Malware analysis Report

Yaohan Teng B00807306

1. Malware Sample Intro:

The malware samples that we are considering are coming from the Big Bing database(http://cybersec.cs.binghamton.edu/bigbing/index.html). The launch year is 2018 and its file-type is Win64 executable. Some details below have been extracted from analyzing the radare2 results.

Sample Basic details:

File type : Win64 EXE

File size : 3.1 MB (3098112 bytes)

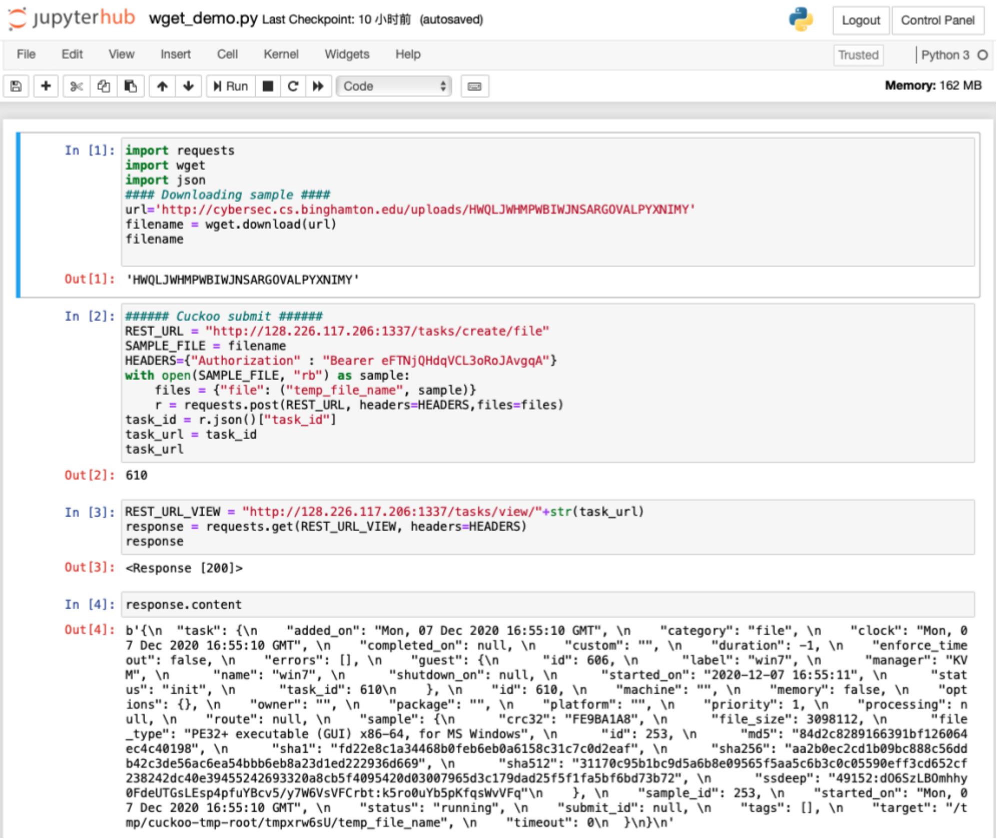
MD5: 84d2c8289166391bf126064ec4c40198

SHA-1: fd22e8c1a34468b0feb6eb0a6158c31c7c0d2eaf

SHA-256: aa2b0ec2cd1b09bc888c56ddb42c3de56ac6ea54bbb6eb8a23d1ed222936d669

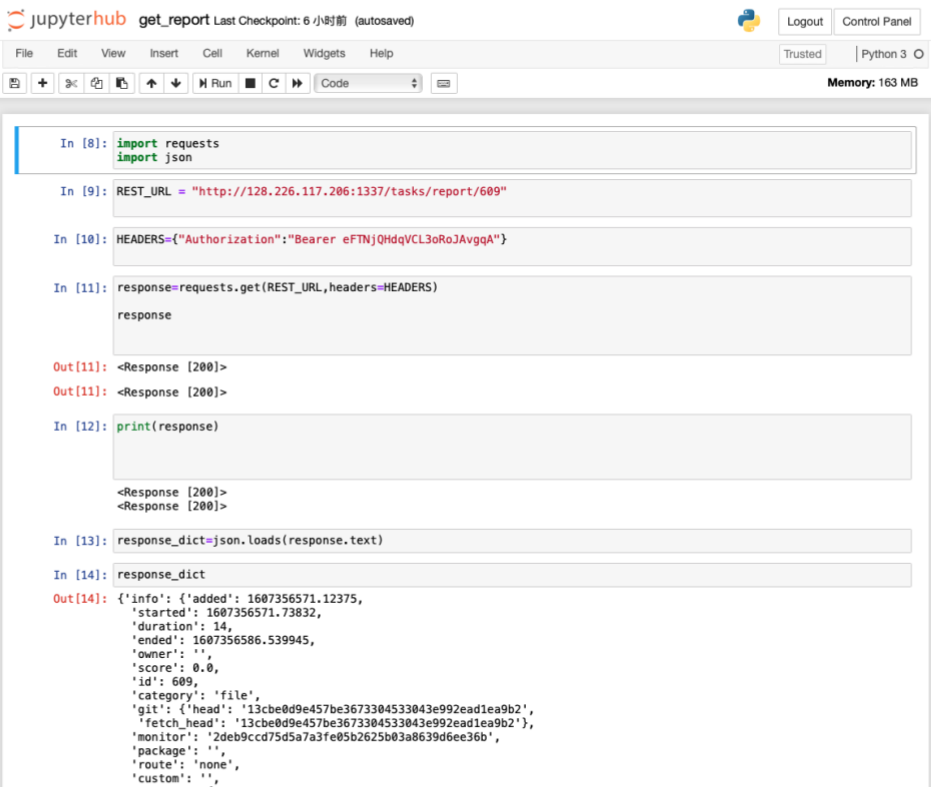
2. Analysis tools: cuckoo, Jupyterhub

(1) we use this script to get malware samples from Big Bing Database, and analysis it automatically by cuckoo.



```
In [1]: import requests
        import wget
        import json
        #### Downloading sample ####
        url='http://cybersec.cs.binghamton.edu/uploads/HWQLJWHMPWBIWJNSARGOVALPYXNIMY'
        filename = wget.download(url)
        filename

Out[1]: 'HWQLJWHMPWBIWJNSARGOVALPYXNIMY'

In [2]: ###### Cuckoo submit ######
        REST_URL = "http://128.226.117.206:1337/tasks/create/file"
        SAMPLE_FILE = filename
        HEADERS={"Authorization" : "Bearer eFTNjQHdqVCL3oRoJAvgqA"}
        with open(SAMPLE_FILE, "rb") as sample:
            files = {"file": ("temp_file_name", sample)}
            r = requests.post(REST_URL, headers=HEADERS,files=files)
        task_id = r.json()["task_id"]
        task_url = task_id
        task_url

Out[2]: 610

In [3]: REST_URL_VIEW = "http://128.226.117.206:1337/tasks/view/"+str(task_url)
        response = requests.get(REST_URL_VIEW, headers=HEADERS)
        response

Out[3]: <Response [200]>

In [4]: response.content

Out[4]: b'{\n  "task": {\n    "added_on": "Mon, 07 Dec 2020 16:55:10 GMT", \n    "category": "file", \n    "clock": "Mon, 0
        7 Dec 2020 16:55:10 GMT", \n    "completed_on": null, \n    "custom": "", \n    "duration": -1, \n    "enforce_time
        out": false, \n    "errors": [], \n    "guest": {\n      "id": 606, \n      "label": "win7", \n      "manager": "KV
        M", \n      "name": "win7", \n      "shutdown_on": null, \n      "started_on": "2020-12-07 16:55:11", \n      "stat
        us": "init", \n      "task_id": 610\n    }, \n    "id": 610, \n    "machine": "", \n    "memory": false, \n    "opt
        ions": {}, \n    "owner": "", \n    "package": "", \n    "platform": "", \n    "priority": 1, \n    "processing": n
        ull, \n    "route": null, \n    "sample": {\n      "crc32": "FE9BA1A8", \n      "file_size": 3098112, \n      "file
        _type": "PE32+ executable (GUI) x86-64, for MS Windows", \n      "id": 253, \n      "md5": "84d2c8289166391bf126064
        ec4c40198", \n      "sha1": "fd22e8c1a34468b0feb6eb0a6158c31c7c0d2eaf", \n      "sha256": "aa2b0ec2cd1b09bc888c56dd
        b42c3de56ac6ea54bbb6eb8a23d1ed222936d669", \n      "sha512": "31170c95b1bc9d5a6b8e09565f5aa5c6b3c0c05590eff3cd652cf
        238242dc40e39455242693320a8cb5f4095420d03007965d3c179dad25f5f1fa5bf6bd73b72", \n      "ssdeep": "49152:d06SzLBOmhhy
        0FdeUTGsLEsp4pfuYBcv5/y7W6VsVFCrbt:k5ro0uYb5pKfqsWvVFq"\n    }, \n    "sample_id": 253, \n    "started_on": "Mon, 0
        7 Dec 2020 16:55:10 GMT", \n    "status": "running", \n    "submit_id": null, \n    "tags": [], \n    "target": "/t
        mp/cuckoo-tmp-root/tmpxrw6sU/temp_file_name", \n    "timeout": 0\n  }\n}\n'
```

(2) Then we use following script to get report that we need to analysis.



```
In [8]: import requests
        import json

In [9]: REST_URL = "http://128.226.117.206:1337/tasks/report/609"

In [10]: HEADERS={"Authorization":"Bearer eFTNjQHdqVCL3oRoJAvgqA"}

In [11]: response=requests.get(REST_URL,headers=HEADERS)

         response

Out[11]: <Response [200]>

Out[11]: <Response [200]>

In [12]: print(response)

         <Response [200]>
         <Response [200]>

In [13]: response_dict=json.loads(response.text)

In [14]: response_dict

Out[14]: {'info': {'added': 1607356571.12375,
           'started': 1607356571.73832,
           'duration': 14,
           'ended': 1607356586.539945,
           'owner': '',
           'score': 0.0,
           'id': 609,
           'category': 'file',
           'git': {'head': '13cbe0d9e457be3673304533043e992ead1ea9b2',
            'fetch_head': '13cbe0d9e457be3673304533043e992ead1ea9b2'},
           'monitor': '2deb9ccd75d5a7a3fe05b2625b03a8639d6ee36b',
           'package': '',
           'route': 'none',
           'custom': '',
```

3. Static analysis:

Here are some of the important static information found by my observation:

1. Files Hashes – MD5, SHA-1, SHA-256

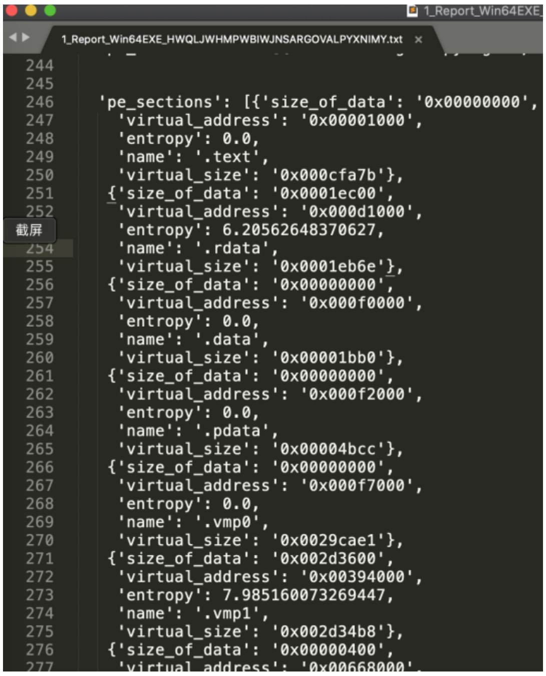
a) As above in the Intro.

2. File-type and size :

a) File-type : Win64 EXE

b) Size = 1.04 MB (1086808 bytes)

3. PE_sections: Storing what kind of files the sample has.



```
'pe_sections': [{'size_of_data': '0x00000000',
  'virtual_address': '0x00001000',
  'entropy': 0.0,
  'name': '.text',
  'virtual_size': '0x000cfa7b'},
 {'size_of_data': '0x0001ec00',
  'virtual_address': '0x000d1000',
  'entropy': 6.20562648370627,
  'name': '.rdata',
  'virtual_size': '0x0001eb6e'},
 {'size_of_data': '0x00000000',
  'virtual_address': '0x000f0000',
  'entropy': 0.0,
  'name': '.data',
  'virtual_size': '0x00001bb0'},
 {'size_of_data': '0x00000000',
  'virtual_address': '0x000f2000',
  'entropy': 0.0,
  'name': '.pdata',
  'virtual_size': '0x00004bcc'},
 {'size_of_data': '0x00000000',
  'virtual_address': '0x000f7000',
  'entropy': 0.0,
  'name': '.vmp0',
  'virtual_size': '0x0029cae1'},
 {'size_of_data': '0x002d3600',
  'virtual_address': '0x00394000',
  'entropy': 7.985160073269447,
  'name': '.vmp1',
  'virtual_size': '0x002d34b8'},
 {'size_of_data': '0x00000400',
  'virtual_address': '0x00668000',
```

4. Dynamic analysis:

Here are some aspects that I used for analysis:

(1) Behavior

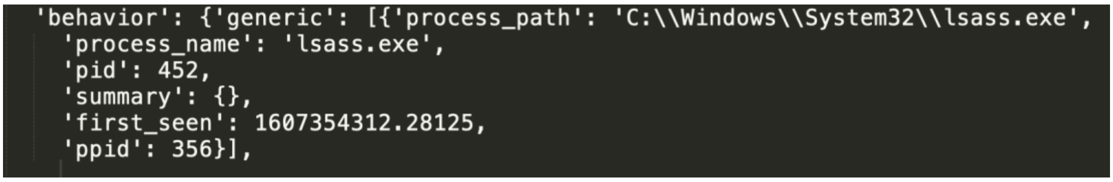
(2) Processes:

(3) Process-tree:

(4) Debug:

(5) Strings

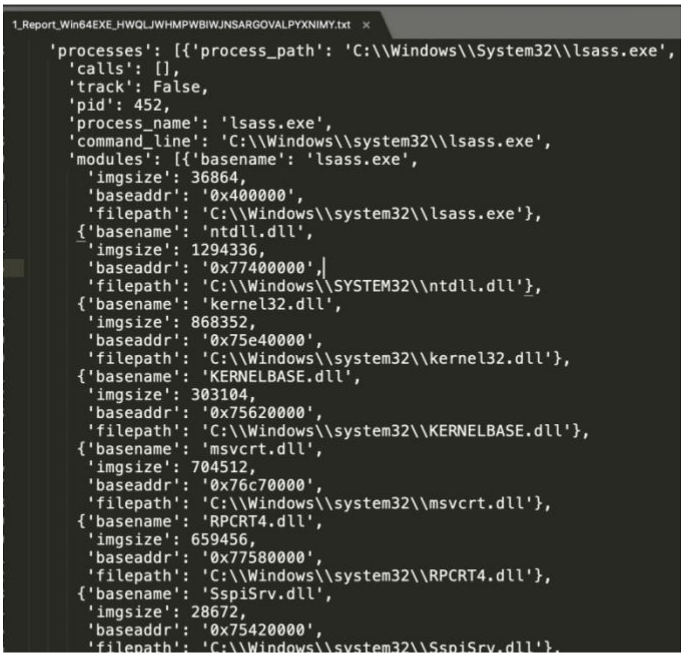
1. Behavior: what did the malware sample do, here it used system call and we can see process name and pid and ppid:



```
'behavior': {'generic': [{'process_path': 'C:\\Windows\\System32\\lsass.exe',
  'process_name': 'lsass.exe',
  'pid': 452,
  'summary': {},
  'first_seen': 1607354312.28125,
  'ppid': 356}],
```

2. Processes: we can see a lot of things in this part:

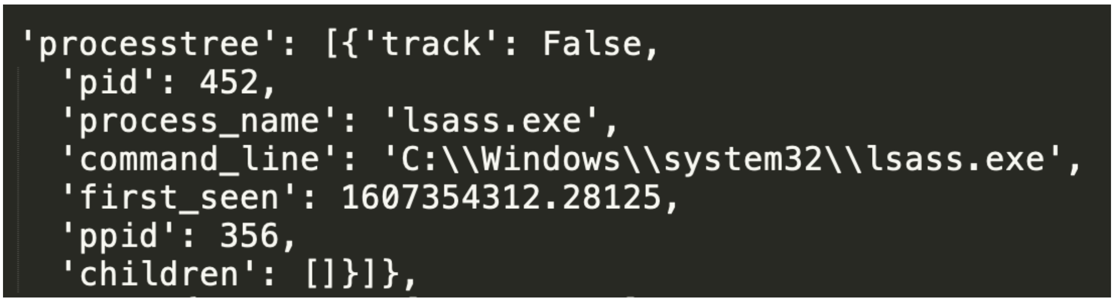
Like what kind of dlls and exes this sample used, we can know their base-name, img-size, base-addr, filepath.



1_Report_Win64EXE_HWQLJWHMPWBIWJNSARGOVALPYXNIMY.txt

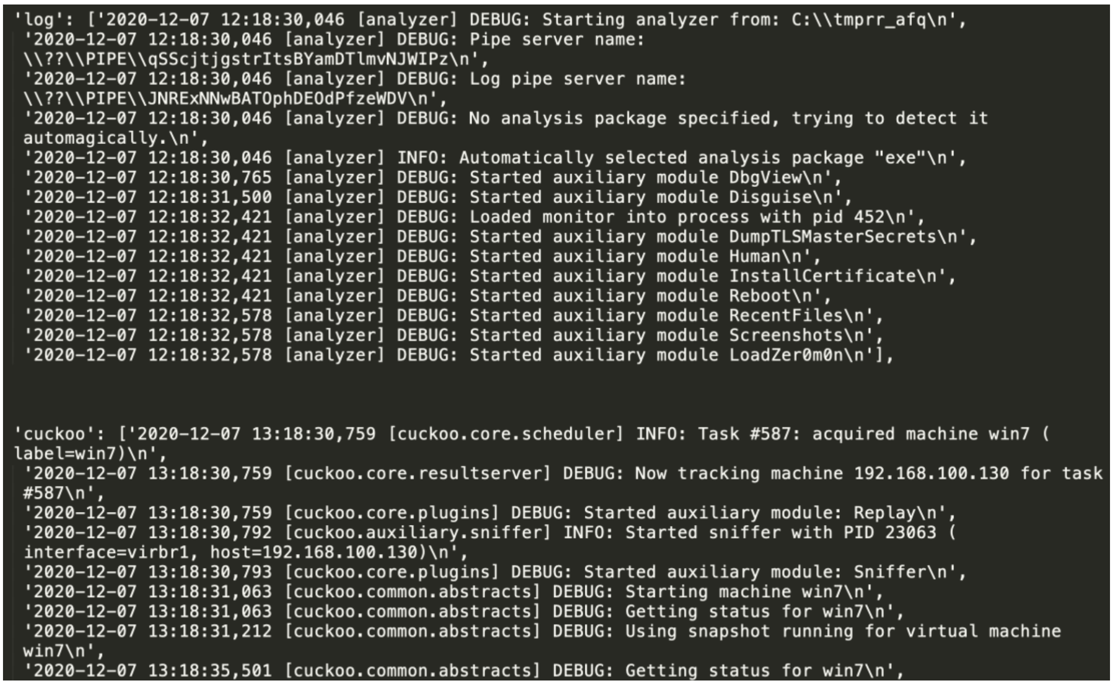
```
'processes': [{'process_path': 'C:\\Windows\\System32\\lsass.exe',
  'calls': [],
  'track': False,
  'pid': 452,
  'process_name': 'lsass.exe',
  'command_line': 'C:\\Windows\\system32\\lsass.exe',
  'modules': [{'basename': 'lsass.exe',
    'imgsize': 36864,
    'baseaddr': '0x400000',
    'filepath': 'C:\\Windows\\system32\\lsass.exe'},
   {'basename': 'ntdll.dll',
    'imgsize': 1294336,
    'baseaddr': '0x77400000',
    'filepath': 'C:\\Windows\\SYSTEM32\\ntdll.dll'},
   {'basename': 'kernel32.dll',
    'imgsize': 868352,
    'baseaddr': '0x75e40000',
    'filepath': 'C:\\Windows\\system32\\kernel32.dll'},
   {'basename': 'KERNELBASE.dll',
    'imgsize': 303104,
    'baseaddr': '0x75620000',
    'filepath': 'C:\\Windows\\system32\\KERNELBASE.dll'},
   {'basename': 'msvcrt.dll',
    'imgsize': 704512,
    'baseaddr': '0x76c70000',
    'filepath': 'C:\\Windows\\system32\\msvcrt.dll'},
   {'basename': 'RPCRT4.dll',
    'imgsize': 659456,
    'baseaddr': '0x77580000',
    'filepath': 'C:\\Windows\\system32\\RPCRT4.dll'},
   {'basename': 'SspiSrv.dll',
    'imgsize': 28672,
    'baseaddr': '0x75420000',
    'filepath': 'C:\\Windows\\system32\\SspiSrv.dll'},
```

3.Process-tree:



```
'processtree': [{'track': False,
  'pid': 452,
  'process_name': 'lsass.exe',
  'command_line': 'C:\\Windows\\system32\\lsass.exe',
  'first_seen': 1607354312.28125,
  'ppid': 356,
  'children': []}]},
```

4. Debug: This part is created by some build in software in cuckoo. For example, we can see log, which recorded what the analyzer did in the recent analysis.



```
'log': ['2020-12-07 12:18:30,046 [analyzer] DEBUG: Starting analyzer from: C:\\tmprr_afq\n',
 '2020-12-07 12:18:30,046 [analyzer] DEBUG: Pipe server name:
 \\??\\PIPE\\qSScjtjgstrItsBYamDTlmvNJWIPz\n',
 '2020-12-07 12:18:30,046 [analyzer] DEBUG: Log pipe server name:
 \\??\\PIPE\\JNRExNNwBATOphDEOdPfzeWDV\n',
 '2020-12-07 12:18:30,046 [analyzer] DEBUG: No analysis package specified, trying to detect it
 automagically.\n',
 '2020-12-07 12:18:30,046 [analyzer] INFO: Automatically selected analysis package "exe"\n',
 '2020-12-07 12:18:30,765 [analyzer] DEBUG: Started auxiliary module DbgView\n',
 '2020-12-07 12:18:31,500 [analyzer] DEBUG: Started auxiliary module Disguise\n',
 '2020-12-07 12:18:32,421 [analyzer] DEBUG: Loaded monitor into process with pid 452\n',
 '2020-12-07 12:18:32,421 [analyzer] DEBUG: Started auxiliary module DumpTLSMasterSecrets\n',
 '2020-12-07 12:18:32,421 [analyzer] DEBUG: Started auxiliary module Human\n',
 '2020-12-07 12:18:32,421 [analyzer] DEBUG: Started auxiliary module InstallCertificate\n',
 '2020-12-07 12:18:32,421 [analyzer] DEBUG: Started auxiliary module Reboot\n',
 '2020-12-07 12:18:32,578 [analyzer] DEBUG: Started auxiliary module RecentFiles\n',
 '2020-12-07 12:18:32,578 [analyzer] DEBUG: Started auxiliary module Screenshots\n',
 '2020-12-07 12:18:32,578 [analyzer] DEBUG: Started auxiliary module LoadZer0m0n\n'],



'cuckoo': ['2020-12-07 13:18:30,759 [cuckoo.core.scheduler] INFO: Task #587: acquired machine win7 (
label=win7)\n',
 '2020-12-07 13:18:30,759 [cuckoo.core.resultserver] DEBUG: Now tracking machine 192.168.100.130 for task
 #587\n',
 '2020-12-07 13:18:30,759 [cuckoo.core.plugins] DEBUG: Started auxiliary module: Replay\n',
 '2020-12-07 13:18:30,792 [cuckoo.auxiliary.sniffer] INFO: Started sniffer with PID 23063 (
 interface=virbr1, host=192.168.100.130)\n',
 '2020-12-07 13:18:30,793 [cuckoo.core.plugins] DEBUG: Started auxiliary module: Sniffer\n',
 '2020-12-07 13:18:31,063 [cuckoo.common.abstracts] DEBUG: Starting machine win7\n',
 '2020-12-07 13:18:31,063 [cuckoo.common.abstracts] DEBUG: Getting status for win7\n',
 '2020-12-07 13:18:31,212 [cuckoo.common.abstracts] DEBUG: Using snapshot running for virtual machine
 win7\n',
 '2020-12-07 13:18:35,501 [cuckoo.common.abstracts] DEBUG: Getting status for win7\n',
```

5. Strings:

a) **There are a lot of strings showed in the report, and I found some interesting strings that I think are important to analysis:**

b) **Here are some examples:**

(1) This URL is related to a game platform called steam, I'm familiar to it.

'https://steamcommunity.com/tradeoffer/new/?partner=140443144&token=yVL0mOB6',

(2) This string maybe related to bitcoincash:

bitcoincash:qrdeh9fzdt4uu55rp4fs2y89p37fg0jr95z03e4m0w',

(3) Here are some names of some files:

'config.txt', 'pools.txt', 'amd.txt', 'nvidia.txt', 'cpu.txt', 'cache.txt',

(4) Here are some executable files of the games on steam:

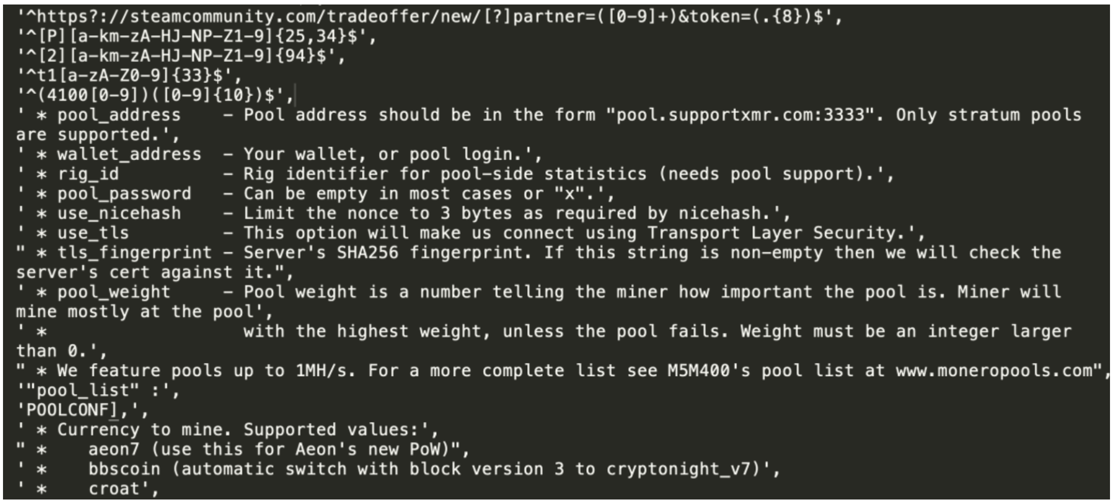
dota2.exe, csgo.exe, payday.exe, Minecraft.exe', 'TheDivision.exe', GTA5.exe',

(5) I think this is related to the wallet and trade, like when people buying games in the game platform, then if they use this malicious sample on their computer and steam log in at the meantime, people may lost money.

a)'/getinfo.php?getwallets=russia',

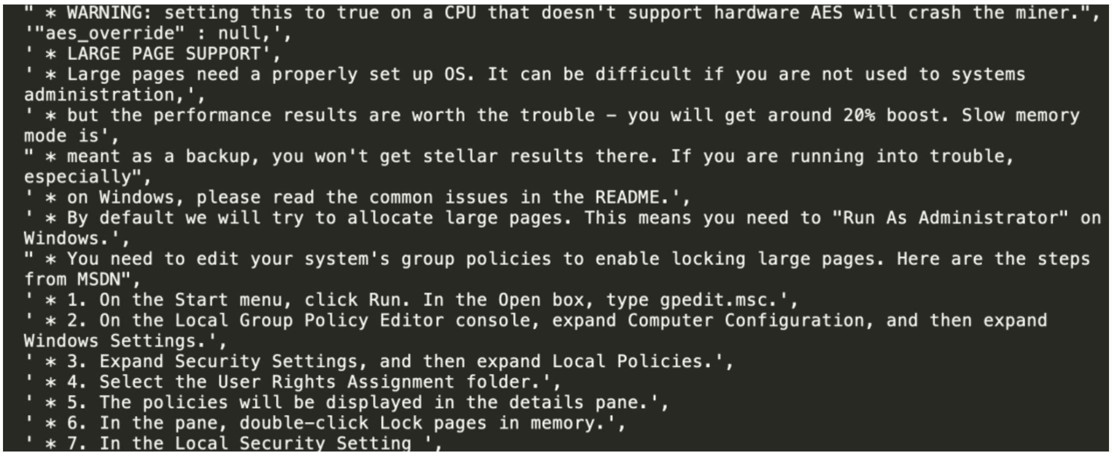
b)'^https?://steamcommunity.com/tradeoffer/new/[?]partner=([0-9]+)&token=(.{8})$',

(6) It seems the malicious sample was trying to get your information about your steam account, which means if you have your steam log in, you may lose your account.



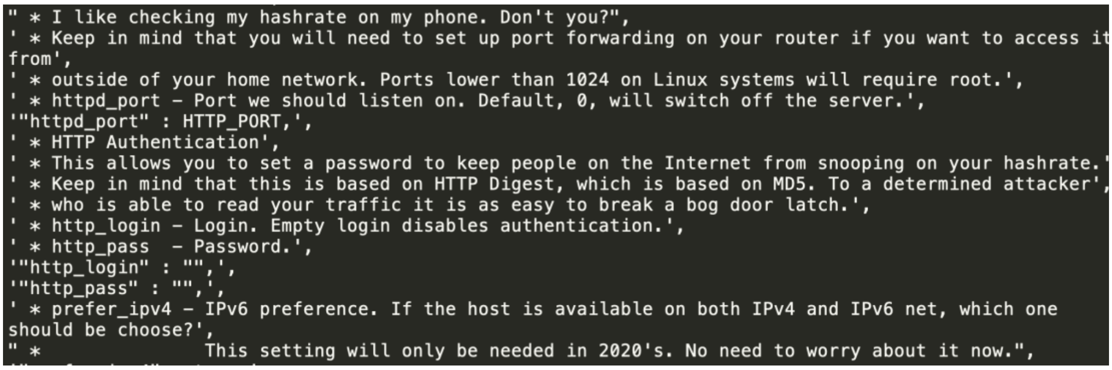
```
'^https?://steamcommunity.com/tradeoffer/new/[?]partner=([0-9]+)&token=(.{8})$',
'^[P][a-km-zA-HJ-NP-Z1-9]{25,34}$',
'^[2][a-km-zA-HJ-NP-Z1-9]{94}$',
'^t1[a-zA-Z0-9]{33}$',
'^(4100[0-9])([0-9]{10})$',
' * pool_address    - Pool address should be in the form "pool.supportxmr.com:3333". Only stratum pools
are supported.',
' * wallet_address  - Your wallet, or pool login.',
' * rig_id          - Rig identifier for pool-side statistics (needs pool support).',
' * pool_password   - Can be empty in most cases or "x".',
' * use_nicehash    - Limit the nonce to 3 bytes as required by nicehash.',
' * use_tls         - This option will make us connect using Transport Layer Security.',
" * tls_fingerprint - Server's SHA256 fingerprint. If this string is non-empty then we will check the
server's cert against it.",
' * pool_weight     - Pool weight is a number telling the miner how important the pool is. Miner will
mine mostly at the pool',
' *                   with the highest weight, unless the pool fails. Weight must be an integer larger
than 0.',
" * We feature pools up to 1MH/s. For a more complete list see M5M400's pool list at www.moneropools.com",
'"pool_list" :',
'POOLCONF],',
' * Currency to mine. Supported values:',
" *    aeon7 (use this for Aeon's new PoW)",
' *    bbscoin (automatic switch with block version 3 to cryptonight_v7)',
' *    croat',
```

(7) In the following pic, you can see something called miner, which is related to what we analysis in last part: bitcoincash. And we can see u can get 20 % boost.

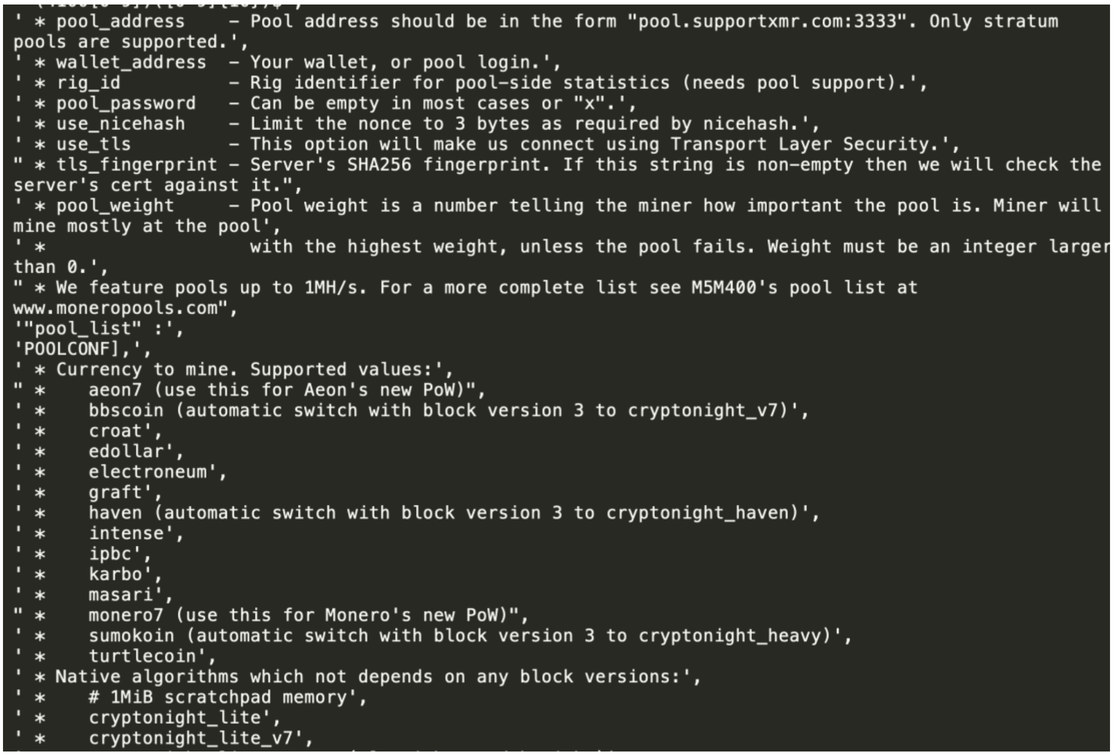


```
" * WARNING: setting this to true on a CPU that doesn't support hardware AES will crash the miner.",
'"aes_override" : null,',
' * LARGE PAGE SUPPORT',
' * Large pages need a properly set up OS. It can be difficult if you are not used to systems
administration,',
' * but the performance results are worth the trouble - you will get around 20% boost. Slow memory
mode is',
" * meant as a backup, you won't get stellar results there. If you are running into trouble,
especially",
' * on Windows, please read the common issues in the README.',
' * By default we will try to allocate large pages. This means you need to "Run As Administrator" on
Windows.',
" * You need to edit your system's group policies to enable locking large pages. Here are the steps
from MSDN",
' * 1. On the Start menu, click Run. In the Open box, type gpedit.msc.',
' * 2. On the Local Group Policy Editor console, expand Computer Configuration, and then expand
Windows Settings.',
' * 3. Expand Security Settings, and then expand Local Policies.',
' * 4. Select the User Rights Assignment folder.',
' * 5. The policies will be displayed in the details pane.',
' * 6. In the pane, double-click Lock pages in memory.',
' * 7. In the Local Security Setting ',
```

(8) We can see the following part is kind of a tutorial for people who use this malware sample and how to avoid being detected.



```
" * I like checking my hashrate on my phone. Don't you?",
' * Keep in mind that you will need to set up port forwarding on your router if you want to access it
from',
' * outside of your home network. Ports lower than 1024 on Linux systems will require root.',
' * httpd_port - Port we should listen on. Default, 0, will switch off the server.',
'"httpd_port" : HTTP_PORT,',
' * HTTP Authentication',
' * This allows you to set a password to keep people on the Internet from snooping on your hashrate.'
' * Keep in mind that this is based on HTTP Digest, which is based on MD5. To a determined attacker',
' * who is able to read your traffic it is as easy to break a bog door latch.',
' * http_login - Login. Empty login disables authentication.',
' * http_pass  - Password.',
'"http_login" : "",',
'"http_pass" : "",',
' * prefer_ipv4 - IPv6 preference. If the host is available on both IPv4 and IPv6 net, which one
should be choose?',
" *              This setting will only be needed in 2020's. No need to worry about it now.",
```



```
' * pool_address    - Pool address should be in the form "pool.supportxmr.com:3333". Only stratum
pools are supported.',
' * wallet_address  - Your wallet, or pool login.',
' * rig_id          - Rig identifier for pool-side statistics (needs pool support).',
' * pool_password   - Can be empty in most cases or "x".',
' * use_nicehash    - Limit the nonce to 3 bytes as required by nicehash.',
' * use_tls         - This option will make us connect using Transport Layer Security.',
" * tls_fingerprint - Server's SHA256 fingerprint. If this string is non-empty then we will check the
server's cert against it.",
' * pool_weight     - Pool weight is a number telling the miner how important the pool is. Miner will
mine mostly at the pool',
' *                   with the highest weight, unless the pool fails. Weight must be an integer larger
than 0.',
" * We feature pools up to 1MH/s. For a more complete list see M5M400's pool list at
www.moneropools.com",
'"pool_list" :',
'POOLCONF],',
' * Currency to mine. Supported values:',
" *    aeon7 (use this for Aeon's new PoW)",
' *    bbscoin (automatic switch with block version 3 to cryptonight_v7)',
' *    croat',
' *    edollar',
' *    electroneum',
' *    graft',
' *    haven (automatic switch with block version 3 to cryptonight_haven)',
' *    intense',
' *    ipbc',
' *    karbo',
' *    masari',
" *    monero7 (use this for Monero's new PoW)",
' *    sumokoin (automatic switch with block version 3 to cryptonight_heavy)',
' *    turtlecoin',
' * Native algorithms which not depends on any block versions:',
' *    # 1MiB scratchpad memory',
' *    cryptonight_lite',
' *    cryptonight_lite_v7',
```

And in the above part, we can see pool_address, pool_password, pool_list, pool_weight, pool_CONF etc. And I found a sentence here : we feature pools up to 1MH/s. For a more complete list see M5M400's pool list at www.moneropools.com", I found a URL here, I think this URL is related to some malicious operations. So I googled this URL and it just like what I thought in the earlier part: It's related to bit coin and digging bit coin.

5. Conclusion:

According to the analysis that I made in the earlier parts.

I have a conclusion:

This is a malware sample that executed on windows operating system, which is used for digging bit coin by using hosts' computers.

People need to execute this malware sample and log in their steam account. then when people are playing games, the malware sample program will use people's PC to dig bit coin at meantime.