White Paper

Bitdefender®
Pacifier APT

Contents

Authors:

- Marius TIVADAR
- Cristian ISTRATE
- Iulian MUNTEAN
- Andrei ARDELEAN

Overview

Bitdefender detected and blocked an ongoing cyber-espionage campaign against Romanian institutions and other foreign targets. The attacks started in 2014, with the latest reported occurrences in May of 2016. The APT, dubbed Pacifier by Bitdefender researchers, makes use of malicious .doc documents and .zip files distributed via spear phishing e-mail.

Documents used range from curriculum vitae, to invitations to social functions or conferences, to second hand car offers and even, in one case, a letter of instructions from a high-ranking official. Some were marked as “urgent”, “important”, “immediate action required” and so on.

Other samples of the same malicious software were detected in Iran, India, Philippines, Russia, Lithuania, Thailand, Vietnam and Hungary.

The high number of variants, in conjunction with the low number of reports and the nature of the affected machines has brought us to the conclusion that we are dealing with an APT.

The malicious payloads delivered evolved over time, becoming stealthier and adding functionality as time went by. Our analysis focuses on three representative variants of the malware used in the attacks, but a number of others, differing by minor details, were found in the wild.

Aside from the analysis, this paper lists hashes of malicious files, as well as other IOCs.

2014-15 Executable Files

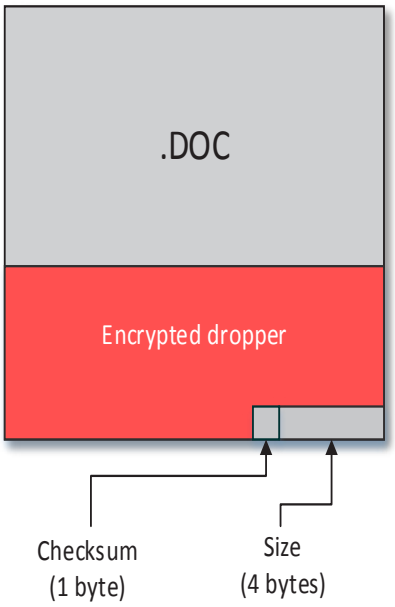
The Infected Document

The infection starts from one infected document.

Analysis started from documents containing droppers. The dropper is encrypted and appended to the end of the document; the document contains a script that reads, decrypts and runs the dropper.

The last dword in the document file represents the size of the executable. The 5th byte from the end of the document is a checksum on the decrypted executable, used for validation. The actions from the script are summarized below:

```
size = last_dword_from_file;
checksum = byte_before_size_dword; // read encrypted dropper in
buffer for (key = 35, i = 0; i < size; i++)
{     buffer[i] = buffer[i] ^ key;
    key = (key ^ 217) ^ (i % 256);
} for (sum = 0, i = 0; i < size; i++)
{     sum = sum ^ buffer[i];
}
if (sum ==checksum)(1 byte)    (4 bytes)
{
    // write and execute the file in:
    // %appdata%\Microsoft\Word\MSWord.exe
}
```

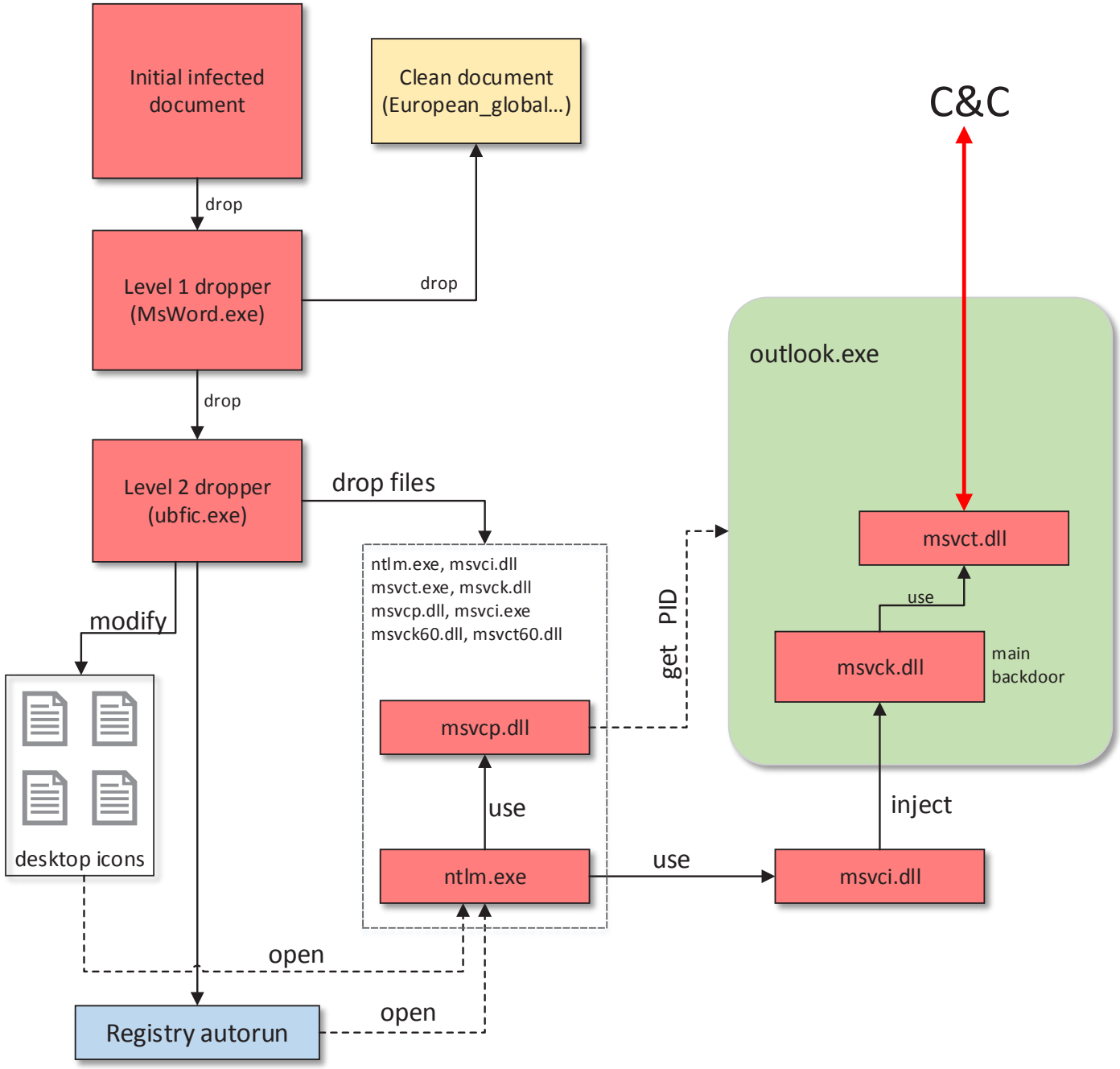


For the script to run, macros must be enabled in Word.

As you can see in Appendix A the content of the infected documents is designed to trick the user into enabling the macros. If the macros are enabled the dropper is executed and opens another document, as expected by the user. For example if the infected document says it is a „protected" document and you must enable macros to view it, then the dropper will open another document with an invitation to a conference as the „protected" document.

In Appendix B you can find some examples of these „pacifier" documents, these are clean and contain no scripts or executables.

Trojan component



Functionality on 32bit Windows

ntlm.exe – startup executable
msvcp.dll – get PID of outlook.exe
msvci.dll – inject msvck.dll in outlook process
msvck.dll – main backdoor
msvct.dll – C&C communication

The Dropper

The script previously loaded from the infected .doc file executes the dropper from:

%appdata%\Microsoft\Word\MSWord.exe

The dropper is a small executable that has the files to be deployed in the overlay encrypted with RC4. It just creates and runs the following files in this order:

- %appdata%\TMP\European_global_navigation_system.doc
- %appdata%\Axpim\ubfic.exe
- %appdata%\Axpim\ anfel.js

The file European_global_navigation_system.doc is a clean document used to distract the user (see Appendix B). The file ubfic.exe is another dropper containing the real payload. The anfel.js file is used for self deletion. The names: Axpim, ubfic, anfel are random generated.

The folder name will contain 4-6 characters and starts with capital letter. The file names contain 4-5 lowercase letters before the extension. The random generator is based on GetTickCount API. The algorithm for creating the names is presented below in python implementation. Practically, it concatenates random vowels and consonants, but with the condition that no more than two of the same type to be consecutive, with the aim of generating names that are somewhat pronounceable and thus may pass as man-made.

This algorithm was also used in some versions of Zeus for file name generation.

```
# name length will be between minLen and maxLen
# flags - 4 name will contain spaces
#       - 2 name will have first letter uppercase
#       - can be combined
def RandomName(minLen,maxLen,flag):
letters = ['aeiouy', 'bcdfghklmnpqrstvwxz']
name = ' ' seed()
index = randint(0,1)
nameLen = randint(minLen, maxLen)
for i in range(0, nameLen):
if i > 0 and i % 2 == 0:
      index = randint(0,1)
            if (flag & 4) == 4 and len(name)>0 and name[-1]!=' ' and randint(0,3)==0:
            name = name + ' '
            elif i % 2 == 0:
            name = name + choice(letters[index])
else:
            name = name + choice(letters[1 - index])
if (flag & 2) == 2:
name = name.title()
return name.rstrip()
# generate folder name
RandomName(4, 6, 2)
# generate file names
RandomName(4, 5, 0)
```

The Second Dropper

The payload dropper, ubfic.exe, contains its files in its .data section and is not encrypted or compressed. It creates the files:

- %temp%\ntlm.exe
- %temp%\msvci.dll
- %temp%\msvcp.dll
- %temp%\msvck.dll
- %temp%\msvct.dll
- %temp%\msvci.exe (64 bit)
- %temp%\msvck60.dll (64 bit)
- %temp%\msvct60.dll (64 bit)

These files make up the payload. The last three are for 64-bit Windows, the rest are for the 32-bit version. The starting point of the payload is the ntlm.exe file which is discussed below. Next, using COM objects, the dropper modifies the .lnk files from the desktop and saves the original links in %temp%\Links folder. The links are modified to start the trojan:

Lnk	Target
Original	*„C:\Program Files\Sysinternals\Filemon.exe"*
Modified	*„C:\Documents and Settings\user\Local Settings\Temp\ntlm.exe"* *C:\Program Files\Sysinternals\Filemon.exe*

The target file of the link is replaced with ntlm.exe and the original target is sent as a parameter to ntlm.exe, which upon execution will execute it. Next, the dropper creates %temp%\startup.bat which adds to the registry:

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svchostUpdate -> %TEMP%\ntlm.exe

HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Devices -> %TEMP%\ntlm.exe

The first key is for starting the trojan, along the .lnk files. Some versions do not have the lnk feature, only the registry keys. The second key is never used in our samples.

Last file created is %temp%\Axpim\selfdel.bat for self-deletion.

msvcp.dll

32 bit library used for returning the PID of one of the processes: iexplore.exe, outlook.exe, firefox.exe, chrome.exe. It hase one export, msvcp, which enumerates running processes and checks their names. It will return the PID of the first one found. If the processes could not be enumerated it returns 0. If no process was found it returns -1. Instead of storing the actual names of those processes in the dll, it stores a byte array as a key and one byte array (result) for each of the 4 names. The checking is done: ProcessName xor key == result, on coresponding bytes. The function may return different PIDs when processes are stopped or started.

msvci.dll

32 bit library used for injecting msvck.dll (the 32 bit backdoor) into a running 32 bit process. The library has one export msvci, which takes one parameter representing the PID of a running process. It allocates a small chunk of memory into that process (260 bytes). In this memory it copies the path to the msvck.dll file, which is found in %temp%\msvck.dll.

Then, from the current process it gets the address of LoadLibraryA function and creates a remote thread in the target process starting at that address. The parameter sent for the thread function is the address of the new allocated string containing the path to msvck.dll. As a result, in the target process, a thread is created which just executes LoadLibraryA(„path to msvck.dll").

This method will work even if ASLR is enabled, because the random loading offset for dlls is calculated once per boot and by default one dll is loaded at the same address in different processes. msvci.exe

64 bit application used for injecting msvck60.dll (the 64 bit backdoor) into a running 64bit process. The executable takes a commandline parameter representing a PID. The functionality is identical to msvci.dll library. ntlm.exe

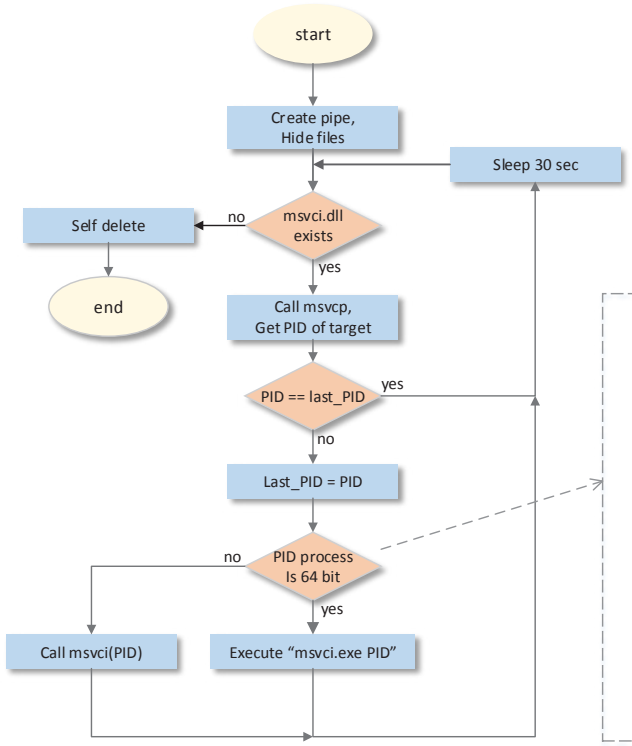
This is the file that starts the trojan. First, it creates a named pipe \\.\pipe\bc367 used as a mutex, as the file will probably be executed many times from the shortcuts. The name of the pipe and all other strings in the file and in the rest of the dlls are encrypted with RC4. It sets the other files of the trojan (msvci.dll, msvcp.dll, msvck.dll, msvct.dll, msvci.exe, msvck60.dll, msvct60.dll) as hidden, ntml.exe will not be hidden.

Afterwards, the executable enters an infinite loop.

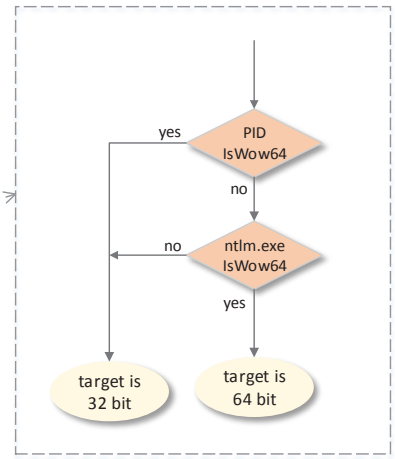
It checks if the file %temp%\msvci.dll exists, if it does not exist it will do a self destroy: deletes all components, deletes registry keys, restores the original lnk files from %temp%\Links, closes the pipe. This behaviour is used by the backdoor component later for self deleting.

Next it loads the msvcp.dll library and call msvcp export from it. This function returns the PID of one of the processes: iexplore.exe, outlook. exe, firefox.exe, chrome.exe, and 0 or -1 if none of those processes was found. If a process was found, its PID is saved into a variable, then the program checks whether that process is a 32 bit or 64 bit process.

If it is a 32 bit process it calls msvci export from msvci.dll with one parameter, the PID. If the process is a 64 bit process it creates a



ntlm.exe functionality



Check if a process is 64bit knowing that ntlm.exe is 32bit

process from msvci.exe with the commandline parameter being the PID converted to ascii, base 10 (ex: msvci.exe 728).

The purpose of msvci.dll and msvci.exe (64bit) is to inject the payload into a process. After this it sleeps 30 seconds and repeats indefinitely.

The saved PID is checked so that it doesn't inject again into the same process. This mecanism is flawed because another process from the list could be found and it will inject into it also, or it is possible to inject in the same process twice, for example outlook.exe opened first and injected then iexplore.exe opened second and injected then iexplore.exe closed and outlook.exe injected again.

Another bug can appear if msvcp.dll will be deleted because when calling LoadLibrary and GetProcAddress it doesn't check the return values and the program will crash when calling a NULL pointer.

msvct.dll

32 bit library contains functions for communicating with the C&C, using WinINet API. The backdoor does not contain any C&C addresses or networking logic, it just uses the exports from msvct.dll, namely:

- BOOL CI(void) – Checks for internet connection. Returns true if a http request („GET /1") to www.google.com succeeds; returns false if not.

- BOOL SHR(char *ServerAddr, char *ServerScript, void *ID, void *SendBuff, void *RecvBuff, void *extra) – Sends and receives data to/ from C&C. The communication is encrypted through HTTPS, port 443.

- Extra flags are used for the connection: INTERNET_FLAG_IGNORE_ CERT_DATE_INVALID, INTERNET_FLAG_IGNORE_CERT_ CN_INVALID, SECURITY_FLAG_IGNORE_UNKNOWN_CA to ignore errors caused by invalid certificates. ServerAddr and ServerScript make the address of the C&C, these are found using the CS export. ID is a structure made from a buffer (string) and its length. The ID string will be put into the HTTP headers. SendBuff is the same type of structure like the ID. SendBuff contains data that is sent to the C&C. RecvBuff is a structure that contains 4 members, 3 being pointers: a data buffer, a buffer length and two strings. This structure will be filled with data coming from C&C. The data buffer can contain batch commands or whole files to be written to disk. The first string will contain the Content-Type from the headers and this will be the command for the backdoor. The second string will contain the Content-Location from the headers and will have the name of a file for download/upload commands. The last parameter, extra, is again a structure from a buffer and its length. It is optional. The string that it contains is sent through the HTTP header. This is used by the CS export and then it contains that „Check: RandomNr" string for C&C validation. It is also used in the „upload" backdoor command. The function returns true or false.

- BOOL CS(void *ID, char *ServerAddr, char *ServerScript) – Check C&C connectivity. Returns true if it founds a valid C&C and ServerAdd, ServerScript (output parameters) will contain the address and the page/script of the C&C. ID (input parameter) is a structure that contains a string and its length. The string represents an ID identifying the infected system. This function tries two hardcoded C&C addresses: 88.208.0.130/rss.php, 78.47.51.238/rss.php (other variants used different addreses). It generates a random number from 0 to 32767 and then creates a string with it, such as: „Check: 1352". This string will be sent in the http headers using SHR function to the C&C. If a C&C is alive it must respond with the string „1352" back. If none of the two C&Cs are alive the function returns false.

msvct60.dll

64 bit version of msvct.dll msvck.dll
32 bit library containing the main functionality, the actual backdoor. It has no exports and will only execute if injected into iexplore.exe, outlook.exe, firefox.exe or chrome.exe.
First, it checks the internet connection using the CI export from msvct.dll. If it has no internet access the execution ends.

Next it creates an ID of the infected system as a string such as:

„MyCookie: {eceee5c0-1eca-11de-abc9-806d6172696f}{3559831177}" – the GUID is obtained using GetCurrentHwProfile API and the second bracketed number is the volume serial number. If GetCurrentHwProfile fails the ID will be:

„MyCookie: UserName{3559831177}" – with the username from GetUserNameA API and again the volume serial number.

The ID created will be used to check the connection to the C&C with the CS export from msvct.dll. The connection is checked in an infinite loop with a sleep of 28 minutes after each check, until a valid, active C&C server is found. The C&C addresses are contained in msvct.dll and one of them is returned by CS function on success. It can be seen that the loop was meant to only check 3 times for the connection (like other samples do), but, maybe because this is a intermediary version or by negligence, the code actually loops indefinitely.

After this follows the code for a regular backdoor which receives commands from the C&C. The commands are received and the results are sent back with the SHR export from msvct.dll. The backdoor will be identifying the computer with the ID it created. After 3 successful commands received it will sleep for 28 minutes. After 3 consecutive failed commands (SHR returns false) it will again perform a C&C validation with the CS function in an infinite loop. Here it may receive the other C&C address. If a command is received but it is not recognized it will sleep again for 28 minutes. The commands are text strings and are described below:

„cmd" - Creates the file %temp%\xmlupd.bat which will contain batch commands. It creates a process with xmlupd.bat but with stderr and stdout redirected to the file %temp%\1. It waits maximum 30 seconds for it to finish then it will kill the process. After that it sends back to the C&C the content of the %temp%\1 file. Some examples of commands received:

```
systeminfo

set

netstat -ano

dir/a %programfiles%

dir /a %programfiles(x86)%
```

„download" – Receives a file and writes it to %temp%. The name of the file is also received from C&C. It will not execute the file.

„upload" – Receives a file name, reads the file, converts the content to base64 and sends it to the C&C with an extra HTTP header „File: filename". If it doesn't find the file it sends back the last error (GetLastError for CreateFileA).

„text/html" – This looks like a normal Content-Type HTTP header value (the commands arrive in the Content-Type header). When this is received it resets the number of consecutive failed commands and sleeps 28 minutes. The backdoor will continue after.

„close" – Stops the execution of the backdoor, the injected thread will terminate.

„selfdestroy" – Deletes the file %temp%\msvci.dll then ntlm.exe will take care of deleteing all other components. The backdoor ends execution after the command.

msvck60.dll

64 bit version of msvck.dll.

2014-15 Browser Extension

The Infected Document

The infection start from a document cv_Mate.Dimitrescu.doc . The document is constructed in the same way as the documents containing the other variant of the malware. The script in it has the same functionality, it will create and execute the dropper %appdata%\Microsoft\Word\MSWord.exe

The Dropper

The dropper looks the same as the other droppers, only smaller in size. The files that it contains are encrypted with RC4 in overlay. Only two files will be dropped:

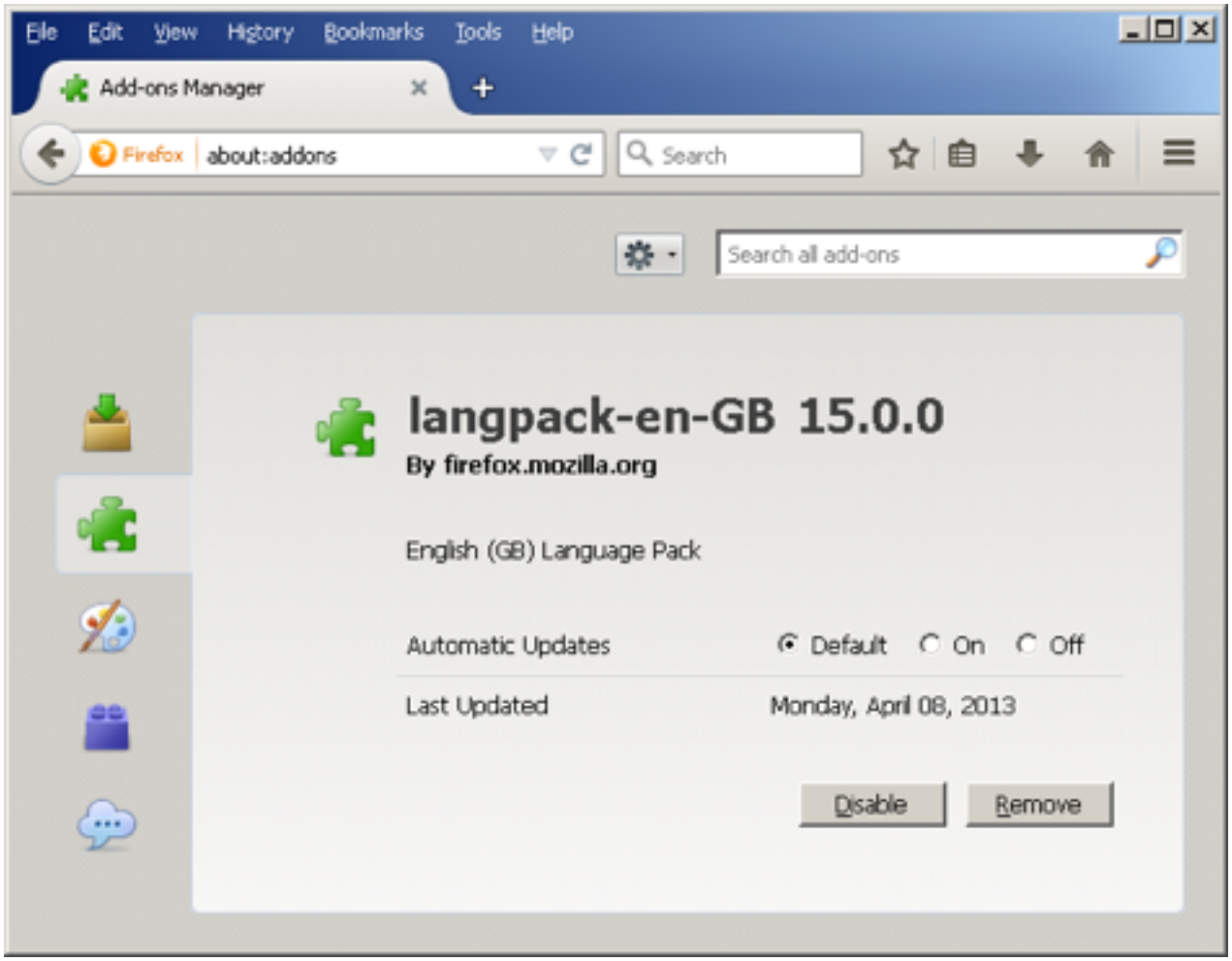
%appdata%\Aggea\ivotp.xpi

%appdata%\Aggea\ylir.js

The names Aggea, ivotp.xpi, ylir.js are random generated. No clean document is present in the dropper and the initial infected document will not close. There is no second dropper either. The javascript file is executed, installs the xpi file as an extension in Firefox and then deletes the directory %appdata%\Aggea\.

The Firefox Extension

The extension file will be renamed to {285364ef-e70c-4386-8e5c-2aa93a78daad}.xpi then will be installed in Firefox. In the browser it will appear with the name „langpack-en-GB 15.0.0" as in the picture below.



We tested it in Firefox 35.0, in some newer versions it didn't work. In this version of the malware the extension will work as the backdoor. The functionality is contained in 3 files in the extension package: 1.js, 2.js, main.js.

1.js

The file 1.js contains encryption and decryption routines. Algorithm used is AES, and the implementation seems to be copied from https:// github.com/chrisveness/crypto/blob/master/aes.js and ran through an obfuscator.

2.js

The file 2.js contains the C&C addresses and the network functionality. All data to and from the C&C will be sent through normal HTTP but it will be encrypted and converted to base64. The encryption key is generated once when the extension is started but it can be changed if the C&C requests as we will see later. In this variant we find 6 C&C addresses (presumably these are previously-compromised machines):

http://reckless.dk/wp-includes/class-pomo.php
http://reckless.dk/wp-includes/class.wp-db.php
http://fishstalk.esy.es/wp-content/plugins/bbpress/includes/common/menu.php
http://fishstalk.esy.es/wp-includes/SimplePie/Net/IPv4.php
http://77-ufo.com/wp-includes/class-menu.php
http://77-ufo.com/pma/db_table.php

The strings in the files are not visible because the javascript files are highly obfuscated. The function that sends the data to the C&C also receives the response. The function will select each time a C&C that responds from the list. This file would be the equivalent of the msvct. dll file from the earlier version.

main.js

The file main.js is the actual backdoor (msvck.dll equivalent). The malware starts by creating an ID for the system. The ID is a Md5 hash on some data colected from the system. The ID is sent in every request to the C&C as in the previous version with executable files. This is an example of the data collected for the ID from a test machine:

{285364ef-e70c-4386-8e5c-2aa93a78daad}C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\2gmaw237. defaultC:\Documents and Settings\user\Desktop{ec8030f7-c20a-464f-9b0e-13a3a9e97384}

There are no line separators, it is just a long string. The ID in this case will be „5815da5d0d5565f342474d976f507807". The gathered data represents:

{285364ef-e70c-4386-8e5c-2aa93a78daad} – GUID for the extension

C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\2gmaw237.default

Firefox profile folder where the extension was placed

C:\Documents and Settings\user\Desktop – desktop folder of the active user

{ec8030f7-c20a-464f-9b0e-13a3a9e97384} - GUID for Firefox

After the ID was generated, an encryption/decryption key will be created. It concatenates the hardcoded string „7201895b632dc5044c02ea98b0dbd371" with the string containing the ID. In the case of our example this will make the string "7201895b632dc5044c02ea98b0dbd3715815da5d0d5565f342474d976f507807". Then it makes the Md5 hash on the string which will result in "ec398e010a0cb6b6e4f48722dc07eaa3".

This final buffer (containing a C like string with the md5) is the decryption key. The key is subsequently used for encryption and decryption of every message sent and received from the C&C.

Next it gathers more data about the system. Again, an example from the test machine:

5815da5d0d5565f342474d976f507807 {285364ef-e70c-4386-8e5c-2aa93a78daad} winnt x86

Mozilla Firefox {ec8030f7-c20a-464f-9b0e-13a3a9e97384} 35.0

C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\2gmaw237.default

C:\Documents and Settings\user\Desktop

0040|||C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\2gmaw237.default\user.js

Lines are separated by line feed (0x0A). The data represents:

5815da5d0d5565f342474d976f507807 – System ID

{285364ef-e70c-4386-8e5c-2aa93a78daad} – GUID for the extension winnt x86 – Operating system

{ec8030f7-c20a-464f-9b0e-13a3a9e97384} – GUID for Firefox

35.0 – Firefox version

C:\Perl... - %path%

C:\Documents and Settings\user\Application Data\Mozilla\Firefox\Profiles\2gmaw237.default – Path to Firefox profile

C:\Documents and Settings\user\Desktop – Desktop folder of the logged user

C:\Documents and Settings\user\ApplicationData\Mozilla\Firefox\Profiles\2gmaw237.default\user.js – Config file for Firefox which can override normal settings, it will try to delete it. If is still present after deletion it will put 0041 instead 0040 as the status.

The data is encrypted with the key that was generated and sent to the C&C. Finally the malware sets a timer which calls a function that sends to C&C, receives from C&C and processes the backdoor commands every five seconds. The commands are made up from 3 strings separated by „|||",like „nr|||string1|||string2".

nr will be ‚0' to ‚6' and represents the backdoor command. string1 and string2 contain file names, urls and other things used by the commands. In some commands string2 is not used. The commands are:

„1|||file commandlineOptions”

Executes the file „file” with command line parameters „commandlineOptions”. stdout and stderr are redirected to a string and the content of the string will be sent to the C&C. If the file to be executed does not exists it sends back to the C&C „0011|||file”

„2|||url|||file”

Downloads the file from „url” and writes it with the name „file”. Returns to the C&C „0020|||file” if the file was successfuly written, „0021|||file” if the file was not written or „0051|||url|||errCode” if the download failed.

„3|||file”

Searches the file „file” and sends it to the C&C. In case of an error it will return „0034|||file” if the file is a folder, „0033|||file|||size” if the file size is greater than 5000000 bytes, „0031|||file” if the file exists but it couldn’t get information about it, „0032|||file” if the file does not exists, „0051|||url|||errCode” if the file could not be sent due to some network problems.

„4|||file”

Deletes the file „file”. Returns to the C&C „0040|||file” if the file was deleted or „0041|||file” if the file could not be deleted.

„5|||string1|||file”

Sends back „string1” to the C&C. The C&C responds with a buffer which will be written to the file „file”. It sends back to the C&C „0020|||file” if the file was successfuly written, „0021|||file” if the file could not be written or „0051|||url|||errCode” in case of network problems.

„6|||path|||depth”

Lists files and directories from „path” recursively until „depth” level, then it sends the list to the C&C. An example:

0060|||c:\0
1|c:\0\Documents|d|ct=NaN|lat=1442919637000|lmt=1442919637000|
1|c:\0\main.js|17394|ct=NaN|lat=1442919639000|lmt=1442405697000|
1|c:\0\main1.js.js_format|9486|ct=NaN|lat=1442919639000|lmt=1442404160000|
2|c:\0\Documents\rec.doc|12|ct=NaN|lat=1442919637000|lmt=1442919637000|
...
0061|||path_failed|||error_code
...
0061|||c:\0|||10000

The „path” listed and a return code would be „0060|||c:\0”. „1” and „2” from the start of the lines is the level of a file or directory. The count starts at 1 and the maximum level would be „depth”+1This is followed by a file or directory path. After the path a „d” follows, if the path specifies a directory, or something like „17394” in case of a file, which is the size of the file. „ct=NaN” is the creation time of a file. There is a typo in the code and because of that the creation time is never actually retrieved („winBirtdhDate” instead of „winBirthDate”). „lat=1442919637000” is the last access time. „lmt=1442919637000” is last modified time. „0061|||path_failed|||error_code” is optional and may appear multiple times, contains a path at which the file enumeration failed. „00061|||c:\0|||10000” at the end is optional and appears only if the listing so far contains more than 10000 characters, then no more lines will be added.

„0|||key”

Sets a new encryption/decryption key with the C string „key”

Other 2015 variants

We found different versions of the files with almost identical functionality and only minor differences. The most notable difference is that C&C addresses vary. Another interesting fact is where the samples were spotted.

More C&C servers

- reckless.dk/wp-includes/class-pomo.php
- reckless.dk/wp-includes/class.wp-db.php
- fishstalk.esy.es/wp-content/plugins/bbpress/includes/common/menu.php
- fishstalk.esy.es/wp-includes/SimplePie/Net/IPv4.php
- 77-ufo.com/wp-includes/class-menu.php
- 77-ufo.com/pma/db_table.php
- scientific.otzo.com/rss.php

Documents

The documents differ in what they present but they are identical in where the dropper resides and how the script operates.

Droppers

The most common level 1 droppers contain the files encrypted with RC4 in the overlay and level 2 droppers have the files in clear in .data section. Some level 1 and level 2 droppers are just selfextracting winrar archives, but they have the same functionality.

Firefox addon

Only one version was found.

ntlm.exe, svchost.exe, dwms.exe

Different names for the starting executable. Some variants do not have the link files functionality and rely only on the registry key to start. All variants use the key HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and the value is named „svchostUpdate" or „dwm service".

msvci.dll

Some versions check if the process into which they inject is 64 bit or 32 bit. If it is 64 bit „msvci.exe PID" is executed. This looks like an early version of the code because the functionality for 32/64 bit is in ntlm.exe file and if we have a 64 bit process to inject, msvci.dll would not be loaded in the first place.

msvct.dll

Different C&C addresses.

msvck.dll

Some versions check the C&C in an infinite loop other will check it only 3 times and if no connection could be established the execution ends. The computer ID has a GUID obtained with GetCurrentHwProfile in some versions and in others the GUID is taken from the registry key HKLM\SOFTWARE\Microsoft\Cryptography\ MachineGuid

Strings

Almost all strings in the files are encrypted. They are encrypted with RC4 in all versions, only the decryption keys are different.

2016 attack wave

In May 2016 we have encountered a new wave of attacks. They came, at least in some known cases, as spear phishing emails containing various documents: topics like Oil conferences, international politics, budget calculations, simple guidelines on how to interview for a job in foreign affairs.

The attackers moved away from using documents containing macro scripts to employing a zip archive containing a java-script file that would in turn drop a clean document and the actual malware. The archived file has a double extension, something like urgent-document.doc.js. This method is probably more efficient, as the victim doesn't have to enable macros in Word Viewer.

Generally speaking, the components are slightly different from the 2015 variants but they achieve the same results. As a general rule, we observed that these variants tend to be stealthier than past years' variants because malware is only injected in legitimate processes, so that no new main executable will be seen in the process list after injection. The components are outlined below.

Document containing malicious macro script

The new infected documents have a different structure, as seen in the picture. The clean document is encrypted directly in the infected document, not contained in the dropper as with previous versions. The macro from the infected document will decrypt and run the clean document and the dropper.

At the end of the document there is a dword which contains the total size (document+dropper+4). In front of the encrypted document is a dword which represents its size. After the encrypted document there is one byte, the checksum.

The encrypted dropper has the same structure. The checksum algorithm remains the same but the encryption is slightly different than before:

```
// decrypt
for (key = 75, i = 0; i < size; i++)
{
    buffer[i] = buffer[i] ^ key;
    key = (key ^ ((200 + i) % 256) ^ (i % 256);
}
```

The Dropper

The droppers are self-extracting Winrar archives. They do not contain the clean document any more. The component files are extracted to:

- %appdata%\Microsoft\VisualStudio\11.0\dws.exe
- %appdata%\Microsoft\VisualStudio\11.0\msi.dll
- %appdata%\Microsoft\VisualStudio\11.0\msi32.dll
- %appdata%\Microsoft\VisualStudio\11.0\msk.dll (optional)
- %appdata%\Microsoft\VisualStudio\11.0\msp.dll
- %appdata%\Microsoft\VisualStudio\11.0\mst.dll

- %appdata%\Microsoft\VisualStudio\11.0\msi.exe (64 bit)
- %appdata%\Microsoft\VisualStudio\11.0\msi60.dll (64 bit)
- %appdata%\Microsoft\VisualStudio\11.0\msk60.dll (64 bit, optional)
- %appdata%\Microsoft\VisualStudio\11.0\msp60.dll (64 bit)
- %appdata%\Microsoft\VisualStudio\11.0\mst60.dll (64 bit)
- %appdata%\Microsoft\VisualStudio\11.0\msvci60.dll (64 bit)

msk.dll and msk60.dll are not always present in which case their functionality is taken by another dll. After the files are dropped the file dws.exe is executed without any parameters.

msp.dll

Similar to the old msvcp.dll. Returns the PID of some processes. It has 3 important exports: p, p1, p2.

- p export has the functionality as the older variant, returns the PID of one of iexplore.exe, firefox.exe, chrome.exe, outlook.exe.
- p1 export returns the PID of dwm.exe (Windows Desktop Manager).
- p2 returns the PID of taskhostex.exe (Host Process for Windows Tasks) or sihost.exe (Shell Infrastructure Host). These are legitimate processes that can be found in Windows 8 and Windows 10. Some variants do not search for sihost.exe.
- Some variants also have another export kp which kills a process, but it's never invoked.

msp60.dll

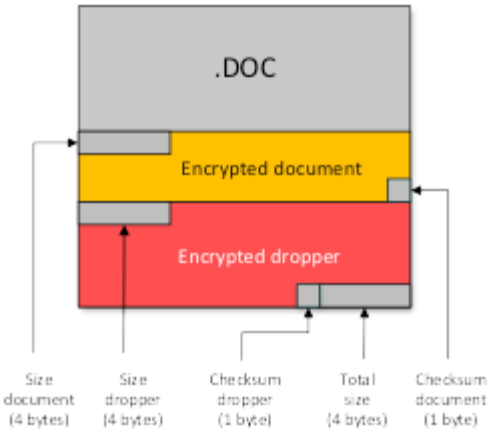
64 bit variant of msp.dll.

msi.dll

This is similar to the old msvci.dll. It has one export msvci with two parameters, a path to a file and a PID. It will inject the file from the path into the process with that PID. For 32 bit processes.

msvci60.dll

64 bit variant of msi.dll.



msi.exe

Similar to msi.dll, for 64 bit processes. It will look for 1 or 2 command line parameters. The first is a process PID. The second parameter can be anything and is just a flag for knowing what library to inject: 2 parameters – will inject msi60.dll, 1 parameter – will inject msk60.dll. The injection is done with the msvci export from msvci60.dll.

dws.exe

The starting module. If it was started with a parameter and the parameter is not a number (as an ascii string), it will create a process with the executable taken from the parameter; this looks like a reminiscent of the .lnk files functionality.

If the parameter is a number, the number will be interpreted as a PID and it will check if the process with that PID is a 32/64 bit process. If it is a 64 bit process, dws.exe will create a new process with the command line "msi.exe PID". If it is a 32 bit process, dws.exe will load msi.dll and call its export as msvci("msvck",pid). In this case (parameter as number received) the execution stops here.

If the file msp.dll is not present it will delete all components of the trojan. Otherwise it will call from msp.dll the export p1 or p2 (only on Windows 8/10 or Server 2012) to get the PID of dwm.exe, taskhostex.exe or sihost.exe. If the returned PID is for a 64 bit process

will run “msi.exe PID 1”; two parameters are passed, the last one being dummy and used by msi.exe just to know to take a different path of execution. If the returned PID is for a 32 bit process, it will load msi.dll and call msvci(“msi32.dll”, PID). After this the program terminates (the older ntlm.exe would have stayed in a loop here).

msi32.dll

Code from this library will execute injected in dwm.exe, taskhostex.exe. Sets the same registry autorun key as the 2015 variants. It will also set a scheduled task:

schtasks /create /SC DAILY /ST 12:00 /TN update /F /TR %appdata%\Microsoft\VisualStudio\11\dws.exe

which will make sure that main module will start daily.

Checks if msp.dll exists, if it does not, it will execute dws.exe without parameters, which in turn will self delete the Trojan. It uses from msp.dll the p export to get a PID. If that PID is a 32 bit process it will run “dws.exe PID” and if it is a 64 bit process it will run “msi.exe PID”. Those processes with parameters will inject the actual backdoor (msk.dll). It will stay in a loop and try to find targeted processes in order to inject in them. This variant of the Trojan is stealthier than the previous one in which ntlm.exe would stay in a loop and try to inject, in which case a suspicious process (ntlm.exe) would be visible.

msi32.dll – with backdoor functionality

In some droppers msi32.dll has another variant different enough to be described separately. In this case msi32.dll would contain backdoor functionality along the functionality described earlier. It will function in 2 ways (backdoor or earlier msi32.dll) based on the name of the process from which it runs. If the containing process is dwm.exe, taskhostex.exe or sihost.exe it will function like the usual msi32.dll and also will copy itself as msk.dll for later use as the backdoor. If the containing process is another process then it will function as the backdoor (identical to msk.dll).

msi60.dll

64 bit variant of msi32.dll.

msk.dll

The backdoor component, similar to the old msvck.dll, it has the same functionality. For selfdelete it will delete msp.dll. It also has a new backdoor command “st” which sets the time in milliseconds for Sleep, time value received from the C&C.

msk60.dll

64 bit variant of msk.dll.

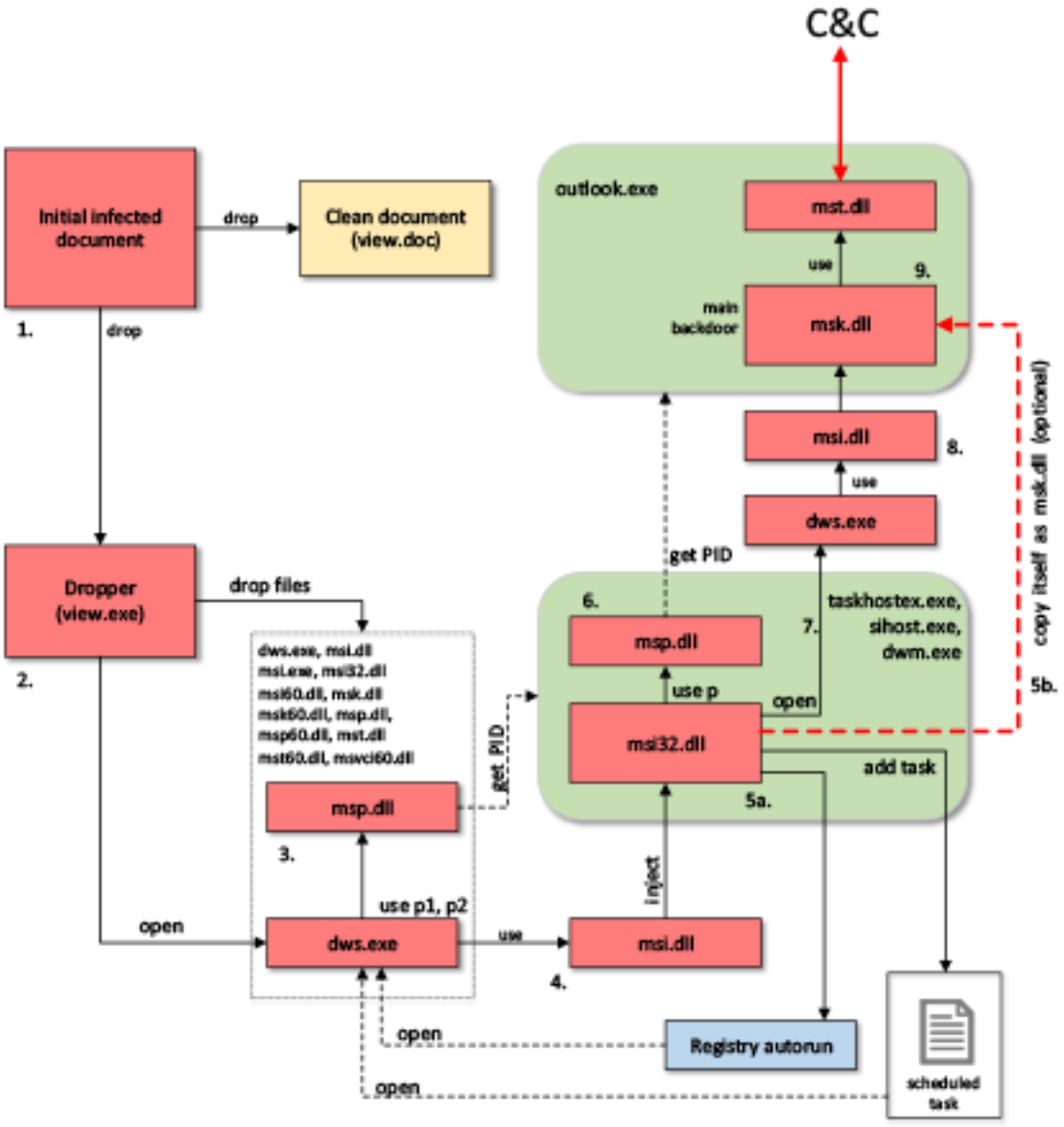
mst.dll

Library used for communication with the C&C. Very similar to the old msvct.dll. Internet connection is checked with “go.microsoft.com” .

mst60.dll

64 bit variant of mst.dll.

Functionality diagram



Functionality on 32bit Windows (2016 version)

Functionality summary:

1. The macro from the infected document will drop and open a dummy clean document and a dropper.
2. The dropper will drop the component files in the folder %appdata%\Microsoft\VisualStudio\11\ and will open dws.exe without parameters.
3. dws.exe will use the function p2 (only on Windows 8 or Windows Server 2012) or the function p1 from msp.dll. p1 will return the PID of dwm.exe, p2 will return the PID of taskhostex.exe.

4. dws.exe will load msi.dll and will call its export as msvci("msi32.dll", PID), with the PID returned from step 3. This will inject msi32.dll into dwm.exe or taskhostex.exe depending on the OS. After this dws.exe process terminates.
5. a. msi32.dll will add a run key in registry and a scheduled task, both will open dws.exe

   b. msi32.dll will copy itself to msk.dll. This step is done only in some versions where msi32.dll has the backdoor functionality and the initial backdoor msk.dll is missing.
6. msi32.dll will use the function p from msp.dll which will return the PID of one of iexplore.exe, outlook.exe, firefox.exe, chrome.exe.
7. msi32.dll will create a new process with dws.exe with a parameter, the PID returned at step 6. If at step 6. the file msp.dll was not found (selfdelete from backdoor), it will create a new process with dws.exe but without parameters, which in turn will delete all components. msi32.dll will stay in a loop repeating from step 6 (the backdoor variant of msi32.dll will exit).
8. dws.exe with a parameter will function differently and will call the export from msi.dll as msvci("msk.dll", PID), with the PID returned at step 6., received as a command line parameter. This will inject msk.dll in the specified process. After this dws.exe process terminates. At this step it does not matter if msk.dll is the msi32.dll variant or not.

9. msk.dll is the backdoor program and will use exports from mst.dll to communicate with the C&C. In case that msk.dll is msi32.dll variant the selfdelete will be done here (start dws.exe) and not in step 7. because msi32.dll will no longer run in dwm.exe, taskhostex.exe or sihost.exe.

Zip file containing malicious java-script

In this variant victim is lured to double click on a file with double extension .doc.js this way java-script file gets executed, will decode a clean Word document and a malware executable file, both are embedded in java-script. Next, a windows task is created to run the malware, and clean document is opened. Malware execution follows as previously described.

C&C

We have 4 unique C&Cs for 2016 variants, hosted in Netherlands, New York and Germany. The machines used are most likely compromised web servers.

Binary difference between 2014/2015 and 2016 variants

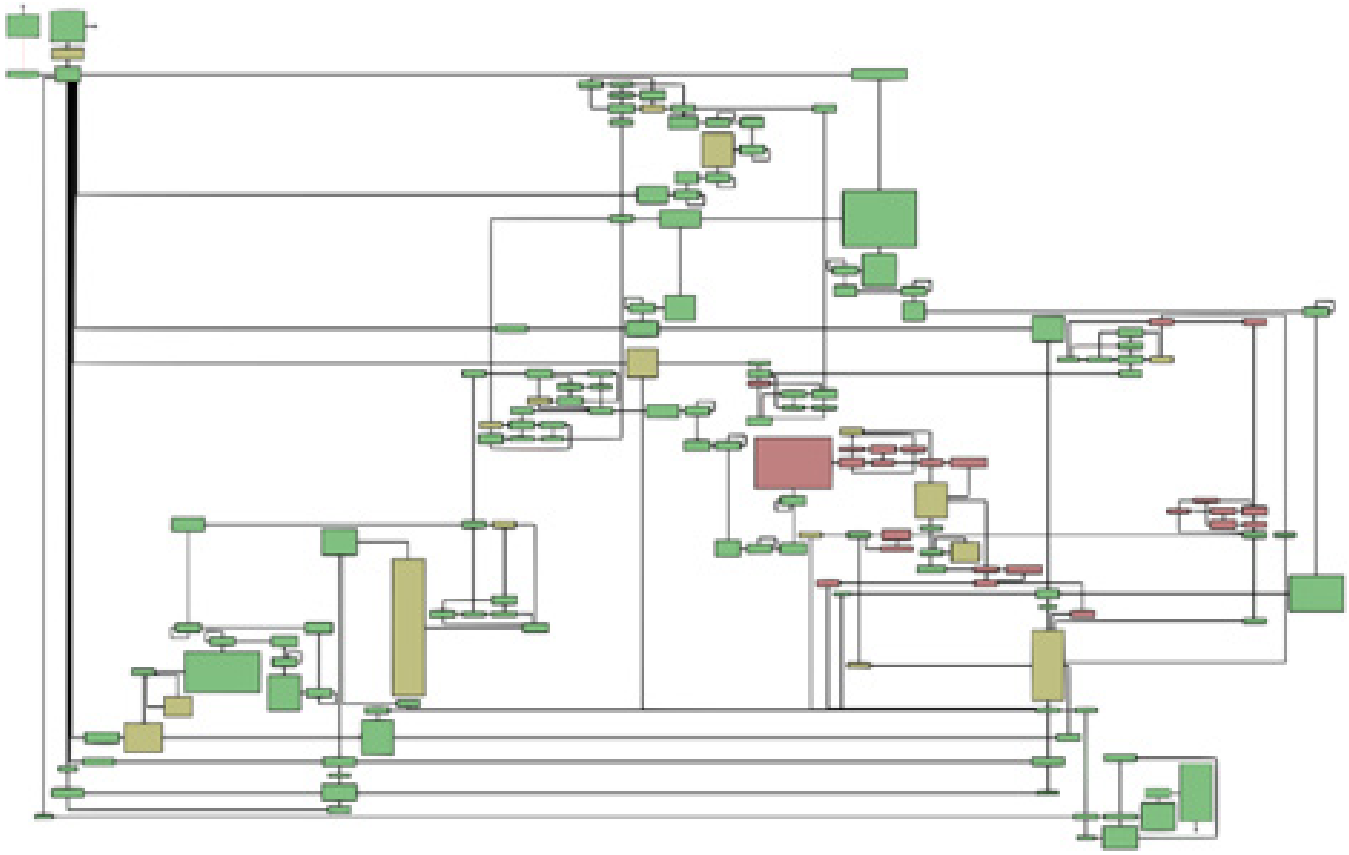


diagram representing backdoor main function

Yellow blocks represent partial code modifications compared to 2015 versions. Red blocks represent added functionality. We can see that little functionality was added to backdoor component.

IOCs

File paths

```
%APPDATA%\Microsoft\Word\MSWord.exe
%APPDATA%\Axpim\ubfic.exe (random)
%APPDATA%\Axpim\anfel.js  (random)
%APPDATA%\Nuuw\ilebi.xpi  (random)
%APPDATA%\Nuuw\yqyra.js   (random)

%TEMP%\ntlm.exe
%TEMP%\msvci.dll
%TEMP%\msvcp.dll
%TEMP%\msvck.dll
%TEMP%\msvct.dll
%TEMP%\msvci.exe   (64bit)
%TEMP%\msvck60.dll (64bit)
%TEMP%\msvct60.dll (64bit)

%APPDATA%\Microsoft\VisualStudio\11.0\dws.exe
%APPDATA%\Microsoft\VisualStudio\11.0\msi.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msi.exe
%APPDATA%\Microsoft\VisualStudio\11.0\msi32.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msi60.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msk.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msk60.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msp.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msp60.dll
%APPDATA%\Microsoft\VisualStudio\11.0\mst.dll
%APPDATA%\Microsoft\VisualStudio\11.0\mst60.dll
%APPDATA%\Microsoft\VisualStudio\11.0\msvci60.dll
%APPDATA%\Axpim\selfdel.bat
%TEMP%\xmlupd.bat
```

pipes

```
\\.\pipe\bc367
\\.\pipe\bc31a7
```

Registry paths

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svchostUpdate
-> %TEMP%\ntlm.exe
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Devices
-> %TEMP%\ntlm.exe
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\svchostUpdate
-> %TEMP%\svchost.exe
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Devices
-> %TEMP%\svchost.exe
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\dwm service
-> %TEMP%\dwms.exe
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Devices
-> %TEMP%\dwms.exe
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\dwupdate
-> %APPDATA%\Microsoft\VisualStudio\11.0\dws.exe
```

tasks

update

• command schtasks /create /SC DAILY /ST 12:00 /TN update /F /TR %APPDATA%\Microsoft\VisualStudio\11.0\dws.exe

network activity

2014-2015 variants:

reckless.dk/wp-includes/class-pomo.php

reckless.dk/wp-includes/class.wp-db.php

fishstalk.esy.es/wp-content/plugins/bbpress/includes/common/menu.php

fishstalk.esy.es/wp-includes/SimplePie/Net/IPv4.php

77-ufo.com/wp-includes/class-menu.php

77-ufo.com/pma/db_table.php

scientific.otzo.com/rss.php

SHA1 hashes of all known variants

0641f22e1b4e15cc23660b2e8bbf42623e997dfb
0af1a6d6c487e78aa252ae2f5921606a8a379206
100241519698bb013f668ff49d3d0d4fdab6a584
14014f810a0c07b6dde48b7a8954b56c409ae7f3
16c6d317fd7c361623c62cf5652a6b7937f58e0a
22c565e2cfb8adadd022b0ec281bb2b6ed62dca2
23ce92fd1d4d2d42389a66869434fb578aa3f539
261a8fc8e0e396298120a7bc15c32a37f3ce5b94
2a9c8639215faf08593f17b930f83757324dfbee
2ad7262ad52320399aa54cd8482c30e7a480bebc
2eb5a075b710155c409e727e7f74fdc3be63b58c
325b1075b4544ecc2c5741a7a06a9df00f0965da
336d59579094879900 33a3432d0347be34db044a
368b746daf5448812b231aed67bd795dfb5a605d
38d16c19b54bf2c94e0ad81fca207de062181b31
4880a13c4e1cde0343c233f5e107abf4e3d00664
49f0569886e5e6ba4b32b7f118dc35f9e5916dc2
4eecebf5c9720c8e85347e0dcf55a844a6d01b08
5374b898dbb618aa84d92f7a3e9d166e9e819960
54ac8caeae8046e01301379602041c74ee527dfc
5617c1414cb79411c64883ee72d219d52123fa30
583036a7c9b210508c222c7dfdd9b8321feca7df
58952be65d0ed53490f69f566485c699f246dcc0
5a6b14fad221ab65a086b1ee7c97eb63ff38480e
5aaa055fa5eb9a436ca0e643bf2ada268bcd6f33
5bcc6da122b3aa88c766d80eb7774c2c6e9e25d5
623185a651a1962538141d7ffefdc2f2445a9201
66a7642abaf3d05d5ab14e83dfd52eca0c17acc6
67e9e098c2b39b5847f6cd3aa5a3f86917602f5f
6a2d12adc541c9c5aaa1096d7e59c72c489cdd59
713855aa5680154324bfcbac638aa1c12681e3c3
7674f680fd0c24c222c027976c40ffe1e08c6f2e
7abf407b9a19dd9ee528fa6e5a099ea1c8ba2f98
80091e1b7b4dd404c83a9c54fda9e6951b2689b1
852dc73ca9e6d92b3da96500d27ab44b7f9a4ea4
85c03c6fa5e3803e55a46f17d6981992181de57b
88af035dc34f730c884b5a11c8be666974a1a6eb
88fd1ee6fb78385a1c5e462dd0768bc34b8188a3
8c4dd73cdd48908ddf5039c5a99e719dfd44ff41
8d40a65a2bca1378eb6e009c1842aa0e45ae289e
a5359856742d09d1596e5c7fde407856d72046db
a9239572afe4fbdfe077a262c9699eb1d22a9c87
acc2250be782063f268b87bd0f798549c5838b95
aecf66120861b71c92a2d1f0015fc9228c02ee88
b2700f16e4494ef7eba26b88a800728621adffea
b4afc5e0002201ce052466cba9061018474b1de0
b55dac24f646dd5e0ea856d6ed7891ad8c8acdc1
b84ef6480d888b560b071e1f97e78f06080dae89
c340534b8eafed85fc6e9950033b0b9e696d5cb0
c4b06021c6c925c837dab3ba42c6b76eb77ad30b
c5166d1a574bc5e374490846f2584f94f755d90b
c9b1208be2aa2c5cfbcbfcb9b1a45c36854414b8
ce234ed0899c8f97e3f2085215b842723a773368
d80d5ccb9d37d971a408d3c91f803e40b8421a2c
d83d7de186fa6c7abe4676eb568ba4dc62a7c931
e20b0f03f6708118bca9f408b156b210ba083b54
eacadedc31af04ef86470aec62ad3eccc9a35332
eb0f02e36e77221366becabc60e78dd43368ab9d
eb1b83825ff28de7f13812bfce273ad7fb1994fb
ede8ec9f3efeb515859becd1f430f82933b42dd9
edf96c42f4e1cf43fbaab3f0bbf54280fc8e311d
eea9fec97dca5d122069adf6dd71628bd6d9c2fd
f9af4a51616db485adc577ad600b60e77916cace
fbd538cf432f2576b37e2770f860b70b009c3cf3
01e2e16be5828ca03c6b78f253bd962bfaa5ccbf
09df1b0abd32791c3b0d5d657cd956f81e2dacb2
0a9dd2b71df68ba088d7d868d7e191875755e34c
14b6f2bc2b869d3417619201c7205e240a93d2ef
1ae10d6ec5d33b704c32ef52c3ee9671f4298d5f
1e49924afe56e3c782893118a51256ca5f247fba
23d5cc54641f56f554890bbd55d580e5c564e197
26f8d64038439c006f12ec34b035b1dee1c56b31
272c42bcdcc88adba1e01e60a931fbe5f5800883
2a84f90ed23a569defee7b37f4650aca4021a767
2bf06a003a9bd56d2ed91770966a7aee7d9784b9
321ca51b4c250515bc3075abe735e360a57dee22
33f57151a52666ca055f1dc66ef04e2f9cb09918
3e10fd3e8d4c4a7900e603aee7660c83441d998e
3fbaf98c75992db9db11d29ae20c13b7b0f50470
443551d822eba6a81b8ac3177e31e210c99934d0
45c7f3f065cf015289ab17161a1880eb638b508a
46f1b8722f8f094015c749599e94a3e44850df0a
4f35665e689bea4f116505f81ae2906fd1517128
547f525f57f3f47222ae3ab253635df936bd355a
585550816539b73dfdc3cee80cc60e1cdc1cdb3e
5d492ae763bfc227db9eea46e560124128ff925b
5ff776d23e6c6af47619ad2e7333a434b79e19df
621698f821a2bafccad026f9f5d2fe1ac46a39ce
66ec04c005d0a1ebc218455915e31d2a2b6dd459
686ada60c898782b57ca993141b64f7c7a531c50
6c68a9df2d710187d067ecb2d0cc04358d570b52
6e070e01076a4a92f08924a405f389436003d927
78499e4694f847972576960a04f8177691a7c911
840563929f13ab05e45a8d3fb2d11e70e3cdccca
840de34aa767131eb34069e6f936dea3a48c024e
85a6e3a3fcee71ffa2aad90336960132fa8f4c4d
88f473f3d7a7eb2637754a8d0856ab888066ab08
8f8d7cd742fb843ba8cb16c2b2d6349436049ed8

8ffd436182f8d2a7ec0a66c0d6d43f71222f62b5
92731e4ed149c59a25c233635c55a87a8a22b19f
96d9cf7296f02bf4e49c0540fb84981493b61a93
9957af2dbfa04bca2a5319a216852ce4f4a17682
9b0effd20ea7239275b6cf1e02280eb67eced701
a5daecfd57f006acd15486bd544f40e4cdce3801
a753de6b2e6d3d5735fc5e90a879f1ad7e93fb0f
b0b9215e236bb47f5f0a108be97b24d20898d2fc
b35b07ad4f42493ecb19f66aba83da8e74c1bb5a
b4e867893d9d6f8b52de98ab6b41513d61f20472
ccf0a302eb264cbb5db726d61ad18ebdc0d3d012
d53eb2a6904d1fb7982bb876916cd3723c3dc9b1
d6d3d9a56513b83db497a8d4701c2ac7270d78eb
d7218e80261517badd8090d3a5ba0a1ed21c21a2
d74d8ec530c02b1eb94203de1f641e15a72faf8d
e32832e3f0e0b8450e7bdded16c441951b171130
eda30afac2c1fa0ed2c80e8859e2556ea3dfe2ef
ee1f5ba06400fa192664f984d71b1a0cdba96d75
f781e603c55558708ac3101d0bfee2c1752693c2
fdb9d026502aa64aa23b1acb96f6d0013ef874b4

b719e1d03e860235a68dda4168f29ac4988d25de
ba29c29a35d15a668ea2ea79d1d4e56c2d67553f
bca5accb9f1d0806f8603cf74ce0ebe9519f5004
be10c837af1f25ee67440f3a33da8c650f5ab54a
c34a68c1a2d2beddbbe8ee8bd125cce14d0dc377
c3bc94b065449879c25a541d740346e060d9d6fe
c414ba1dd1f281a63e58c60eb1d8cb4ac3c4e7f0
c7accc1c4ceedc756c30ebb2f1ff9f0dbd0255b0
c8395601ea301ba083cb530dad7a44c8048eeb77
ca07bbfc5e8c15c4258f92e6e6c328b86b7b19a5

Clean documents opened by droppers

Invitation to event organized by the UK embassy in Ashgabat:

British Embassy
Ashgabat

Note No 054/2015

Her Britannic Majesty's Embassy presents its compliments to the Ministry of Foreign Affairs of Turkmenistan and to all Diplomatic Missions and International Organizations accredited in Turkmenistan, and has the honour to confirm its intention to celebrate the official birthday of Her Britannic Majesty, Queen Elizabeth II, at a reception on Tuesday 16 June 2015. The Embassy will confirm the venue for this event in due course.

Her Britannic Majesty's Embassy avails itself of this opportunity to renew to the Ministry of Foreign Affairs of Turkmenistan, and to all Diplomatic Missions and International Organizations accredited in Turkmenistan, the assurances of its highest consideration.

British Embassy
Ashgabat
23 February 2015

Car for sale:

CAR FOR SALE

BMW X5

60 000$

Model, Caroseric:	BMW X5, SUV
Fabricate yeahr:	2014
VIN:	WBAZW61070L570883
Km:	1023 km
Motor:	Diesel 3.0
Transmission:	automatic
CO2:	198
Norma Euro:	euro5
Garantie	Yes
City	Kiev
Color	Black

23rd International Caspian Oil & Gas Conference

Presentation for a real conference that took place on 1-4 June 2016 in Azerbaijan, Baku. Picture and text are taken from official page of conference organizers.

23rd International Caspian Oil & Gas Conference

THE LEADING OIL & GAS EVENT IN CASPIAN REGION

Key conference topics include:

- The role of the Caspian region in the global energy supply
- The dynamics of the gas market in Europe, the CIS and Asia
- Discussing TAP and TANAP, the gas transportation projects
- International cooperation to increase oil and gas production
- Environmental standards within the oil and gas sector
- Prospects for and development of the petrochemical industry in the Caspian region
- Investing in the Caspian Sea oil and gas industry. Opportunities for foreign investors in the region
- Shah Deniz (SD) Stage 2 Project
- Diversifying routes for transporting Caspian oil and gas to Europe
- New challenges and future prospects for Azerbaijan's oil and gas sector
- Implementing the SOCAR POLYMER project to produce polypropylene
- Turkey's role in transporting oil
- Import substitution in the oil and gas industry
- Exploring and producing oil in the Caspian Sea
- The relationship between human resources and natural resources in the Caspian Sea oil industry

Australia - Korea Foundation, foreign affairs position, interview guidelines

Data seems to be taken from Australian Government, Department of Foreign Affairs and Trade.

Topics for the interview:

Australia's engagement with the Korean peninsula

the current importance of the Australia – Republic of Korea Relationship
truth brief history of Australia`s relations with Korea

Inter - governmental relations

government and parlamentary visits
Republic of Korea Government visits to Australia
Commonwealth Government visits to the Republic of Korea
government cooperation
parlamentary delegations

Security relations

The Republic of Korea security posture
Australia-Republic of Korea shared security interests
senior level defence visits
peacekeeping
consequence management
defence industry cooperation
exercise observation and participation
defence educational exchange

The Australia-Korea Foundation

International politics

Text is a Bloomberg news story from October 24th: Bulgaria, Romania and Serbia Ready to Close Borders for Migrants.

Bulgaria, Romania and Serbia are concerned that possible closing of borders for migrants by some European Union countries may cause a bottleneck stranding millions of refugees on their territory, forcing the three states to close their borders as well.

Bulgarian Prime Minister Boyko Borissov met Romanian and Serbian counterparts Victor Ponta and Aleksandar Vucic in Sofia Saturday to coordinate their policies before EU leaders gather on Sunday to forge plans to control the region's worst migrant crisis since World War II.

"If Germany and Austria or other states close their borders for migrants, we won't allow our countries to become a buffer zone for millions of migrants stranded between Turkey and the new barriers that may follow," Borissov told reporters in Sofia. "We're also prepared to close our borders immediately."

With more than a million migrants set to reach the EU this year and cold weather settling in, national authorities have taken unilateral decisions to close borders and send asylum-seekers to neighboring countries. Civil-war-torn Syria, origin of many of the refugees, could become an even bigger exodus point as Russia pursues a bombing campaign in support of Syrian President Bashar al-Assad.

Closing Gates

"If you look at the map, Bulgaria, Serbia and Romania are the gate to Europe," Romania's Ponta said at the same briefing. "We have to act together and with all other EU states, but when someone behind these three countries raises a new gate, we'll have to reconsider our policy."

Slovenia earlier this week gave its army extra powers to help police border posts while Hungary closed its borders with Serbia and Croatia. The action has led to bottlenecks with thousands of refugees, mainly from the Middle East and north Africa, massing in outdoor border areas overnight. Bulgaria is setting up a wire fence since last year along its 160-kilometer (99-mile) border with Turkey, of which about a 30-kilometer section is in place.

"If someone thinks that we're the place, where there will be two or three million refugees, that's an unrealistic position," Serbia's Vucic said at the briefing. "Our three states agreed that they can not accept such a scenario."

The main flow of migrants fleeing conflict-stricken nations changed from a route through southern Europe to one leading from Turkey to Greece and through former war-torn nations including Croatia, Serbia and Slovenia.

Budget plan template

This one looks as a budget calculation template taken from “Relations internationals et Francophonie” of Québec.

Fiche BUDGET GLOBAL du projet **Format suggéré. Les dépenses et revenus indiqués ci-dessous ne le sont qu'à ti*

Normes du programme e admissibles

Pour connaître les dépenses admissibles, voir document

TITRE DU PROJET :

PARTIE FRANÇAISE

REVENUS	**An 1**	**An 2**
Contributions financières		
Demande de soutien déposé à la CPCFQ	- €	- €
Autre source de financement	- €	- €
Autre source de financement	- €	- €
Autre source de financement	- €	- €
Autre source de financement	- €	- €
Autre source de financement	- €	- €
	- €	- €
	- €	- €
	- €	- €
Sous-Total	- €	- €

Bitdefender is a global security technology company that delivers solutions in more than 100 countries through a network of value-added alliances, distributors and reseller partners. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a leading security provider in virtualization and cloud technologies. Through R&D, alliances and partnership teams, Bitdefender has elevated the highest standards of security excellence in both its number-one-ranked technology and its strategic alliances with the world's leading virtualization and cloud technology providers. More information is available at http://www.bitdefender.com/

FOR MORE INFORMATION VISIT: enterprise.bitdefender.com

BD-Business-Jun.30.2016-Tk#: 70585