PWNCLOUDOS v1.1 — Release Update

Zsh Configuration for Non-root User with Custom Prompt and Color Scheme

ZSH root prompt

PowerShell Profile Configuration for normal user

PowerShell Profile Configuration for sudo user (root)

Hostname and username are now displayed directly in the PowerShell prompt for better session visibility.

ZSH - Python Virtual Environment has been fixed!

PowerShell - Python Virtual Environment has been fixed!

sudo user format for Zsh and PowerShell

OpenVPN plugin and Network Manager installed — allowing users to easily connect to VPNs through a graphical plugin.

Fixed
Connect-MgGraph authentication issues.

AADInternals Launcher guide updated with improved usage instructions.

Pacu updated to the latest version.

AWeSomeUserFinder updated to the latest version.

Session Manager plugin Installed

Added Support for JSON Parsing via jq

Added Steampipe - Multicloud tool

Added Powerpipe - Multicloud Tool
Added seamlesspass tool
Installed impacket 0.12.0 using Python 3.11.2

Zsh Configuration for Non-root User with Custom Prompt and Color Scheme

```
┌──[pwnedlabs@cloud]─[~]
└─────-$ cat ~/.zshrc
# ~/.zshrc: executed by zsh for interactive shells.

# Location of starship config File
export STARSHIP_CONFIG="$HOME/.config/starship/starship.toml"

# Initialize Starship prompt
eval "$(starship init zsh)"

# Enable colors in prompt
autoload -U colors && colors

# Define the prompt (Kali custom prompt and PwncloudOS prompt)
PROMPT=$' ┌──[%F{#FF33FF}pwnedlabs%f@%F{#7EC8E3}cloud%f]─[%F{#6FF

# If not running interactively, don't do anything
[[ -o interactive ]] || return

# History configurations
HISTCONTROL=ignoreboth
HISTSIZE=10000
HISTFILESIZE=10000
HISTFILE=~/.zsh_history
SAVEHIST=10000
setopt SHARE_HISTORY
setopt hist_expire_dups_first # delete duplicates first when HISTFILE size exceed
setopt hist_ignore_dups       # ignore duplicated commands history list
```

```
setopt hist_ignore_space      # ignore commands that start with space
setopt hist_verify            # show command with history expansion to user before r

# Append to the history file, don't overwrite it
alias history="history 0"

# Set variable identifying the chroot you work in (used in the prompt below)
if [[ -z "$debian_chroot" ]] && [[ -r /etc/debian_chroot ]]; then
    debian_chroot=$(< /etc/debian_chroot)
fi

# Enable color support of ls and also add handy aliases
if [[ -x /usr/bin/dircolors ]]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)
    alias ls='ls --color=auto'
fi

# Some more ls aliases with color customization for light and dark terminals
export LS_COLORS="di=36:ln=36:so=32:pi=33:ex=38;5;203:bd=36:cd=36:su=3
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# Enable programmable completion features
autoload -Uz compinit
compinit

# Custom paths (from both files)
export PATH="$PATH:/home/pwnedlabs/.local/bin"
export PATH="$PATH:/opt/azure_tools/azure_hound"
export PATH="$PATH:/root/.local/bin"  # pipx
export PATH="$PATH:/usr/local/go/bin" # Go
export PATH="$PATH:/home/pwnedlabs/go/bin" # CloudFox
export PATH="$PATH:/opt/mssql-tools/bin" # MS_SQL

# Plugins
```

```
plugins=(git zsh-autosuggestions zsh-syntax-highlighting)

# Enable zsh-autosuggestions
if [[ -f /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh ]]; then
    source /usr/share/zsh-autosuggestions/zsh-autosuggestions.zsh
    ZSH_AUTOSUGGEST_HIGHLIGHT_STYLE='fg=#808080'  # Gray color
fi

# Enable zsh-syntax-highlighting
if [[ -f /usr/share/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh ]]; then
    source /usr/share/zsh-syntax-highlighting/zsh-syntax-highlighting.zsh
fi

# Enable command-not-found if installed
if [ -f /etc/zsh_command_not_found ]; then
    . /etc/zsh_command_not_found
fi

# Additional key bindings and settings
setopt autocd              # change directory just by typing its name
setopt interactivecomments # allow comments in interactive mode
setopt magicequalsubst     # enable filename expansion for arguments of the form
setopt nonomatch           # hide error message if there is no match for the pattern
setopt notify              # report the status of background jobs immediately
setopt numericglobsort     # sort filenames numerically when it makes sense
setopt promptsubst         # enable command substitution in prompt

# Configure key bindings
bindkey -e                                # emacs key bindings
bindkey ' ' magic-space                   # do history expansion on space
bindkey '^U' backward-kill-line           # ctrl + U
bindkey '^[[3;5~' kill-word               # ctrl + Supr
bindkey '^[[3~' delete-char               # delete
bindkey '^[[1;5C' forward-word            # ctrl + →
bindkey '^[[1;5D' backward-word           # ctrl + ←
bindkey '^[[5~' beginning-of-buffer-or-history    # page up
```

```
bindkey '^[[6~' end-of-buffer-or-history        # page down
bindkey '^[[H' beginning-of-line                # home
bindkey '^[[F' end-of-line                      # end
bindkey '^[[Z' undo                             # shift + tab undo last action

# History configurations
alias history="history 0"

# Set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# Uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
        # a case would tend to support setf rather than setaf.)
        color_prompt=yes
    else
        color_prompt=
    fi
fi

# Custom prompt configurations
configure_prompt() {
    prompt_symbol=㉿
    case "$PROMPT_ALTERNATIVE" in
        twoline)
            PROMPT=$'%F{%(#.blue.green)}┌──${debian_chroot:+($debian_chroot)
            ;;
```

```
        oneline)
            PROMPT=$'${debian_chroot:+($debian_chroot)}${VIRTUAL_ENV:+($(bas
            RPROMPT=
            ;;
    esac
    unset prompt_symbol
}

# Set a color prompt for supported terminals
if [ "$color_prompt" = yes ]; then
    VIRTUAL_ENV_DISABLE_PROMPT=1
    configure_prompt
fi

# Custom functions and aliases
toggle_oneline_prompt() {
    if [ "$PROMPT_ALTERNATIVE" = oneline ]; then
        PROMPT_ALTERNATIVE=twoline
    else
        PROMPT_ALTERNATIVE=oneline
    fi
    configure_prompt
    zle reset-prompt
}
zle -N toggle_oneline_prompt
bindkey ^P toggle_oneline_prompt

# End of file
```

ZSH root prompt

```
┌──[root@cloud]─[/home/pwnedlabs]
└─── # cat /root/.zshrc
```

```
# Created by `pipx` on 2025-01-01 07:56:22
export PATH="$PATH:/root/.local/bin"

# Load colors for zsh
autoload -U colors && colors

# Define the prompt (similar to your current user prompt)
PROMPT=$' ┌─[%F{#FF6B6B}root%f@%F{#FF6B6B}cloud%f]─[%F{#6FFF4F}%

# Enable color support for ls and set LS_COLORS
export LS_COLORS="di=36:ln=36:so=32:pi=33:ex=38;5;203:bd=36:cd=36:su=3
alias ls='ls --color=auto'

# Ensure history is enabled, as it is for your user
HISTCONTROL=ignoreboth
HISTSIZE=10000
HISTFILESIZE=10000
HISTFILE=~/.zsh_history
SAVEHIST=10000
setopt SHARE_HISTORY
```

PowerShell Profile Configuration for normal user

```
┌─ PS [pwnedlabs@cloud]─[/home]
└──╼>  cat /home/pwnedlabs/.config/powershell/Microsoft.PowerShell_profile.p
# PowerShell Profile Configuration
# Custom prompt with Zsh-like styling and true color support

# ANSI Color Definitions (24-bit RGB)
$ESC = [char]27
$magenta = "$ESC[38;2;255;51;255m"   # #FF33FF (pwnedlabs)
$blue = "$ESC[38;2;126;200;227m"     # #7EC8E3 (cloud)
$green = "$ESC[38;2;111;255;79m"     # #6FFF4F (directory)
$promptBlue = "$ESC[38;2;88;93;245m" # #585df5 (prompt lines)
```

```
$venvRed = "$ESC[38;2;217;33;33m"    # #D92121 (virtualenv)
$white = "$ESC[38;2;255;255;255m"    # White for brackets
$reset = "$ESC[0m"

# Zsh-like colors for items (close to the image)
$dirColor = "$ESC[38;5;141m"    # Light Blue directory color
$exeColor = "$ESC[38;5;203m"        # Light red (ex=38;5;203 in Zsh)
$symlinkColor = "$ESC[36m"          # Cyan (ln=36 in Zsh)
$otherColor = "$ESC[37m"            # Gray (default)

# Custom Colored Listing
function Get-ChildItemColor {
    param([string]$Path = ".")
    Get-ChildItem $Path @args | ForEach-Object {
        if ($_.PSIsContainer) {
            Write-Host ($dirColor + $_.Name + $reset) -NoNewline
            Write-Host "/" -ForegroundColor White
        }
        elseif ($_.Extension -match '\.(exe|sh|ps1)$') {
            Write-Host ($exeColor + $_.Name + $reset)
        }
        elseif ($_.LinkType -eq "SymbolicLink") {
            Write-Host ($symlinkColor + $_.Name + $reset) -NoNewline
            Write-Host "@" -ForegroundColor White
        }
        else {
            Write-Host ($otherColor + $_.Name + $reset)
        }
    }
}

# Custom PowerShell Prompt Function
function prompt {
    # Detect if the current user is root using whoami
    $user = if ((whoami) -eq 'root') { 'root' } else { 'pwnedlabs' }
    $hostname = "cloud"
```

```
    $dir = (Get-Location).Path.Replace($HOME, '~')  # Shorten home path

    # Virtual Environment Detection
    $venv = if ($env:VIRTUAL_ENV) {
        " $venvRed($([System.IO.Path]::GetFileName($env:VIRTUAL_ENV)))$reset"
    }

    # Set prompt colors
    $userColor = if ($user -eq 'root') { "$ESC[31m" } else { "$magenta" }  # Red fo
    $hostnameColor = if ($user -eq 'root') { "$ESC[31m" } else { "$blue" }  # Red f
    $dirColor = "$green"
    $promptSymbolColor = if ($user -eq 'root') { "$ESC[31m" } else { "$promptBlu

    # Set the prompt symbol (`#` for root, `>` for non-root)
    $promptSymbol = if ($user -eq 'root') { "#" } else { ">" }

    # Build Prompt Lines with all brackets [ ] in white
    $line1 = "$promptBlue ┌─ PS $white[$userColor$user$reset@$hostnameColor
    $line2 = "$promptBlue └──╼$promptSymbol$reset "

    # Print Prompt
    Write-Host $line1
    Write-Host $line2 -NoNewline

    return " "
}

# Enable UTF-8 for ANSI color support
$OutputEncoding = [Console]::OutputEncoding = [Text.UTF8Encoding]::New()
[Console]::InputEncoding = [Text.UTF8Encoding]::New()

# Fixed Aliases
function ll { Get-ChildItem -Force -Hidden @args }  # ls -al equivalent
function la { Get-ChildItem -Force @args }          # ls -a equivalent
function l { Get-ChildItem -Force @args }           # Basic listing
```

```
# Set Aliases with AllScope
Set-Alias ll ll -Option AllScope
Set-Alias la la -Option AllScope
Set-Alias l l -Option AllScope

# Quality-of-Life Improvements
$MaximumHistoryCount = 4096
Set-PSReadLineOption -PredictionSource History
Set-PSReadLineKeyHandler -Chord "Ctrl+LeftArrow" -Function BackwardWord
Set-PSReadLineKeyHandler -Chord "Ctrl+RightArrow" -Function ForwardWord

# Set Alias for ls
Set-Alias ls Get-ChildItemColor -Option AllScope

# Auto-Reload Profile Shortcut
function Reload-Profile { & $PROFILE }
Set-Alias reload Reload-Profile -Option AllScope


# End of PowerShell profile configuration
```

PowerShell Profile Configuration for sudo user (root)

```
┌──[pwnedlabs@cloud]─[~]
└─── $ sudo cat /root/.config/powershell/Microsoft.PowerShell_profile.ps1
[sudo] password for pwnedlabs:
# PowerShell Profile Configuration
# Custom prompt with Zsh-like styling and true color support

# ANSI Color Definitions (24-bit RGB)
$ESC = [char]27
$magenta = "$ESC[38;2;255;51;255m"   # #FF33FF (pwnedlabs)
$blue = "$ESC[38;2;126;200;227m"     # #7EC8E3 (cloud)
```

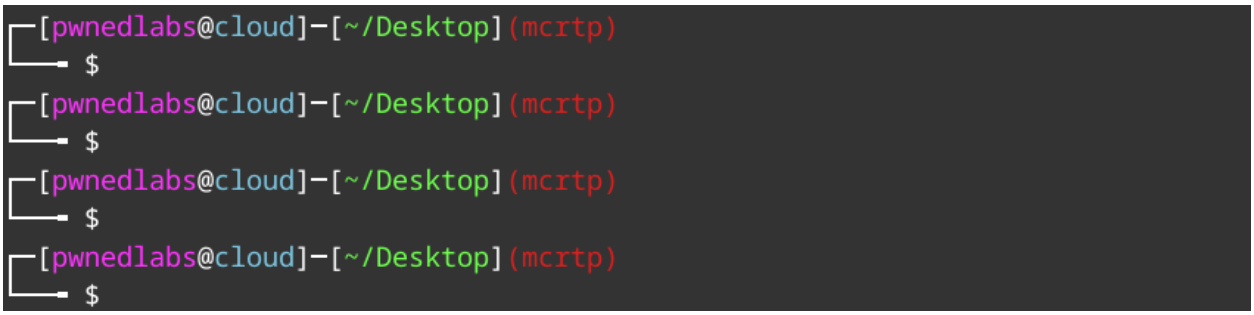
```
$green = "$ESC[38;2;111;255;79m"     # #6FFF4F (directory)
$promptBlue = "$ESC[38;2;88;93;245m" # #585df5 (prompt lines)
$venvRed = "$ESC[38;2;217;33;33m"    # #D92121 (virtualenv)
$white = "$ESC[38;2;255;255;255m"    # White for brackets
$reset = "$ESC[0m"

# Light red for root user display
$lightRed = "$ESC[38;5;9m"

# Zsh-like colors for items (close to the image)
$dirColor = "$ESC[38;5;141m"    # Light Blue directory color
$exeColor = "$ESC[38;5;203m"        # Light red (ex=38;5;203 in Zsh)
$symlinkColor = "$ESC[36m"          # Cyan (ln=36 in Zsh)
$otherColor = "$ESC[37m"            # Gray (default)

# Custom Colored Listing
function Get-ChildItemColor {
    param([string]$Path = ".")
    Get-ChildItem $Path @args | ForEach-Object {
        if ($_.PSIsContainer) {
            Write-Host ($dirColor + $_.Name + $reset) -NoNewline
            Write-Host "/" -ForegroundColor White
        }
        elseif ($_.Extension -match '\.(exe|sh|ps1)$') {
            Write-Host ($exeColor + $_.Name + $reset)
        }
        elseif ($_.LinkType -eq "SymbolicLink") {
            Write-Host ($symlinkColor + $_.Name + $reset) -NoNewline
            Write-Host "@" -ForegroundColor White
        }
        else {
            Write-Host ($otherColor + $_.Name + $reset)
        }
    }
}
```

```
# Custom PowerShell Prompt Function
function prompt {
    # Detect if the current user is root using whoami
    $user = if ((whoami) -eq 'root') { 'root' } else { 'pwnedlabs' }
    $hostname = "cloud"
    $dir = (Get-Location).Path.Replace($HOME, '~')  # Shorten home path

    # Virtual Environment Detection
    $venv = if ($env:VIRTUAL_ENV) {
        " $venvRed($([System.IO.Path]::GetFileName($env:VIRTUAL_ENV)))$reset"
    }

    # Set prompt colors (Light Red for root)
    $userColor = if ($user -eq 'root') { "$lightRed" } else { "$magenta" }  # Light R
    $hostnameColor = if ($user -eq 'root') { "$lightRed" } else { "$blue" }  # Light
    $dirColor = "$green"
    $promptSymbolColor = if ($user -eq 'root') { "$lightRed" } else { "$promptBlue

    # Set the prompt symbol (`#` for root, `>` for non-root)
    $promptSymbol = if ($user -eq 'root') { " $lightred#" } else { ">" }

    # Build Prompt Lines with all brackets [ ] in white
    $line1 = "$promptBlue ┌─ PS $white[$userColor$user$reset@$hostnameColor
    $line2 = "$promptBlue └──-$promptSymbol$reset "

    # Print Prompt
    Write-Host $line1
    Write-Host $line2 -NoNewline

    return " "
}

# Enable UTF-8 for ANSI color support
$OutputEncoding = [Console]::OutputEncoding = [Text.UTF8Encoding]::New()
[Console]::InputEncoding = [Text.UTF8Encoding]::New()
```

```
# Fixed Aliases
function ll { Get-ChildItem -Force -Hidden @args }  # ls -al equivalent
function la { Get-ChildItem -Force @args }          # ls -a equivalent
function l { Get-ChildItem -Force @args }           # Basic listing

# Set Aliases with AllScope
Set-Alias ll ll -Option AllScope
Set-Alias la la -Option AllScope
Set-Alias l l -Option AllScope

# Quality-of-Life Improvements
$MaximumHistoryCount = 4096
Set-PSReadLineOption -PredictionSource History
Set-PSReadLineKeyHandler -Chord "Ctrl+LeftArrow" -Function BackwardWord
Set-PSReadLineKeyHandler -Chord "Ctrl+RightArrow" -Function ForwardWord

# Set Alias for ls
Set-Alias ls Get-ChildItemColor -Option AllScope

# Auto-Reload Profile Shortcut
function Reload-Profile { & $PROFILE }
Set-Alias reload Reload-Profile -Option AllScope

# End of PowerShell profile configuration
```

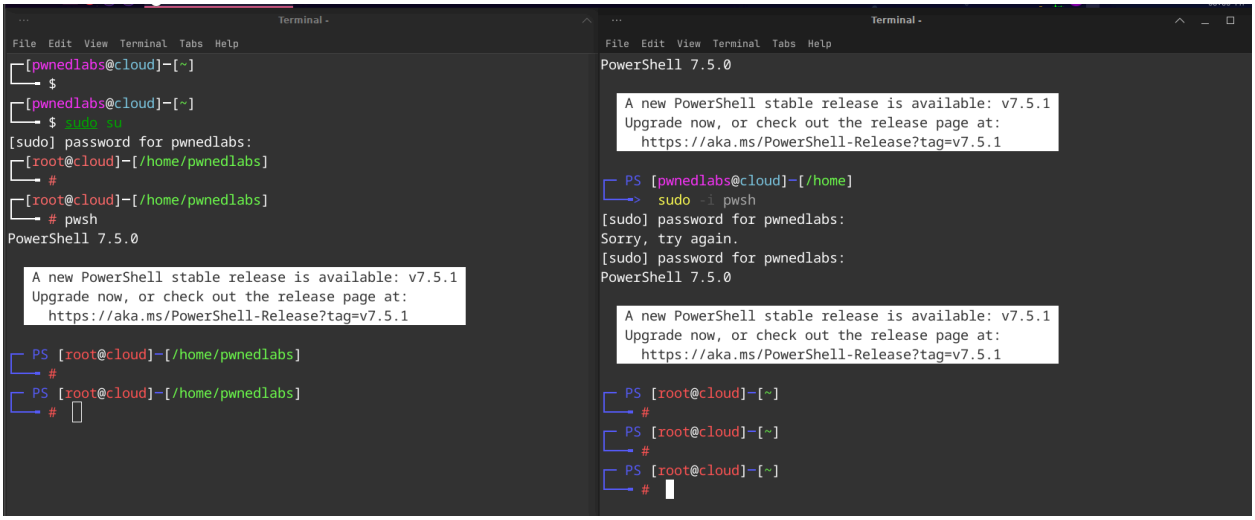
Hostname and username are now displayed directly in the PowerShell prompt for better session visibility.

ZSH - Python Virtual Environment has been fixed!

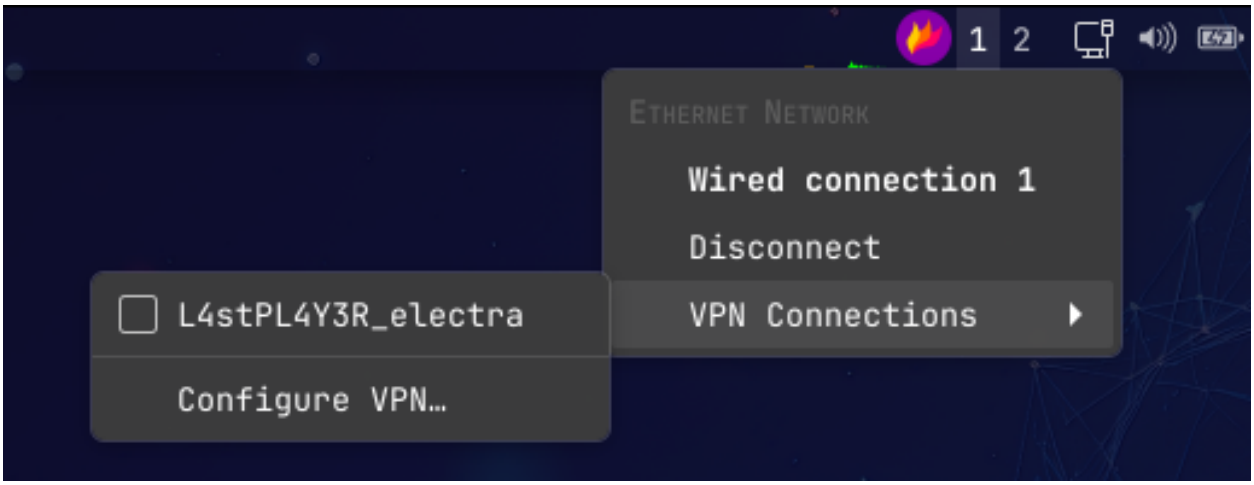


PowerShell - Python Virtual Environment has been fixed!

sudo user format for Zsh and PowerShell



OpenVPN plugin and Network Manager installed — allowing users to easily connect to VPNs through a graphical plugin.




```
sudo apt install network-manager-openvpn network-manager-openvpn-gnome c

sudo systemctl restart NetworkManager
```

Fixed
Connect-MgGraph authentication issues.

```
┌──[pwnedlabs@cloud]─[~]
└──╼ $ sudo chown -R $USER:$USER ~/.local/share/powershell/Modules/Micro
┌──[pwnedlabs@cloud]─[~]
└──╼ $ sudo chown 700  ~/.local/share/powershell/Modules/Microsoft.Graph
┌──[pwnedlabs@cloud]─[~]
└──╼ $ timedatectl status | grep -i "system clock synchronized"
System clock synchronized: yes
```

```
┌─ PS [/home/pwnedlabs]
└──>/>  Connect-MgGraph -TenantId "2590ccef-687d-493b-ae8d-441cbab63a72" -UseDeviceCode
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CDQTCLJ8E to authenticate.
Welcome to Microsoft Graph!

Connected via delegated access using 14d82eec-204b-4c2f-b7e8-296a70dab67e
Readme: https://aka.ms/graph/sdk/powershell
SDK Docs: https://aka.ms/graph/sdk/powershell/docs
API Docs: https://aka.ms/graph/docs

NOTE: You can use the -NoWelcome parameter to suppress this message.

┌─ PS [/home/pwnedlabs]
└──>/>
┌─ PS [/home/pwnedlabs]
└──>/>  Get-MgUser -UserId alee@megabigtech.com | fl
AboutMe                          :
AccountEnabled                   :
Activities                       :
AgeGroup                         :
AgreementAcceptances             :
AppRoleAssignments               :
AssignedLicenses                 :
AssignedPlans                    :
Authentication                   : Microsoft.Graph.PowerShell.Models.Micro
                                   softGraphAuthentication
AuthorizationInfo                : Microsoft.Graph.PowerShell.Models.Micro
                                   softGraphAuthorizationInfo
Birthday                         :
BusinessPhones                   : {}
Calendar                         : Microsoft.Graph.PowerShell.Models.Micro
                                   softGraphCalendar
CalendarGroups                   :
CalendarView                     :
Calendars                        :
Chats                            :
```

AADInternals Launcher guide updated with improved usage instructions.

Option 0: Install from PowerShell Gallery (if needed): Install-Module AADInternals

```
To use the AADInternals module, you can install or import it using the following options:
Option 0: Install from PowerShell Gallery (if needed): Install-Module AADInternals
Option 1 (recommended): Import-Module ./AADInternals.psd1
Option 2: Import-Module ./AADInternals.psm1
```

Pacu updated to the latest version.

```
┌──[pwnedlabs@cloud]─[~]
└──╼ $ pacu --version
Pacu 1.6.0
```

AWeSomeUserFinder updated to the latest version.

https://github.com/dievus/AWeSomeUserFinder

Session Manager plugin Installed

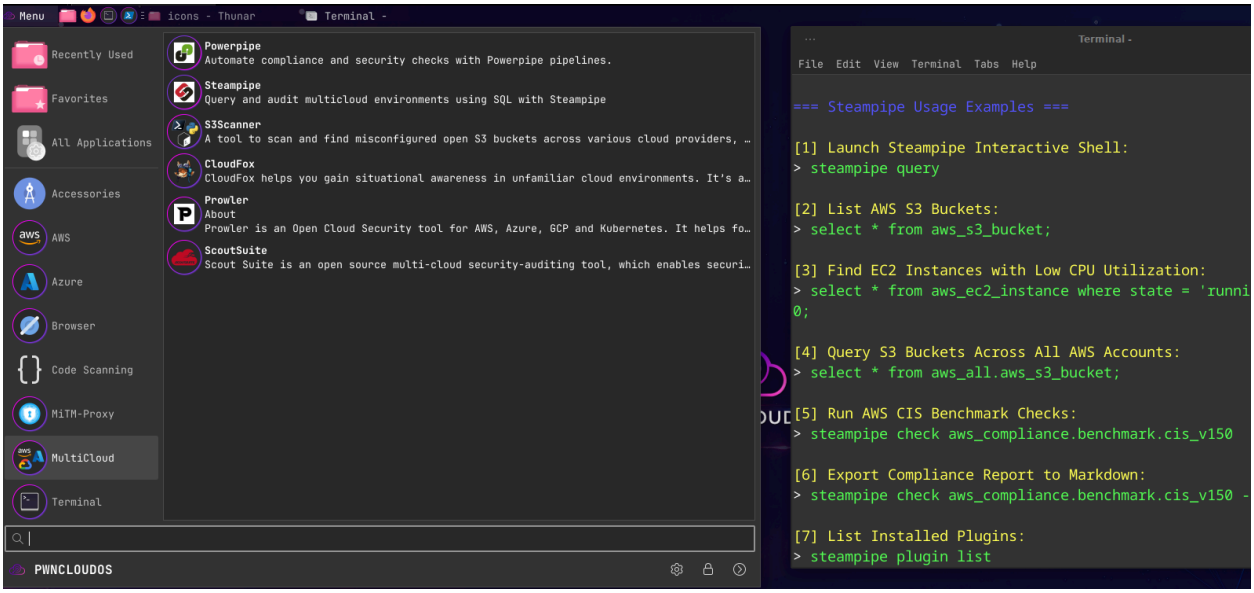
```
┌─[pwnedlabs@cloud]─[~]
└──╼ $ session-manager-plugin

The Session Manager plugin was installed successfully. Use the AWS CLI to start
a session.
```

Added Support for JSON Parsing via jq

```
┌─[pwnedlabs@cloud]─[/opt/aws_tools/aws_enumerator]
└──╼ $ echo '{"name": "PwnCloudOS", "version": 1.1}' | jq '.name'

"PwnCloudOS"
```

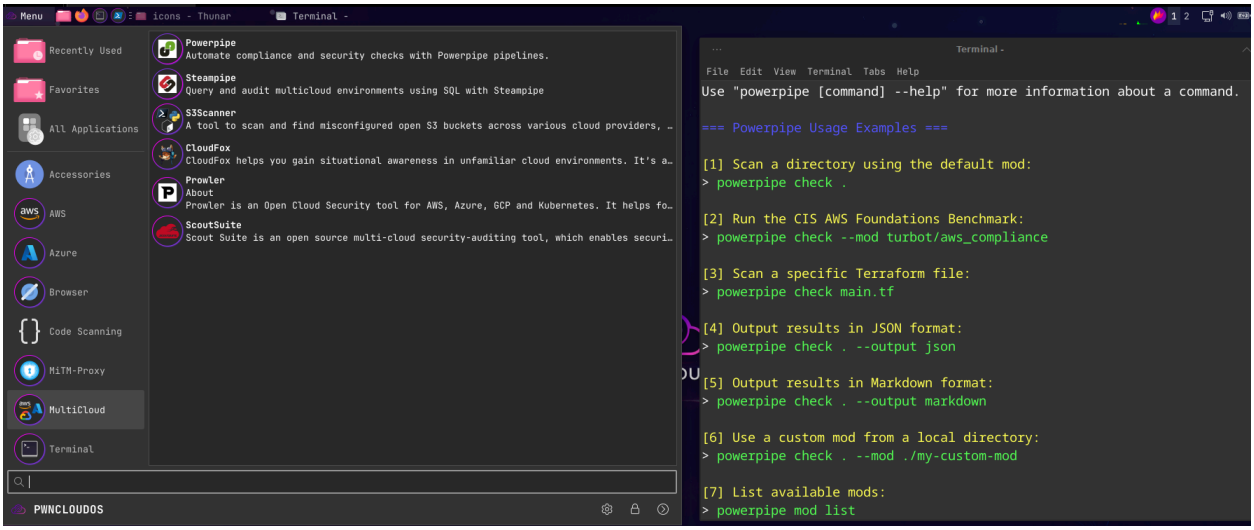
Added Steampipe - Multicloud tool

Query and audit multicloud environments using SQL with Steampipe



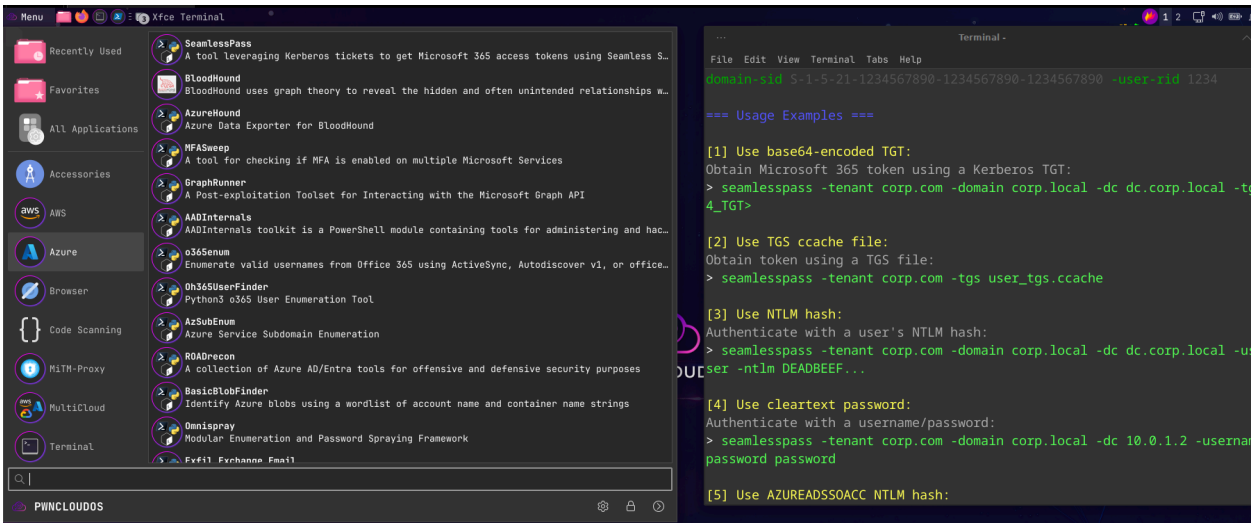
Added Powerpipe - Multicloud Tool

Automate compliance and security checks with Powerpipe pipelines.



Added seamlesspass tool

A tool leveraging Kerberos tickets to get Microsoft 365 access tokens using Seamless SSO



Installed impacket 0.12.0 using Python 3.11.2

Reference: https://pypi.org/project/impacket/

```
┌─[pwnedlabs@cloud]─[~]
└─► $ pipx install impacket
  installed package impacket 0.12.0, installed using Python 3.11.2
  These apps are now globally available
    - DumpNTLMInfo.py
    - Get-GPPPassword.py
    - GetADComputers.py
    - GetADUsers.py
    - GetLAPSPassword.py
    - GetNPUsers.py
    - GetUserSPNs.py
    - addcomputer.py
```