



EDIT: 04.02.2026:

I have YARA Rules available for detection, contact me at contact@robin-dost.de if you need them.

► *After publishing this article, I received technical feedback regarding the root cause of CVE-2026-21509. Based on that input, I corrected several parts of the analysis.*

Since the beginning of this year, we have again observed an increased number of attacks by **APT28** targeting various European countries. In multiple campaigns, the group actively leverages the Microsoft Office vulnerability **CVE-2026-21509** as an initial access vector.

This article focuses on how CVE-2026-21509 is used in practice, how relevant IOCs can be extracted efficiently from weaponized Word documents and how the actors own geofencing can be leveraged to infer operational target regions.

Before diving into the analysis, a brief look at CVE-2026-21509 itself.

► **Understanding CVE-2026-21509** (Click)

tldr;

APT28 abuses CVE-2026-21509 by embedding a forgotten OLE browser object (`Shell.Explorer.1`) into RTF documents.

Office happily instantiates it, the object navigates to a remote `.lnk`, and that's your execution path. An allowlist gap that somehow survived for years.

The documents themselves contain no payload.

They only exist to get Office into a state where external shortcut files can be fetched. From there, the real infection chain starts.

Analyzed Samples

For this analysis, I looked at the following samples:

Hash	Download
c91183175ce77360006f964841eb4048cf37cb82103f2573e262927be4c7607f	Download
5a17cfaea0cc3a82242fdd11b53140c0b56256d769b07c33757d61e0a0a6ec02	Download
b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546	Download
fd3f13db41cd5b442fa26ba8bc0e9703ed243b3516374e3ef89be71cbf07436b	Download
969d2776df0674a1cca0f74c2fccbc43802b4f2b62ecccecc26ed538e9565eae	Download

When I receive potentially malicious Word documents, my first step is usually to run `oleid`. In most common malicious documents, this already reveals macros, external references or other active content.

In this case, `oleid` reports a clean file. No macros, no external relationships, no obvious indicators.

Indicator	Value	Risk	Description
File format	Rich Text Format	info	
Container format	RTF	info	Container type
Encrypted	False	none	The file is not encrypted
VBA Macros	No	none	RTF files cannot contain VBA macros
XLM Macros	No	none	RTF files cannot contain XLM macros
External Relationships	0	none	External relationships such as remote templates, remote OLE objects, etc

This is expected.

The document is not a classic OLE container but an RTF file. In RTF, embedded objects are stored as hexadecimal data inside the document body using control words such as `\object` and `\objdata`. These objects do not exist as real OLE structures until Word parses the document and reconstructs them in memory.

`oleid` operates at the container level. It can only detect features that already exist as structured objects in the file. Since the embedded OLE data is still plain text at this stage, there is nothing for `oleid` to flag.

The exploit surface of CVE-2026-21509 only becomes visible after this reconstruction step. Tools like `rtfobj` replicate this part of WordS parsing logic by extracting and rebuilding the embedded objects from the RTF stream.

```
rtfobj -s all b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc
```

```

=====
File: 'b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc' - size: 1831352 bytes
-----+-----+-----
id |index      |OLE Object
-----+-----+-----
0  |001AC71Bh  |Not a well-formed OLE object
-----+-----+-----
1  |001ADD69h  |format_id: 2 (Embedded)
   |           |class name:
   |           |b'file\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
   |           |data size: 2560
   |           |MD5 = '1751572fbf7dd8f50dce6f1c0702761f'
   |           |CLSID: EAB22AC3-30C1-11CF-A7EB-0000C05BAE0B
   |           |unknown CLSID (please report at
   |           |https://github.com/decalage2/oletools/issues)
-----+-----+-----
2  |001AF30Ch  |format_id: 2 (Embedded)
   |           |class name:
   |           |b'file\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
   |           |data size: 2560
   |           |MD5 = 'b52524715e989b23ec39846956535230'
   |           |CLSID: EAB22AC3-30C1-11CF-A7EB-0000C05BAE0B
   |           |unknown CLSID (please report at
   |           |https://github.com/decalage2/oletools/issues)
-----+-----+-----
3  |001B08C2h  |format_id: 2 (Embedded)
   |           |class name: b'Word.Document.12'
   |           |data size: 25088
   |           |MD5 = 'cad4f8ce48d31d6c10253ddb00a993'
   |           |CLSID: F4754C9B-64F5-4B40-8AF4-679732AC0607
   |           |Microsoft Word Document (Word.Document.12)
-----+-----+-----
Saving raw data in object #0:
  saving object to file b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc_object_001AC71B.raw
  md5 f989477552d2a42f0e17d0f13a4b159d
Saving file embedded in OLE object #1:
  format_id = 2
  class name = b'file\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
  data size = 2560
  saving to file b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc_object_001ADD69.bin
  md5 1751572fbf7dd8f50dce6f1c0702761f
Saving file embedded in OLE object #2:
  format_id = 2
  class name = b'file\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
  data size = 2560
  saving to file b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc_object_001AF30C.bin
  md5 b52524715e989b23ec39846956535230
Saving file embedded in OLE object #3:
  format_id = 2
  class name = b'Word.Document.12'
  data size = 25088
  saving to file b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.doc_object_001B08C2.doc
  md5 cad4f8ce48d31d6c10253ddb00a993

```

Once reconstructed, the embedded objects resolve to **Shell.Explorer.1**. Some tools flag the CLSID as unknown, but Windows loads it normally. The containers themselves are valid OLE objects. The vulnerability is triggered solely because this specific ProgID is still allowed.

After extracting the embedded objects, I inspected the resulting files using **xxd**. At this stage, **strings** did not yield anything particularly useful, which is not surprising given that the document is not designed to carry a readable payload.

Why file:///.../davwwwroot/...lnk appears in these samples

Some in-the-wild samples use paths like:

```
file://wellnessmedcare.org/davwwwroot/pol/Downloads/document.lnk
```

This is more an operational choice, then a technical requirement of CVE-2026-21509.

The same behavior can be triggered using plain HTTP or HTTPS URLs.

The exploit primitive is simple: the embedded `Shell.Explorer.1` object calls `Navigate()` to a remote URI.

What happens next is handled by the legacy Internet Explorer engine (`ieframe.dll`), which does not implement modern protections such as SmartScreen or Smart Application Control.

WebDAV mainly provides delivery convenience.

It exposes remote files as filesystem-like objects via the Windows WebClient service, but it does not change the exploit mechanics.

As already mentioned, the Word document itself contains no payload and performs no execution. Its only purpose is to instantiate `Shell.Explorer.1` and trigger navigation to a remote shortcut file.

The `.lnk` becomes the actual execution vector. When accessed, the user is prompted to open or save the file, and any follow-on activity happens outside the document.

The query parameter is client-side only and used to avoid caching. It has no functional relevance for the server.

Identifying Targets

While analyzing the documents and extracted URLs, it became apparent that they reference potential target regions:

- `/cz/` -> Czech Republic
- `/buch/` -> Bucharest / Romania
- `/pol/` -> Poland

Additional indicators inside the Word documents further support this assessment:

Bună ziua,

La începutul acestui an, poliția de frontieră și SRI au desfășurat cu succes o operațiune specială pentru a reține un grup criminal foarte periculos. Ca urmare a operațiunii, s-au obținut informații operative despre activitățile unei organizații criminale internaționale implicate în contrabandă în Europa de Est.

Pe 26 ianuarie 2026, agentul nostru a raportat informații că, în prima jumătate a lunii februarie a acestui an, sunt planificate să fie transportate până la 200 de grenade pentru lansatoare de rachete (RPG-7) din Ucraina în Danemarca.

Destinația finală este necunoscută.

Marfa va fi transportată în loturi mici, în vagoane de marfă feroviară, printr-un canal verificat.

Traseul și datele sunt în curs de clarificare.

Se știe că este implicat în această livrare un cetățean ucrainean, [REDACTED], născut în [REDACTED]

Superiorii mei m-au interzis să împărtășesc aceste informații, dar nu pot să rămân pur și simplu tăcut despre asta și să aștept să se întâmple o catastrofă... Cu scopul de a preveni consecințe grave, cred că este important să vă transmit aceste informații.

Cu respect,

Ofiter al Poliției de Frontieră Române
[REDACTED]

Good afternoon,

Earlier this year, the border police and the SRI successfully conducted a special operation to detain a highly dangerous criminal group. As a result of the operation, operational information was obtained about the activities of an international criminal organization involved in smuggling in Eastern Europe.

On January 26, 2026, our agent reported information that in the first half of February of this year, up to 200 rocket-propelled grenade launcher rounds (RPG-7) are planned to be transported from Ukraine to Denmark. The final destination is unknown.

The cargo will be transported in small batches in freight railway cars via a verified channel. The route and dates are being clarified.

It is known that a Ukrainian citizen [REDACTED], born in [REDACTED] is involved in this delivery.

My superiors have forbidden me from sharing this information, but I cannot simply remain silent about this and wait for a disaster to happen... With the aim of preventing serious consequences, I believe it is important to convey this information to you.

Respectfully,

Officer of the Romanian Border Police
[REDACTED]

Consultation Topics
EU / COREPER-Aligned Discussion Points on the Situation in
Ukraine

Topic
Overall Political and Security Developments
Military Situation and Defence Capabilities
EU Military and Security Assistance
Prospects for Negotiations and Diplomatic Pathways
Energy Situation and Energy Security
Economic Stability and Macro-Financial Assistance
Sanctions Policy and Effectiveness
Humanitarian Situation and Civilian Protection
Reconstruction and Recovery
EU Enlargement and Ukraine's European Perspective
Strategic Communication and Countering Disinformation
Role of the EU and Transatlantic Coordination

- Romanian language content
- References to Ukraine
- Mentions of Slovenia
- EU-related context

None of this is accidental.

At this point, the next step is validation. Russian threat actors are known to rely heavily on geofencing and APT28 is no exception. Fortunately, this behavior can be turned into a useful source of intelligence for us ^-^

Turning Geofencing into Intelligence

The first step was to take a closer look at the domains extracted from the samples:

wellnessmedcare.org 193.187.148.169

freefoodaid.com 159.253.120.2

% Abuse contact for '193.187.148.0 - 193.187.148.255' is 'abuse@m247.ro'

inetnum: 193.187.148.0 - 193.187.148.255
netname: RO-M247RO-20181127
country: DE
descr: M247 LTD Frankfurt Infrastructure
org: ORG-MES6-RIPE
geoloc: 50.1202 8.7364
admin-c: GBXS-RIPE
tech-c: GBXS-RIPE
status: ALLOCATED PA
mnt-by: M247-EU-MNT
mnt-by: RIPE-NCC-HM-MNT
remarks: Geofeed found at <https://geoip.m247.ro/geofeeds.csv>
remarks: ----- LEGAL CONCERNS -----
remarks: For any legal requests, please send an email
remarks: to ro-legal@m247.com for a maximum 48hours response.
remarks: ----- LEGAL CONCERNS -----
created: 2025-04-08T14:21:17Z
last-modified: 2025-04-28T10:03:08Z
source: RIPE

organisation: ORG-MES6-RIPE
org-name: M247 Europe SRL
country: RO
org-type: LIR
address: 11B Sos. Fabrica de Glucoza, 1st floor District 2
address: 020331
address: Bucharest
address: ROMANIA
phone: +4 031 080 0700
abuse-c: ME5262-RIPE
mnt-ref: RIPE-NCC-HM-MNT
mnt-ref: M247-EU-MNT
mnt-by: RIPE-NCC-HM-MNT
mnt-by: M247-EU-MNT
created: 2014-01-06T11:23:03Z
last-modified: 2021-08-18T08:33:50Z
source: RIPE # Filtered

role: GLOBALAXS NOC
remarks: M247 - Network Management Centre
address: 1 Ball Green, Cobra Court
address: M32 0QT, Manchester - United Kingdom
tech-c: JB3482-RIPE

```

inetnum:      159.253.120.0 - 159.253.120.127
org:          ORG-AS895-RIPE
netname:      AlexHost
country:      MD
admin-c:      SZ3268-RIPE
tech-c:       SZ3268-RIPE
status:       ASSIGNED PA
mnt-by:       IPSMAIN
created:      2022-09-20T13:11:42Z
last-modified: 2022-09-20T13:11:42Z
source:       RIPE
mnt-domains:  IPSMAIN
mnt-domains:  CLOUDATAMD-MNT
mnt-lower:    CLOUDATAMD-MNT
mnt-routes:   CLOUDATAMD-MNT
mnt-routes:   IPSMAIN

organisation: ORG-AS895-RIPE
org-name:     ALEXHOST SRL
org-type:     OTHER
address:      str. C. Brancusi nr. 3, Chisinau, Moldova
abuse-c:      AR18916-RIPE
mnt-ref:      FREENET-MNT
mnt-ref:      IPSMAIN
mnt-ref:      FIRSTDC-MNT
mnt-by:       IPSMAIN
created:      2021-02-08T19:58:24Z
last-modified: 2025-09-25T13:06:05Z
source:       RIPE # Filtered

person:       AlexHost SRL
address:      str. Constantin Brancusi nr. 3, Chisinau, Moldova
phone:        +37379600002
nic-hdl:      SZ3268-RIPE
mnt-by:       CLOUDATAMD-MNT
created:      2014-03-21T14:17:01Z
last-modified: 2023-03-03T08:12:53Z
source:       RIPE # Filtered

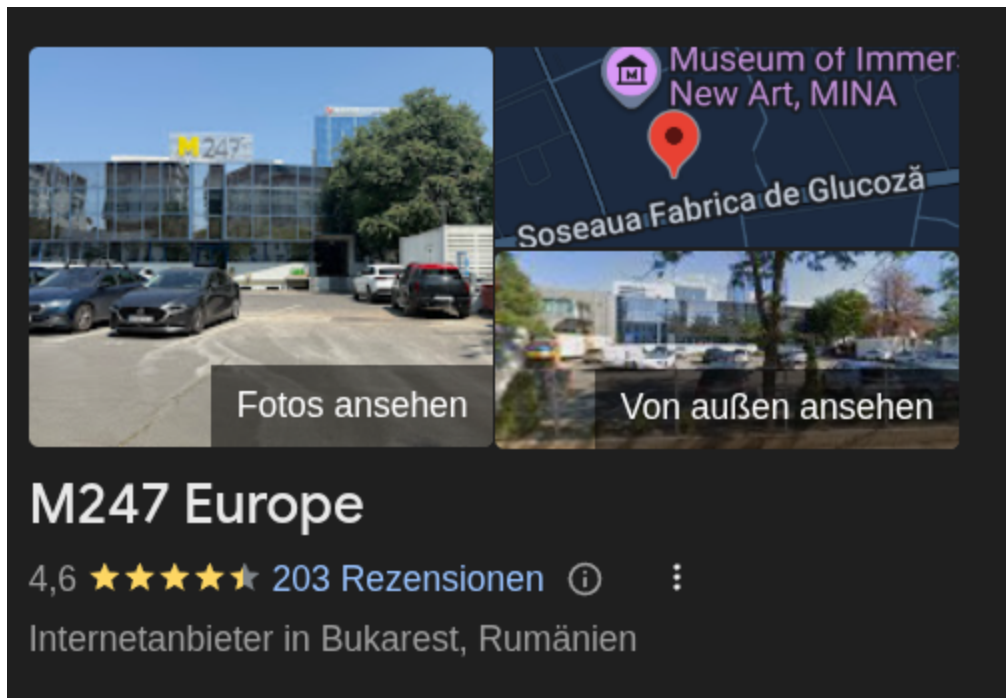
% Information related to '159.253.120.0/24AS200019'

route:        159.253.120.0/24
origin:       AS200019
mnt-by:       VPLAB-MNT
created:      2022-09-17T11:24:32Z
last-modified: 2022-09-17T11:24:32Z
source:       RIPE

```

What stands out here is the choice of hosting locations.

Both IP addresses resolve to providers in Romania and Moldova. It is reasonable to assume that these locations were selected based on the campaigns intended target regions.



AlexHost

If you are looking for a convenient, anonymous and high-quality hosting service, then alexhost.com

IT-Dienstleistungen und IT-Beratung · Chisinau, Chisinau · 348 Follower:innen · 2-10 Beschäftigte

Next, I attempted to replicate the WebDAV requests generated by Windows in order to test the observed geofencing behavior.

To do this, I executed the document in a sandbox and captured the resulting network traffic.

Endpoint	Request	URL	Data
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: DavCint translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:443 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/cz	OPTIONS /cz HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:443 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/cz	OPTIONS /cz HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:443 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/cz	OPTIONS /cz HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:443 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/cz	OPTIONS /cz HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details
193.187.148.169:80 (wellnessmedcare.org)	⚠️ OPTIONS	wellnessmedcare.org/	OPTIONS / HTTP/1.1 Connection: Keep-Alive User-Agent: Microsoft-WebDAV-MiniRedir/10.0.16299 translate: f Host: wellnessmedcare.org 403 Forbidden More Details

Geofence Analysis

To validate the geofencing, I needed to determine which proxy locations were required to access the malicious resources without being blocked.

After identifying suitable proxies, I performed test requests using a custom script, once without a proxy and once using a Romanian proxy.

Without proxy:

```

02.02.2026 python3 download2.py

=== OPTIONS http://wellnessmedcare.org:80/ ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

=== OPTIONS http://wellnessmedcare.org:80/ ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

=== OPTIONS https://wellnessmedcare.org:443/cz ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

=== GET https://wellnessmedcare.org:443/cz/Downloads/document.Lnk?init=1 ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

=== GET https://wellnessmedcare.org:443/davwwwroot/cz/Downloads/document.Lnk?init=1 ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

```

With proxy:

```

02.02.2026 python3 download2.py --proxy "http://
edcare.org --outfile document.Lnk" --host wellnessmedcare.org

=== OPTIONS http://wellnessmedcare.org:80/ ===
HTTP/1.1 200 OK
Server: Apache
DAV: 1,2
DAV: <http://apache.org/dav/propset/fs/1>
Allow: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK
Content-Type: httpd/unix-directory

=== OPTIONS http://wellnessmedcare.org:80/ ===
HTTP/1.1 200 OK
Server: Apache
DAV: 1,2
DAV: <http://apache.org/dav/propset/fs/1>
Allow: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK
Content-Type: httpd/unix-directory

=== OPTIONS https://wellnessmedcare.org:443/cz ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

=== GET https://wellnessmedcare.org:443/cz/Downloads/document.Lnk?init=1 ===
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1

```

The result is fairly clear. Requests originating from outside the expected regions are rejected with HTTP 403, while requests routed through a Romanian proxy succeed. This pattern can be used to validate likely operational target regions.

```
[X] Seychelles      -> 403 Forbidden
[X] Singapore      -> 403 Forbidden
[X] Slovakia        -> 403 Forbidden
[X] South Africa    -> 403 Forbidden
[X] South Korea     -> 403 Forbidden
[X] Spain           -> 403 Forbidden
[X] Sweden          -> 403 Forbidden
[X] Switzerland     -> 403 Forbidden
[X] Syria           -> 403 Forbidden
[X] Taiwan          -> 403 Forbidden
[X] Tajikistan      -> 403 Forbidden
[X] Thailand        -> 403 Forbidden
[X] Togo            -> 403 Forbidden
[X] Tunisia         -> 403 Forbidden
[X] Turkey          -> 403 Forbidden
[X] Uganda          -> 403 Forbidden
[X] Ukraine         -> 403 Forbidden
[X] United Arab Emirates -> 403 Forbidden
[X] United Kingdom  -> 403 Forbidden
[X] United States   -> 403 Forbidden
[X] Uruguay         -> 403 Forbidden
[X] Uzbekistan      -> 403 Forbidden
[X] Venezuela       -> 403 Forbidden
[X] Vietnam         -> 403 Forbidden
```

```
=====
SCAN SUMMARY
=====
```

```
Scanned countries: 114
```

```
Successful access:
```

- Czech Republic
- Poland
- Romania

Out of 114 tested countries, only three were allowed access: **Czech Republic**, **Poland** and **Romania**. This aligns perfectly with the indicators observed earlier in the documents and URLs.

As this example shows, defensive measures such as geofencing can provide valuable intelligence when analyzed properly. Even access control mechanisms can leak information about an actors operational focus if you know where to look.

The second domain, freefoodaid.com, was already offline at the time of analysis. Given how short-lived APT28 infrastructure tends to be, this is hardly surprising. It is reasonable to assume that similar geofencing behavior would have been observable there as well, but for demonstration purposes, the remaining data is more than sufficient.

How to protect against these attacks

Update Microsoft Office and enforce a structured update routine.

Treat unexpected Word documents as untrusted and have them analyzed before opening them.

(or stop using windows :3)

Conclusion

CVE-2026-21509 works because it fits neatly into how Office processes documents today.

The exploit relies on Office instantiating an allowed OLE object during normal parsing, not on macros or embedded payloads, which makes it easy to overlook during initial analysis.

The tradecraft follows a familiar pattern.

Remote shortcut files and strict geofencing have been used by APT28 before and continue to show up in current campaigns.

WebDAV appears here mainly as a delivery detail.

The technique is stable, requires little user interaction, and sidesteps many modern Office protections by falling back to legacy browser behavior.

At the same time, this setup exposes useful signals.

Geofencing decisions, hosting locations and access behavior provide insight into intended target regions when tested systematically. In this case, the infrastructure behavior aligns closely with the indicators found inside the documents.

From an analytical POV, the value lies less in the exploit itself and more in what can be inferred from how it is deployed and constrained.