disclosure.

Conclusion

KRACK was easily the most significant vulnerability discovered in 2017. Affecting nearly every internet user and limited only by being in range of a wireless access point, it had the

[8]Bruce Schneier, *Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'.*

farthest-reaching scope of any such bug in recent memory. However, it has not had nearly as large of an impact as its potential implies. A significant mitigating factor in this and all software exploits is the manner in which the Reporter chooses to disclose the vulnerability, specifically the full disclosure aspect. KRACK was by all means disclosed in a "responsible" manner, but no response was enacted until the public became fully aware of the problem. If vendors were alerted early, they should have fixed it. They didn't, so the "responsible" response of the KRACK reporters was useless. We only saw patches after full disclosure, and this response is not at all out of the ordinary. Until Vendors can truly be held accountable for securing their software, the only truly responsible and ethical way to disclose vulnerabilities is fully to the public.

References

Christey, Steve & Chris Wysopal. *Responsible Vulnerability Disclosure Process*. https://tools.ietf.org/html/draft-christey-wysopal-vuln-disclosure-00.

Cimpanu, Catalin. *New KRACK Attack Breaks WPA2 WiFi Protocol*. https://www.bleepingcomputer.com/news/security/new-krack-attack-breaks-wpa2-wifi-protocol/.

IEEE Standards. *IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements*. http://standards.ieee.org/getieee802/download/802.11i-2004.pdf.

Schneier, Bruce. *Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'*. https://www.schneier.com/essays/archives/2007/01/schneier_full_disclo.html.

Shepherd, Stephen A. *Vulnerability Disclosure: How do we define Responsible Disclosure*. https://www.sans.org/reading-room/whitepapers/threats/define-responsible-disclosure-932.

Vanhoef, Mathy. *Key Reinstallation Attacks Breaking WPA2 by forcing nonce reuse*. https://www.krackattacks.com/.