On the *Krack Attack*: Reproducing Vulnerability and a Software-Defined Mitigation Approach

Ramon dos Reis Fontes[1][2], Christian Esteve Rothenberg[1]
[1] Federal Institute of Education, Science and Technology of Bahia (IFBA), Salvador, Bahia, Brazil
[2] University of Campinas (UNICAMP), Campinas, Sao Paulo, Brazil
Email: ramonrf,chesteve@dca.fee.unicamp.br

***Abstract*—Among the claimed features of Software-Defined Wireless Networking (SDWN) is the ability to address network security attack vectors by provisioning fine-grain flow rules in addition to smart, reactive methods combining security monitoring, analysis and response systems. On the automation foreground, SDWN proposals include to automate security services by diverting specific network flows to special enforcement points or security services after anomaly detection. This is the context where our research is inserted: more concretely, we developed an application on top of SDN controller which is capable of protecting the network infrastructure against the "Krack Attack", one of the most popular and recent security threats targeting WPA2/WiFi vulnerabilities. Using the Mininet-WiFi emulation environment, we are able to reproduce the vulnerability and evaluate our proof-of-concept implementation with regard to the proactive detection of the vulnerability and a strawman mitigation approach that may inspire further innovative ideas around SDWN security.**

I. Introduction

Mobile devices are an indispensable part of our daily lives. More and more personal information and work related data are processed on mobile systems connected to wireless networks. The development and advancement of secure mobile operating systems, applications, devices and cellular networks is considered a critical aspects in order to keep up with the ever-growing mobile devices usage. When connecting to a Wi-Fi hotspot, one should assume that it is an untrusted network environment and that there are vulnerabilities that could compromise your wireless experience. When connected to an unsecured Wi-Fi network, for instance, data eavesdropping is a well-known risk.

Going deeper into this subject, the recently unveiled *Krack* vulnerability (also known as "*Krack attack*") [1] has been recognized as a potential security threat against modern encryption techniques that have been used to secure Wi-Fi networks for the last 15 years: WPA2 (*Wi-Fi Protected Access 2*). Publicly available information on the *Krack attack* include those related to the attack itself and how big companies are trying to mitigate the risk factors; in general, by creating security patches for their systems. However, there is no guarantee that all devices will be patched and in fact become immune to such attack from any network attached point.

Given this scenario, we propose Software Defined Wireless Networking (SDWN) [3] to protect the network infrastructure against vulnerabilities such as the *Krack attack*. In the same spirit of SDN [4], SDWN aims at providing programmatic centralized control (by means of so-called *controllers*) of the network outside the wireless access points (APs) which enforce the received instructions (policy decisions) and remain responsible for the transmission and reception of the traffic over the wireless link.

We expect that our work could also inspire other researchers and network administrators to comprehend how the vulnerability on 802.11r Fast-BSS Transition (FT) can be tested for pursuing further innovative ideas by adapting our approach to other security threats of mobile networks. We use Mininet-WiFi [2] as the emulation platform to reproduce the vulnerability and evaluate the proposed mitigation method leveraging its realistic and rich experimentation features.

II. A SDWN Approach to Krack Attack

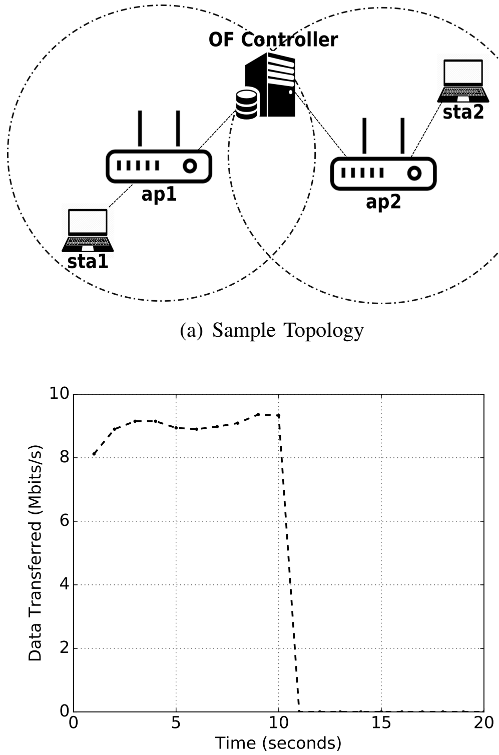
The main security vulnerability approached by the author of *Krack attack* targets 802.11r Fast-BSS Transition (FT), which affects access points. Fast Basic Service Set (`BSS`) Transition (`FT`) reduces the time needed for a device to transition to an AP that supports IEEE 802.11r of the same protected network (i.e. of the same *Basic Service Set*). The attack over the IEEE 802.11r FT handshake is comprehensively covered in the Krack attack's paper [1].

Our first step was to effectively reproduce the *Krack attack* in Mininet-WiFi. In support of SDWN research, Mininet-WiFi [2] offers a software-centric wireless network emulation that allows completely abandoning wireless network hardware as it packages the most common wireless facilities/tools, such as, *wpa_supplicant*, *hostapd*, *iw* (*iwconfig*), and others.

Within 48 hours after the attack went public (16-Oct-2017), we were able to reproduce the attack and publish a video[1] exploring the Key Reinstall in FT Handshake (802.11r) pointing to the whether an implementation is vulnerable to attacks or not.

Our second step was to design and implement a control plane solution to protect the network against the attack. We extended the Ryu controller[2] code in the reproducibility demo in order to react when the vulnerability is detected. Our experience is a first-hand showcase of the researcher-friendly and feature rich emulation capabilities of Mininet-WiFi in the context of SDWN research.

[1] https://www.youtube.com/watch?v=aA4notyZph0
[2] https://github.com/osrg/ryu



(a) Sample Topology

(b) Data Transferred between sta1 and sta2

Fig. 1. Topology and Data Transferred

A. Proof-of-Concept Experiment

We present now a Proof-of-Concept where the OpenFlow Controller can detect whether the access point attached to it is vulnerable and consequently take a mitigation decision. As shown in Fig.1(a), the topology consists on two access points, two stations, and one OpenFlow (OF) controller. Stations sta1 and sta2 are associated with ap1 and ap2, consequently. Both ap1 and ap2 are connected to the OF controller, which is equipped with a Wi-Fi interface working in monitor mode. This interface is responsible for capturing beacons that will contain the message responsible for detecting whether the access point is vulnerable to key re-installation attacks.

After loading the application we developed using the Ryu controller, network traffic was generated between the OF controller and sta2 in order to receive encrypted data frames. Next, we force the association of the OF controller with ap2 to trigger the FT handshake which is only performed when a station roams from one AP to another. Once the vulnerability is detected on ap2, the OF Controller sends a command to turn ap2 off. Multiple methods could be considered in addition to (*i*) switching off the AP; such as (*ii*) isolating the AP dataplane; (*iii*) warning the user / stations (e.g. HTTP/DNS redirection); among others. Noteworthy, the script we use in this work is not an attack script because the nodes require network credentials in order to test whether ap2 is affected by the attack.

As today, the current version of wpa_supplicant[3] (2.6) and that is also supported by Mininet-WiFi does not include the patch that fixing the issue behind the vulnerability. However, as expected, code updates can be observed on the hostap source code repository[4] so that the next version of wpa_supplicant (and also Mininet-WiFi) will certainly include the security patch any time soon.

B. Results Discussion and Future Work

The results presented in Figure 1(b) show that our SDN control application (started at *t=10sec*) can protect the network by first detecting the *Krack* vulnerability and then isolating the AP from the rest of the network infrastructure. Future work should consider other 3 attacks described in the *Krack's* paper that directly affect clients. The SDWN should reach the edge, where clients could be notified about the vulnerability. Source code, data, configuration, and instructions enabling experiment reproducibility are publicly available.[5]

III. Conclusions

This work presents a practical SDWN security approach that extends a SDN controller in order to receive and process beacons generated by wireless nodes. We exploit the knowledge on a recent vulnerability around the 802.11r Fast-BSS Transition (FT) and are able to run all PoC experiments on the Mininet-WiFi emulation platform without requiring physical access points implementing IEEE 802.11r protocol (note that hybrid physical and virtual experiments are also possible [5]). Basically, no additional effort was required to reproduce the *Krack attack* approach against the 802.11r Fast-BSS Transition (FT) and no changes were required to tools like *hostapd* and *wpa_supplicant* as well as wireless utilities like *iw/iwconfig*. Altogether, the experience with this lightning effort on a security use case shows the potential of SDWN and the value of Mininet-WiFi as an experimental platform allowing the fast prototyping evaluation of new designs in support of academic and industrial research, new networking technologies, troubleshooting, change planning of deployed networks, and so on.

References

[1] Mathy Vanhoef and Frank Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. CCS17, October 30 November 3, 2017, Dallas, TX, USA. https://doi.org/10.1145/3133956.3134027

[2] Fontes, Ramon R., et al. ”Mininet-WiFi: Emulating software-defined wireless networks.” Network and Service Management (CNSM), 2015 11th International Conference on. IEEE, 2015.

[3] S.Costanzo, L.Galluccio, G.Morabito, and S.Palazzo. Software Defined Wireless Networks: Unbridling SDNs. Proc. of EWSDN 2012. Darmstadt, Germany, October 2012.

[4] D. Kreutz, F. M. V. Ramos, P. E. P. Verissimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig Software-defined networking: A comprehensive survey, Proc. IEEE, vol. 103, no. 1, 2015.

[5] Fontes, Ramon R., et al. ”Mininet-WiFi: A Platform for Hybrid Physical-Virtual Software-Defined Wireless Networking Research.” SIGCOMM Demo Session. ACM, 2016.

[3] https://w1.fi/wpa_supplicant/

[4] https://w1.fi/cgit/hostap/commit/?id=a00e946c1c9a1f9cc65c72900d2a444ceb1f872e

[5] https://github.com/ramonfontes/reproducible-research/tree/master/mininet-wifi/krack-2017