# TABLE OF CONTENT

1. Targets 5
2. Infection vectors 8
   2.1. Is the malware related to Buhtrap? 9
3. Evolution 10
   3.1. Versioning 10
   3.2. Timeline 11
4. Technical analysis 11
   4.1. Installation and persistence 12
   4.3. Network 14
   4.4. Fingerprint 17
   4.5 Monitoring 19
   4.6. Uninstall 20
   4.7. Config 21
   4.8. Other functions 21
5. Conclusion 24
6. Indicators of Compromise 25
   6.1. ESET detection names 25
   6.2. File names 25
   6.3. Registry indicators 25
   6.4. Hashes 25
   6.5. C&C server domain names 27
   6.6. C&C server IP addresses 27
   6.7. PDB paths 27
   6.8. Strings 28
7. Bibliography 30
LIST OF TABLES

Table 1  Code-signing certificates used by RTM  10
Table 2  List of botnet prefixes  10
Table 3  Possible values of the action field  16
Table 4  Targeted software  18
Table 5  Targeted URLs  18
Table 6  Identifier for a banking related machine  19
Table 7  Banking windows targeted  19
Table 8  Configuration keys  21

LIST OF FIGURES

Figure 1  RTM telemetry  5
Figure 2  Decoy document used in recent RTM campaigns  6
Figure 3  “1C: Enterprise 8” software export window  7
Figure 4  lc_to_kl.txt example  7
Figure 5  Infection vectors timeline  8
Figure 6  Code-signing certificate used by both RTM and Buhtrap  9
Figure 7  Timeline of the RTM malware  11
Figure 8  DLL name for sample F4C746696B0F5BB565D445EC49DD912993DE6361  12
Figure 9  Additional step in the RC4 algorithm  13
Figure 10  Encrypted strings structure  14
Figure 11  C&C feed hosted on livejournal  14
Figure 12  C&C requests protocol  15
Figure 13  C&C replies protocol  16
Figure 14  Login page for the RTM panel  17
Figure 15  Cleaning DLL name  20
Figure 16  Configuration values xor key  21
Figure 17  Bogus registry check dialog  22
Figure 18  Fake registry error  23
Figure 19  Actual UAC prompt from Windows  23
There are several groups actively and profitably targeting businesses in Russia. A trend that we have seen unfold before our eyes lately is these cybercriminals’ use of simple backdoors to gain a foothold in their targets’ networks. Once they have this access, a lot of the work is done manually, slowly getting to understand the network layout and deploying custom tools the criminals can use to steal funds from these entities. Some of the groups that best exemplify these trends are Buhtrap, Cobalt and Corkow.

The group discussed in this white paper is part of this new trend. We call this new group RTM [1]; it uses custom malware, written in Delphi, that we cover in detail in later sections. The first trace of this tool in our telemetry data dates back to late 2015. The group also makes use of several different modules that they deploy where appropriate to their targets. They are interested in users of remote banking systems (RBS), mainly in Russia and neighboring countries.

In this paper, we cover the details of their tools, whom they target, and offer a rare glimpse into the type of operation they are carrying out.

1. TARGETS

That this group is mostly targeting businesses is apparent from the processes they are looking for on a compromised system. They look for software that is usually only installed on accountants’ computers, such as remote banking software or tools to help with accounts payable. The full list of the processes they target is detailed in Section 4.4.2.

While both RTM and Buhtrap [2] are looking for a quite similar process list, the infection vectors are quite different. While the latter mostly used spear phishing campaigns, this group uses typical crimeware distribution channels such as drive-by downloads and spam. From our telemetry data, we see this threat actor focusing mostly on Russia and neighboring countries, with a few hits elsewhere. However, due to the mass spreading mechanism used, seeing detections outside the presumed primary region of interest is not surprising.

Figure 1  RTM telemetry
The total number of detections we have for this threat is low and the complexity of the malware is high, indicating that this group is targeting specific, high-value targets in the Russian region. In fact, the latest campaigns we have seen using the RTM malware contained interesting decoy documents as depicted in Figure 2.

Figure 2  Decoy document used in recent RTM campaigns

We found several types of decoy documents used by this group, such as contracts, invoices or tax-related forms. The nature of these decoy documents combined with the type of software targeted by this group lead us to believe they are going after accounting departments in Russian firms. These targets were also attacked by the Buhtrap group in 2014-2015.

While researching this group, we were able to interact with a few command and control (C&C) servers. The full list of commands that can be received from a C&C server will be covered in the following sections, but for now we can say that the client sends keylogger logs directly to the C&C, which then may respond with additional commands.

However, the days where you could just connect to the C&C and gather all the interesting interactions you wanted are mostly gone. To receive commands from the server, we had to provide realistic logs. We were rewarded for our efforts by seeing several relevant commands.

The first one was a request for the bot to hand over a specific file: 1c_to_kl.txt. This file is an export from popular accounting software called “1C: Enterprise 8” which, conveniently enough, is present in the process list that the bot is actively seeking. This software can work in tandem with different Remote Banking Systems (RBS) by exporting transfer data to a text file. This text file can then be imported into an RBS to automate the payment order as seen in Figure 3.
As the data in this file contain information about the transfer, it is possible for the fraudsters to modify the information in the file in order to steal money from the victim.
About a month after these files were requested from the C&C server, we saw the apparition of a new module: 1c_2_kl.dll. This module was pushed as a plugin to compromised systems and has the capacity to parse the export file automatically by injecting itself in the accounting software process. This module will be detailed in a later section.

Interestingly, FinCERT, a Russian CERT responsible for fighting cybercrime targeting Russian financial institutions, issued a statement warning potential victims against criminals going after 1c_to_kl.txt export files in late 2016. Also, the developers of “1C: Enterprise 8” are already aware of this scheme and issued a statement [3] acknowledging the problem and enumerating the possible protective measures the user can take to prevent this attack.

We also saw additional modules being downloaded once the bots reported to the C&C, such as a VNC module. This VNC module has both a 32 and 64-bits variants and is similar to a VNC module already seen in used in Dridex campaigns. We hypothesize that once the bot installs this VNC module, the criminals will use it to connect to the computer remotely and look around to see what they could do to take full advantage of the system. Organized groups like these usually then try to move laterally in the network, recovering account passwords through tools such as mimikatz and using other offensive tactics to learn more about the network and make the malware’s presence as persistent as possible.

2. INFECTION VECTORS

The following figure shows the different infection vectors’ first-seen date gathered throughout our tracking.

![Infection vectors timeline](image)

Figure 5  Infection vectors timeline
This group has used a large array of infection vectors, mostly revolving around drive-by downloads and spam. These channels are well suited for targeted attacks, since for the former, cybercriminals can pick websites likely to be visited by their targets, and for the latter, send emails with attachments directly to businesses or individuals the cybercriminals are attacking.

The fact that their malware is distributed through several different channels – such as the RIG [4] and Sundown exploit kits or spam runs – also shows that this group has strong ties with criminals in the underground market who are selling these services.

2.1. Is the malware related to Buhtrap?

As RTM exhibits some operational similarities to what we have seen with Buhtrap, one might legitimately wonder whether they are related.

In September 2016, we saw an RTM sample being distributed by a Buhtrap downloader. We also saw two code-signing certificates used to sign samples of both Buhtrap and RTM. The first one, issued to a company allegedly called TOV DNISTER-M, was used to sign a Delphi second stage (SHA-1: 025C718BA31E43DB1B87DC13F94A61A9338C11CE) as well as a Buhtrap DLL (SHA-1: 1E2642B454A2C889B6D41116CCDBA83F6F2D4890).

![Code-signing certificate used by both RTM and Buhtrap](image)

The second one, issued to a company allegedly called Bit-Tredj, was used to sign Buhtrap downladers (SHA-1: 7C1B6B1713BD923FC243DFEC80002FEB9B3EB292 and B74F71560E448488D2153A2F851207A0AC206E2B) fetching and installing an RTM payload. Table 1 highlights key characteristics of these certificates.

Although they shared the code-signing certificate with other malware families, they also used one that, according to our telemetry, was used only by them. It was issued to a company allegedly called Kit-SD and was used to sign some RTM malware (e.g. SHA-1: 42A4B04446A20993DDAE98B2BE6D5A797376D4B6).
Because they used the same Buhtrap downloader, they also shared some network indicators, since an RTM payload was downloaded from Buhtrap infrastructure. That said, as RTM has been using several different ways to distribute their malware besides the Buhtrap downloader, we believe that the groups using these two malware families are not the same. However, they are using a similar modus operandi. They are both targeting businesses using accounting software, are fingerprinting systems of interest similarly, are looking for smart card readers, and finally, they deploy an array of malicious tools to spy on their victims.

3. EVOLUTION

In this section, we will cover the different versions of the malware we have seen during our tracking, but first we need to define a way to follow its evolution.

3.1. Versioning

RTM stores configuration data in registry keys and a particularly interesting piece of information is called botnet-prefix. A list of all the values we have seen in the samples we analyzed is presented in Table 2.

We are not sure of the purpose of this value. It may be used to record a version of the malware. However, we observed very few changes between consecutive versions such as bit2 and bit3 or 0.1.6.4 and 0.1.6.6. Moreover, one of the prefixes, main, has existed since the beginning and even evolved from typical C&C domains to .bit domains, as we will explain in Section 4.3.
3.2. Timeline

Using telemetry data, we created a timeline of the first appearance of the samples we have seen. This is presented in Figure 7.

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>October</td>
<td>• First known appearance of the RTM malware.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Main prefix.</td>
</tr>
<tr>
<td>2016</td>
<td>March</td>
<td>Appearance of the KL2 prefix.</td>
</tr>
<tr>
<td>2016</td>
<td>April</td>
<td>• Switch to .bit C&amp;C domains for all the prefixes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Appearance of the bit2 and bit3 prefixes.</td>
</tr>
<tr>
<td>2016</td>
<td>July</td>
<td>First known appearance of the hit3 prefix. Does not use .bit C&amp;C domains</td>
</tr>
<tr>
<td>2016</td>
<td>December</td>
<td>First known appearance of the 0.1.7.6 prefix.</td>
</tr>
<tr>
<td>2016</td>
<td>October</td>
<td>First known appearance of the 0.1.6.6 prefix.</td>
</tr>
<tr>
<td>2016</td>
<td>September</td>
<td>First known appearance of the kosmos and 0.1.6.4 prefixes.</td>
</tr>
<tr>
<td>2017</td>
<td>January</td>
<td>First known appearance of the 0.1.9.0 prefix.</td>
</tr>
</tbody>
</table>

Figure 7  Timeline of the RTM malware

4. TECHNICAL ANALYSIS

In this section, we describe the main functions of the RTM banker. These include the persistence mechanism, their custom version of the RC4 algorithm, the network protocol, the spying abilities and some other miscellaneous functions. In particular, we will focus on the samples SHA-1 AA0FA4584768CE9E16D67D8C529233E99FF1BBF0 and 48BC113EC8BA20B8B80CD5D4DA92051A19D1032B.
4.1. Installation and persistence

4.1.1. Executable

The core of the RTM malware is a DLL but it is dropped onto the disk by a .EXE. That executable file is generally packed and contains the DLL code. When it is launched it only extracts the DLL and runs it using the following command:

```
rundll32.exe "%PROGRAMDATA%\Winlogon\winlogon.lnk",DllGetClassObject host
```

4.1.2. DLL

The main DLL is always dropped on the disk as `winlogon.lnk` in the `%PROGRAMDATA%\Winlogon` folder. This file extension usually refers to a shortcut [5] but this file is actually a DLL written in Delphi, which was called `core.dll` by its developers as shown in Figure 8.

![Figure 8](core.dll DllGetClassObject)

Just after being launched, the malware sets up its persistence mechanism. It can do this in two different ways, depending on the victim's level of user privileges. That is, if running as admin, it can add an entry in the `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\` registry under the name 'Windows Update'. The commands that are in the data of the 'Windows Update' value will be executed at the start of the user session.

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Windows Update = rundll32.exe "%PROGRAMDATA%\winlogon.lnk",DllGetClassObject host
```

It will also try to add a task in the Windows Task Scheduler. This task will launch the `winlogon.lnk` DLL with the same export and parameter as explained before.

If running as an ordinary user, it will add a value in the `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\` registry under the same name, 'Windows Update' and with the same data:

```
rundll32.exe "%PROGRAMDATA%\winlogon.lnk",DllGetClassObject host
```

4.2. Modified RC4 algorithm

Despite its known weaknesses, the RC4 algorithm is regularly used by malware authors. However, RTM's programmers have decided to modify it slightly, possibly to slow down analysis by security researchers. This variant of RC4 is widely used in RTM in its malware to encrypt strings, network data, configuration and modules.
4.2.1. Differences

The original RC4 algorithm is divided into two main parts: the Key-Scheduling Algorithm (KSA) and the Pseudo-Random Generation Algorithm (PRGA) [6]. The KSA part consists of the initialization of the $s$ table using the key. In the PRGA part, the plain text is processed, using the $s$ table, to obtain the ciphertext.

In the RTM malware, the authors added an intermediate step between the initialization of the $s$ table and the encryption of the plaintext, meaning that it comes between the KSA and the PRGA. Before encrypting data, the $s$ table is XORed with a four-byte number. This additional key is variable and is given at the same time as the data to encrypt or decrypt. The function that performs this additional step is given in Figure 9.

![Figure 9](image)

Additional step in the RC4 algorithm

4.2.2. String encryption

At first glance, there are few strings readable in the main DLL. The others are encrypted with the previous algorithm using the structure described in Figure 10. We found more than 25 different RC4 keys used for string encryption in the samples we analyzed while the XOR key is different for each string. The value of the integer field separating each string structure is always 0xFFFFFFFF.

At the beginning of its execution, the malware decrypts the strings into a global variable. Then, when it needs to access a string, it dynamically computes the address of the decrypted strings based on a base address and an offset.
The strings contain interesting information about the functions of the malware. Some examples of these strings are found in Section 6.8.

```c
struct struct_encrypted_string
{
    int separator; //constant - 0xFFFFFFFF
    int size;
    int xor_key[4];
    char encrypted_data[size];
};
```

4.3. Network

The way the malware contacts the C&C has changed over the different versions. The first few samples, from October 2015 to April 2016, used traditional domain names along with an RSS feed hosted on livejournal.com to update the C&C list. Starting in April 2016, we observed in our telemetry data a move toward the use of .bit domains, which provides, as explained in Section 4.3.2, more resilience against domain takedown. The move is confirmed by the registration date of the C&C domains: the first RTM C&C .bit domain, fde05d0573a.bit, was registered on March 13th, 2016.

All C&C URLs we saw during our tracking shared a common path: /r/z.php. As this path is uncommon, it helps to identify RTM requests in network flows.

4.3.1. C&C feed

Older samples used a feed to update their list of C&C servers. This is hosted on livejournal.com and at the time of writing it is still up at the URL https://f72ba81c921.(.)livejournal.(.)com/data/rss. Livejournal [7] is a US/Russian company providing social network services such as a blog-hosting platform. The RTM operators simply created a blog on livejournal and posted an article containing the feed of encrypted C&Cs. A screenshot of this blog is presented in Figure 11.
The C&C strings are encrypted with the RC4 algorithm variant presented in Section 4.2. The actual version (November 2016) of the feed contains the following C&C addresses:

- hxxp://cainmoon(.)net/r/z.php
- hxxp://rtm(.)dev/0-3/z.php
- hxxp://vpntap(.)top/r/z.php

### 4.3.2 .bit domains

In the most recent versions of the RTM malware, the authors switched to C&C domains that use the .bit Top-Level Domain (TLD) [8]. This is not in the Internet Corporation for Assigned Names and Numbers (ICANN) list of TLDs [9]. Instead, it relies on the Namecoin cryptocurrency, a fork of the well-known Bitcoin. Malware authors do not frequently use the .bit TLD for their C&C domains, though an example of such use was previously seen in a version of the Necurs botnet [10].

Unlike Bitcoin, Namecoin blockchain users are able to store data. One of the main uses for this feature is the top level domain .bit. It is possible to register domains that will be stored in the blockchain. The associated record in the blockchain also includes the IP addresses the domain is resolving to. Thus, this TLD is said to be “censorship-resistant” because only the registrant can change the resolution of a .bit domain. That means it is much more difficult to disrupt a malicious domain using this kind of TLD.

However, the RTM malware does not embed any of the software needed to read the Namecoin blockchain. It relies on DNS tier servers like dns.dot-bit.org or OpenNic servers to resolve the .bit domains. Hence, it has the same resilience as the DNS servers it uses. We observed that some C&C domains stopped being resolved after they were mentioned in a blogpost.

Another advantage to the attackers of the .bit TLD is the cost. To register a domain, the operators only need to pay 0.01NC which corresponds to 0.00185 USD (as of December 5th, 2016). By comparison, a .com domain costs at least 10 USD.

#### 4.3.3. Protocol

To contact its C&C, the malware uses HTTP POST requests with data formatted using a custom protocol. The path is always /r/z.php and User-Agent set to Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0).

In the requests to the C&C server, the data are formatted as follows, where the offset values are in bytes:

![C&C requests protocol](image)

The bytes 0 to 6 are not encrypted while the bytes starting at 6 are encrypted with the RC4 algorithm variant.
The packet structure of the C&C’s replies is simpler. The bytes from 4 to packet size are encrypted.

![C&C replies protocol](image)

Figure 13  **C&C replies protocol**

A list of the possible values of the action byte is provided in **Table 3**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OK (ACK)</td>
</tr>
<tr>
<td>1</td>
<td>Core. Optional: a string that contains a command.</td>
</tr>
<tr>
<td>3</td>
<td>Unrecognized request type</td>
</tr>
<tr>
<td>4</td>
<td>Unexpected request type</td>
</tr>
<tr>
<td>5</td>
<td>Install module</td>
</tr>
<tr>
<td>6</td>
<td>Execute downloaded executable</td>
</tr>
<tr>
<td>7</td>
<td>Execute downloaded executable. If the process does not have administrative rights, it will try to elevate its privileges using the method presented in Section 4.8.2.</td>
</tr>
<tr>
<td>8</td>
<td>Load downloaded DLL</td>
</tr>
<tr>
<td>9</td>
<td>Load and execute an MZ executable in memory without using a file.</td>
</tr>
<tr>
<td>10</td>
<td>Add a certificate to the windows store</td>
</tr>
<tr>
<td>11</td>
<td>Self-Update</td>
</tr>
<tr>
<td>12</td>
<td>Write a file into the common app data folder. (Usually C:\ProgramData)</td>
</tr>
</tbody>
</table>

The malware always computes the CRC32 of the decrypted data and compares it with the one provided in the packet. If they differ, it will simply drop the packet.
The optional data can contain different objects such as a PE file, a file to search in the file system or a new C&C URL.

4.3.4. Panel
Interestingly, we note that a panel is available on the C&C servers used by RTM. A screenshot of the login page is shown in Figure 14.

![Figure 14 Login page for the RTM panel](image)

4.4. Fingerprint
The RTM malware is a banking Trojan so it is not surprising that its operators need information about the victim's machine. The information gathered by the bot can be divided into two parts. On one hand, we find common information about the OS attributes. On the other hand, the malware checks whether the compromised machine contains artefacts related to Russian remote banking systems.

4.4.1. Common information
After the malware is installed or launched after a reboot, a report containing common information is sent to the C&C server. It contains:
- The time zone
- The default language identifier
- The privileges of the logged-in user
- The process integrity level
- The username
- The computer name
- The OS version
- The additional modules installed
- The security software installed
- The list of smart card readers
4.4.2. Remote banking system

A typical target for banking Trojans is the remote banking system [11][2]. RTM is not an exception, as one of its classes is called TBdo. Bdo is the Russian translation for RBS (Remote Banking System) so it is clear that RBS is a target for this malware. This class is able to perform several tasks including the scanning of drives and of browsing history.

When the malware scans the drive, its sole aim is to determine whether banking software is installed on the machine. A full list of targeted software is available in Table 4. If it finds an interesting file, it reports the information to the C&C server. The next actions performed by the malware depend on the logic, which is on the C&C server-side.

Table 4

<table>
<thead>
<tr>
<th>Targeted software</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ftcgpk.exe</td>
</tr>
<tr>
<td>1cv7.exe</td>
</tr>
<tr>
<td>1cv7l.exe</td>
</tr>
<tr>
<td>1cv7s.exe</td>
</tr>
<tr>
<td>1cv8.exe</td>
</tr>
<tr>
<td>1cv8c.exe</td>
</tr>
</tbody>
</table>

The RTM malware also looks for banking URL patterns in IE’s browsing history and in the opened tabs. For the history, it browses using the functions `FindNextUrlCacheEntryA` and `FindFirstUrlCacheEntryA` and for each entry, it checks to see if the URL matches one of the patterns shown in Table 5.

For opened tabs, it connects to Internet Explorer and Firefox through the Dynamic Data Exchange (DDE) [12] mechanism to access the current URL of tabs opened. For each tab, it also checks if that matches a pattern shown in Table 5.

The browsing history and the opened tabs check is executed in a WHILE loop with a sleep of 1 second between the checks. The other data monitored in real time will be detailed in Section 4.5.

Table 5

<table>
<thead>
<tr>
<th>Targeted URLs</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsi.dll?</td>
</tr>
<tr>
<td>online.payment.ru</td>
</tr>
<tr>
<td>bankline.ru</td>
</tr>
<tr>
<td>/ic/login.zhtml</td>
</tr>
<tr>
<td>/servlets/ibc</td>
</tr>
</tbody>
</table>

If a pattern is found, it is reported to the C&C using the list of strings presented in Table 6.
4.5 Monitoring

When the malware starts, various fingerprint information is sent to the C&C server, including the presence of banking software. This fingerprinting occurs only when the malware first starts monitoring the system, just after the initial scan is finished.

4.5.1. Remote banking system

The previously discussed class TBdo is also responsible for monitoring banking-related events. As in the initial scan, it uses DDE to check tabs in Firefox and Internet Explorer; another class, TShell, is also used to monitor shell windows (Internet Explorer and Explorer windows).

This class uses the COM interfaces IShellWindows, iWebBrowser, DWebBrowserEvents2 and IConnectionPointContainer to monitor the windows. When the user navigates to a new webpage, the malware is notified. It then checks the URL against the patterns provided in Table 5. If a match is found, it will take six consecutive screenshots, waiting 5 seconds between each one. Finally, it sends the screenshots to the C&C server.

It also checks for some window titles, related to banking software. A full list of these titles is available in Table 7.

<table>
<thead>
<tr>
<th>Window name</th>
<th>Class name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClBank</td>
<td>SunAwtFrame</td>
</tr>
<tr>
<td>ClBank</td>
<td>SunAwtDialog</td>
</tr>
<tr>
<td>Логин (Login)</td>
<td>TLoginWindow</td>
</tr>
<tr>
<td>Null</td>
<td>TfmiISClient</td>
</tr>
<tr>
<td>Ключ электронной подписи</td>
<td>TInitialForm</td>
</tr>
<tr>
<td>(Key Electronic Signature)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6  Identifier for a banking related machine

<table>
<thead>
<tr>
<th>BSS</th>
<th>RosBank</th>
<th>BiCrypt</th>
<th>SberBank_Fiz</th>
<th>ISCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSS_PC</td>
<td>SberBank_BO</td>
<td>VTB24</td>
<td>CFT</td>
<td>WebMoney</td>
</tr>
<tr>
<td>iBank2_PC</td>
<td>INIST</td>
<td>SGB</td>
<td>WinPost</td>
<td>XTC</td>
</tr>
<tr>
<td>Faktura</td>
<td>Inversion</td>
<td>Raiffeisen</td>
<td>SBIS</td>
<td>iFOBS</td>
</tr>
<tr>
<td>PCB</td>
<td>Interbank</td>
<td>HandyBank</td>
<td>CIBank</td>
<td>TRANSAQ</td>
</tr>
<tr>
<td>InterPro</td>
<td>iBank2</td>
<td>WU</td>
<td>QiwiCashier</td>
<td>OSMP</td>
</tr>
</tbody>
</table>
4.5.2. Smart card

The RTM malware has the ability to monitor smart card readers connected to the infected computers. These devices are used in some countries to validate banking related orders. So, if this kind of device is attached to the computer, it may indicate that these machines are used to perform banking transactions. However, unlike other banking Trojans [13], it is not able to interact with these smartcards. Such features may be included in an additional module we have not yet seen.

4.5.3. Keylogger

An important part of the monitoring of the victim’s computer is to record the keystrokes. Moreover, it seems that the RTM developers did not want to miss any information, as they monitor not only the keyboard but also the virtual keyboard and the clipboard.

To do so, they use the SetWindowsHookExA function to hook the keyboard and the mouse. When an event occurs, they log the key pressed, or the key corresponding to the click on the virtual keyboard, along with the software name and the date. Then, the buffer is sent to the C&C. For the clipboard, they use the SetClipboardViewer function. They only log the clipboard content when the data is text. As for the keyboard, they also log the software name and the date before sending the buffer to the C&C.

4.5.4. Screenshots

The last monitoring feature is the ability to make screenshots. This is used when the shell monitoring class detects an interesting website or when a banking software window is detected. The screenshot is taken using the GDI library before being sent to the C&C.

4.6. Uninstall

The C&C can send a command to stop all the malware activities and clean the computer. It will clean all the files and the registry entries created during the execution of RTM on the victim's machine. Then, using an additional DLL encrypted in the malware, it will kill the malware and delete the winlogon file. Finally, it shuts down the computer. As shown in Figure 15, this DLL was called erase.dll by its developers.

![Figure 15 Cleaning DLL name](image)

A more destructive command can be sent to the malware, called uninstall-lock. If the malware has administrative rights it will try to erase the MBR. If it fails, it tries to move the MBR to a random path. If it succeeds, the computer will no longer boot after the shutdown. In this event, it would probably lead to a full reinstallation of the machine, erasing all possible forensic evidence. If the program has low privileges, the malware writes a .EXE that is encrypted in the main RTM DLL. This .EXE is really small and only implements the code needed to shut down the computer. It will also register this executable in the HKCU\CurrentVersion\Run registry key. Thus, each time the user starts a session, the computer will immediately shut down.
4.7. Config

By default, the RTM malware doesn’t really have a config file. However, the C&C can push config values that will be stored in the registry and used by the malware. A list of configuration keys is presented in Table 8.

<table>
<thead>
<tr>
<th>Configuration keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>keylogger.last-data</td>
</tr>
<tr>
<td>botnet-id</td>
</tr>
<tr>
<td>cc.url.1</td>
</tr>
<tr>
<td>keylogger.last-wnd-caption</td>
</tr>
<tr>
<td>cc.connect-interval</td>
</tr>
<tr>
<td>cc.url.2</td>
</tr>
<tr>
<td>keylogger.last-exe-path</td>
</tr>
<tr>
<td>scan-files</td>
</tr>
<tr>
<td>scards.monitoring-interval</td>
</tr>
<tr>
<td>botnet-prefix</td>
</tr>
<tr>
<td>post-install-report</td>
</tr>
<tr>
<td>dbo.detected</td>
</tr>
<tr>
<td>scan-files</td>
</tr>
<tr>
<td>multiinstance-off</td>
</tr>
<tr>
<td>post-install-report-url</td>
</tr>
<tr>
<td>modules-data.%modulename%</td>
</tr>
<tr>
<td>keylogger-off</td>
</tr>
<tr>
<td>dbo-detector-off</td>
</tr>
<tr>
<td>scard-off</td>
</tr>
<tr>
<td>modules-off</td>
</tr>
<tr>
<td>modules.%modulename%</td>
</tr>
</tbody>
</table>

The configuration is stored in the registry Software\[Pseudo-random string\] key. In this registry key, each value corresponds to one of the strings presented in the previous table. However, the values and the associated data are encrypted using RTM’s custom RC4 algorithm.

The data has the same structure as the network or the strings. The four-byte XOR key is added at the beginning of the encrypted data.

For the configuration values, the XOR key is different and depends on the size of the value. It can be computed as follows:

\[
\text{xor\_key} = (\text{len}(\text{config\_value}) \ll 24) | (\text{len}(\text{config\_value}) \ll 16) | \text{len}(\text{config\_value}) | (\text{len}(\text{config\_value}) \ll 8)
\]

4.8. Other functions

This malware has many other functions that we present here.

4.8.1. Additional modules

It can run additional modules that are DLL files. They are sent over the network by the C&C server and could be run as external programs. They can also be mapped in memory and launched in new threads.

For storage, they are saved in .dtt files and encrypted using the custom RC4 algorithm with the same key used for the network communications.

At this time, we have seen only the VNC module (896631988249077C21F66A8354E2CBAA0370464), the browser data exfiltration module (03DE8622BE6B2F75A364A275995C3411626C4D9F) and the 1c_2_k1 module (B1EE562E1F69EFC6FBA588B88753BE7D0B3E4CFAB) being installed on an infected machine.
For the VNC module, the C&C server later sent a command, asking to connect to a VNC server at a specific IP address on port 44443.

The browser data exfiltration plugin implements the class `TBrowserDataCollector`, which is able to read the IE's browsing history. It then sends the full list of visited URLs to the C&C server.

The last discovered module is called `1c_2_kl` and is able to interact with the 1C software suite, which was described in Section 1. This module is divided in two parts: a main DLL and two agents (32 and 64 bits) that will be injected in every process by registering a hook to the `WH_CBT` event. If the agent is injected in a process of the 1C suite, it hooks the `CreateFile` and `WriteFile` functions.

Whenever the hooked `CreateFile` function is called, it keeps in memory the file path should its filename be `1c_to_kl.txt`. Then, when a call to `WriteFile` is intercepted, it calls the genuine `WriteFile` function and sends the `1c_to_kl.txt` file's path to the main module DLL by sending it a crafted `WM_COPYDATA` Windows message.

Finally, the main module DLL opens and parses the file to identify payment orders. It extracts both the total amount and number of transactions contained in the file. Ultimately, it sends this information to the C&C server. At this time, we believe this module is still in development as it contains debug messages and it cannot modify automatically the `1c_to_kl.txt` file.

4.8.2. Privilege elevation

This malware can try to elevate its privileges by showing fake error messages. First, it mimics a registry check by using a similar design to typical Windows progress bars, as shown in Figure 17. It also uses the real `regedit` icon. We could also note the spelling mistake in “whait”. After a few seconds, the scan is finished and a fake error message is displayed.

This fake error message, shown in Figure 18, would easily fool a typical end user. Once again, the design is similar to that of a regular Windows error message, but again there are telltale grammatical errors not found in real Windows error dialogs. If the user clicks on one of the two choices, it will try to elevate the malware's privileges.

Immediately after the user clicks on one of the restore choices, the malware launches the DLL using the `runas` option in the `ShellExecute` function. This will run the program with administrative privileges. However, the user will be prompted with a real UAC prompt coming from the Windows system, shown in Figure 19, asking the user to allow the privilege elevation. If thus allowed, the malware will run with administrative privileges.
Thus, rather than depending on a vulnerability, it tries to confuse the user. Moreover, depending on the machine's default language, error messages will be displayed in English or in Russian.

![Fake registry error](image1)

**Figure 18** Fake registry error

![Actual UAC prompt from Windows](image2)

**Figure 19** Actual UAC prompt from Windows

### 4.8.3. Certificates

The malware can also add certificates to the Windows store and it is able to validate the addition by automatically clicking “Yes” on the csrss.exe dialog box. This behavior is not new, as the Retefe banking Trojan is also able to self-validate a new certificate installation [14].

### 4.8.4. Backconnect

The malware authors also implemented a Backconnect TCP tunnel. We have not seen any usage of this functionality but it may be used to control the infected machines remotely.

### 4.8.5. Host file management

The C&C can send a command to the malware to modify Windows’ `etc/hosts` file. The `hosts` file is used on Windows systems to make custom DNS resolutions.
4.8.6. Find and send a file
The C&C can ask the malware to find a file on the file system and to upload it to the server. For instance, during our investigation, we received a command from the C&C server asking for the file 1c_to_kl.txt. As explained before, this file can be generated by the accounting software 1C: Enterprise 8.

4.8.7. Update
Finally, the authors can update the malware by sending a new DLL that will replace the current version.

5. CONCLUSION

Our research on the RTM malware shows that the Russian banking system is still a target of choice for criminals. Other groups, such as Buhtrap, Corkow and Carbanak, were already known to target and successfully steal money from financial institutions and their customers in Russia. Thus, RTM is another player in this profitable game.

According to our telemetry, this piece of malware has been used since the end of 2015 or earlier. Even though it does not use previously unknown techniques, it implements a full range of spying capabilities including smart card reading, keylogging and real-time monitoring of banking-related activities. It can also search for export files created by accounting software such as 1C: Enterprise 8. Finally, its command and control infrastructure is highly resilient, since it uses the decentralized and non-censurable .bit top-level domain.

The Indicators of Compromise (IOCs) are detailed in Section 6 and on GitHub. If you have any questions related to this investigation, please contact us at threatintel@eset.com.

Acknowledgements

Special thanks to our colleagues Anton Cherepanov and Jan Matušík for their help in this research.
6. INDICATORS OF COMPROMISE

6.1. ESET detection names

Win32/Spy.RTM.A
Win32/Spy.RTM.B
Win32/Spy.RTM.C
Win32/Spy.RTM.D
Win32/Spy.RTM.E
Win32/Spy.RTM.F
Win32/Spy.RTM.G
Win32/Spy.RTM.H
Win32/Hvnc.AD
Win64/Spy.RTM.A

6.2. File names

C:\ProgramData\Winlogon\winlogon.lnk

6.3. Registry indicators

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run :
  "Windows Update = rundll32.exe "%PROGRAMDATA%\Winlogon\winlogon.lnk",DllGetClassObject"

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run :
  "Windows Update = rundll32.exe "%PROGRAMDATA%\Winlogon\winlogon.lnk",DllGetClassObject"

6.4. Hashes

6.4.1. DLLs

30c8b60cc66eaefb4c861584f45fe80dab71ee22
471a8f0d0a32ce61cf5e4bece9555571db234ed6
42b99033477b22578b0a35adda62c0bc02a09a5
be06b838eb84b2e6b5f9ceaaafa3fbbb4ebd522
f9138b6e29fe0c3467fc591a9b9b5fe9975027
ca80c6a8397c16695e1352a48c538fb02f1c1b6
e942145c0f3549bf7be79cbe5a4031cfc6614af19
0705ba1a9696b05130e5768ea8efdaa864ddaff0
c75273cd886c3ea1a5be7e99b11044f888db3ef
e0f377551d5b653eedf9a0c3ef23eabfc7a937e
e7777db52fc9d34d7573db242f9c195d24836d8
ef7de8746c414a8925aa6a01f7130cfc7eac2df
00fe6cf9c85821a2a2479083acb538ee49c8c141
5d6a9646ee6f015b296d9b0d4cb3e095957d0aad
912dd756893fe9e68b2ddc85613e736412a25ef
5f357ffbb6aae8327f7a0fcca824fa4026db4000a0a
ebb585edfe6f9359a38c7fb65871f418cc633
c33b4a75f4e160a153c3baada042b2eaca5d60618
f675519445ce89f61df8ad6aaf2bf49f040224a
4a670cc34e59ea94e88c19eef64a4106cf551124
5153886f6e6c6fed81501e68678286633bc564ec
4d83f2f601036bc77085796a1b0170aafb6927
be83d9b2b69bb2fafa9e2e35734d3bfc6f635166
f4c746696b0f5b5b56d445ec49d9d12993de6361
d8f3c6a1bb4d014fa34aae41a8def7c3b6
31b7215c892a064a6f59c16de6a1defcf39012a9
5521ccee3e68eb6b8f7fa129d4af31514363b2d
da0c2e636909ea861bb24794ee8ff4c051ade64
822e05f99f55d727da5a63d06273da507ef5f135
6.4.2. Droppers/Downloaders

0961119783365e884dfdf12df7c7ae9f7388a140c0
d6b4dae4ab07e7c95f4a58cf0be0f0281c81c67
858f6667a7057304e7e2aef78d81a2aee42a41dc0
42a4b0444a02093daee8b2c6d54da92051a19d10320
026c75a1ba3e43dbbb8d7c13f944a1a933c1ce
059a114c2e5643251c0ce7c74b28418353cf29a
4a084e70f2a04510c24586c9b2560a0e3598969e6
77a7735d063e815e0285150d9ea15b5c5f699
83df646510e6f403cf28ed714bdf5e21bb8fb41
8b77419e600c03f078e776e61ca21547bb8bd1
9ac461fe98483674f46b66469d46de9554afcc66
aic7b1574ad7c144a0f4d70098a4585ef1f24a5c4
a6d3e9d7382d17d5890ca10a248aaccd73a2b39
9e9fe3259bceb852e1cb8e5a011ff9ebe3b08fbc
f539f53f847d60e6b6b6e32be76f6e1f9b29b60
3e6f88d67cd5a09b2c5ed530cbf2657ff7aad4a90
d125a86b3a009a390490d22d4016e159a2a50a9
92a1c9f90697446534d3733ebf8669a84351b9
9d2fd31c086f0e982fbdf6e3951173ab69db0b2
c9a9ade798a61450adbcac5e433dade811867b9
fca3d0a53e66897599f7f2b0c3008a254a508
312a487b2830c62845f6feaf11d4a7c25783f1a
c4844acdb88eb10a40a5775e475bc848e0a2583e7
c824ee171382002a71274c0312f5c1355f0b44
85aad39688b03bac2c755a502f6bf104643a3
d0390f4bdc50a952cc01f2db034f87d88ed3d44
d43ac4c63da2c11242e910f4e6e359579504642
5sb38be12dc5e21f9f5ef01e8a457004c9f978a6d
c6e3aaa12a52762bf6290b97cc79148eddd01e0
136185555755c537522e53cc8aad7487dc9dcfffa
7c4b8114447776541302206df9ae22be8490c45ba
f89e56dd9ca7cecc02d0a2b95803843c59234082
1e3061c49cf62821ca7b83577f88d98a3bb6c2
f6e67d94acbc6d970eea079b96b678d23265cad
49994863abhba440212bd24232df21f9f3d812
7c1b2b173b9232c234dfec80002f9e993eb292
b74f751560e48488d2153ae2f512070a0ac260e2b
6.4.3. Modules

03DE8622BE6B2F75A364A275995C3411626C4D9F
8966319B82494077C21F66A8354E2C8CA0370464
B1EE5622E1F69EFC6FBA50B88753BE7D0B3E4CFAB

6.5. C&C server domain names

f72bba81c921(.)livejournal(.)com/data/rss
webstatisticaonline(.)tech
vpntap(.)top
rtm(.)dev
caimmoon(.)net
micro4n(.)top
ssdc0ol(.)top
cash-money-analitica(.)bit
money-cash-analitica(.)bit
vpnomnet(.)bit
vpnkeep(.)bit
dfe05d0573da(.)bit
d47ea26b7faa(.)bit
feb96eb2aa89(.)bit

6.6. C&C server IP addresses

5.154.190(.)167
5.154.190(.)168
5.154.190(.)169
5.154.191(.)157
5.154.191(.)154
5.154.191(.)174
5.154.191(.)225
37.1.206(.)78
88.208.28(.)147
91.207.7(.)69
91.215.153(.)31
93.170.168(.)218
93.190.139(.)66
95.183.52(.)182
109.236.82(.)1150
109.248.32(.)1152
131.72.138(.)169
138.201.104(.)161
154.70.153(.)125
158.255.6(.)150
158.255.208(.)197
185.61.149(.)70
185.61.149(.)78
185.82.201(.)45
185.82.216(.)114
185.128.42(.)237
185.141.27(.)249
185.169.229(.)42
188.138.71(.)117
200.74.240(.)80
200.74.240(.)134
212.48.90(.)155
213.184.127(.)137
217.23.6(.)29

6.7. PDB paths

getapulab.pdb
6.8. Strings

7. BIBLIOGRAPHY


3. «Информация для пользователей и партнеров №22531 от 20.01.2017,» [Online].

4. «Rig Exploit Kit via EiTest delivers malicious payloads,» 19 10 2016. [Online].

5. «Shortcut (computing),» 07 12 2016. [Online].


