Cloud Atlas: RedOctober APT is back in style

Two years ago, we published our research into RedOctober, a complex cyber-espionage operation targeting diplomatic embassies worldwide. We named it RedOctober because we started this investigation in October 2012, an unusually hot month.

After our announcement in January 2013, the RedOctober operation was promptly shut down and the network of C&Cs was dismantled. As usually happens with these big operations, considering the huge investment and number of resources behind it, they don't just "go away" forever. Normally, the group goes underground for a few months, redesigns the tools and the malware and resume operations.

See:

- RedOctober Part 1
- RedOctober Part 2

Since January 2013, we've been on the lookout for a possible RedOctober comeback. One possible hit was triggered when we observed Mevade, an unusual piece of malware that appeared late in 2013. The Mevade C&C name styles as well as some other technical similarities indicated a connection to RedOctober, but the link was weak. It wasn't until August 2014 that we observed something which made us wonder if RedOctober is back for good.

Meet Cloud Atlas

In August 2014, some of our users observed targeted attacks with a variation of CVE-2012-0158 and an unusual set of malware. We did a quick analysis of the malware and it immediately stood out because of certain unusual things that are not very common in the APT world.

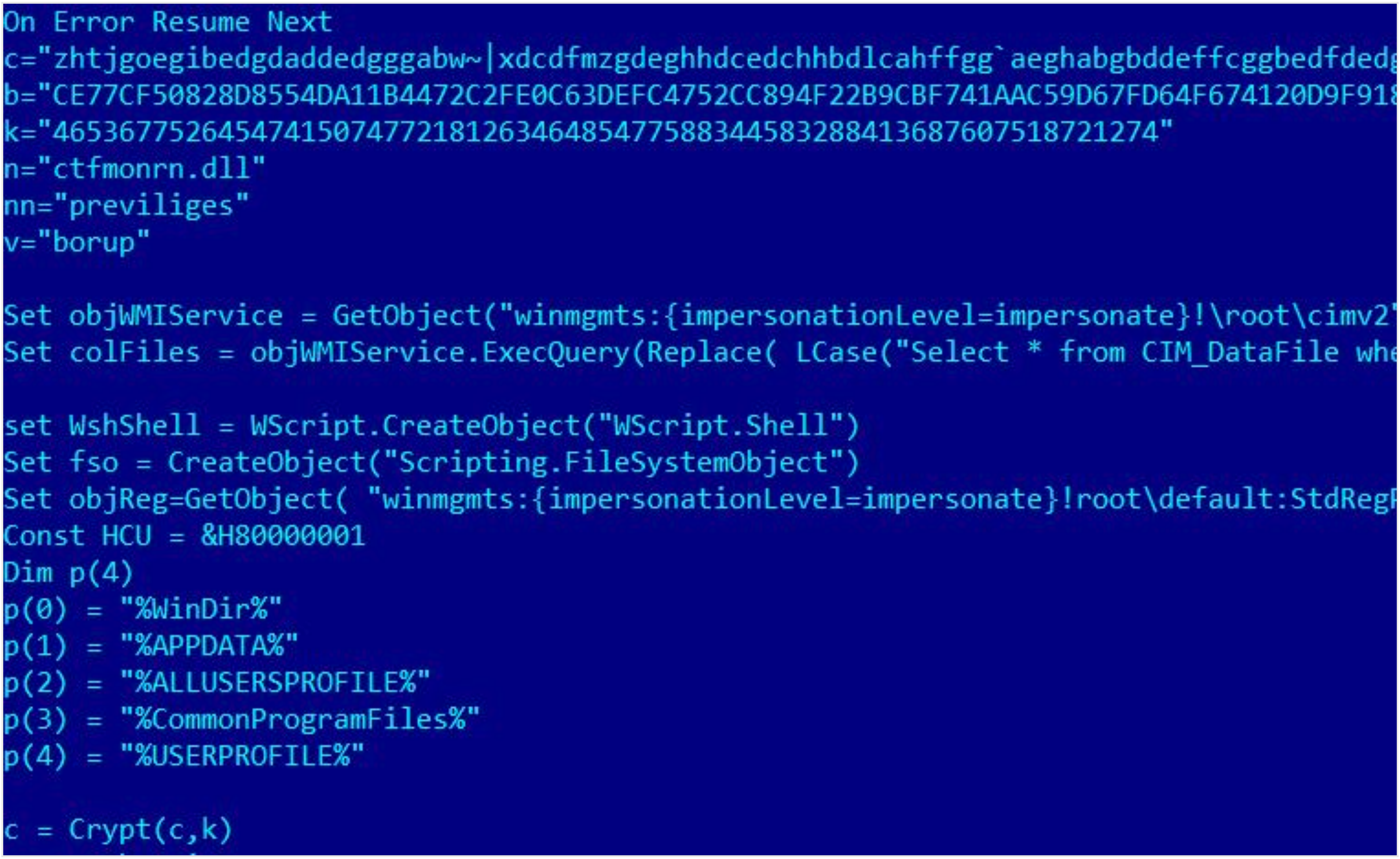
Some of the filenames used in the attacks included:

- FT - Ukraine Russia's new art of war.doc
- Катастрофа малайзийского лайнера.doc
- Diplomatic Car for Sale.doc
- МВКСИ.doc
- Organigrama Gobierno Rusia.doc
- Фото.doc
- Информационное письмо.doc
- Форма заявки (25-26.09.14).doc
- Информационное письмо.doc

- Письмо_Руководителям.doc
- Прилож.doc
- Car for sale.doc
- Af-Pak and Central Asia's security issues.doc

At least one of them immediately reminded us of **RedOctober**, which used a very similarly named spearphish: "Diplomatic Car for Sale.doc". As we started digging into the operation, more details emerged which supported this theory.

Perhaps the most unusual fact was that the Microsoft Office exploit didn't directly write a Windows PE backdoor on disk. Instead, it writes an encrypted Visual Basic Script and runs it.



```
On Error Resume Next
c="zhtjgoegibedgdaddedgggabw~|xdcdfmzgdeghhdcedchhbdlcahffgg`aeghabgbddeffcggbedfded
b="CE77CF50828D8554DA11B4472C2FE0C63DEFC4752CC894F22B9CBF741AAC59D67FD64F674120D9F91
k="465367752645474150747721812634648547758834458328841368760751872127 4"
n="ctfmonrn.dll"
nn="previliges"
v="borup"

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\root\cimv2
Set colFiles = objWMIService.ExecQuery(Replace( LCase("Select * from CIM_DataFile whe

set WshShell = WScript.CreateObject("WScript.Shell")
Set fso = CreateObject("Scripting.FileSystemObject")
Set objReg=GetObject( "winmgmts:{impersonationLevel=impersonate}!root\default:StdReg
Const HCU = &H80000001
Dim p(4)
p(0) = "%WinDir%"
p(1) = "%APPDATA%"
p(2) = "%ALLUSERSPROFILE%"
p(3) = "%CommonProgramFiles%"
p(4) = "%USERPROFILE%"

c = Crypt(c,k)
```

Cloud Atlas exploit payload - VBScript

This VBScript drops a pair of files on disk - a loader and an encrypted payload. The loader appears to be different every time and internal strings indicate it is "polymorphically" generated. The payload is always encrypted with a unique key, making it impossible to decrypt unless the DLL is available.

We observed several different spear-phishing documents that drop uniquely named payloads. For instance, the "**qPdoaKJu.vbs**" file MD5:

E211C2BAD9A83A6A4247EC3959E2A730 drops the following files:

DECF56296C50BD3AE10A49747573A346 - bicorporate - encrypted payload
D171DB37EF28F42740644F4028BCF727 - ctfmonrn.dll - loader

The VBS also adds a registry key:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\ setting the key "bookstore" to the value "regsvr32 %path%\ctfmonrn.dll /s", which ensures the malware runs every time at system boot.

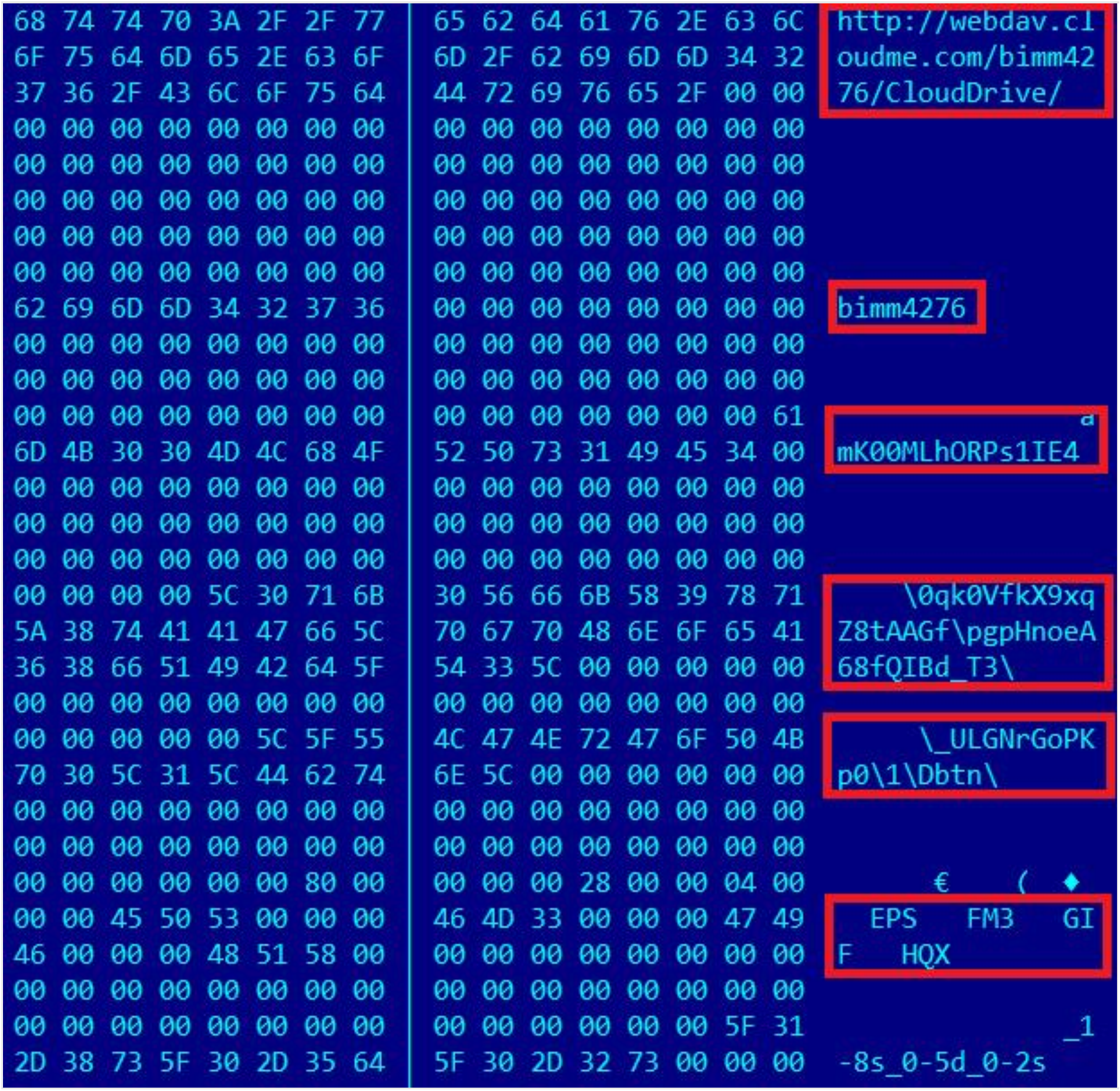
Some of the DLL names we observed include:

f4e15c1c2c95c651423dbb4cbe6c8fd5 - bicorporate.dll
649ff144aea6796679f8f9a1e9f51479 - fundamentive.dll
40e70f7f5d9cb1a669f8d8f306113485 - papersaving.dll
58db8f33a9cdd321d9525d1e68c06456 - previliges.dll
f5476728deb53fe2fa98e6a33577a9da - steinheimman.dll

Some of the payload names include:

steinheimman
papersaving
previliges
fundamentive
bicorporate
miditiming
damnatorily
munnopsis
arzner
redtailed
roodgoose
acholias
salefians
wartworts
frequencyuse
nonmagyar
shebir
getgoing

The payload includes an encrypted configuration block which contains information about the C&C sever:



```
68 74 74 70 3A 2F 2F 77  65 62 64 61 76 2E 63 6C  http://webdav.cl
6F 75 64 6D 65 2E 63 6F  6D 2F 62 69 6D 6D 34 32  oudme.com/bimm42
37 36 2F 43 6C 6F 75 64  44 72 69 76 65 2F 00 00  76/CloudDrive/
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
62 69 6D 6D 34 32 37 36  00 00 00 00 00 00 00 00  bimm4276
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 61                 a
6D 4B 30 30 4D 4C 68 4F  52 50 73 31 49 45 34 00  mK00MLhORPs1IE4
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 5C 30 71 6B  30 56 66 6B 58 39 78 71      \0qk0VfkX9xq
5A 38 74 41 41 47 66 5C  70 67 70 48 6E 6F 65 41  Z8tAAGf\pgpHnoeA
36 38 66 51 49 42 64 5F  54 33 5C 00 00 00 00 00  68fQIBd_T3\
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 5C 5F 55  4C 47 4E 72 47 6F 50 4B       \_ULGNrGoPK
70 30 5C 31 5C 44 62 74  6E 5C 00 00 00 00 00 00  p0\1\Dbtn\
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 80 00  00 00 00 28 00 00 04 00        €    (  ♦
00 00 45 50 53 00 00 00  46 4D 33 00 00 00 47 49    EPS   FM3   GI
46 00 00 00 48 51 58 00  00 00 00 00 00 00 00 00  F   HQX
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 5F 31                _1
2D 38 73 5F 30 2D 35 64  5F 30 2D 32 73 00 00 00  -8s_0-5d_0-2s
```

The information from the config includes a WebDAV URL which is used for connections, a username and password, two folders on the WebDAV server used to store plugins/modules for the malware and where data from the victim should be uploaded.

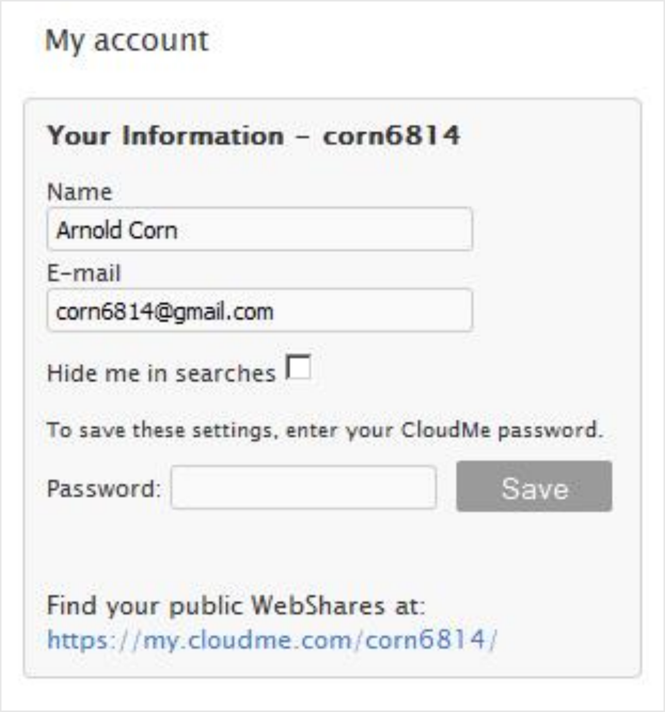
C&C communication

The Cloud Atlas implants utilize a rather unusual C&C mechanism. All the malware samples we've seen communicate via HTTPS and WebDav with the same server "cloudme.com", a cloud services provider. According to their website, CloudMe is owned and operated by CloudMe AB, a company based in Linköping, Sweden.

(*Important note: we do not believe that CloudMe is in any way related to the Cloud Atlas group - the attackers simply create free accounts on this provider and abuse them for command-and-control*).

Each malware set we have observed so far communicates with a different CloudMe account though. The attackers upload data to the account, which is downloaded by the implant, decrypted and interpreted. In turn, the malware uploads the replies back to the server via the same mechanism. Of course, it should be possible to reconfigure the malware to use any Cloud-based storage service that supports WebDAV.

Here's a look at one such account from CloudMe:



My account

Your Information - corn6814

Name: Arnold Corn

E-mail: corn6814@gmail.com

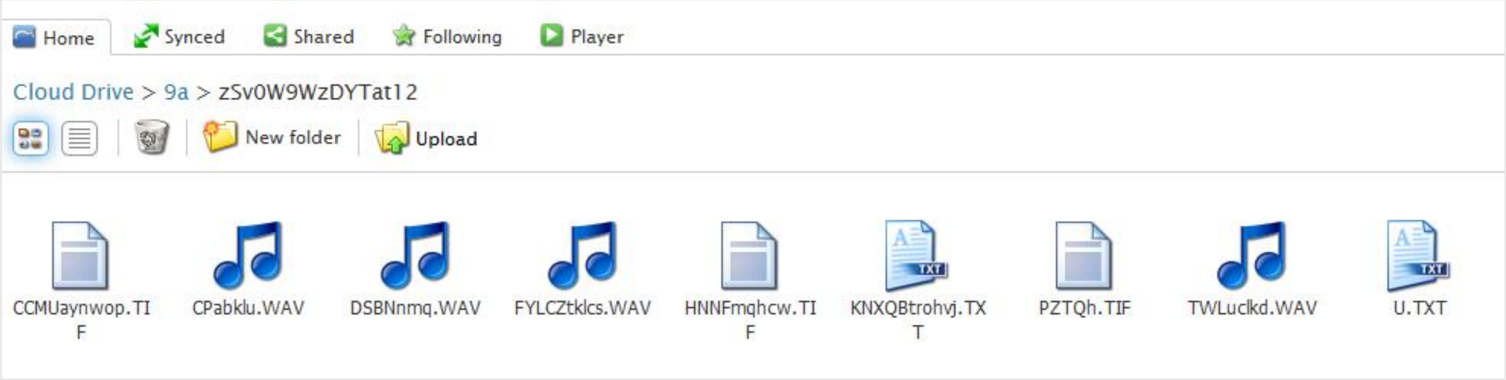
Hide me in searches □

To save these settings, enter your CloudMe password.

Password:

Find your public WebShares at:
https://my.cloudme.com/corn6814/

The data from the account:



The files stored in the randomly named folder were uploaded by the malware and contain various things, such as system information, running processes and current username. The data is compressed with LZMA and encrypted with AES, however, the keys are stored in the malware body which makes it possible to decrypt the information from the C&C.

We previously observed only one other group using a similar method – ItaDuke – that connected to accounts on the cloud provider **mydrive.ch**.

Victim statistics: top 5 infected countries

Similarities with RedOctober

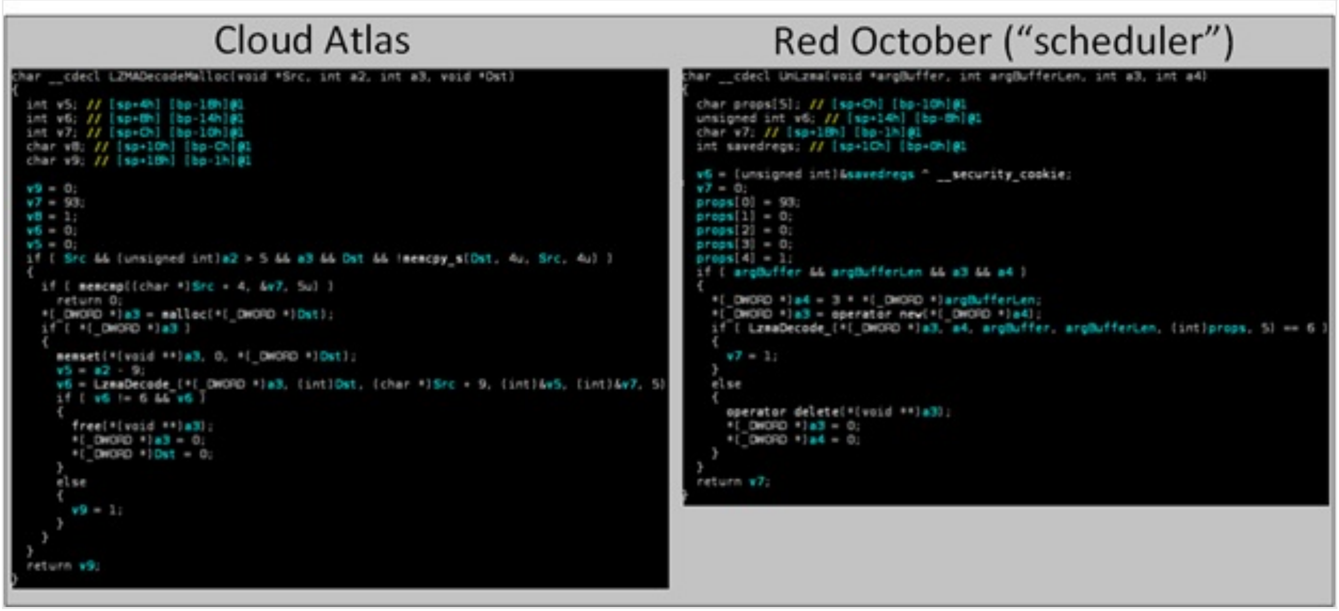
Just like with RedOctober, the top target of Cloud Atlas is Russia, followed closely by Kazakhstan, according to data from the Kaspersky Security Network (KSN). Actually, we see an obvious overlap of targets between the two, **with subtle differences which closely account for the geopolitical changes in the region that happened during the last two years**.

Interestingly, some of the spear-phishing documents between Cloud Atlas and RedOctober seem to exploit the same theme and were used to target the same entity at different times.

Both Cloud Atlas and RedOctober malware implants rely on a similar construct, with a loader and the final payload that is stored encrypted and compressed in an external file. There are some important differences though, especially in the encryption algorithms used – RC4 in RedOctober vs AES in Cloud Atlas.

The usage of the compression algorithms in Cloud Altas and RedOctober is another interesting similarity. Both malicious programs share the code for LZMA compression algorithm. In CloudAtlas it is used to compress the logs and to decompress the decrypted payload from the C&C servers, while in Red October the "scheduler" plugin uses it to decompress executable payloads from the C&C.

It turns out that the implementation of the algorithm is identical in both malicious modules, however the way it is invoked is a bit different, with additional input sanity checks added to the CloudAtlas version.



Cloud Atlas

```
char __cdecl LZMADecodeMalloc(void *Src, int a2, int a3, void *Dst)
{
  int v5; // [sp+4h] [bp-18h]@1
  int v6; // [sp+8h] [bp-14h]@1
  int v7; // [sp+Ch] [bp-10h]@1
  char v8; // [sp+10h] [bp-Ch]@1
  char v9; // [sp+18h] [bp-1h]@1

  v9 = 0;
  v7 = 93;
  v8 = 1;
  v6 = 0;
  v5 = 0;
  if ( Src && (unsigned int)a2 > 5 && a3 && Dst && !memcpy_s(Dst, 4u, Src, 4u) )
  {
    if ( memcmp((char *)Src + 4, &v7, 5u) )
      return 0;
    *(_DWORD *)a3 = malloc(*(_DWORD *)Dst);
    if ( *(_DWORD *)a3 )
    {
      memset(*(void **)a3, 0, *(_DWORD *)Dst);
      v5 = a2 - 9;
      v6 = LzmaDecode_(*(_DWORD *)a3, (int)Dst, (char *)Src + 9, (int)&v5, (int)&v7, 5)
      if ( v6 != 6 && v6 )
      {
        free(*(void **)a3);
        *(_DWORD *)a3 = 0;
        *(_DWORD *)Dst = 0;
      }
      else
      {
        v9 = 1;
      }
    }
  }
  return v9;
}
```

Red October ("scheduler")

```
char __cdecl UnLzma(void *argBuffer, int argBufferLen, int a3, int a4)
{
  char props[5]; // [sp+Ch] [bp-10h]@1
  unsigned int v6; // [sp+14h] [bp-8h]@1
  char v7; // [sp+18h] [bp-1h]@1
  int savedregs; // [sp+1Ch] [bp+0h]@1

  v6 = (unsigned int)&savedregs ^ __security_cookie;
  v7 = 0;
  props[0] = 93;
  props[1] = 0;
  props[2] = 0;
  props[3] = 0;
  props[4] = 1;
  if ( argBuffer && argBufferLen && a3 && a4 )
  {
    *(_DWORD *)a4 = 3 * *(_DWORD *)argBufferLen;
    *(_DWORD *)a3 = operator new(*(_DWORD *)a4);
    if ( LzmaDecode_(*(_DWORD *)a3, a4, argBuffer, argBufferLen, (int)props, 5) == 6 )
    {
      v7 = 1;
    }
    else
    {
      operator delete(*(void **)a3);
      *(_DWORD *)a3 = 0;
      *(_DWORD *)a4 = 0;
    }
  }
  return v7;
}
```

Another interesting similarity between the malware families is the configuration of the build system used to compile the binaries. Every binary created using the Microsoft Visual Studio toolchain has a special header that contains information about the number of input object files and version information of the compilers used to create them, the "Rich" header called so by the magic string that is used to identify it in the file.

We have been able to identify several RedOctober binaries that have "Rich" headers describing exactly the same layout of VC 2010 + VC 2008 object files. Although this doesn't necessarily mean that the binaries were created on the same development computer, they were definitely compiled using the same version of the Microsoft Visual Studio up to the build number version and using similar project configuration.

Number of object files, **CloudAtlas loader**	Number of object files, **Red October Office plugin**	Number of object files,**Red October Fileputexec plugin**	HEX compiler version	Decoded compiler version
01	01	01	009D766F	VC 2010 (build 30319)
01	01	01	009B766F	VC 2010 (build 30319)
22	2E	60	00AB766F	VC 2010 (build 30319)
5B	60	A3	00010000	–
05	07	11	00937809	VC 2008 (build 30729)
72	5C	AD	00AA766F	VC 2010 (build 30319)
20	10	18	009E766F	VC 2010 (build 30319)

To summarize the similarities between the two:

	Cloud Atlas	**RedOctober**
Shellcode marker in spearphished documents	PT@T	PT@T
Top target country	Russia	Russia
Compression algorithm used for C&C communications	LZMA	LZMA
C&C servers claim to be / redirect to	BBC (mobile malware)	BBC
Compiler version	VC 2010 (build 30319)	VC 2010 (build 30319) (some modules)

Finally, perhaps the strongest connection comes from targeting. Based on observations from KSN, **some of the victims of RedOctober are also being targeted by CloudAtlas**. In at least one case, the victim's **computer was attacked only twice in the last two years**, with only **two malicious programs** – **RedOctober** and **Cloud Atlas**.

These and other details make us believe that CloudAtlas represents a rebirth of the RedOctober attacks.

Conclusion

Following big announcements and public exposures of targeted attack operations, APT groups behave in a predictable manner. Most Chinese-speaking attackers simply relocate C&C servers to a different place, recompile the malware and carry on as if nothing happened.

Other groups that are more nervous about exposure go in a hibernation mode for months or years. Some may never return using the same tools and techniques.

However, when a major cyber-espionage operation is exposed, the attackers are unlikely to **completely shut down everything**. They simply go offline for some time, completely reshuffle their tools and return with rejuvenated forces.

We believe this is also the case of RedOctober, which makes a classy return with Cloud Atlas.

Kaspersky products detect the malware from the Cloud Atlas toolset with the following verdicts:

Exploit.Win32.CVE-2012-0158.j
Exploit.Win32.CVE-2012-0158.eu
Exploit.Win32.CVE-2012-0158.aw
Exploit.MSWord.CVE-2012-0158.ea
HEUR:Trojan.Win32.CloudAtlas.gen

HEUR:Trojan.Win32.Generic

HEUR:Trojan.Script.Generic

Trojan-Spy.Win32.Agent.ctda

Trojan-Spy.Win32.Agent.cteq

Trojan-Spy.Win32.Agent.ctgm

Trojan-Spy.Win32.Agent.ctfh

Trojan-Spy.Win32.Agent.cter

Trojan-Spy.Win32.Agent.ctfk

Trojan-Spy.Win32.Agent.ctfj

Trojan-Spy.Win32.Agent.crtk

Trojan-Spy.Win32.Agent.ctcz

Trojan-Spy.Win32.Agent.cqyc

Trojan-Spy.Win32.Agent.ctfg

Trojan-Spy.Win32.Agent.ctfi

Trojan-Spy.Win32.Agent.cquy

Trojan-Spy.Win32.Agent.ctew

Trojan-Spy.Win32.Agent.ctdg

Trojan-Spy.Win32.Agent.ctlf

Trojan-Spy.Win32.Agent.ctpz

Trojan-Spy.Win32.Agent.ctdq

Trojan-Spy.Win32.Agent.ctgm

Trojan-Spy.Win32.Agent.ctin

Trojan-Spy.Win32.Agent.ctlg

Trojan-Spy.Win32.Agent.ctpd

Trojan-Spy.Win32.Agent.ctps

Trojan-Spy.Win32.Agent.ctpq

Trojan-Spy.Win32.Agent.ctpy

Trojan-Spy.Win32.Agent.ctie

Trojan-Spy.Win32.Agent.ctcz

Trojan-Spy.Win32.Agent.ctgz

Trojan-Spy.Win32.Agent.ctpr

Trojan-Spy.Win32.Agent.ctdp

Trojan-Spy.Win32.Agent.ctdr

Trojan.Win32.Agent.idso

Trojan.Win32.Agent.idrx

HEUR:Trojan.Linux.Cloudatlas.a

Trojan.AndroidOS.Cloudatlas.a

Trojan.IphoneOS.Cloudatlas.a

Parallel research:

- Blue Coat Exposes Inception Framework