Anything related to windows

security stuff, scripts, how tos and other things

- security scripts
  - uber basic check for kelogers.py
  - start-win-defender.bat
  - check for systemstats.py v1
  - check for pc's pids names and usage stats and send them to a file to search later .py v2
  - pid killer (ranges too)
  - block all .zip .rar .mov top level domains

- How to disable Start up Apps
  - Via Task Manager
  - Via Settings
  - Via System Configuration:
  - Via Registry Editor

- How to admin people Via Run
- change another users passwd without knowing it
- fix windows files
- every tool for windows in a single file :D
- download file to sshed windows server
- How to Remove 'Show More Options' From the Windows 11 Context Menu - Command Prompt
- firefox yt better audio scaling for when im studying
- uhhh websites to visit
- windows 10 exploerer in windows 10

- usefull windows hotkeys

security scripts

security scripts

uber basic check for kelogers.py

```
import psutil
import os
import sys

def find_suspicious_processes():
    suspicious_processes = []
    for process in psutil.process_iter(['pid', 'name', 'exe', 'cmdline']):
        try:
            if process.info['exe'] and process.info['cmdline']:
                exe_name = os.path.basename(process.info['exe']).lower()
                cmdline = ' '.join(process.info['cmdline']).lower()

                if 'keylogger' in exe_name or 'keylogger' in cmdline:
                    suspicious_processes.append(process)
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            pass
    return suspicious_processes

def main():
    suspicious_processes = find_suspicious_processes()

    if not suspicious_processes:
        print("No suspicious processes found.")
    else:
        print("Suspicious processes found:")
        for process in suspicious_processes:
            print(f"PID: {process.info['pid']} - Name: {process.info['name']} - Exe: {process.info['exe']}
- Cmdline: {' '.join(process.info['cmdline'])}")

if __name__ == '__main__':
    main()
```

security scripts

start-win-defender.bat

```
@echo off
echo Starting Windows Defender malware scan...
"%ProgramFiles%\Windows Defender\MpCmdRun.exe" -Scan -ScanType 1
echo Scan complete.
pause
```

security scripts

check for systemstats.py v1

```
import psutil
import datetime


def check_high_memory_usage(threshold=50):
    high_memory_usage_processes = []
    total_memory = psutil.virtual_memory().total
    for proc in psutil.process_iter(['pid', 'name', 'memory_info']):
        try:
            memory_percent = (proc.info['memory_info'].rss / total_memory) * 100
            if memory_percent > threshold:
                high_memory_usage_processes.append((proc, memory_percent))
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            pass

    return high_memory_usage_processes


def write_processes_to_file():
    now = datetime.datetime.now()
    file_name = f"processes_{now:%Y-%m-%d_%H-%M-%S}.txt"
    with open(file_name, 'w') as f:
        try:
            f.write(f"List of highest CPU usage processes on {now}:\n\n")
            for proc in sorted(psutil.process_iter(['pid', 'name', 'memory_percent'
, 'cpu_percent']), key=lambda p: p.info['cpu_percent'], reverse=True):
                try:
                    cpu_percent = proc.info['cpu_percent']
                    if cpu_percent > 0.0:
```

```
                        f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name'
]} - CPU%: {cpu_percent:.2f} - Memory%: {proc.info['memory_percent']:.2f}\n")

                        f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f
}%\n")

                        f.write(f"\tNetwork usage: {psutil.net_io_counters().
bytes_sent/1024:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}
KB received\n")

                except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.
ZombieProcess):

                    pass

        except:

            f.write("An error occurred while writing the file.\n")



        f.write("\n\n
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n")

        f.write(f"List of highest memory usage processes on {now}:\n\n")

        for proc, mem_percent in sorted(check_high_memory_usage(), key=lambda p: p[
1], reverse=True):

            f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name']}
 - Memory%: {mem_percent:.2f}\n")

            f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f}%\n")

            f.write(f"\tNetwork usage: {psutil.net_io_counters().bytes_sent/1024
:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}KB received\n")

        f.write("\n\n
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n")

        f.write(f"List of all running processes on {now}:\n\n")

        for proc in psutil.process_iter(['pid', 'name', 'memory_percent',
'cpu_percent']):

            try:

                cpu_percent = proc.info['cpu_percent']

                mem_percent = proc.info['memory_percent']

                if cpu_percent > 0.0:

                    f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name']}
 - CPU%: {cpu_percent:.2f} - Memory%: {mem_percent:.2f}\n")

                    f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f}%\n
")
```

```
                    f.write(f"\tNetwork usage: {psutil.net_io_counters().bytes_sent
/1024:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}KB received\n")

            except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess
):

                pass



def main():

    write_processes_to_file()



if __name__ == '__main__':

    print("checking")

    main()

    print("done")
```

security scripts

check for pc's pids names and usage stats and send them to a file to search later .py v2

```
import os
import psutil
import datetime

def check_high_memory_usage(threshold=50):
    high_memory_usage_processes = []
    total_memory = psutil.virtual_memory().total
    for proc in psutil.process_iter(['pid', 'name', 'memory_info']):
        try:
            memory_percent = (proc.info['memory_info'].rss / total_memory) * 100
            if memory_percent > threshold:
                high_memory_usage_processes.append((proc, memory_percent))
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            pass

    return high_memory_usage_processes

def write_processes_to_file():
    current_pid = os.getpid()
    now = datetime.datetime.now()
    file_name = f"processes_{now:%Y-%m-%d_%H-%M-%S}.txt"
    with open(file_name, 'w') as f:
        f.write(f"List of all running processes on {now}:\n\n")
```

```
        for proc in psutil.process_iter(['pid', 'name', 'memory_percent',
'cpu_percent']):

            try:

                if proc.info['pid'] != current_pid:  # Exclude the current script

                    cpu_percent = proc.info['cpu_percent']

                    mem_percent = proc.info['memory_percent']

                    f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name']}
 - CPU%: {cpu_percent:.2f} - Memory%: {mem_percent:.2f}\n")

                    f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f}%\n
")

                    f.write(f"\tNetwork usage: {psutil.net_io_counters().bytes_sent
/1024:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}KB received\n")

            except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess
):

                pass



        f.write("\n\n
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n")

        f.write(f"List of highest CPU usage processes on {now}:\n\n")

        for proc in sorted(psutil.process_iter(['pid', 'name', 'memory_percent',
'cpu_percent']), key=lambda p: p.info['cpu_percent'], reverse=True):

            try:

                if proc.info['pid'] != current_pid:  # Exclude the current script

                    cpu_percent = proc.info['cpu_percent']

                    if cpu_percent > 0.0:

                        f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name'
]} - CPU%: {cpu_percent:.2f} - Memory%: {proc.info['memory_percent']:.2f}\n")

                        f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f
}%\n")

                        f.write(f"\tNetwork usage: {psutil.net_io_counters().
bytes_sent/1024:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}
KB received\n")

            except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess
):

                pass
```

```
        f.write("\n\n
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=\n\n")

        f.write(f"List of highest memory usage processes on {now}:\n\n")

        for proc, mem_percent in sorted(check_high_memory_usage(), key=lambda p: p[
1], reverse=True):

            f.write(f"PID: {proc.info['pid']} - Name: {proc.info['name']}
 - Memory%: {mem_percent:.2f}\n")

            f.write(f"\tDisk usage: {psutil.disk_usage('/').percent:.2f}%\n")

            f.write(f"\tNetwork usage: {psutil.net_io_counters().bytes_sent/1024
:.2f}KB sent/{psutil.net_io_counters().bytes_recv/1024:.2f}KB received\n")



def main():

    write_processes_to_file()



if __name__ == '__main__':

    main()
```

security scripts

pid killer (ranges too)

```
import psutil



while True:

    pids = input(
"Type the PID(s) you want to kill, separated by commas, or specify a range with a da
)

    if pids.lower() == "exit" or pids.lower() == "stop":

        confirm = input("Are you sure you want to stop the script? (Y/N): ")

        if confirm.lower() in ["y", "yes"]:

            break

        else:

            continue

    if '-' in pids:

        start, end = pids.split('-')

        pids_list = [str(pid) for pid in range(int(start), int(end) + 1)]

        yes_all = input(
"Do you want to kill all processes in the range without confirmation? (Y/N): ")

        if yes_all.lower() in ['y', 'yes']:

            response = 'y'

        elif yes_all.lower() in ['n', 'no']:

            response = ''

        else:

            response = ''

    else:

        pids_list = pids.split(",")

        response = ''



    for pid in pids_list:
```

```
        try:

            process = psutil.Process(int(pid))

            name = process.name()

            mem_usage = process.memory_info().rss / 1024 / 1024

            cpu_usage = process.cpu_percent()

            net_io_counters = psutil.net_io_counters(pernic=False)

            network_usage = net_io_counters.bytes_sent / 1024 / 1024 +
net_io_counters.bytes_recv / 1024 / 1024

            disk_usage = process.io_counters().write_bytes / 1024 / 1024 + process.
io_counters().read_bytes / 1024 / 1024

            if response.lower() in ['y', 'yes']:

                process.kill()

                print(f"Process with PID {pid} ({name}) terminated.")

            elif response.lower() in ['n', 'no']:

                response = input(f"Y/N are you sure that you want to kill PID {pid}
 ({name}) current: mem {mem_usage:.2f}MB, CPU {cpu_usage:.2f}%, net {network_usage
:.2f}MB, disk {disk_usage:.2f}MB? ")

                if response.lower() in ['y', 'yes', 'yeah', 'yep', 'sure', 'ok',
'okay', 'fine', 'affirmative', 'positive']:

                    process.kill()

                    print(f"Process with PID {pid} ({name}) terminated.")

                else:

                    print(f"Skipped terminating process with PID {pid} ({name}).")

            else:

                response = input(f"Y/N are you sure that you want to kill PID {pid}
 ({name}) current: mem {mem_usage:.2f}MB, CPU {cpu_usage:.2f}%, net {network_usage
:.2f}MB, disk {disk_usage:.2f}MB? ")

                if response.lower() in ['y', 'yes', 'yeah', 'yep', 'sure', 'ok',
'okay', 'fine', 'affirmative', 'positive']:

                    process.kill()

                    print(f"Process with PID {pid} ({name}) terminated.")

                else:

                    print(f"Skipped terminating process with PID {pid} ({name}).")

        except (psutil.NoSuchProcess, psutil.AccessDenied, ValueError) as e:
```

```
            print(f"Error: Cannot kill process with PID {pid}. Reason: {e}")
```

security scripts

block all .zip .rar .mov top level domains

current research
impossible

How to disable Start up Apps

Via Task Manager

1. press Ctrl+Shift+Esc.
    - or
2. Open the Task Manager by right-clicking on the taskbar and selecting "Task Manager"
    1. Click on the "Startup" tab to see a list of programs that start automatically with Windows.
        1. Disable the ones you don't need by right-clicking on them and selecting "Disable".
        2. this one is only a small potion of the auto start up tasks and may no longer work on newer windows versions

Via Settings

1. in the search bar or the settings app search for startup apps
    - Some apps may have an option in their settings to disable the automatic startup.
    - you "may" find something there i literally never have
    - stg this is useless but hay its "an option"

Via System Configuration:

1. Windows key + R to open "Run"
    1. type the fallowing program name then click Enter.
2. msconfig
3. Click on the "Startup" tab to see a list of programs that start automatically with Windows.
    1. Disable the ones you don't need by unchecking the checkbox next to them.
    2. if you are running into extensive issues you may need to enable diagnostic startup and see if you still have problems

Via Registry Editor

- "DISCLAIMER" This method should only be used if you have experience with editing the Windows registry.

1. Open the Registry Editor by pressing the Windows key + R and typing "regedit" in the Run dialog box.
2. paste these into the URL bar
    1. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    2. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
    3. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    4. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

    - OBVIOUSLY local and current users are different
    - HKEY_CURRENT_USER
        - is for the logged in user and
    - HKEY_CURRENT_USER
        - but you also have
    - HKEY_USERS

    - this is for other users on the pc

- HKEY_USERS\useridgoeshere\SOFTWARE\Microsoft\Windows \CurrentVersion\Run

    - if you have allot of users on your PC u might just want to loginto there acct by admining them and changing there password

- Remember to only disable apps that you're sure you don't need, as disabling the wrong program could cause problems with your system.

How to admin people Via Run

1. key + R to open the Run dialog box.
2. Type "control userpasswords2" in the Run dialog box and press Enter.
    - control userpasswords2
3. click on user
4. click properies
5. click group membership
6. click administrator
7. click apply

change another users passwd without knowing it

1. open cmd as admin by searching it in the search bar then clicking run as administrator
    1. paste this command into the terminal
2. net user [username] [new password]
    - replace username with the users username
    - and new password with your preferred password
3. ***disclaimer*** don't do this without there permission

fix windows files

1. start CMD or PowerShell in admin mode
2. type or past
   - sfc /scannow
3. restart pc
4. regardless if it fixes stuff, then do
   - DISM /Online /Cleanup-Image /RestoreHealth
5. when its done restart
6. then do
   - sfc /scannow
     - again
7. then restart
   1. and boom all of your corrupt files should be fixed
8. u could do
   - DISM /Online /Cleanup-Image /RestoreHealth
   - again and restart but that's up to you

or do

```
sfc /scannow; DISM /Online /Cleanup-Image /RestoreHealth
```

like a bad boy in powershell

every tool for windows in a single file :D

1. change the name of a folder in your desktop to
   - GodMode.{ED7BA470-8E54-465E-825C-99712043E01C}

download file to sshed windows server

1. Open your SSH client and log in to the Windows server.
2. Navigate to the directory where you want to download the file.
3. Use the "curl" command followed by the URL of the file you want to download.
4. curl -LJO
    1. example location:[
    2. curl -LJO https://github.com/Fictiverse/bark/releases/download/0.2/Bark_WebUI.7z
5. Press Enter to start the download. The file will be downloaded to your current directory on the server.

what does -LJ0 mean?

- `-L` or `--location`: Follow HTTP(S) redirects. This flag tells `curl` to automatically follow any HTTP redirects that the server may send, allowing it to download the file from the correct URL even if the server responds with a redirect.
- `-J` or `--remote-header-name`: Use the remote header name for the downloaded file. This flag tells `curl` to use the suggested filename in the Content-Disposition header, which is typically set by the server when you download a file. In your case, the server sets the filename to `Bark_WebUI.7z`, and this flag tells `curl` to use that filename instead of just saving the file with a generic name like `download`.
- `-O` or `--remote-name`: Use the remote file name for the downloaded file. This flag tells `curl` to save the downloaded file with the same name as it has on the server.

So in summary, `-LJO` tells `curl` to follow redirects and use the filename suggested by the server, and to save the file with the same name it has on the server.

How to Remove 'Show More Options' From the Windows 11 Context Menu - Command Prompt

Open Windows Terminal, Command Prompt, or PowerShell.

Disable:
reg add "HKCU\Software\Classes\CLSID\{86ca1aa0-34aa-4e8b-a509-50c905bae2a2}\InprocServer32" /f /ve

Enable:
reg delete "HKEY_CURRENT_USER\Software\Classes\CLSID\{86ca1aa0-34aa-4e8b-a509-50c905bae2a2}" /f

firefox yt better audio scaling for when im studying

```
    // ==UserScript==
    // @name         Youtube Music fix volume ratio
    // @namespace    http://tampermonkey.net/
    // @version      0.4
    // @description  Makes the YouTube music volume slider exponential so it's easier to
select lower volumes.
    // @author       Marco Pfeiffer <git@marco.zone>
    // @icon         https://music.youtube.com/favicon.ico
    // @match        https://music.youtube.com/*
    // @run-at       document-start
    // @grant        none
    // ==/UserScript==

    (function() {
        'use strict';

        // manipulation exponent, higher value = lower volume
        // 3 is the value used by pulseaudio, which Barteks2x figured out this gist here:
https://gist.github.com/Barteks2x/a4e189a36a10c159bb1644ffca21c02a
        // 0.05 (or 5%) is the lowest you can select in the UI which with an exponent of 3
becomes 0.000125 or 0.0125%
        const EXPONENT = 3;

        const storedOriginalVolumes = new WeakMap();
        const {get, set} = Object.getOwnPropertyDescriptor(HTMLMediaElement.prototype,
'volume');
        Object.defineProperty(HTMLMediaElement.prototype, 'volume', {
            get () {
                const lowVolume = get.call(this);
                const calculatedOriginalVolume = lowVolume ** (1 / EXPONENT);
```

```
                // The calculated value has some accuracy issues which can lead to problems
for implementations that expect exact values.
                // To avoid this, I'll store the unmodified volume to return it when read
here.
                // This mostly solves the issue, but the initial read has no stored value and
the volume can also change though external influences.
                // To avoid ill effects, I check if the stored volume is somewhere in the same
range as the calculated volume.
                const storedOriginalVolume = storedOriginalVolumes.get(this);
                const storedDeviation = Math.abs(storedOriginalVolume -
calculatedOriginalVolume);

                const originalVolume = storedDeviation < 0.01 ? storedOriginalVolume :
calculatedOriginalVolume;
                // console.log('manipulated volume from', lowVolume.toFixed(2), 'to  ',
originalVolume.toFixed(2), storedDeviation);
                return originalVolume;
            },
            set (originalVolume) {
                const lowVolume = originalVolume ** EXPONENT;
                storedOriginalVolumes.set(this, originalVolume);
                // console.log('manipulated volume to  ', lowVolume.toFixed(2), 'from',
originalVolume.toFixed(2));
                set.call(this, lowVolume);
            }
        });
    })();
```

uhhh websites to visit

```
# Define the URLs for the security news websites
$csoUrl = "https://www.csoonline.com"
$krebsUrl = "https://krebsonsecurity.com"
$darkReadingUrl = "https://www.darkreading.com"

# Function to fetch and display headlines from a URL
function Get-NewsHeadlines {
    param (
        [string]$url
    )

    try {
        # Fetch content from the URL
        $response = Invoke-WebRequest -Uri $url -UseBasicParsing

        # Parse and extract headlines - Adjust selectors as needed for each site
        $headlines = $response.Content | Select-String -Pattern "<title>(.*?)</title>" -
AllMatches | ForEach-Object { $_.Matches.Groups[1].Value }

        # Display the headlines
        $headlines | Select-Object -First 5 # Adjust the number to control how many headlines
are displayed
    }
    catch {
        Write-Host "Error fetching news from $url. Please check the URL or network
connectivity."
    }
}

# Fetch and display news from each website
Write-Host "CSO Online Latest Headlines:"
Get-NewsHeadlines -url $csoUrl

Write-Host "`nKrebs on Security Latest Headlines:"
Get-NewsHeadlines -url $krebsUrl
```

```
Write-Host "`nDark Reading Latest Headlines:"
Get-NewsHeadlines -url $darkReadingUrl
```

windows 10 exploerer in windows 10

Open File Explorer and navigate to Control Panel from the Address bar.

From Control Panel, open the navigation again from the Address bar and click on Home.

sometimes if you type explorer in it it works

source

usefull windows hotkeys

Quick Rename

- f2

while renaming to go to next file without needing to click f2 again

- tab