

Forum:[Hack Forums](#)

Thread: [Red & Blue Team Winget Installation Arsenal](#)

Clumsy - August 10, 2025, 10:03 am

Run this powershell script as admin.

Code # List of desired tools with winget IDs (some entries may have alternative IDs)

```
$tools = @(
    @{Name='Wireshark'; WingetId='WiresharkFoundation.Wireshark'},
    @{Name='Nmap'; WingetId='Nmap.Nmap'},
    @{Name='Npcap'; WingetId='Npcap.Npcap'},
    @{Name='BurpSuiteCommunity'; WingetId='PortSwigger.BurpSuiteCommunity'},
    @{Name='OWASP ZAP'; WingetId='OWASP.ZAP'},
    @{Name='Hashcat'; WingetId='Hashcat.Hashcat'},
    @{Name='John The Ripper'; WingetId='Openwall.JohnTheRipper'},
    @{Name='Aircrack-ng'; WingetId='Aircrack-ng.Aircrack-ng'},
    @{Name='Ghidra'; WingetId='NationalSecurityAgency.Ghidra'},
    @{Name='IDA-Free'; WingetId='Hex-Rays.IDA-Free'},
    @{Name='Radare2'; WingetId='Radare.Radare2'},
    @{Name='SysinternalsSuite'; WingetId='Microsoft.Sysinternals'},
    @{Name='Sysmon'; WingetId='Microsoft.Sysmon'},
    @{Name='ProcessHacker'; WingetId='ProcessHacker.ProcessHacker'},
    @{Name='PowerShell'; WingetId='Microsoft.PowerShell'},
    @{Name='Python'; WingetId='Python.Python.3'},
    @{Name='Go'; WingetId='GoLang.Go'},
    @{Name='OpenJDK17'; WingetId='Microsoft.OpenJDK.17'},
    @{Name='NodeJS'; WingetId='NodeJS.NodeJS'},
    @{Name='Rust'; WingetId='Rustlang.Rust'},
    @{Name='Perl'; WingetId='StrawberryPerl.Pperl'},
    @{Name='PHP'; WingetId='PHP.PHP'},
    @{Name='Ruby'; WingetId='RubyInstallerTeam.Ruby'},
    @{Name='Git'; WingetId='Git.Git'},
    @{Name='GnuPG'; WingetId='GnuPG.GnuPG'},
    @{Name='Postman'; WingetId='Postman.Postman'},
    @{Name='VSCode'; WingetId='Microsoft.VisualStudioCode'},
    @{Name='Notepad++'; WingetId='Notepad++.Notepad++'},
    @{Name='SublimeText'; WingetId='SublimeHQ.SublimeText.4'},
    @{Name='DockerDesktop'; WingetId='Docker.DockerDesktop'},
    @{Name='VirtualBox'; WingetId='Oracle.VirtualBox'},
    @{Name='VMwarePlayer'; WingetId='VMware.WorkstationPlayer'},
    @{Name='GoogleChrome'; WingetId='Google.Chrome'},
    @{Name='Firefox'; WingetId='Mozilla.Firefox'},
    @{Name='Brave'; WingetId='Brave.Brave'},
    @{Name='TorBrowser'; WingetId='TorProject.TorBrowser'},
    @{Name='Slack'; WingetId='SlackTechnologies.Slack'},
    @{Name='Discord'; WingetId='Discord.Discord'},
    @{Name='Zoom'; WingetId='Zoom.Zoom'},
    @{Name='OpenVPN'; WingetId='OpenVPNTechnologies.OpenVPN'},
    @{Name='WireGuard'; WingetId='WireGuard.WireGuard'},
```

```

@{Name='Malwarebytes'; WingetId='Malwarebytes.Malwarebytes'},
@{Name='ClamWin'; WingetId='ClamWin.ClamWin'},
@{Name='PuTTY'; WingetId='PuTTY.PuTTY'},
@{Name='Terminus'; WingetId='Terminus.Terminus'},
@{Name='MobaXterm'; WingetId='MobaXterm.MobaXterm'}
# Add more as needed, leaving out unavailable packages
)

# Function to check if winget package exists
function Test-WingetPackageExists {
    param ($Id)
    $searchResult = winget search --id $Id 2>$null
    return $searchResult -match $Id
}

# Function to install with fallback
function Install-Tool {
    param ($tool)

    Write-Host "Attempting to install $($tool.Name)..."

    if (Test-WingetPackageExists $tool.WingetId) {
        winget install --id $tool.WingetId --exact --accept-source-agreements
--accept-package-agreements --silent
        if ($LASTEXITCODE -ne 0) {
            Write-Warning "$($tool.Name): Winget install failed."
        } else {
            Write-Host "$($tool.Name) installed successfully."
            return
        }
    }
    else {
        Write-Warning "$($tool.Name): Package not found in winget repository."
    }

    # Fallback: Python packages (pip) for certain tools
    $pipTools = @('sqlmap','dirsearch','ffuf','Recon-ng','SpiderFoot')
    if ($pipTools -contains $tool.Name.ToLower()) {
        Write-Host "Trying to install $($tool.Name) with pip..."
        pip install $tool.Name.ToLower()
        return
    }

    # Additional fallback mechanisms can be added here for chocolatey, scoop, manual installer, etc.

    Write-Error "$($tool.Name) installation failed on all methods."
}

# Main loop - install tools sequentially or parallel with throttling if desired
foreach ($tool in $tools) {
    Install-Tool -tool $tool
}

```

