

Paranoia Troll Script Use Guide

Overview

Paranoia is a script intended to create a unique and unsettling experience on your computer. It can randomly:

- Create pop-up windows
- Talk to you using text-to-speech
- Make random beeping sounds
- Eject your CD/optical drives
- Flip your mouse buttons
- Make your keyboard toggle lights (capslock, numlock etc.) blink
- Change the brightness of your monitor (only on certain monitors)
- Change your wallpaper
- Send keyboard inputs
- Write spooky text files on your desktop
- Create spooky word documents (if you have Microsoft Word)
- Open the Windows Camera app
- Make any connected printers print spooky text

These payloads are intended to create a kind of "digital psychological horror" experience, giving off a feeling that you are hallucinating, stuck inside a dream/simulation, or that your computer is haunted by something.

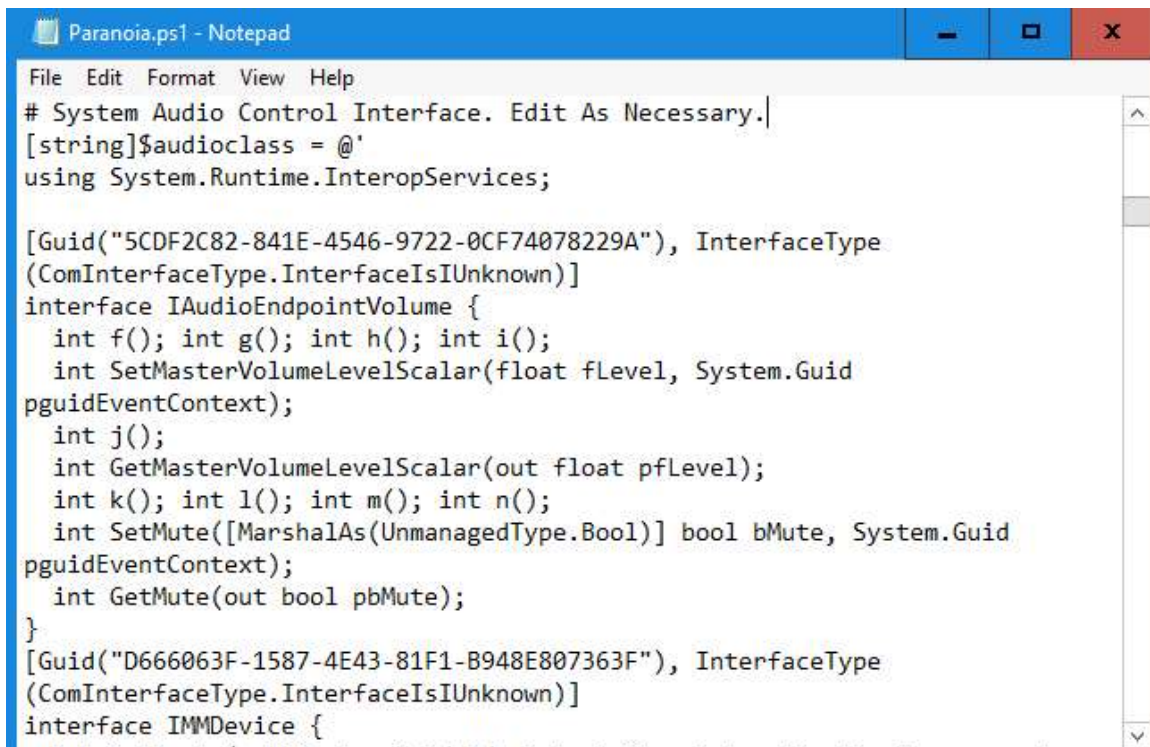
Configuration

Paranoia has various settings that can be configured by opening **Paranoia.ps1** in a text editor such as Notepad. These settings include:

- Enabling/Disabling payloads
- Setting payload chances
- Setting payload intervals
- Miscellaneous settings such as volume settings for audio based payloads, font settings for the word document payload, and the wallpaper to use for the wallpaper payload

These settings can be found in the **Settings** section of **Paranoia.ps1**.

Paranoia uses the Windows Core Audio API to control the system volume and mute settings for audio based payloads. Based on the target system, it may be necessary to reconfigure certain parts of the Audio API Interface for these parts of the script to properly run. The Core Audio API Interface code may be found in the **Script** section of **Paranoia.ps1**, from lines **186-229**:



```
File Edit Format View Help
# System Audio Control Interface. Edit As Necessary.
[string]$audioclass = '@'
using System.Runtime.InteropServices;

[Guid("5CDF2C82-841E-4546-9722-0CF74078229A"), InterfaceType
(ComInterfaceType.InterfaceIsIUnknown)]
interface IAudioEndpointVolume {
    int f(); int g(); int h(); int i();
    int SetMasterVolumeLevelScalar(float fLevel, System.Guid
pguidEventContext);
    int j();
    int GetMasterVolumeLevelScalar(out float pfLevel);
    int k(); int l(); int m(); int n();
    int SetMute([MarshalAs(UnmanagedType.Bool)] bool bMute, System.Guid
pguidEventContext);
    int GetMute(out bool pbMute);
}

[Guid("D666063F-1587-4E43-81F1-B948E807363F"), InterfaceType
(ComInterfaceType.InterfaceIsIUnknown)]
interface IMMDevice {
```

Certain payloads, such as the popup window and text to speech payloads, select their options randomly from a preset list. You may modify these presets by modifying the preset list found in the payload's code block:

Window Payload Preset List

```

$windowtype = get-random -max 78;
if ($windowtype -eq 0) {$windowtext = "Everything is fine. Go back to
work."; $windowtitle = "You are not being watched";}
elseif ($windowtype -eq 1) {$windowtext = "Do you hear the voices?";}
elseif ($windowtype -eq 2) {$windowtext = "Everything is fine. Return to
your work."; $windowtitle = "I am watching";}
elseif ($windowtype -eq 3) {$windowtext = "Everything is fine. Nothing is
behind you."; $windowtitle = "Do not look";}
elseif ($windowtype -eq 4) {$windowtext = "Do not look behind you. There is
nothing there.";}
elseif ($windowtype -eq 5) {$windowtext = "Do not look behind you.";
$windowtitle = "I am not watching you";}
elseif ($windowtype -eq 6) {$windowtext = "There is nothing behind you.";}
elseif ($windowtype -eq 7) {$windowtext = "Look behind you.";}
elseif ($windowtype -eq 8) {$windowtext = "He is here. Look behind you.";}
elseif ($windowtype -eq 9) {$windowtext = "I am here. Look behind you";}
elseif ($windowtype -eq 10) {$windowtext = "He is here. Do not look behind
you."; $windowtitle = "You are not safe";}

```

Text To Speech Payload Preset List

```

$audiotype = get-random -max 57;
if ($audiotype -eq 0) {$speech = "Everything is fine. Nothing is wrong. You
do not hear anything. Go back to work.";}
elseif ($audiotype -eq 1) {$speech = "You do not hear anything. Keep
working.";}
elseif ($audiotype -eq 2) {$speech = "Everything is fine. Return to your
work.";}
elseif ($audiotype -eq 3) {$speech = "Everything is fine. Nothing is behind
you.";}
elseif ($audiotype -eq 4) {$speech = "Do not look behind you. There is
nothing there.";}
elseif ($audiotype -eq 5) {$speech = "Do not look behind you.";}
elseif ($audiotype -eq 6) {$speech = "There is nothing behind you.";}
elseif ($audiotype -eq 7) {$speech = "Look behind you.";}
elseif ($audiotype -eq 8) {$speech = "He is here. Look behind you.";}
elseif ($audiotype -eq 9) {$speech = "I am here. Look behind you";}
elseif ($audiotype -eq 10) {$speech = "He is here. Do not look behind
you.";}
elseif ($audiotype -eq 11) {$speech = "I am here. Do not look behind you";}

```

When modifying payload preset lists, be sure to change the **max** parameter of the random number generator to accomodate all possible values in the list:

```
$audiotype = get-random -max 57;
```

Remember that the range for **get-random** is **0 to (MAX - 1)**, this should be the range used in the preset list's if statements.

Use Guide

Paranoia can be activated by running **Gaslight.bat**. This file must be in the same directory as **Paranoia.ps1**.

Paranoia can be deactivated by running **RemoveGaslight.bat**. This file does **not** need to be in the same directory as **Paranoia.ps1**.

Some settings may require Paranoia to be run as **Administrator** to function.

You can do this by **right-clicking Gaslight.bat** and clicking **Run As Administrator**.

If the **\$runonstartup** setting is set to **\$true**, Paranoia will automatically run on startup once it has been activated, as long as you are logged into the same account it was run on. This will stop once **RemoveGaslight.bat** has been run.

Paranoia's mouse swapping payload changes a system setting, and will remain after Paranoia itself has been stopped, which you may find difficult to fix. To fix this, **RemoveGaslight.bat** can call **FixMouse.ps1**, if available, which will restore the mouse settings to the **system default (primary button left, secondary button right)**. If your normal mouse settings are the **inverse (primary button right, secondary button left)**, then you must

change it back using either:

**Control Panel => Hardware and Sound => Devices and Printers => Mouse
=> Switch primary and secondary buttons**

OR

Settings => Devices => Mouse => Select your primary button

This will change your mouse button settings.

I hope you enjoy!

HTX Venom