Vulnerabilidade do sistema operacional Windows com inserção de malware por HID

Nicolas de Freitas Azevedo, Ricardo Araújo Pereira, Vinicius Galasse Semente, Jean Carlo Wagner

Escola de Engenharia, Arquitetura e Tecnologia – Departamento da Computação

Universidade Anhembi Morumbi (UAM)

***Resumo* — Ao conectar um HID (*Human Interface Device*), o computador vincula o dispositivo sem grandes validações, dessa forma, pode-se usar um *hardware* contendo um microcontrolador que após ser conectado ao computador, demonstra uma vulnerabilidade pouco conhecida, através da porta USB conectada, a qual é lida pelo sistema operacional, reconhecendo a placa como um teclado, demonstrando a falha do sistema em reconhecer a placa como um teclado de acesso, podendo afetar o próprio computador, bem como caixas eletrônicos, urnas eletrônicas entre outros.**

***Palavras-Chave* — HID. Microcontrolador. Computador. *Hardware*. Segurança.**

I. INTRODUÇÃO

Os computadores são tidos como um pilar importante da sociedade e muito se fala sobre segurança da informação, mas um tópico pouco explorado é a vulnerabilidade de computadores quando conectados dispositivos USB nos mesmos. Neste trabalho será utilizada uma placa com microcontrolador oculto dentro de um teclado usb, assim a pessoa que conectar o dispositivo na porta usb do *desktop/notebook,* não percebe a existência de outro aparelho conectado, uma vez que o teclado funcionará normalmente. Após a conexão física, o código desenvolvido em C++ que reproduz o *drive* de teclado, utilizará este recurso para operar comandos no computador, podendo assim explorar vários vetores de ataque, como: Inserção de *Malware*, *Keylogger*, *Reverse Shell* e entre outros.

E desta forma, é possível testar e validar diversos conjuntos de ataques utilizados nos computadores, especialmente, no Windows.

II. PROBLEMA DE PESQUISA E OBJETIVOS

Estudo sobre a vulnerabilidade do sistema operacional Windows ao utilizar-se um HID *(Human Interface Device*).

É bem difundido na internet o risco de ataques através de *e-mails* de *phishing*, *links* maliciosos, vírus em programas baixados em *sites* desconhecidos, porém, pouco falado sobre essa possibilidade, um simples dispositivo conectado à porta USB do computador. O objetivo é demonstrar que essa forma de ataque, usando um dispositivo USB malicioso escondido em um teclado, pode ser perigosa.

Para o usuário será um simples teclado, porém, ao conectar no computador o dispositivo irá injetar códigos maliciosos, que podem afetar o uso do sistema operacional, inserir *malwares*, danificar ou até mesmo roubar dados pessoais.

III. METODOLOGIA DE PESQUISA

A. Assunto abordado.

Como o tema de pesquisa é pouco explorado no meio acadêmico, decidiu-se por utilizar a metodologia exploratória, sendo assim, muitos métodos utilizados serão baseados em experiências dos membros do grupo e aprovação do orientador. Então foi decidido atuar nesta vasta gama de HID com ênfase em explorar as vulnerabilidades do Sistema Operacional Windows com *Malwares* inseridos por USB.

B. Tipos de Hardwares.

Existem diversos *Hardwares* HID que podem ser adicionados facilmente em um computador e fazer toda a ação de executar códigos na inicialização pré-OS (Operational System), com o objetivo de acessar e interagir com *prompts pré-boot*, *menus* da BIOS (Basic Input/Output System) e entre outros. Existem diversos dispositivos que podem ser usados com a

finalidade de atingir um sistema, como o *bad USB*, *rubber ducky*, *bash bunny*, e entre outros.

C. Mecanismos de proteção e Integridade Windows.

Para que o Sistema Operacional opere de maneira correta, foram desenvolvidos mecanismos de defesa e manutenção do sistema, como o *Windows update*, que permite baixar novas atualizações dos *drivers* e do sistema; o *TrustedInstaller*, que é responsável pelo gerenciamento das atualizações do sistema; o UAC (*User Account Control*), que restringe alterações e acessos no sistema sem antes notificar e pedir permissão ao usuário para ter o acesso liberado. Além destes, ainda possui *firewall* e antivírus, que conseguem proteger o sistema contra alguns ataques de *malwares*, como os vírus, *spywares* (*Keyloggers*), *ransomwares*, cavalos de troia, alguns tipos de ataque de *phishing* e *softwares* maliciosos (*exploits*).

Uma das maneiras do *Windows defender* ou *Microsoft Defender* proteger o sistema é através da detecção de ameaças baseadas em assinaturas, quando um *software* malicioso ao ser muito usado pelas pessoas acaba sendo detectado pelo antivírus, e entrando em uma lista de possíveis ameaças, sendo assim, impedido automaticamente de executar.

Detecções de ameaças com base em assinatura examinam o conteúdo de um arquivo de um computador infectado e compara com assinaturas de códigos com ameaças já conhecidas como *malwares* e *softwares* mal-intencionados(*exploits*).

No entanto uma forma de burlar esse sistema de detecção de assinatura é alterar o conteúdo do *exploit* e o local de hospedagem, assim, gerando outra assinatura. Ao alterar coisas simples, como novas funções, excluir ou incluir mais comentários no código, é possível gerar outras assinaturas, permitindo a execução do *exploit*.

D. Sistemas Operacionais.

O sistema operacional é um *software*, ou um conjunto de *softwares*, que tem como papel gerenciar e administrar todos os recursos presentes em um sistema. Isso envolve desde os componentes do *hardware*, sistemas de arquivos e programas terceiros, é o sistema operacional que faz essa conexão entre o usuário e o computador ou *notebook*.

Para que fique mais claro, quando o usuário aciona um botão para ligar o dispositivo, o sistema operacional realiza uma bateria de testes, e isso é essencial para que o computador se certifique de que tudo irá funcionar dentro do esperado, envolvendo desde componentes físicos até o *hardware*. Então somente após este processo feito é que o sistema operacional é iniciado.

No mercado existem diversos tipos de sistemas operacionais, não existe apenas um padrão, cada sistema operacional apresenta uma particularidade, benefício e propostas diferentes, como: Windows, Linux, Android, MacOs, iOS, Chrome OS, dentre outros.

E. Sobre o PowerShell.

O PowerShell é um ambiente de comando orientado a objetos interativos com recursos de linguagem de *script* que utiliza pequenos programas chamados cmdlets para simplificar a configuração, administração e gerenciamento de ambientes heterogêneos em tipologias autônomas e em rede, utilizando protocolos remotos baseados em padrões.

O PowerShell pode ser executado no Windows, Linux e macOS. Como uma linguagem de *script*, PowerShell é normalmente utilizado para automatizar o gerenciamento de sistemas, compilar, testar e implantar soluções, geralmente em ambientes de CI/CD (Continuous Integration/Continuous Delivery).

Um dos principais benefícios do PowerShell é de se conectar remotamente a outro sistema, ou seja, um administrador pode usar uma sessão remota para se conectar a um servidor que não está no mesmo local físico e executar comandos como se estivesse trabalhando diretamente neste servidor.

F. O que é um Malware.

Malware é um termo utilizado para classificar todo tipo de *software* ou código malicioso usado para causar algum dano, seja ele financeiro, interceptar dados, danificar sistemas instalados ou irritar usuários, que afeta tanto computadores, celulares e redes inteiras.

Hostil, geralmente assumindo o controle parcial das operações de um dispositivo, assim como a gripe

para os humanos, o *malware* interfere no funcionamento normal.

G. O que é Keylogger.

O *Keylogger* é um *script* malicioso que grava/registra as teclas pressionadas em um teclado, normalmente de maneira que o usuário nem saiba que suas ações estão sendo monitoradas. Um *Keylogger* pode também existir no *hardware*, que da mesma maneira que o programa, pode capturar tudo que está sendo digitado no teclado por meio de uma conexão USB.

H. O que é um Reverse Shell.

Para obter controle sobre um sistema comprometido, os invasores visam obter acesso ao *shell* do computador para execução de comandos. Com esses acessos, é possível tentar elevar privilégios para ter controle total do sistema operacional. A maioria dos sistemas estão atrás de *firewalls* e as conexões de *shell* remotas diretas são impossíveis, então um dos métodos usados para contornar isso é o *Reverse Shell*. Em um cenário típico de acesso ao sistema remoto, o usuário é o cliente e a máquina de destino é o servidor, ao utilizar o *reverse shell* os papéis são opostos, então a máquina destino inicia a conexão com o usuário, e o computador do usuário escuta as conexões de entrada em uma porta específica. A principal razão de uso frequente do *Reverse Shell* por invasores é a maneira como a maioria dos *firewalls* é configurada.

I. Sobre Exploits.

É a prova de que a vulnerabilidade existe, são um subconjunto de *malware*. Seu objetivo é explorar as vulnerabilidades do alvo e acessar o que antes não era permitido.

I.1 *Payload*: É o código malicioso que executa uma ação destrutiva no sistema alvo, faz parte do *exploit* (ou compilado independente). Com o *payload* é possível, por exemplo, criar um usuário, apagar arquivos, obter o controle da *Shell* do sistema, entre outros.

I.2 *Shellcod*: É um código malicioso que também faz parte do *exploit*, tem como objetivo injetar códigos no sistema alvo, normalmente acompanhado do *payload*.

IV. TECNOLOGIAS DO PROJETO

Para o desenvolvimento e pesquisa é utilizada a placa Digispark, um *HUB* USB e um teclado USB simples. O código foi escrito utilizando a linguagem C++ e para gravação do código, o ambiente de desenvolvimento integrado (IDE) Arduino.

Foi escolhido o sistema operacional Windows por ser o mais popular no Brasil, porém, a mesma abordagem utilizando apenas códigos C++ diferentes, poderia ser utilizada para atacar diferentes sistemas operacionais. O protótipo é inserido dentro do teclado juntamente com o Digispark e, para que o teclado continue funcionando perfeitamente, será utilizado o *HUB* USB, ou seja, visualmente será vendido apenas o teclado, porém com a placa Digispark acoplada no *HUB* USB, sendo inseridos em um computador, conectando ambos os dispositivos. A prova do conceito se dá através da conexão do protótipo USB a um computador, sem que o usuário perceba, sendo capaz de injetar um código em seu sistema operacional, como por exemplo, a abertura e uma porta de rede ou instalação de algum código via *CURL (Cliente URL)*.

A. Inserção e acessos.

A conexão da porta USB, é composta de um teclado *HUB* USB e uma placa Arduino Digispark acoplada ao teclado. Ao conectar no computador, inicia-se a instalação do *driver*, e uma janela *fake* "*Windows Update*" aparece para o usuário (esta janela não é obrigatória para o funcionamento, apenas uma estética melhor agregando ao trabalho), então o terminal faz o *download* de um arquivo *Shell* no GitHub. O *download* do código considera a limitação de memória, abordada no tópico "Limitações", de forma que se tenha mais versatilidade na alteração de código, uma vez que essa placa está acoplada internamente no teclado. GitHub do código: https://github.com/araujoricardo/Digispark

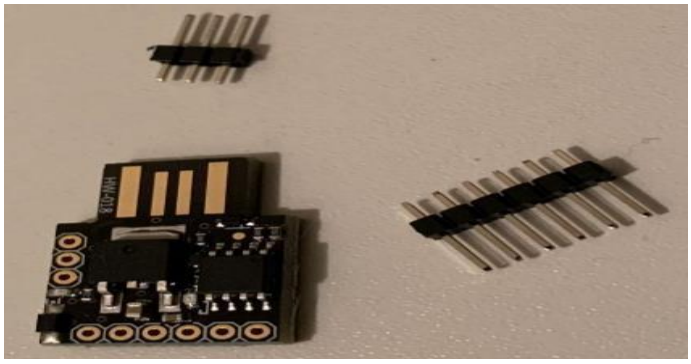

Fig. 1. Placa Mini Arduino Digispark Kickstarter Attiny85 USB.

Fonte: Autoria própria, 2022.

A placa Mini Arduino Digispark Kickstarter Attyni85 usb encontrada em diversos sites de informática e vendas, com diversos preços, e optada pelo custo desse produto que é de R$ 26,50, acessado em 05/04/2022.

B. Placa Digispark e Teclado USB.

A placa utiliza um microcontrolador Attiny85 tendo apenas 6 pinos para serem usados, conta também com um regulador de tensão para 5V, uma entrada USB e memória *flash* de 8kB e aproximadamente 6kB após instalação do *Bootloader*.

Na figura 2. abaixo temos a especificação da placa Digispark. O teclado Multilaser *HUB* USB utilizado para acoplar a placa Digispark com os *scripts* maliciosos inseridos no Sistema Operacional.

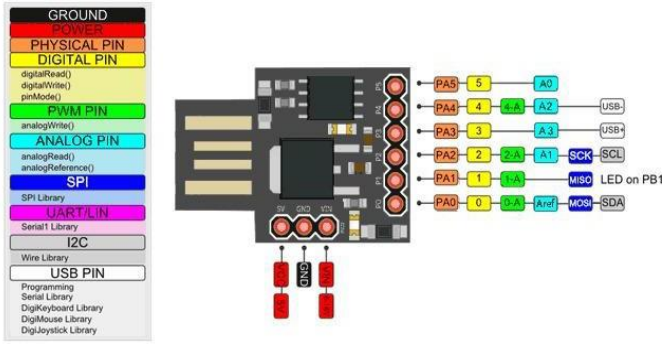

Fig. 2. Placa Mini Arduino Digispark Kickstarter Attiny85 Usb.

Fonte: Autoria própria, 2022.

Fig. 3. Teclado Multilaser *HUB* USB.

Fonte: Google imagens, 2022.

C. Fluxograma e sequência de ação.

Os *scripts* maliciosos mais conhecidos como *Keylogger* e *Reverse Shell*, o próprio terminal do Windows bloqueia, mesmo que o usuário seja o administrador da máquina, impedindo que seja feita alguma coisa nociva ao computador, porém, com comandos e sequências de ações da placa Digispark mais simples, é encontrada uma vulnerabilidade dentro do sistema operacional. Certas ações de eventos nocivos e *scripts* maliciosos injetados, ainda têm algumas falhas e vulnerabilidades.

Abaixo na Fig. 4 o fluxograma do processo:

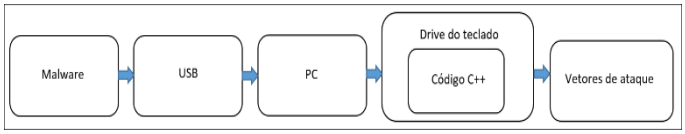

Fig. 4. Fluxograma e sequência de ação Digispark.

Fonte: Autoria própria, 2022.

D. Powershell bloqueando ataques conhecidos.

Ao executar um código de *script* malicioso *Keylogger* ou *Reverse Shell*, o Windows Defender embutido no Powershell identifica que se trata de um *script* malicioso e executa o bloqueio. Abaixo, na Fig. 5 podemos ver um caso em que foi identificado como um *script* malicioso, e o bloqueio foi feito pelo Windows defender.

Fig. 5. Powershell bloqueando *Keylogger*.

Fonte: Autoria Própria, 2022.

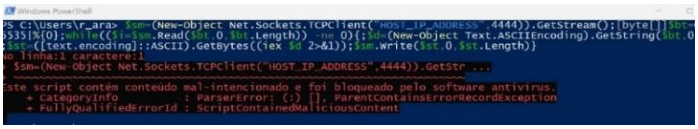

Fig. 6. Powershell bloqueando *Reverse Shell*.

Fonte: Autoria Própria, 2022.

E. Executando Script como Administrador.

O *script* é executado, e o usuário da máquina está com o acesso de Administrador e, como resultado do teste o Windows Defender fez o bloqueio automaticamente.

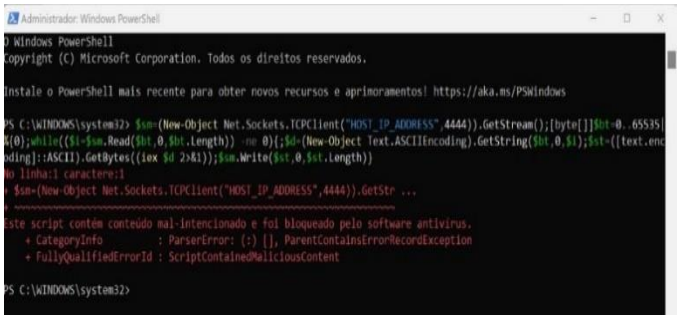

Fig. 7. Powershell com Admin Windows Defender.

Fonte: Autoria Própria, 2022.

F. Início de execução dos scripts.

Ao conectar o HID (*Human Interface Device*) na porta USB do computador, o Digispark inicia com um simples comando e executa um dos *scripts* inseridos na placa simulando uma atualização *fake* para o usuário.

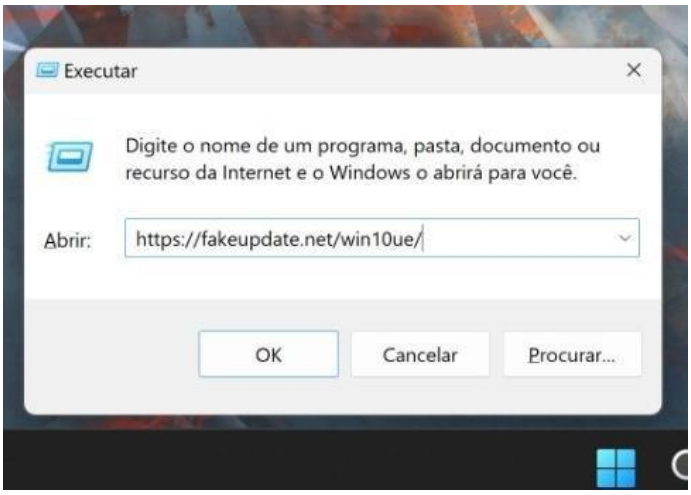

Fig. 8. Comando para abertura atualização *fake*.

Fonte: Autoria Própria, 2022.

G. Fake Update Iniciado.

É aberta a tela de *fake update* para que o usuário imagine que o Windows esteja sendo atualizado.

Abaixo na Fig. 9 podemos ver a tela simulando uma atualização *fake* sendo exibida para o usuário.

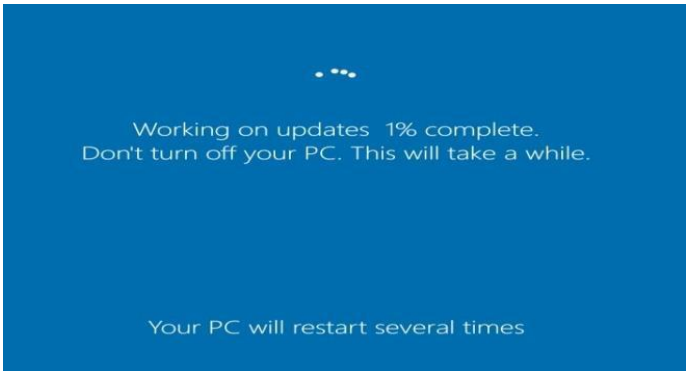

Fig. 9. *Fake Update* na tela do Computador.

Fonte: Autoria Própria, 2022.

H. Abrindo o PowerShell.

Ao manter a tela de atualização *fake* na tela do usuário, é aberto o PowerShell por meio de um comando no código inserido na placa, dando sequência ao *download* do código e execução dos *scripts* maliciosos na máquina.

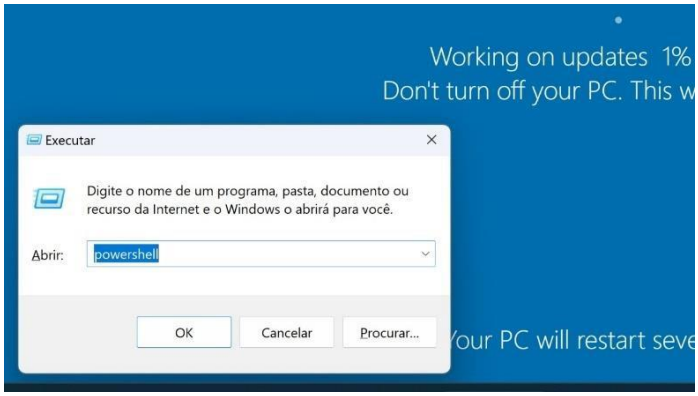

Fig. 10. Abrindo do Powershell.

Fonte: Autoria Própria, 2022.

I. Execução dos Scripts.

Com o Powershell aberto, os s*cripts* maliciosos, que estão armazenados no GitHub, foram executados. Após isso, é fechada a janela do Powershell e os *scripts* passam a ser todos executados em segundo plano, sem que o usuário perceba que algo está acontecendo com sua máquina. Desta forma, é verificado que o *Hardware/Software* não têm nenhuma operação e nenhum tipo de proteção contra esses ataques recebidos via HID.

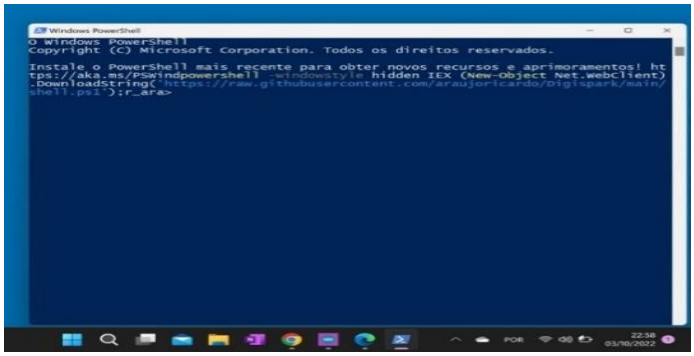

Fig. 11. PowerShell aberto.

Fonte: Autoria Própria, 2022.

J. Teste no Laboratório UAM.

Ao realizar os testes nos computadores da Universidade Anhembi Morumbi, constatou-se uma demora na execução do *script* por HID (*Human Interface Device*), o *Software* da Digispark executou rapidamente e não havia tempo suficiente para o *driver* de teclado ser instalado no computador, ou seja, ainda não estava plugado para injetar o comando. Para o teste nos computadores da UAM foi necessário aumentar o tempo até o início dos ataques no sistema, o que corresponde a 20 segundos, tendo em vista que as configurações de computador podem acarretar sintomas diferentes.

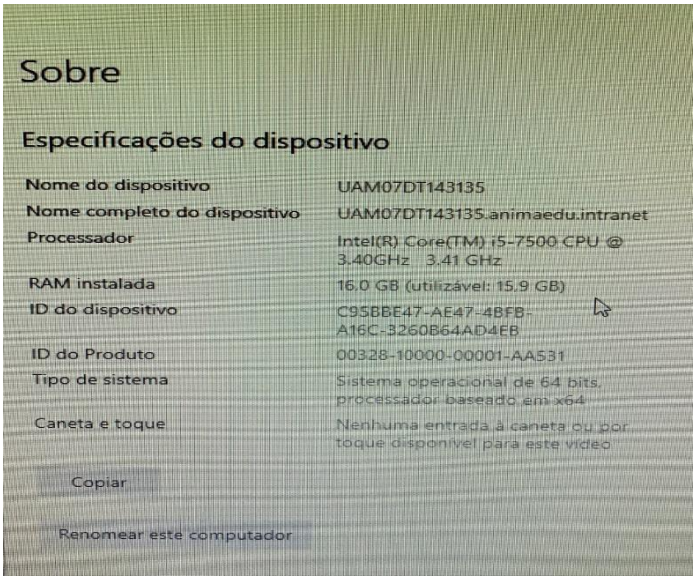

Fig. 12. Configurações computador UAM.

Fonte: Autoria Própria, 2022.

V. LIMITAÇÕES.

A. Memória do Digispark.

A memória da placa Digispark é de 8kB, porém o bootloader utiliza 2kB, sobrando apenas 6kB, sendo assim, não se pode colocar códigos extensos e complexos, tendo que baixar o código de um *link* GitHub, ou seja, uma vez que não se tem internet, há uma limitação de funcionamento da solução Digispark.

B. Layouts de teclados.

Existem diversos *layouts* para teclados sendo assim, os rearranjos das teclas são alterados para cada tipo, sejam alguns deles: ABNT, ABNT2, PT, espanhol, americano e entre outros. A placa Digispark criada para um tipo de *layout* de teclado não funcionariam em *layouts* diferentes, ou seja, se criado para um *layout* ABNT, não funcionaria num *layout* de teclado internacional.

C. Vetores de ataques.

Os vetores de ataques para cada sistema operacional são criados de formas diferentes, os ataques devem ser baseados com foco em apenas um sistema operacional, os códigos podem ser parecidos, mas com mudanças em alguns aspectos na lógica de programação, nas instalações de *drivers* no computador e nos componentes que são utilizados para que funcione da forma correta diante do ataque realizado.

CONCLUSÕES

O sistema operacional Windows possui uma falha de segurança quando se trata de dispositivos USB e/ou HID (*Human Interface Device*). Acredita-se que esta mesma falha exista em todos os sistemas operacionais, porém, como não testamos, não é possível afirmar. Apesar da Microsoft estar melhorando o Windows defender, o que pode ser constatado em nossos testes, é possível utilizar *exploits* do Powershell para atacar o computador no qual o dispositivo foi conectado, sendo assim, o bloqueio e parte da segurança deveriam ser em nível de sistema operacional no momento que é conectado o *hardware* malicioso, já que uma vez que estes têm acesso ao terminal, são viáveis vários vetores de ataques perigosos ao usuário.

Os objetivos foram atingidos, provando através de testes reais utilizando o Digispak via conexão USB HID, que é possivel inserir *malwares*, códigos e outros tipos de *exploits* para atacar o computador de qualquer usuário.

REFERÊNCIAS

[1] TANENBAUM, Andrew S. Organização Estruturada de Computadores. 6a ed. São Paulo. Pearson Prentice Hall, 2013.

[2] https://thehackernews.com/2021/07/hackers-turning-to-exotic-programming.html
Acessado em: 10/2022

[3] https://www.mcafee.com/blogs/enterprise/fileless-malware-execution-with-powershell-is-easier-than-you-may-realize/
Acessado em: 10/2022

[4] https://www.buscape.com.br/notebook/conteudo/o-que-e-sistema-operacional
Acessado em: 10/2022

[5] https://tecnoblog.net/responde/o-que-e-o-powershell-do-windows/
Acessado em: 10/2022

[6] https://br.malwarebytes.com/malware/
Acessado em: 10/2022

[7] https://canaltech.com.br/seguranca/O-que-e-keylogger/
Acessado em: 10/2022

[8] https://www.linkedin.com/pulse/elastic-siem-detectando-reverse-shell-em-sistemas-linux-souza-/?originalSubdomain=pt
Acessado em: 10/2022