

APT28 : 地理围栏作为目标信号 (CVE-2026-21509 攻击活动)

 ctfiot.com/302421.html



APT28 利用 CVE-2026-21509 漏洞，将一个被遗忘的 OLE 浏览器对象嵌入 Shell.Explorer.1 到 RTF 文档中。Office 会正确实例化该对象，该对象会导航到远程服务器 .lnk，这就是攻击者的执行路径。

这是一个不知何故存在了多年的白名单漏洞。

这些文档本身不包含任何有效载荷。

它们存在的唯一目的是让 Office 进入可以获取外部快捷方式文件的状态。真正的感染链由此开始。

在大多数常见的恶意文档中，这已经可以检测出宏、外部引用或其他活动内容。

在这种情况下，oleid 报告显示文件干净。没有宏，没有外部关联，也没有明显的异常迹象。

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

该文档并非传统的 OLE 容器，而是 RTF 文件。在 RTF 中，嵌入对象以十六进制数据的形式存储在文档主体中，并使用诸如 `

oleid它运行在容器级别，只能检测文件中已作为结构化对象存在的特征。由于此时嵌入的 OLE 数据仍为纯文本，因此无法oleid标记任何内容。

CVE-2026-21509 的漏洞利用面只有在完成这一重构步骤后才会显现。诸如此类的工具rtfobj通过从 RTF 流中提取并重建嵌入对象来复制 WordS 解析逻辑的这一部分。

rtfobj-sa11b2ba51b4491da8604ff9410d6e004971e3cd9a321390d0258e294ac42010b546.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\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\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\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\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

```

重构后，嵌入对象解析为Shell.Explorer.1。某些工具会将 CLSID 标记为未知，但 Windows 可以正常加载它。容器本身是有效的 OLE 对象。此漏洞的触发仅仅是因为此特定 ProgID 仍然被允许。

提取嵌入对象后，我使用检查了生成的文件xxd。在这个阶段，strings并没有发现任何特别有用的信息，考虑到该文档并非设计用于携带可读有效载荷，这并不奇怪。

根据这些数据，可以提取出以下字符串：

```
file://wellnessmedcare.org/davwwwroot/pol/Downloads/document.Lnk?  
init=1file://wellnessmedcare.org/buch/pol/Downloads/document.Lnk?  
init=1file://wellnessmedcare.org@ssl/cz/Downloads/document.Lnk?  
init=1file://freefoodaid.com@80/documents/2_2.lnk?init=1
```

为什么这些样本中会出现 `file://.../davwwwroot/...lnk`

一些实际使用的样本路径如下：

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

这更多的是一种操作选择，而非 CVE-2026-21509 的技术要求。

使用普通的 HTTP 或 HTTPS URL 也能触发相同的行为。

利用方式很简单：嵌入 `Shell.Explorer.1` 对象调用 `Navigate()` 远程 URI。

接下来的操作由旧版 Internet Explorer 引擎处理 `ieframe.dll`，该引擎没有实现 `SmartScreen` 或 `Smart Application Control` 等现代保护机制。WebDAV

主要提供传输便利。

它通过 Windows WebClient 服务将远程文件作为类似文件系统的对象公开，但不会改变利用机制。

如前所述，Word 文档本身不包含有效载荷，也不会执行任何操作。

它的唯一目的是实例化 `Shell.Explorer.1` 并触发对远程快捷方式文件的导航。

该快捷.lnk方式文件成为实际的执行向量。当用户访问该文件时，系统会提示用户打开或保存，任何后续操作都在文档之外进行。

查询参数仅在客户端有效，用于避免缓存。它对服务器端没有任何功能意义。

确定目标

在分析文档和提取的URL时，我们发现它们指向潜在的目标区域：

```
/cz/捷克共和国/buch/-> 布加勒斯特 / 罗马尼亚/pol/波兰
```

Word文档中其他指标也进一步佐证了这一评估：

罗马尼亚语内容提及乌克兰提及斯洛文尼亚欧盟相关背景

接下来就是验证了。众所周知，俄罗斯威胁行为者非常依赖地理围栏技术，APT28 也不例外。幸运的是，这种行为可以转化为我们有用的情报来源。

将地理围栏转化为智能

第一步是仔细研究从样本中提取出的结构域：

```
wellnessmedcare.org193.187.148.169freefoodaid.com159.253.120.2
```

值得注意的是服务器托管位置的选择。

这两个IP地址都解析到罗马尼亚和摩尔多瓦的服务器提供商。可以合理推断，选择这些位置是基于广告活动的目标区域。

复现 Windows 生成的 WebDAV 请求，以测试观察到的地理围栏行为。

为此，我在沙箱环境中执行了该文档，并捕获了由此产生的网络流量。

地理围栏分析

为了验证地理围栏的有效性，我需要确定访问恶意资源而不被屏蔽所需的代理服务器位置。

在找到合适的代理服务器后，我使用自定义脚本进行了测试请求，一次不使用代理服务器，一次使用罗马尼亚代理服务器。

无需代理

使用代理

结果非常明确。来自预期区域之外的请求会被拒绝，返回 HTTP 403 错误，而通过罗马尼亚代理服务器路由的请求则会成功。这种模式可用于验证可能的运营目标区域。

在接受测试的114个国家中，只有三个国家被允许访问：捷克共和国、波兰和罗马尼亚。这与之前在文档和URL中观察到的指标完全吻合。

正如这个例子所示，地理围栏等防御措施如果分析得当，可以提供宝贵的情报。即使是访问控制机制，只要掌握了方法，也能泄露攻击者的行动重点。

第二个域名freefoodaid.com在分析时已经离线。考虑到APT28基础设施的生命周期通常很短，这并不令人意外。可以合理推断，类似的地理围栏行为也应该在那里出现，但就演示而言，剩余的数据已经足够了。

结论

CVE-2026-21509 之所以有效，是因为它巧妙地契合了 Office 目前处理文档的方式。

该漏洞利用依赖于 Office 在正常解析过程中实例化一个允许的 OLE 对象，而非宏或嵌入式有效载荷，因此在初始分析阶段很容易被忽略。

远程快捷方式文件和严格的地理围栏技术此前已被 APT28 使用，并且在当前的攻击活动中也屡见不鲜。WebDAV

该技术稳定可靠，几乎不需要用户交互，并且通过回退到旧版浏览器的行为来绕过许多现代 Office 的安全防护措施。

通过系统地测试地理围栏决策、托管位置和访问行为，可以深入了解攻击者的目标区域。

| 本篇文章来源于微信公众号: 威胁情报Z分析