Demonstração Experimental de Vulnerabilidades em Dispositivos Internet das Coisas

Raphael L. C. Nascimento, Ryan M. S. Leal e Iguatemi E. Fonseca

[1]Centro de Informática – Universidade Federal da Paraiba (UFPB)
Caixa Postal 15.064 – 58051-900 – João Pessoa – PB – Brazil

raphaelleite720@gmail.com, ryanleal@cc.ci.ufpb.br, iguatemi@ci.ufpb.br

Abstract. *This paper presents the BLE technology and protocols, as well, vulnerabilities and attacks performed in BLE devices. Attacks were performed in two BLE devices to demonstrate the use of technology sweyntooth and vulnerabilities of BLE devices. In these devices, sweyntooth was used, which in turn, could capture between 12 to 18 different vulnerabilities. This security testing exposed flaws, ranging from the mildest to the most serious.*

Resumo. *Este artigo apresenta a tecnologia e protocolos BLE, bem como vulnerabilidades e ataques realizados em dispositivos BLE. Ataques foram realizados em dois dispositivos BLE para demonstrar o uso da tecnologia sweyntooth e vulnerabilidades dos dispositivos BLE. Nesses dispositivos foi utilizado o sweyntooth, que por sua vez, conseguiu capturar entre 12 a 18 vulnerabilidades diferentes. Esse teste de segurança expôs falhas, desde as mais leves até as mais graves.*

1. Introdução

A Internet das Coisas (IoT - *Internet of Things*) possibilitou a automatização de processos por meio da comunicação de dispositivos de pequeno porte e consequentemente baixo poder energético e computacional. Por essa característica, tornou-se necessário o uso de uma solução de conectividade que fosse focada no desempenho energético [SILVA 2020]. A partir dessa necessidade surgiu o *Bluetooth Low Energy* (BLE) ou *Bluetooth Smart*, como também é chamado, um protocolo que está sendo amplamente utilizado nos principais dispositivos do mercado. Em 2019, existiam cerca de 14,2 bilhões de dispositivos IoT vinculados, com uma expectativa de que até 2021 tenham aumentado para 25 bilhões [Barua et al. 2022]. As limitações geradas pela bateria e o tamanho de suas peças restringem as ações que estes dispositivos podem realizar, reduzindo o número de opções disponíveis para se defender de possíveis ataques [Ryan 2013].

Com o objetivo de auxiliar no desenvolvimento seguro desse tipo de aplicações, a segurança BLE se tornou alvo de pesquisas [Kwon et al. 2016] e demonstrações experimentais de ataques e ferramentas em convenções e eventos [Cauquil]. Assim, visando testar os dispositivos BLE do mercado, foi criado o Sweyntooth [Matheus E. Garbelini; Chundong Wang and A*Star 2020], uma ferramenta que expõe falhas em implementações específicas de Kits de Desenvolvimento de Software (*SDKs - Software Development Kits*) BLE de seis principais fornecedores de Sistemas em um Chip (*SoC - System-On-a-Chip*) do mercado. Sendo composta de 18 vulnerabilidade,

essa ferramenta permite que atacantes consigam causar travamentos, deadlocks e estouros de buffer ou em alguns casos ignorar algumas medidas de segurança dos dispositivos. Esses produtos potencialmente afetados pelo Sweyntooth atuam em diferentes áreas como Smart-Homes [Prakash et al. 2017], logística [Faragher and Harle 2015] e saúde [Zegeye 2015], transportando dados sensíveis de seus usuários. A segurança dos dispositivos BLE é de suma importância para que a adoção das aplicações IoT seja feita sem comprometer a privacidade e até mesmo a saúde dos usuários, assim, esse trabalho visa demonstrar experimentalmente falhas de segurança em dispositivos reais do mercado usando o Sweyntooth.

O artigo está dividido em seção 2, onde são analisados os problemas gerais de segurança, bem como as principais vulnerabilidades e técnicas de invasão usadas em dispositivos BLE, seção 3, na qual são abordados os cenários e tecnologias utilizadas, além dos resultados obtidos nos experimentos realizados em dispositivos do mercado e por último a seção 4, onde tendo em vista os resultados, são feitas as conclusões.

2. Vulnerabilidades em dispositivos BLE

A maioria dos problemas de segurança dos dispositivos BLE decorrem de decisões de implementação focadas em desempenho e não na segurança dos produtos. Um dos exemplos mais conhecidos vem do uso do método de pareamento *Just Work* [dos Santos 2017]. Neste caso, o dispositivo não tem uma interface de exibição de dados, ou seja, ele só é capaz de se comunicar via Bluetooth, então sua senha de conexão é um valor estático: 000000, tornando-se trivial conectar-se a esses dispositivos. Consequentemente, sendo mais suscetível a diversos tipos de ataques. Outro exemplo é o *Injection-Free* [Aellison C. T. Santos and Fonseca 2019], que explora a forma como os desenvolvedores tratam a lista de vínculo dos dispositivos quando cheia, podendo levar os dispositivos até mesmo a negação de serviço, simplesmente pelo fato de passar do limite esperado de vínculos. Nas seções a seguir são exploradas as vulnerabilidades e técnicas de invasão do SweynTooth em dispositivos BLE.

De acordo com o comportamento gerado nos dispositivos alvos, as vulnerabilidades Sweyntooth são classificadas em Crash, Deadlock e Security Bypass.

As vulnerabilidades do tipo Crash acontecem principalmente devido ao comportamento inesperado do código ou uma corrupção de memória causada por um estouro de buffer. Em geral, os dispositivos reiniciam mas a situação pode ser agravada dependendo de como o mecanismo de tratamento de falhas foi implementado, impedindo o dispositivo de reiniciar e passando a entrar no estado de deadlock.

As vulnerabilidades de Deadlocks fazem com que o usuário fique preso em algum ponto do código devido a uma sincronização inapropriada entre o firmware do SDK do fornecedor e o código do usuário, então para retornar ao funcionamento, o usuário precisar reiniciar o dispositivo manualmente, visto que os mecanismos de tratamento de falhas estão indisponíveis.

E por fim as vulnerabilidades mais críticas são as de *Security Bypass*. Nesse caso, o atacante consegue gerar desvios de segurança nos modos de pareamento seguro BLE, permitindo que ele leia e escreva nas funções do dispositivo, atividade que só deveria ser feita por usuários autorizados.

2.1. Tecnicas de invasão

Dentre as técnicas de invasão Sweyntooth, destacam-se 3 ataques utilizados nas demonstrações experimentais.

O primeiro ataque utilizado foi o CC2540 Connection, nesse caso vai ser enviada a solicitação de conexão com seu tamanho truncado para 26 bytes, então, o dispositivo alvo aceita o pacote truncado e assume 0 para os parâmetros de conexão ausentes, gerando comportamentos inesperados no aparelho.

O segundo ataque utilizado foi o Knob Tester BLE, no qual é verificado se o periférico aceita que o tamanho de chave seja reduzido para 7 bytes. Alguns dispositivos podem ser programados para aceitar as chaves que estejam dentro do tamanho de chave BLE padronizado (7-16 Bytes), causando problemas no código da aplicação. Alguns aplicativos mais seguros rejeitam estritamente qualquer tamanho de chave inferior a 16 bytes, impedindo o ataque.

O terceiro ataque utilizado foi o *Link Layer Length Overflow*, essa vulnerabilidade permite que o atacante, que atua como dispositivo central, gere um estouro de buffer ao enviar um pacote com o campo de comprimento da camada de Link modificado, fazendo com que seja preenchido com mais bytes do que são esperados para esse tipo, assim, ultrapassando os limites esperados na memória do dispositivo, gerando travamentos no alvo.

3. Experimentos e Resultados

Foram realizados testes usando o Sweyntooth em dispositivos BLE do mercado. Os ataques foram feitos em um notebook Acer Aspire A514-53G com 8GB de RAM, processador Intel Core i5-1035G1 de 8 núcleos e um distribuição linux Ubuntu 22.04 LTS. Os detalhes e resultados da execução são explorados nas seções abaixo.

3.1. Cenários de testes e tecnologias

Para executar o SweynTooth, foi utilizado o nRF52840 Dongle da Nordic Semiconductor no envio e recebimento de pacotes da camada de link bruto para o periférico testado, o Dongle pode ser visto na Figura 1. Foi utilizado como um dos alvos dos ataques uma pulseira inteligente disponível no mercado, como exibida na Figura 1, a qual utiliza a versão 4.0 do BLE e oferece os serviços de monitoramento cardíaco, qualidade de sono, alerta de medicamentos e outros. O outro alvo dos ataques foi a cinta medidora de batimentos vista na Figura 1, que utiliza também a Versão 4.0 do BLE e funciona junto a um aplicativo de monitoramento de atividades físicas instalado no smartphone, nos testes foi usado o aplicativo *Map My Run* da Under Armour em conjunto com a cinta.

3.2. Resultados

Com a execução do ataque CC2540 Connection na pulseira inteligente, houve o truncamento da solicitação de conexão e nesse caso o dispositivo aceitou o pacote truncado, assim, resultando no crash do dispositivo, o resultado da execução pode ser visto na Figura 2. O mesmo aconteceu com a cinta medidora de batimentos cardíacos, nesse caso, gerando uma má formação da conexão e levando ao crash do dispositivo, tornando-se necessário reiniciar a conexão, o resultado pode ser visto na Figura 2.

Figura 1. nRF52840 Dongle, Pulseira Inteligente e Cinta Medidora de Batimentos

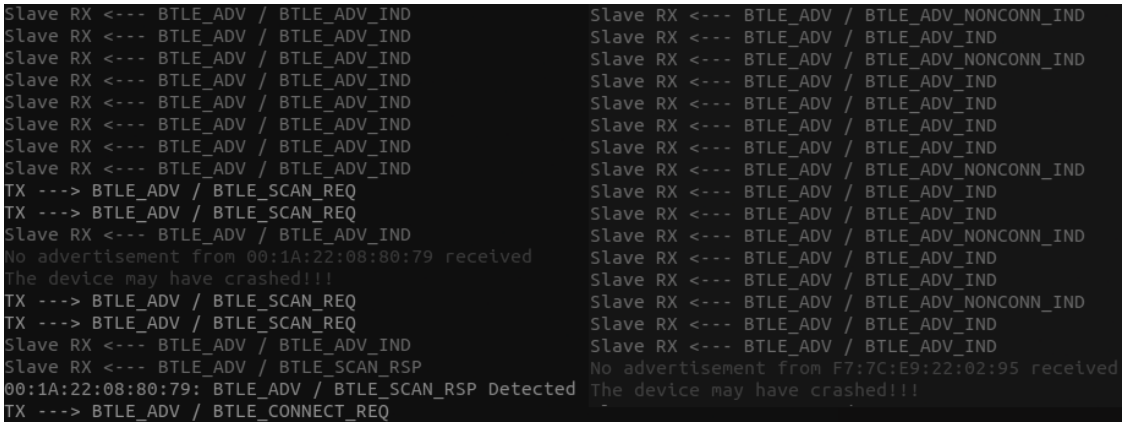

Figura 2. Ataque CC2540 Connection na pulseira inteligente e na cinta medidora de batimentos.

Ao executar o *Knob Tester* BLE na pusleira inteligente, as chaves de tamanho entre 7 e 16 bytes foram aceitas, como visto na Figura 3, conseguindo manter a conexão com o dispositivo, porém, quando é enviada uma chave maior que esse intervalo, a chave é rejeitada e a conexão é reiniciada. Já o mesmo não ocorre na cinta, como visto na Figura 3, ao enviar chaves com tamanho de 7-16 bytes, a conexão com o dispositivo foi perdida, tornando necessário o restabelecimento manual da conexão.

Com a execução do ataque *Link Layer Length Overflow*, tanto na pulseira inteligente quanto na cinta, há o envio dos pacotes com muito mais bytes do que o esperado para os periféricos, só que antes dos dispositivos travarem por completo e terem de ser reiniciados, eles reconhecem a má formação dos pacotes e desfazem a conexão, esperando receber novos pacotes, como visto nas Figuras 4 e 5.

4. Conclusão

Foi possível observar que o foco em implementações energeticamente eficientes promove implementações com falhas de segurança, muitas vezes simples, porém de grande impacto. Como observável nas vulnerabilidades SweynTooth, que apresentam modificações simples em pacotes e que geram consequências como travamentos e deadlocks, que podem ser extremamente prejudiciais aos usuários, visto que envolvem dados sensíveis usados em aplicações como na área da saúde. Por se tratarem de dispositivos utilizados

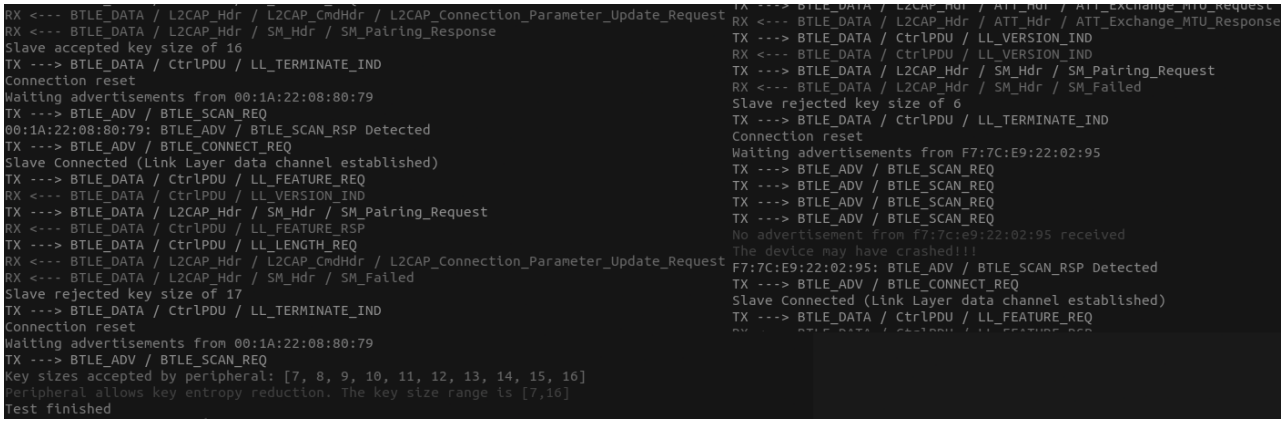

Figura 3. Ataque Knob Tester BLE na pulseira inteligente e na Cinta medidora de Batimentos.

Figura 4. Ataque Link Layer Length Overflow na pulseira inteligente.

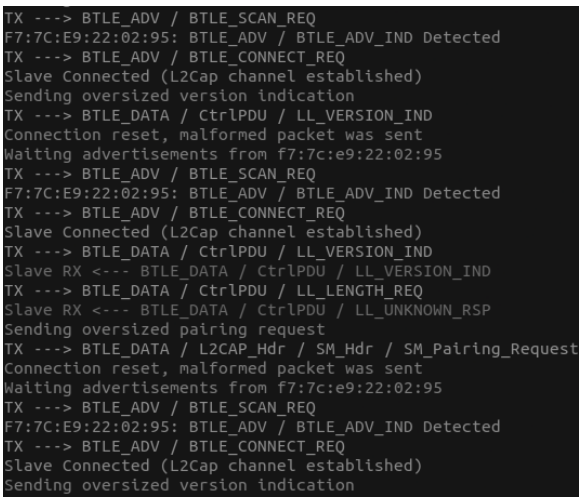

Figura 5. Ataque Link Layer Length Overflow na cinta medidora de batimentos.

atualmente no mercado, os resultados dos experimentos mostram o perigo e a criticidade das vulnerabilidades SweynTooth afetando potencialmente algumas centenas de produtos

IoT da atualidade. Evidenciando também falhas concretas no processo de certificação da BLE stack. Portanto torna-se evidente a necessidade de promover testes de segurança nos dispositivos BLE, assim, impedindo de que os dispositivos IoT sejam explorados por atacantes para fins maliciosos.

Referências

Aellison C. T. Santos, José L Soares Filho, S. S. V. N. and Fonseca, I. E. (2019). *BLE Injection-free Attack: a Novel Attack on Bluetooth Low Energy Devices*. Elsevier Journal of Ambient Intelligence and Humanized Computing.

Barua, A., Al Alamin, M. A., Hossain, M. S., and Hossain, E. (2022). Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey. *IEEE Open Journal of the Communications Society*.

Cauquil, D. Btlejuice: the bluetooth smart mitm framework, def con 24 internet of things village, 2016.

dos Santos, A. C. T. (2017). *Ameaças à Internet das Coisas: Explorando Vulnerabilidades do Bluetooth Low Energy e Suas Defesas*. Universidade Federal da Paraíba.

Faragher, R. and Harle, R. (2015). Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428.

Kwon, G., Kim, J., Noh, J., and Cho, S. (2016). Bluetooth low energy security vulnerability and improvement method. In *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 1–4. IEEE.

Matheus E. Garbelini; Chundong Wang, S. C. S. S. and A*Star, E. K. (2020). *SweynTooth: Unleashing Mayhem over Bluetooth Low Energy*. usenix the advanced computing systems association.

Prakash, Y., Biradar, V., Vincent, S., Martin, M., and Jadhav, A. (2017). Smart bluetooth low energy security system. In *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 2141–2146. IEEE.

Ryan, M. (2013). *Bluetooth: With Low Energy comes Low Security*. usenix the advanced computing systems association.

SILVA, Ávilla I. S.; SANTOS, A. C. T. M. M. A. F. G. P. H. R. S. R. M. N. V. F. I. E. (2020). *Segurança em Aplicações de Internet das Coisas: Bluetooth Low Energy, casos de uso e vulnerabilidades*. Livro de minicursos SBRT 2020.

Zegeye, W. K. (2015). Exploiting bluetooth low energy pairing vulnerability in telemedicine. In *International Telemetering Conference Proceedings. International Foundation for Telemetering*.