Bandook: Signed & Delivered

research.checkpoint.com/2020/bandook-signed-delivered

November 26, 2020

November 26, 2020

Introduction

Check Point Research recently observed a new wave of campaigns against various targets worldwide that utilizes a strain of a 13-year old backdoor Trojan named Bandook.

Bandook, which had almost disappeared from the threat landscape, was featured in 2015 and 2017 campaigns, dubbed “Operation Manul” and “Dark Caracal“, respectively. These campaigns were presumed to be carried out by the Kazakh and the Lebanese governments, as uncovered by the Electronic Frontier Foundation (EFF) and Lookout.

During this past year, dozens of digitally signed variants of this once commodity malware started to reappear in the threat landscape, reigniting interest in this old malware family.

In the latest wave of attacks, we once again identified an unusually large variety of targeted sectors and locations. This further reinforces a previous hypothesis that the malware is not developed in-house and used by a single entity, but is part of an offensive infrastructure sold by a third party to governments and threat actors worldwide, to facilitate offensive cyber operations.

In this publication, we showcase the latest evolution of the infection chain offered by this unknown third-party, compare the different Bandook variants, and share the various techniques its creators use to hinder analysis and detection of all the components in the attack flow.

Infection Chain

As the infection chain is constantly evolving, we describe the one used by the attackers from as early as July, to the present day.

The full infection chain of the attack can be broken down into three main stages. The first stage starts, as in many other infection chains, with a malicious Microsoft Word document delivered inside a ZIP file. Once the document is opened, malicious macros are downloaded using the external template feature. The macros’ code in turn drops and executes the second stage of the attack, a PowerShell script encrypted inside the original Word document. Finally, the PowerShell script downloads and executes the last stage of the infection: the Bandook backdoor.

💡 The names of the various artifacts described below may vary from one infection to the next.

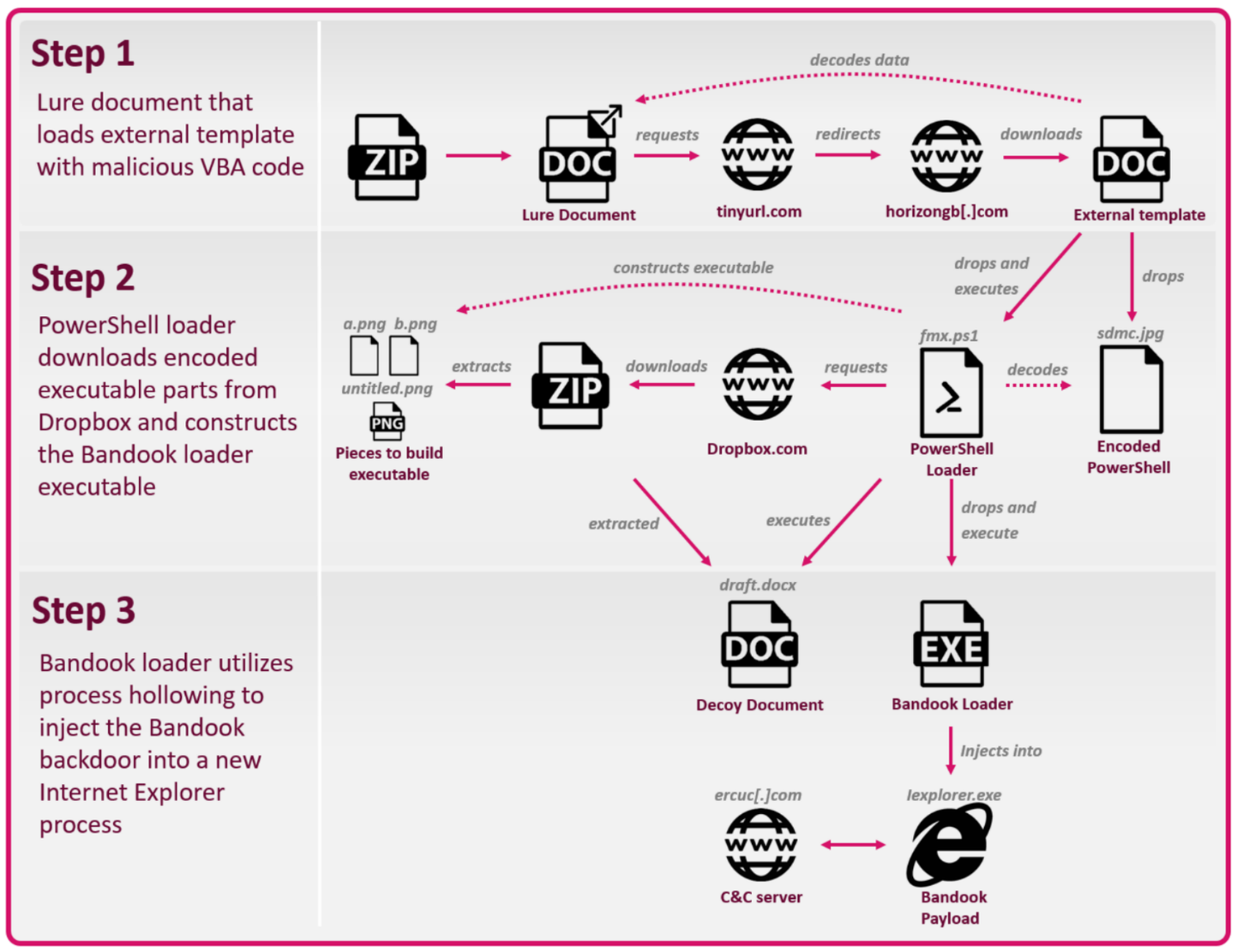

Figure 1: Full infection chain.

First Stage – Lure Documents

The first stage starts with a Microsoft Word document with embedded encrypted malicious script data and an external template that points to a document containing malicious VBA macros.

The external template is downloaded via a URL shortening web service like **TinyURL** or **Bitly**, which redirects to another domain controlled by the attacker.

The external template document contains a VBA code that runs automatically, decrypts the embedded data from the original lure document, and drops the decoded data into two files in the local user folder: `fmx.ps1` (the next stage PowerShell) and `sdmc.jpg` (base64 encoded PowerShell code).

To allow this behavior, the attackers use a combination of two techniques: encrypted data is embedded inside a shape object within the original document (hidden from view by a small font size and white foreground), and is accessed from the external template code by using the following code:

```
o4QQLW7zXjLbj = ActiveDocument.Shapes(1).TextFrame.TextRange.Text
```

For proper analysis, both the original document and the external template must be located, which makes things a bit more difficult for investigators.

We observed and analyzed multiple pairs of documents and external templates. Different lure images were used, alongside different encryption keys.

Examples of lure documents:

Figure 2: Lure documents used to convince the user to enable the macros.

Examples of external templates with macros:

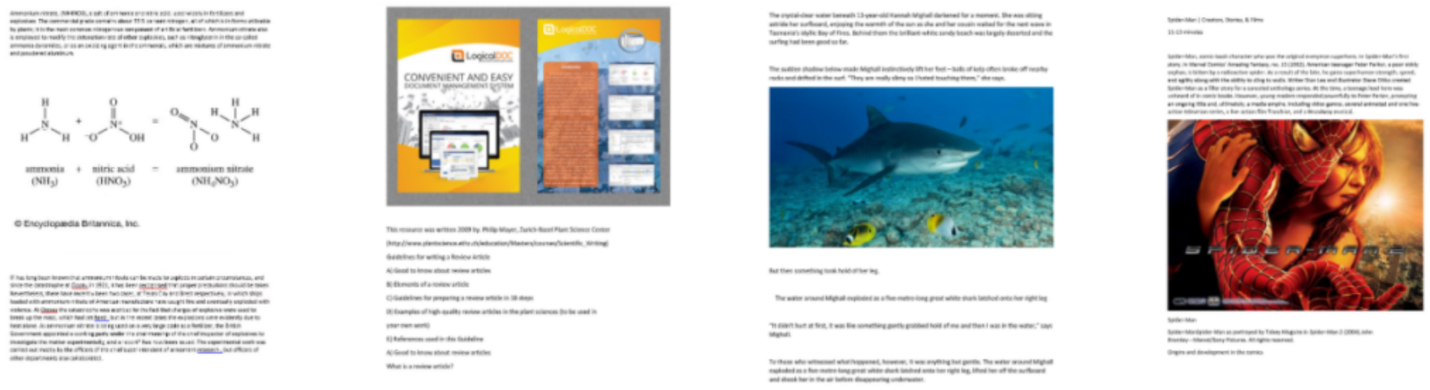

Figure 3: External templates containing malicious macros.

The external templates are not visible to the victim. Their only purpose is to provide malicious macros.

Interestingly, with each attack, after a certain amount of time, the attacker switched the malicious external template to a benign one, further muddying our analysis of the infection chain.

Here again, the external templates look like random benign documents:

Figure 4: Benign external templates.

The themes of the documents are often of cloud-based services like Office365, OneDrive and Azure that contain images of other documents supposedly available once the victim clicks “Enable Content.”

For example, one of the documents that specifically got our attention depicts an Office365 logo and a preview of a certificate issued by the government of Dubai. **JAFZA – Jebel Ali Free Zone**, featured at the top of the document, is an industrial area surrounding the port of Jebel Ali in Dubai, where more than 7,000 global companies are based.

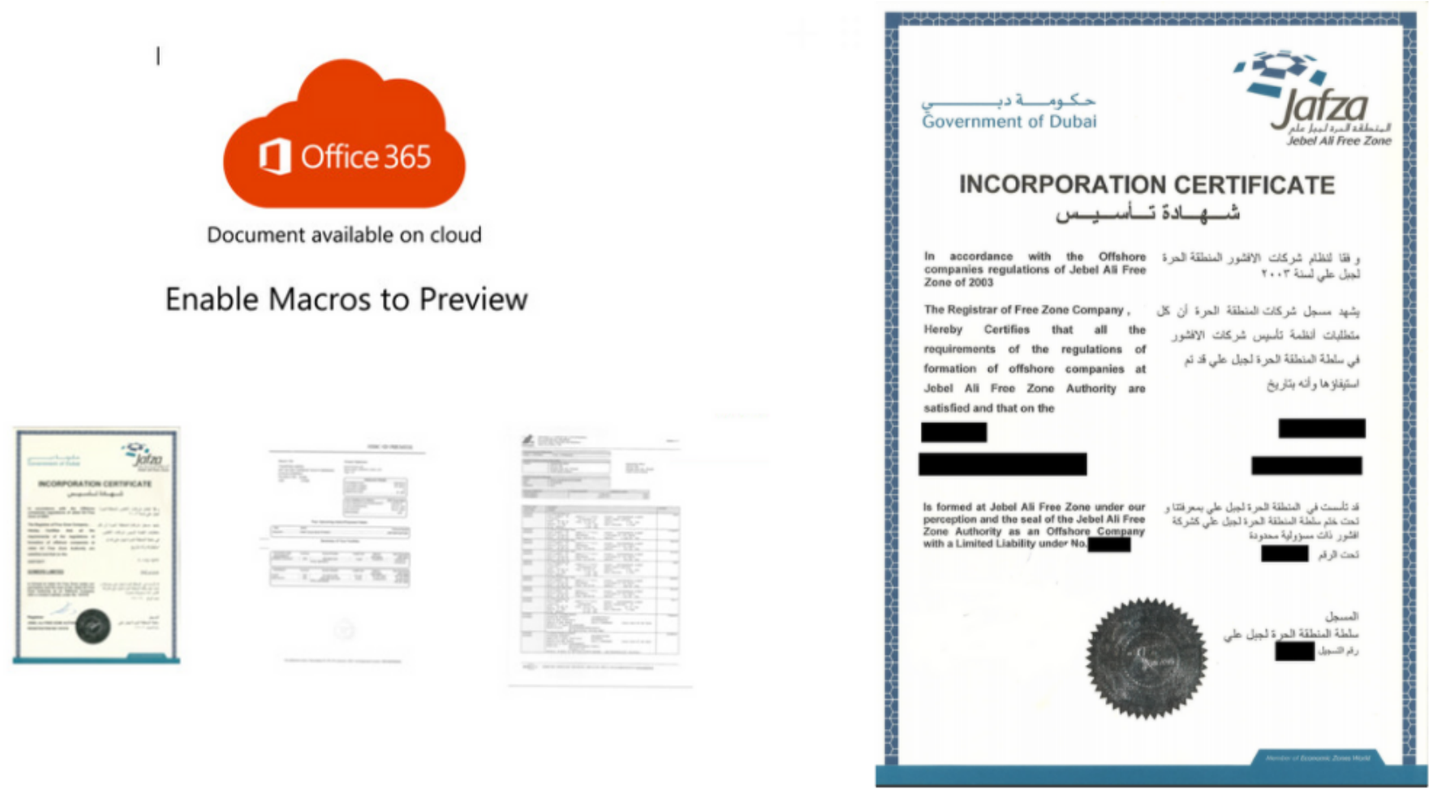

Figure 5: Lure document (left) and an example of a similar publicly available certificate (right).

Sample document file names:

- Malaysia Shipment.docx
- Jakarta Shipment.docx
- malta containers.docx
- Certified documents.docx
- Notarized Documents.docx
- bank statement.docx
- passport and documents.docx
- Case Draft.docx
- documents scan.docx

Second Stage – PowerShell Loader

After the VBA code drops the two files (`fmx.ps1` and `sdmc.jpg`), it invokes `fmx.ps1` .

`fmx.ps1` is a short PowerShell script that decodes and executes a base64 encoded PowerShell stored in the second dropped file (`sdmc.jpg`).

First, the decoded PowerShell script downloads a zip file containing four files from a cloud service such as Dropbox, Bitbucket or an S3 bucket. The zip file is stored in the user's `Public` folder, and the four files are locally extracted.

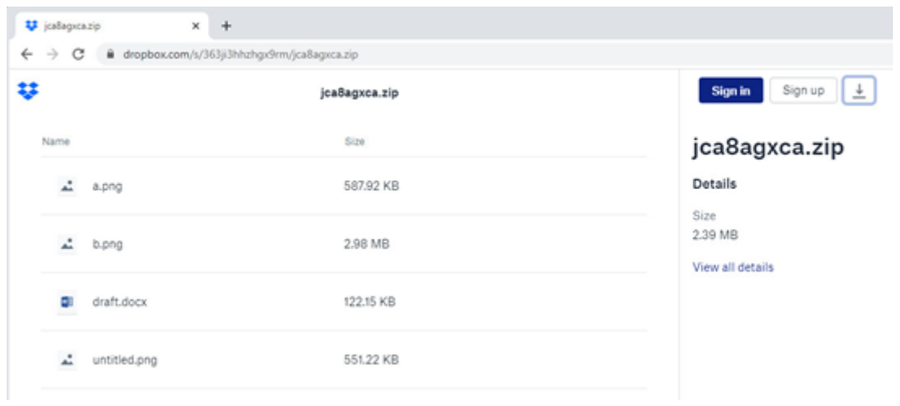

Figure 6: Malware components stored on Dropbox.com.

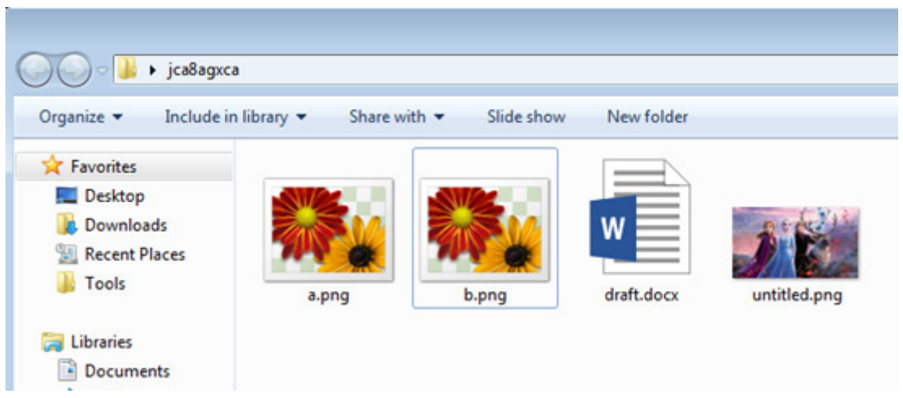

Figure 7: Malware components after being extracted on the victim's device.

Three of the files, `a.png` , `b.png` and `untitled.png` , are used by the PowerShell script to generate the malware payload in the same folder. `untitled.png` , unlike the other two files, is in a valid image format. It contains a hidden RC4 function encoded in the RGB values of the pixels, created using a known tool named invoke-PSImage.

The final executable payload is concatenated from the following files:

- `a.png` – After it is decrypted using RC4 and stored as `aps.png` .
- `b.png` – As is.

Finally, the PowerShell script executes the malware, opens `draft.docx` , and deletes all previous artifacts from the `Public` folder.

`draft.docx` is a benign document whose sole purpose is to convince the victim that the document is no longer available, and that the overall execution was successful.

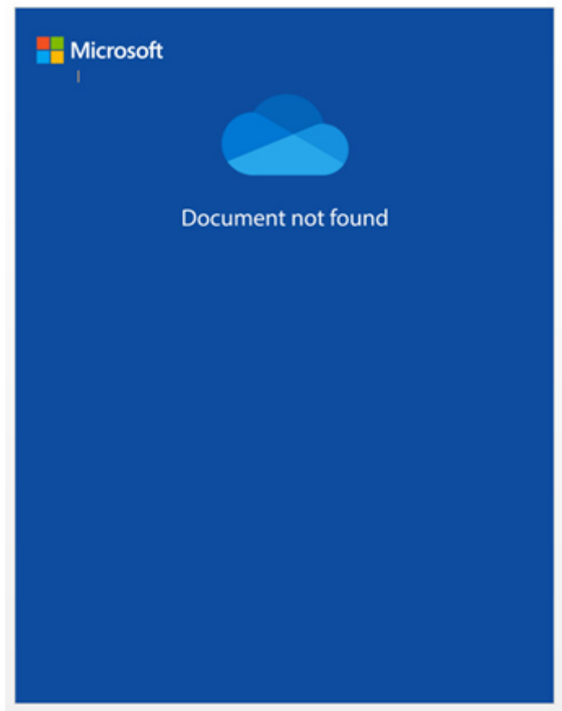

Figure 8: Final document shown to the user post-infection.

Third Stage – The Bandook Loader

The final payload in this infection chain is a variant of an old full-featured RAT named Bandook. Written in both Delphi and C++, Bandook has a long history, starting in 2007 as a commercially available RAT that was developed by a Lebanese individual nicknamed PrinceAli. Over time, several variants of the malware builder were leaked to the Web, and the malware became publicly available for download.

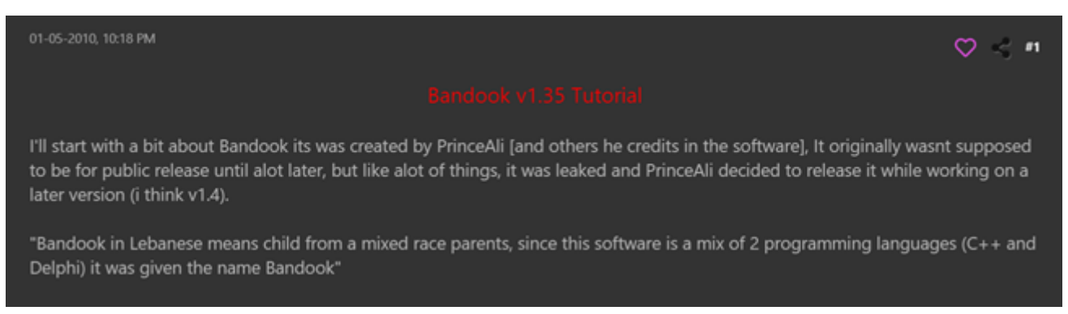

Figure 9: Bandook's history described on a hacking forum.

Bandook's execution flow starts with a loader, written in Delphi, that uses the Process Hollowing technique to create a new instance of an Internet Explorer process and inject a malicious payload into it. The payload contacts the C&C sever, sends basic information about the infected machine, and waits for additional commands from the server.

The variant of the Bandook malware we observed in this attack was not one of the variants whose builder was previously leaked to the Web (which supported a range of more than 100 commands).

In this attack, the threat actor utilized a custom, slimmed-down version of the malware with only 11 supported commands, including:

- File operations
- Taking screenshots
- File download
- File upload
- File execution

💡 For a full list of commands and their corresponding request codes, see Appendix A.

In this version, the communication protocol with the C&C server was also upgraded to use AES encryption.

Bandook variants in the wild

After comparing the Bandook variant we observed in the attack with the ones created by different leaked builders, we began hunting for variants more similar to the ones we observed.

Our search led us to tweets by the MalwareHunterTeam (MHT) from 2019-2020 that mention various Bandook samples — all of them digitally signed with certificates that were issued by **Certum**.

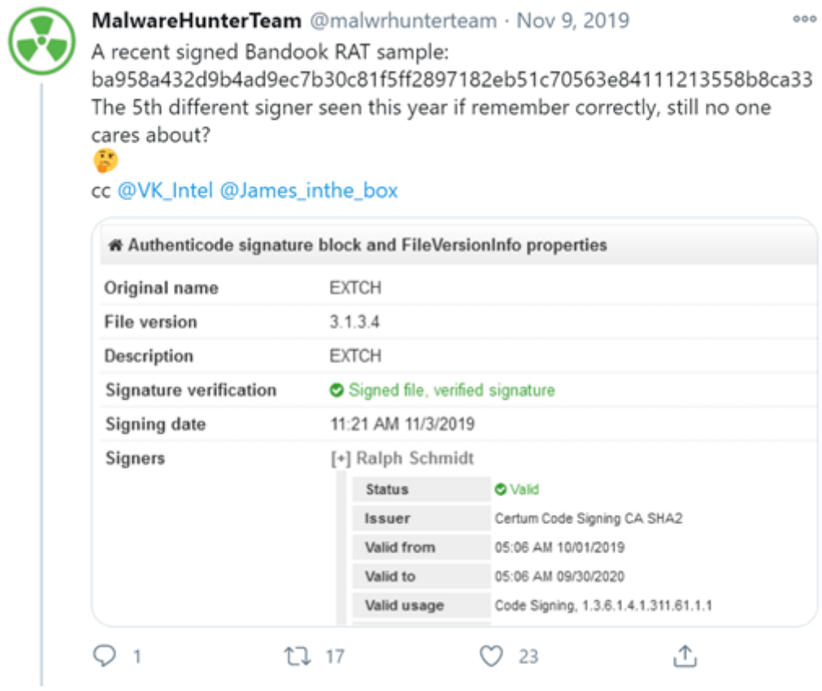

Figure 10: Signed Bandook samples discovered by MHT.

In the newer attack flows we observed, we once again found valid **Certum** certificates were used to sign the Bandook malware executable.

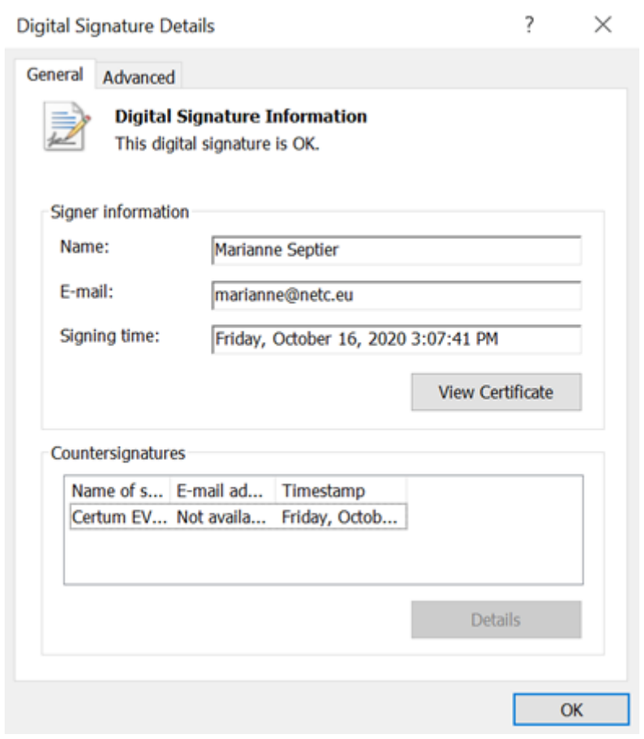

Figure 11: Valid signature information of a newly discovered Bandook sample.

Analyzing all Bandook samples noted by MHT, we discovered that the very first of the samples was compiled in March 2019 and supported around 120 commands. **A sample compiled a few days later – a different signed Bandook variant (with only 11 commands) utilized the very same C&C server.** Since then, all signed samples use only 11 basic commands. The shared C&C provides clear evidence that both the slimmed-down and the fully-fledged variants of the malware are operated by a single attacker.

In addition to the Bandook samples that were reported by MHT, we identified additional samples from the same time period (2019-2020) which were **not digitally signed** and contained about 120 commands. These were the only ITW Bandook samples we were able to locate from this time period.

Several factors led us to believe that these **signed and unsigned** variants are specially crafted Bandook variants, used and developed by the same entity.

- Both use the same domain registration services for their C&C domains: Porkbun or NameSilo.
- They share a similar method of communication, using the **AES** encryption algorithm in **CFB** mode, with a hardcoded IV: 0123456789123456. This feature is not available in the public leaks of this malware.
- They incorporated commands that we did not observe in any other public leak or report. Most notable are the commands to execute **Python** and **Java** payloads.

```
sub_13141940 (
  &Parameters ,
 L "\"%sLib\\dpx.pyc\" \"%ws\" \"%ws\" \"%ws\" \"%ws\" \"%ws\""       ,
  v95 ,
  v93 ,
  v111 ,
  &a68 ,
  lpMultiByteStrb  ,
  retaddr ,
  v100 ,
  v102 ,
  v104 );
sprintf (&a65 , "%spython.exe"  , &MultiByteStr );
v96  = MultiByteToWideChar  (0xFDE9u , 0, &a65 , -1, 0, 0);
v97  = sub_131981C0 (2 * v96  | -(v96  >> 31 != 0));
MultiByteToWideChar  (0xFDE9u , 0, &a65 , -1, v97 , v96 );
v98  = ShellExecuteW (0, L"open" , v97 , &Parameters , 0, 0);
v99  = "1";
if ( !v98 )
  v99  = "0";
sprintf (&a66 , "%s~!%s~!%s~!"  , &unk_13C22CFC  , "@0145" , v99 , v101 , v103 , v105 );
```

Figure 12: Bandook subroutine to execute a precompiled Python from a file named “dpx.pyc”.

At this point, we have three different variants of the malware, which we believe are operated and sold by a single entity, in accordance with their chronological appearance:

1. A full-fledged version with 120 commands (not signed).
2. A full-fledged version (single sample) with 120 commands (signed).
3. A slimmed-down version with 11 commands (signed).

The move to a slimmed-down version with only 11 commands for signed executables may indicate the operators’ desire to reduce the malware’s footprint and maximize their chances for an undetectable campaign against high profile targets (and high paying customers), while continuing the use of the un-signed 120 commands variant for lower profile ones.

Furthermore, such a minimized backdoor might indicate that the slimmed-down variant of Bandook is only utilized as a loader for an additional, more full-featured malware to be downloaded next.

Targeting

As mentioned previously, in this campaign we observed an unusually large variety of targeted sectors and locations. This strengthens a hypothesis made by researchers – that the malware is not being developed and used by a single entity, but an offensive infrastructure is being sold by a third-party, to governments and threat actors worldwide, to facilitate offensive cyber operations.

The different targeted sectors include:

Government, financial, energy, food industry, healthcare, education, IT and legal institutions.

In the following countries:

Singapore, Cyprus, Chile, Italy, USA, Turkey, Switzerland, Indonesia and Germany.

Connection to Dark Caracal

This campaign isn’t the first instance of the Bandook malware incorporated in a targeted attack. As mentioned, in a joint report from Lookout and the EFF, targeted attacks utilizing a Bandook variant, called “Dark Caracal”, were attributed to the Lebanese General Security Directorate.

Some of this campaign’s characteristics and similarities to previous campaigns leads us to believe that the activity we describe in this report is indeed the continuation and evolution of the infrastructure used during the Dark Caracal operation:

- The use of the same certificate provider (**Certum**) throughout the various campaigns.
- The use of the Bandook Trojan, in what appears to be a unique evolving fork from the same source code (which is not known to be publicly available). Samples from the Dark Caracal campaign (2017) utilized around 100 commands, compared to the current 120 command version we analyzed.
- This wave of attacks shares the same anomalous characteristics for targeted attacks – an extreme variance in the selected targets, both in their industry and their geographic spread.

Finally, EFF researchers who first disclosed the Dark Caracal operation also believe that the same attacker “is back at it” again.

Conclusion

All evidence points to our belief that the mysterious operators behind the malicious infrastructure of “Operation Manul” and “Dark Caracal” are still alive and operational, willing to assist in the offensive cyber operations to anyone who is willing to pay.

Although not as capable, nor as practiced in operational security like some other offensive security companies, the group behind the infrastructure in these attacks seems to improve over time, adding several layers of security, valid certificates and other techniques, to hinder detection and analysis of its operations.

Check Point SandBlast Agent protects against this APT attack, and prevents it from the very first steps.

Appendix A: Bandook Commands

Below is a list of commands that the slimmed-down version of Bandook supports.

Command	Functionality
@0001	Download and Execute file via HTTP
@0002	Download and Execute file via raw TCP socket
@0003	Take a screenshot
@0004	List drives
@0005	List files
@0006	Upload file
@0007	Download file
@0008	Shell execute
@0009	Move File
@0010	Delete file
@0011	Get Public IP

Appendix B: Indicators of Compromise

C&Cs from MHT Samples

Domains
ntsclouds[.]com
jtoolbox[.]org
idcmht[.]com
htname[.]info
vdscloud[.]net
mainsrv[.]top
olex[.]live
branchesv[.]com

C&Cs from additional Samples

Domains
s1[.]megawoc[.]com
s2[.]megawoc[.]com
s3[.]megawoc[.]com
s1[.]fikofiko[.]top
s2[.]fikofiko[.]top
s3[.]fikofiko[.]top
d1[.]p2020[.]club
d2[.]p2020[.]club
pronews[.]icu
p2020[.]xyz
2ndprog[.]monster
ercuc[.]com
tancredis[.]com
ec2[.]mbcde[.]net
nopejohn[.]com

External Templates

Domains
horizongb[.]com
styleco[.]me
ewsdocs[.]com
raysdoor[.]com
vsimperial[.]com
mxtms[.]com

Maldocs

MD5	SHA-1	SHA-256
27f8d8bbbeeda5fc439ee18d9d4da343	e78721fd283b0093fb0556167e1b38b81ed0c7bb	1ad83e9d06428dd87203ab8fcc6142014a9c05f3eb9afd61347834f39082d72a
44584c8d010242fddb44afe5ce860872	500813f95615b25f622e82e6c79431d7f4928bc4	74feaf3aa116a88ef3b10453e77feadefbe4e53dd7a71dd3b8309cc9d76cdec9
a6501c62b3a6ffa8d028a88138fe509f	118633bbe46520c65529c0cd1d6eb52f810f6327	034d8ec8d510033c387bb87cac35d240b7b8daa3b5167732118c755c5e6c1d48
7c15ee5b9a12dacaace8fb62271f12f1	154c16ecfa56b71ce7b6f3fca4be4e0820e34665	072c103759968253b7b25837b43eec546c625ae9c04edd52321d848cf6078b87
4e9e12c98cfbc5f3aa3c1345bd063fa0	1e45d9c3ff9bb7e2ed236384694237dcd956de2b	0750c7cdc538d79d9ffed0d37f5d9a083902b49ec02d75ee88028db9f3668b59
7ef261c151519e66ec369c63e4b1aed4	788489ecd0e43f74c7d8df841bee8367cba2164d	40cc5933e608f7a2a5c13af1066257c9e41528bb85e434e2bc3d1f4802dec24d
6effed1b1bb5e9ed6aafacb075c1d4e2	8790c9c1ffdf7ce7d7c1a0825a73ef75958fd9c5	5900abb869c61928f0ef931d6f9d8b62183b2bab9a69b0ef886551005d6c9622
0475771b8bc3efc28b1834f3add608f3	9087c24b181d58bb57d02a1ce19f8d17d63476b4	8cb1f713761a6b31c9c25dd2c7ae11e575a634c9f052cfd598ada35a61783230
045ce6679ed4086e2ded58470e24c15a	9b47ab36a00d83c119620318e4924ce50cec2512	2ee74ae5b202c8aab288ca167c630e9ee3569240958e984474b960cd560bbe95
28ad9ace11919b57bf540e2b9debf8dd	9d0deca8dbdf25bdb9208772f861c28aa5a4e95f	d217288a046e2739159d0081608a44c2e79d41de12c57ebe88a8591693fa15d5
5a3f7c46748791494e29383d1f58a908	a8f2b1f1a1200d25d751b5559a31c034781ea33e	a9a8b0aa5f137e7353db62dc1609da3c709ca30287a5605c73aafaf4968d1e8d
07776b2dd00bcd0be1c7713c37d41120	b0d64b13407da0c6db1aced2d9e110802c05a6d3	6287fc617ff6881169990e6b877c16d8ca3c199f7e453241a0b18a7907c67ab0
d22b31848b6f17efc87d538dede2f2a7	c4d7332f09c7e917d5dd56930854e735cfce45fa	9de287f9af63f02c51c69d9c8480fee2bd4d4bd3c818f2ba81324b1f8ce495c0
b9b8d6f46ff3a9058ffe4b304604b4e7	cf96410d1cde40aaebfe32f26354282a7773abaf	306238a63896fa8b79b4c9a6d25fd906bb9e4919bc698608ab970677d15b0694
573c7dbf4d3aed421bff58df770610fe	d6b69cb3689341997d93b03fbd4499fe60f29b35	1b0d2d096c5f7fff02a5a4ce623b71b862f63e306a0760722f710c425b4e16ec
f037f3961f7d9fe1eb7afa889b556cb1	49a8149054440e33a6228b42f731e2f8035049e5	9a0ee2430f7c77942d544dad6787ca8a94470f6555f1cb08baa9d099c92f8447

Bandook

Slim Version

MD5	SHA-1	SHA-256
1a3889ded73044f8ba0a00c2f089a3bd	03508cbeec86346d6658da8c9d34638c57dad920	766917fe9b543bf218bd824d55967d63f94b28456f1d4919bc990d8262dc608d
70ff19341dee7973ea6dd8e15c6ba86f	9e33cbc25a8ad9987f88d5e1d181098142579f54	d4cf5c5c60e972cc19782d1f37ec9d47dd1e81cdf481b64dab62f96bac846bb4
d6e524514e0d112015c841b62377d648	a31296f1eed15262f070abb3e89acf1a3917746a	6af6fe3eafd4cf2c82738d45a6a95577d970f3fbbe094afd24d1d4a0bb5ad1b4
3f310215a70d748f9335c767e61a2ab4	47b8d74725f353dad8177c478ffa77d424e9a34e	9a19522b23acfc6705e4fac65640527a8adbbc9719dab436f28101b6cbc140c6
bca04d74261fedfbd191ffd5e7cf6214	f6b1b3fc532b516366de6b3452b5c23441386e17	97ea91fb673f4994da491433751c4fca011993ba10191f09c70ca6c8d2b4f944

Full Version

MD5	SHA-1	SHA-256
d1600f45005aa8b8fcbb446f34f7b9f5	816b2442c17585396b73b54fbc87be624d55276c	06ed3daccfbb30c68a33583a761fc20cc3e21adb8dd64a42d922e6da2a01c0dd
4d7e67ed02713c789336f8804231b1ca	424e3570f36fdb541e9b49f9d6824949ccf96ea6	27c6341554a04bdc792ffbc5cda26511cbcfcc66334fb6ebbc24a14969b4e498
9bcf889b14968c61df95961a161719ba	b1604a158cdd24182d8d4198fb17f4d348b92601	3fda0a5da313886b0339eee65c69c779ed620b303ba079ee0864ca4a1496b0b4
54ad403349831b175a98a429f818f02a	8c4d5618c3ab2d2411ede54f443560891a78986b	408c11caf548048732ac21e88a54e80d47a05b9619c1c16b65fa850e0172f428
83311c960609418d5f0a5160324ceb1d	7f534338d1399221bb2134d917a1e3eef8b309e5	41ccf6de0d51bd29d35be12ae24f04b2f88ec2b202b239424f90c666d25473e8
96f09c5c56f59c733d1a9b01fea0cfb4	7e9fcbd7f31c3a4ddb3221351e8d04ecdce69474	66c86f29afb1152aad8e426ebb6569ad03ce7b69ea3c8a5cc40011c2a3ab973b
b5138c77983dba10c4976c411161bbf9	57c9411b444ce4fddf6ddf210dd7dfd008c156ac	aa868d007c4dfd825104faafb3798b9ab745b29794a57365bef41ec3f6019eea
53b7bdd75776f342bf5f5395d4c46520	8971b585f8bf7d9f086d97d2d640ed4ce7f6b178	add9f9dca97c3b6d52efe7d48ecd3d349a70411eaa3d4aeff6e6215b77f42b90
07111aff7afc052a81f267ea2e83dcef	e98700d562edb0ed29e429d0263ec448daf8a5f2	aed7ab5d0de01c3724c917c034e26a5e9eed3f7fbf4082b024576a41725d66cf
eb402e8dd2cae58476acc8e697ee7171	55d1a679ae7e812d5c91ef78bec4095a2048427c	ba153e449ee926c019b548997c32d0579b9c6f350b1590a025d5d9a216ddbffd
cfea49c577ef865de659d5b8025db3f9	99e724a6941c65eed20ac13c4940e325fd323082	ce8ad96819c814dd1735e621639a8845ae7132375879cc5b5d5f6877cb909a68
17fe9611ea566887b3ef42284f96de03	aad7ec556d8d6e4333552d23588734d339373ab8	ea4792353e0f97968e7c69ffba81c144f22f54382af4e61a1347edd0ae15830f

Malware Hunter Team Samples

MD5	SHA-1	SHA-256
d1600f45005aa8b8fcbb446f34f7b9f5	816b2442c17585396b73b54fbc87be624d55276c	06ed3daccfbb30c68a33583a761fc20cc3e21adb8dd64a42d922e6da2a01c0dd
4d7e67ed02713c789336f8804231b1ca	424e3570f36fdb541e9b49f9d6824949ccf96ea6	27c6341554a04bdc792ffbc5cda26511cbcfcc66334fb6ebbc24a14969b4e498
9bcf889b14968c61df95961a161719ba	b1604a158cdd24182d8d4198fb17f4d348b92601	3fda0a5da313886b0339eee65c69c779ed620b303ba079ee0864ca4a1496b0b4
54ad403349831b175a98a429f818f02a	8c4d5618c3ab2d2411ede54f443560891a78986b	408c11caf548048732ac21e88a54e80d47a05b9619c1c16b65fa850e0172f428
83311c960609418d5f0a5160324ceb1d	7f534338d1399221bb2134d917a1e3eef8b309e5	41ccf6de0d51bd29d35be12ae24f04b2f88ec2b202b239424f90c666d25473e8
96f09c5c56f59c733d1a9b01fea0cfb4	7e9fcbd7f31c3a4ddb3221351e8d04ecdce69474	66c86f29afb1152aad8e426ebb6569ad03ce7b69ea3c8a5cc40011c2a3ab973b
b5138c77983dba10c4976c411161bbf9	57c9411b444ce4fddf6ddf210dd7dfd008c156ac	aa868d007c4dfd825104faafb3798b9ab745b29794a57365bef41ec3f6019eea
53b7bdd75776f342bf5f5395d4c46520	8971b585f8bf7d9f086d97d2d640ed4ce7f6b178	add9f9dca97c3b6d52efe7d48ecd3d349a70411eaa3d4aeff6e6215b77f42b90
07111aff7afc052a81f267ea2e83dcef	e98700d562edb0ed29e429d0263ec448daf8a5f2	aed7ab5d0de01c3724c917c034e26a5e9eed3f7fbf4082b024576a41725d66cf
eb402e8dd2cae58476acc8e697ee7171	55d1a679ae7e812d5c91ef78bec4095a2048427c	ba153e449ee926c019b548997c32d0579b9c6f350b1590a025d5d9a216ddbffd
cfea49c577ef865de659d5b8025db3f9	99e724a6941c65eed20ac13c4940e325fd323082	ce8ad96819c814dd1735e621639a8845ae7132375879cc5b5d5f6877cb909a68
17fe9611ea566887b3ef42284f96de03	aad7ec556d8d6e4333552d23588734d339373ab8	ea4792353e0f97968e7c69ffba81c144f22f54382af4e61a1347edd0ae15830f