PlugX goes to the registry (and India)

By Gabor Szappanos, Principal Researcher
PlugX goes to the registry (and India)

Contents

Overview
PlugX in registry
Peeled Tomato
Multi-staged installer shellcode
Overview
Recently we published a paper about the capabilities of APT groups [https://nakedsecurity.sophos.com/2015/02/03/exploit-this-evaluating-the-exploit-skills-of-malware-groups/].

One of the conclusions of the paper was that the authors behind the targeted attack campaigns usually have little knowledge about the actual exploit they are using to distribute their malware. But at the same time, we warned our readers never to underestimate them, because otherwise they are skilled, and quite capable of developing sophisticated backdoors.

One of the worst performances in our comparison of exploit development belonged to the infamous PlugX malware group(s). However, they recently came out with a couple of significant developments in the backdoor component, demonstrating the point above.

One of the improvements was the introduction of a peer-to-peer communication channel to other infected hosts [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html]. Variants using this technology have previously been spotted in the Rotten Tomato campaign [http://blogs.sophos.com/2014/10/30/the-rotten-tomato-campaign-new-sophoslabs-research-on-aps/].

Now additional samples have shown up from this generation. But in addition to the new communication method, some of them were showing another new characteristic: the payload was not stored as separate files, or embedded within the loader DLL, but instead was saved to the registry.

Malware hiding components in registry is not a revolutionary idea; we have seen that before. Most notably the recent Poweliks Trojan [https://blog.gdatasoftware.com/blog/article/poweliks-the-persistent-malware-without-a-file.html] stored the active script component in the registry. Even some of the APT malware families, like Poison or Frethog, occasionally used the registry as storage for the main payload.

There were precursors even within the criminal groups distributing PlugX: they used this method back in 2013 in a couple of cases for storing the Omdork (a.k.a. Sybin) payload. So it was only a question of when the same would happen to the main PlugX backdoor. And that time arrived this January.
PlugX goes to the registry (and India)

PlugX in registry
The new variants were distributed using two distinguishable classes of exploited carrier documents – though in both cases the CVE-2012-0158 exploit was used.

For the first type the distribution was part of a longer campaign, targeting India. This campaign spanned several months, from September 2014 to February 2015. During this time span different variants of the PlugX backdoor were observed as the final payload. Apparently, this was an ongoing operation, where the actors behind it used the latest available versions, as they came out of the factory. Additionally, a few affiliated malware families were distributed to the targets.

The samples of the second type showed up the first week of February. At this point we don’t have conclusive information about the scope and target of the campaign that used these samples.

PlugX payload in the registry

PlugX backdoors use a specific date parameter at specific places in the code. This constant could be used as a major version identifier: when the backdoor code was only slightly modified, the constant did not change. When the constant was updated, that usually meant a significant change in the code.

In earlier versions this constant was a meaningful date in hexadecimal representation (e.g. 0x20130810 in most of the next generation PlugX samples). In the P2P PlugX version it changed, now being a meaningful date in decimal representation (e.g. 0x13352AF = 20140719 in the case of the Rotten Tomato samples).

In the case of registry stored PlugX variants, this constant was stepped further to 20150108, which indicates a new development from the factory. Less than a month later these new variants were already spotted in targeted campaigns in India.
**Peeled Tomato**

The first campaign we labelled as Peeled Tomato, in reference to the earlier Rotten Tomato case, because they were clearly derived from those samples.

As a reminder, the original structure of the Rotten Tomato samples was the following:

The RTF documents started with an encrypted Zbot Trojan (remainder of the original template used for creating the samples), then a block using the CVE-2012-0158 exploit and the corresponding shellcode. After that, there was a block using the CVE-2014-1761 exploit and the corresponding first stage shellcode, followed by the second stage shellcode from the CVE-2014-1761 exploit, and finally the encrypted PlugX backdoor.

The first stage of the CVE-2014-1761 shellcode used a bad offset for the second stage code, thus this exploit never worked.
Having realized the failure of the attempt, the malware authors removed the CVE-2014-1761 exploit block. But even that was not done completely. As a result, they ended up with documents showing the following structure:

![Diagram showing the structure of the documents]

**Samples**

Not surprisingly, just like with several other campaigns, in this case it was observed that different malware families were distributed using similar carrier documents; only the encrypted payload was replaced at the end of the file. The shellcode used in the carrier was very convenient for this purpose: the length and location of the final payload was stored at the end of the file. It was possible to swap the payload without needing to modify the exploit condition and the shellcode itself. And this is exactly what the malware authors did.
A SophosLabs technical paper - February 2015

PlugX goes to the registry (and India)

9blog
This malware family was described in this blog: [http://www.fireeye.com/blog/technical/malware-research/2013/08/the-curious-case-of-encoded-vb-scripts-apt-nineblog.html]

19e9df0fbb9b10a90b62c12f205f0d3eeef3f14

Original name:
ghozaresh amniyati.doc

System activity:
Dropped to %PROFILE%\Application Data\Erease.vbe

SAV detection:
Troj/DocDrop-CH, VBS/9Blog-A

C&C servers:
www.freetimes.dns05.com
Free Dynamic DNS provider
PlugX goes to the registry (and India)

Smoaler
This malware family was described in this blog: [https://nakedsecurity.sophos.com/2013/07/15/the-PlugX-malware-factory-revisited-introducing-smoaler/], and traditionally has strong ties with PlugX, sharing dropper code and C&C infrastructure.

The samples were observed during the period between November 2014 and January 2015 in Russia.

Original name:
Проекты.doc

System activity:
Dropped to C:\Documents and Settings\All Users\Application Data\Microsoft\Windows\Burn\[COMPUTERNAME].dll and C:\Documents and Settings\All Users\Application Data\Microsoft\Windows\LiveUpdate\Mem\CrtRunTime.log; registered for startup in HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\run → [COMPUTERNAME]

Here [COMPUTERNAME] is the name of the computer, as set in Windows preferences.

SAV detection:
Exp/20120158-A, Troj/Smoaler-F

C&C servers:
lucas1.dnset.com
d746ca9b74fb04782e0e783980f7702a9356f1c7
PlugX goes to the registry (and India)

Original name: телефонная книга и почтовый адрес (2014.10).doc

The decoy document is the same as in the case of the Nineblog sample.

System activity:
Dropped to C:\Documents and Settings\All Users\Application Data\Microsoft\Windows\Burn\{COMPUTERNAME}.dll and C:\Documents and Settings\All Users\Application Data\Microsoft\Windows\LiveUpdate_Mem\CrtRunTime.log; registered for startup in HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\run → {COMPUTERNAME}

Here {COMPUTERNAME} is the name of the computer, as set in Windows preferences.

SAV detection:
Exp/20120158-A, Troj/Smoaler-F
PlugX v2
These samples were distributed in September and October 2014, in India.
6f845ef154a0b456afcf8b562a0387dabf4f5f85

Original name:
Indian Cooking Recipe.doc

**Indian Cooking Recipe: Butter Milk Kadi**

**Ingredients:**
- 2 cups butter milk (thick)
- 1 cup water
- ½ cup coconut gratings
- 4 green chillies
- 1 small piece haldi
- 1 tsp jeera
- 3 tsp ghee
- ½ tsp mustard seeds
- 1 sprig curry leaves
- salt to taste

**Method:**
Grind coconut gratings with haldi smoothly.
While removing masala put green chillies and cumin.
Grind for another 2 minutes.
Put enough water to bring the kadi to desired consistency.
Put salt. Keep it to boil. Then put thick butter milk.
Again bring to boil. Take out from flame.
Season with mustard and curry leaves in ghee.

**System activity:**
Dropped to C:\Documents and Settings\All Users\RasTls\RasTls.exe (digitally signed clean loader by Symantec), C:\Documents and Settings\All Users\RasTls\RasTls.dll (loader) and C:\Documents and Settings\All Users\RasTls\RasTls.dll.msc (payload); registered in HKLM\SYSTEM\CurrentControlSet\Services\RasTls → ImagePath

The payload is next generation PlugX [https://nakedsecurity.sophos.com/2014/06/30/from-the-labs-PlugX-the-next-generation/], date constant is 0x20130524

**SAV detection:**
Troj/DocDrop-CH, Troj/PlugX-AP

**C&C servers:**
supercat.strangled.net
Free dynamic DNS provider

a97827ae54e7969b9cbbe64d9ee81a835f2240
PlugX goes to the registry (and India)

**Original name:**
Calling Off India-Pak Talks.doc

**System activity:**
Dropped to `C:\Documents and Settings\All Users\RasTls\RasTls.exe` (digitally signed clean loader by Symantec), `C:\Documents and Settings\All Users\RasTls\RasTls.dll` (loader) and `C:\Documents and Settings\All Users\RasTls\RasTls.dll.msc` (payload); registered in `HKLM\SYSTEM\CurrentControlSet\Services\RasTls` → `ImagePath`

The payload is next generation PlugX [https://nakedsecurity.sophos.com/2014/06/30/from-the-labs-PlugX-the-next-generation/], date constant is 0x20130524

**SAV detection:**
Troj/DocDrop-CH, Troj/PlugX-AP

**C&C servers:**
nusteachers.no-ip.org
Free dynamic DNS provider

`e8a29bb90422fa6116563073725fa54169998325`
PlugX goes to the registry (and India)

Original name:
Human Rights Violations of Tibet.doc

Tibet: Human Rights Violations
Dr. Parasuram Rangarajan

Examining Tibet today, the first topic of concern to the international community is spread through the voice of H.H. Dalai Lama and Tibetan government-in-exile, human rights. One cannot overlook the frequency of self-immolations being committed by peaceful Tibetan Buddhist monks who seek to bring attention to the situation in Tibet.

Latest figures indicate that over 131 monks have so far immolated themselves in the last two years[1]. These are only reported cases and more would have died in vain. Two points to make on this issue are:

1. The Tibetans are able to immolate themselves for the cause despite very restrictive and strict security measures as well as arrest and imprisonment of the relatives of the victims inside Tibet.

2. The immolations are also taking place outside Tibet proper.

The U.S. Commission on International Religious Freedom (USCIRF) released its annual report on April 30th, 2014 identifying China as a country of concern noting the self-immolations and detention of monks, forced renunciations of faith including the Uighur Muslim, Protestant, and Catholic communities, and discriminating of religious leaders which "merits a seat at the table with economic, security, and other key concerns of U.S. foreign policy."[2]

The Tibetan government-in-exile has found a home in India residing peacefully for the past few decades but the government of India has done little beyond extending basic citizenship in terms of assistance to the Tibetan people to defend their human rights in China. The question is could India do more? How can a resolution in the United Nations, at an agency such as the U.N. Human Rights Council (UNHRC) be introduced to bring it to the world, the desperate situation of the people in Tibet?

System activity:
Dropped to C:\Documents and Settings\All Users\RasTls\RasTls.exe (digitally signed clean loader by Symantec), C:\Documents and Settings\All Users\RasTls\RasTls.dll (loader) and C:\Documents and Settings\All Users\RasTls\RasTls.dll.msc (payload); registered in HKLM\SYSTEM\CurrentControlSet\Services\RasTls → ImagePath

The payload is next generation PlugX [https://nakedsecurity.sophos.com/2014/06/30/from-the-labs-PlugX-the-next-generation/], date constant is 0x20130524

SAV detection:
Troj/DocDrop-CH, Troj/PlugX-AP

C&C servers:
ruchi.mysql.net
Dynamic DNS provider
a7e52cb429ac22cc20be77158f97f6f9dd887e1f

This sample is an outlier, as it was distributed in January 2015, and in Russia. The decoy document is also unconventional, of minimalistic design.

But the carrier document and the C&C server name shows correlation with the rest of the campaign.
PlugX goes to the registry (and India)

Original name:
Calling Off India-Pak Talks.doc

System activity:
Dropped to C:\Documents and Settings\All Users\DRM\usaha\usaha.exe (digitally signed clean loader by Kaspersky) and C:\Documents and Settings\All Users\DRM\usaha\usaha.dll (malware loader) and C:\Documents and Settings\All Users\DRM\usaha\usaha.dll.avp (payload).

Registered for startup in HKLM\SYSTEM\CurrentControlSet\Services\usaha → ImagePath

The payload is next generation PlugX [https://nakedsecurity.sophos.com/2014/06/30/from-the-labs-PlugX-the-next-generation/], date constant is 0x20130810

SAV detection:
Exp/20120158-A, Troj/PlugX-AP

C&C servers:
lucas1.freetcp.com
Free dynamic DNS provider

P2P PlugX
These samples were distributed in January 2015, in India.

147fbdfeed9f0825026b3b3ce558c3ad00410b11
PlugX goes to the registry (and India)

Original name:
Minutes of meeting.doc

System activity:
Dropped to C:\Documents and Settings\All Users\DRM\yEjtQOtPhI\fsguidll.exe (digitally signed clean loader by F-Secure), C:\Documents and Settings\All Users\DRM\yEjtQOtPhI\yslapi.dll (loader) and C:\Documents and Settings\All Users\DRM\yEjtQOtPhI\yslapi.dll.gui (payload),

Registered for startup in HKLM\SYSTEM\CurrentControlSet\Services\gzQkNtWeabrwf →ImagePath

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20141028.

SAV detection:
Troj/DocDrop-CH, Troj/PlugX-AP

C&C servers:
unisers.com
Registrant Name: wang cheng
Registrant Organization: wang cheng
Registrant Street: BeijingDagueROAD136
Registrant City: Beijing
Registrant State/Province: Beijing
Registrant Postal Code: 100001
Registrant Country: CN
Registrant Phone: +86.01085452454
Registrant Phone Ext:
Registrant Fax: +86.01085452454
Registrant Fax Ext:
Registrant Email:bitumberls@163.com
8ee8ab984cb01762dfe6d341278b87a7c83906cf
PlugX goes to the registry (and India)

Original name:
U.S., India to formulate smart city action plans in three months.doc

System activity:
Dropped to C:\Documents and Settings\All Users\DRM\inbjUkRVq\fsguidll.exe (digitally signed clean loader by F-Secure), C:\Documents and Settings\All Users\DRM\inbjUkRVq\fslapi.dll (loader) and C:\Documents and Settings\All Users\DRM\inbjUkRVq\fslapi.dll.gui (payload),

Registered for startup in HKLM\SYSTEM\CurrentControlSet\Services\brwTRsulGqjj →ImagePath

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20141028.

SAV detection:
Troj/DocDrop-CH, Troj/PlugX-AP

C&C servers:
unisers.com
Registrant Name: wang cheng
Registrant Organization: wang cheng
Registrant Street: BeijingDagueROAD136
Registrant City: Beijing
Registrant State/Province: Beijing
Registrant Postal Code: 100001
Registrant Country: CN
Registrant Phone: +86.01085452454
Registrant Phone Ext:
Registrant Fax: +86.01085452454
Registrant Fax Ext:
Registrant Email:bitumberls@163.com
PlugX goes to the registry (and India)

Registry PlugX
These samples were typically distributed in January-February 2015, in India.

a4602a357360b0ed8e9b0814b1322146156f7f6

Original name:
CHINA NEWS BRIEF 09 of 2015.doc

System activity:
Dropped to C:\Documents and Settings\All Users\DRM\sock5proxy\SX.EXE (digitally signed clean loader by Microsoft) and C:\Documents and Settings\All Users\DRM\sock5proxy\SXLOC.DLL; registered in HKLM\SYSTEM\CurrentControlSet\Services\sock5proxy → ImagePath; payload stored in the registry in HKCU\Software\BINARY → SXLOC.ZAP

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05. html], date constant is decimal 20150108.

SAV detection:
Exp/20120158-A, Troj/PlugX-AP

C&C servers:
freemoney.ignorelist.com
Free dynamic DNS provider

03b2a660d68004444a5189173e3b8001f4a7cd0b
PlugX goes to the registry (and India)

**Original name:**
Draft contract CMS Trg System.doc

---

**CONTRACT AGREEMENT**

**BETWEEN**

**THE GOVERNMENT OF THE REPUBLIC OF INDIA, MINISTRY OF DEFENCE**

&

**M/S BHARAT ELECTRONICS**

FOR

**SUPPLY OF CMS TRAINING SYSTEM**

Contract No. ......................

**System activity:**
Dropped to \C:\Documents and Settings\All Users\DRM\sock5proxy\SX.EXE (digitally signed clean loader by Microsoft) and \C:\Documents and Settings\All Users\DRM\sock5proxy\SXLOC.DLL; registered in HKLM\SYSTEM\CurrentControlSet\Services\sock5proxy → ImagePath; payload stored in the registry in HKCU\Software\BINARY → SXLOC.ZAP

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20150108.

**SAV detection:**
Exp/20120158-A, Troj/PlugX-AP

**C&C servers:**
freemoney.ignorelist.com

Free dynamic DNS provider
Multi-staged installer shellcode

This second batch of exploited documents had a different structure. All start with a heading RTF content (which is exactly the same in all of the documents), followed by the block that exploits the CVE-2012-0158 vulnerability, along with the first stage shellcode, followed by the second and third stage shellcodes, and finally the encrypted payload executable.

RTF heading of exploited documents

The shellcode itself is encrypted with a 4 byte XOR algorithm, with a lot of inserted junk instructions:

```
fprem1
add  edi, ebx
jz  short loc_13B
nop
fnclex
fldl2e
nop
and  ebx, ebx
test  eax, eax
fsin
xor  [edi], esi
jp  short loc_14B
f2xml
mov  edx, edx
nop
cld
fst  st(1)
pop  edi
jle  short loc_157
fldpi
fprem1
cmp  edi, esi
fdivrp  st(1), st
```

In the above code sample, only the XOR [EDI]. ESI instruction is meaningful, performing the decryption of the one dword; the rest are only polymorphic junk.
PlugX goes to the registry (and India)

The underlying shellcode is multi-stage and has already been observed in an earlier sample dropping a PlugX v2 variant (SHA1: 9b90d6608ba6167619b5991fd70319dfcd1fa881, date constant 0x20140613), but in that case without the top level cryptor.

After the initial bootstrap code is decrypted, it identifies the carrier by looking for ‘DCBA’ at file offset 0x4e28. If it is found there, it allocates a memory area and decrypts (using one byte XOR algorithm) the next stage starting from right after the ID string.

The second stage code decrypts and drops two files: the self-extracting installer archive M.B and the first stage installer M.T into the %TEMP% folder, then allocates another memory region, decrypts, copies and executes the third stage shellcode there.

The third stage shellcode copies the first stage installer (which is a DLL library) M.T into %WINDOWS%\Tasks\n.dll, then executes by calling LoadLibrary to load it. The Windows loader upon loading the DLL will execute its entry code. This entry code runs the self-extracting installer archive M.B which will do the final malware installation in the system. This final piece of installation process is malware family dependent.

This new shellcode also indicates some heavy development in the PlugX factory. Both this kind of multi-stage shellcode and the external cryptor indicate that although the group is not top class in exploit development, in conventional malware development they show serious skills, which makes them dangerous.

dea6525b696df4643b10eb91381d95eec51479d7
PlugX goes to the registry (and India)

Original name:
paris_declaration january_final.doc

The dropped decoy document is corrupted. On opening it, Word will show a conversion dialog as a result of the incomprehensible content.

System activity:
Dropped to C:\Documents and Settings\All Users\DRM\emproxy\SX.EXE (digitally signed clean loader by Microsoft) and C:\Documents and Settings\All Users\DRM\emproxy\SXLOC.DLL and %WINDOWS%\Tasks\n.dll

Registered for startup in HKLM\SYSTEM\CurrentControlSet\Services\sock5proxy → ImagePath and by dropping n.dll into the Windows Tasks directory.

The n.dll file is a first stage installer, loads M.B, which is dropped into the %TEMP% directory. This installer is a self-extracting WinRAR that contains RasTls.exe and a config file. After the installation, this RAR SFX file is removed from the system.

Payload is stored in the registry in HKCU\Software\BINARY → SXLOC.ZAP

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20150108.

SAV detection:
Troj/DocDrop-CD, Troj/Omdork-E, Troj/PlugX-AP

C&C servers:
sumy2012.jkub.com
Free dynamic DNS provider

6340a7916db67c1b6dc1731014bb440435578c66
**PlugX goes to the registry (and India)**

**Original name:**
Obama against IS.doc

The dropped decoy document is corrupted just like in the previous case.

**System activity:**
Dropped to C:\Documents and Settings\All Users\DRM\emproxy\SX.EXE (digitally signed clean loader by Microsoft) and C:\Documents and Settings\All Users\DRM\emproxy\SXLOC.DLL and %WINDOWS%\Tasks\n.dll

Registered for startup in HKLM\SYSTEM\CurrentControlSet\Services\sock5proxy → ImagePath and by dropping n.dll into the Windows Tasks directory.

The n.dll file is a first stage installer, loads M.B, which is dropped into the %TEMP% directory. This installer is a self-extracting WinRAR that contains RasTls.exe and a config file. After the installation, thisRAR SFX file is removed from the system.

Payload is stored in the registry in HKCU\Software\BINARY → SXLOC.ZAP

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20150108.

**SAV detection:**
Troj/DocDrop-CD, Troj/Omdork-E, Troj/PlugX-AP

**C&C servers:**

dheeraj_gaurav.mooo.com
Free dynamic DNS provider

739405cad3650ed0447a475f50f814f7c9787ff4
PlugX goes to the registry (and India)

**Original name:**
N/A

On execution this dropper displays a blank decoy document.

**System activity:**
Dropped to `C:\Documents and Settings\All Users\DRM\RdeGL\fsguidll.exe` (digitally signed clean loader by F-Secure) and `C:\Documents and Settings\All Users\DRM\RdeGL\fsapi.dll` (malware loader) and `C:\Documents and Settings\All Users\DRM\RdeGL\fsapi.dll.gui` (payload) and `%WINDOWS%\Tasks\n.dll`

Registered for startup in `HKLM\SYSTEM\CurrentControlSet\Services\dUuNvGfDQkAll → ImagePath` and by placing `n.dll` in the Windows Tasks directory.

The payload is next generation P2P PlugX [http://blog.jpcert.or.jp/2015/01/analysis-of-a-r-ff05.html], date constant is decimal 20141028.

The `n.dll` file executes a backup installer, M.B, which is dropped into the `%TEMP%` directory. The only problem is that this file is never created.

**SAV detection:**
Troj/DocDrop-CD, Troj/Omdork-E, Troj/PlugX-AP

**C&C servers:**
www.notebookhk.net

Registrant Name: lee stan
Registrant Organization: lee stan
Registrant Street: xianggangdiqiu
Registrant City: xianggangdiqiu
Registrant State/Province: xianggang
Registrant Postal Code: 796373
Registrant Country: HK
Registrant Phone: +0.04375094543
Registrant Phone Ext:
Registrant Fax: +0.04375094543
Registrant Fax Ext:
Registrant Email: stanlee@gmail.com

56b3f0f03ae12b56c000df67c1153d518c8a66fc

This sample is an outlier. It does not distribute PlugX, but uses a strikingly similar persistence method, with exactly the same file names that are used with PlugX installations. Only the final payload is a different backdoor, Omdork, which has earlier been observed in PlugX related distribution channels.
PlugX goes to the registry (and India)

Original name:
United Nations Security Council Committee Pursuant to Resolutions1267.doc

System activity:
Dropped to C:\Documents and Settings\All Users\FlashUpdate\RasTls.exe and C:\Documents and Settings\All Users\FlashUpdate\msi.dll.mov (encrypted payload) and %WINDOWS%\Tasks\n.dll.

The persistence is achieved by two methods: RasTls.exe is registered in HKCU\Software\Microsoft\Windows\CurrentVersion\Run → msusr, and the n.dll is dropped to the Windows Tasks directory for automatic execution.

While the file names are the same as in the case of many PlugX deployments, the files themselves are very different.

RasTls.exe is not digitally signed, it is the loader Trojan, that loads the encrypted payload from a resource. This payload itself contains a loader code, and an embedded executable, that is the final payload.

The n.dll file executes a backup installer, M.B, which is dropped into the %TEMP% directory. This installer is a self-extracting WinRAR that contains RasTls.exe and a config file.

There are still reasons to believe that this malware is strongly connected to the PlugX group:

- It uses the same filenames as some of the PlugX deployments
- It uses the same carrier document as the other PlugX variants in this campaign, including the unique shellcode
- The same n.dll is used in both the Omdork and PlugX deployments
PlugX goes to the registry (and India)

SAV detection:
Troj/DocDrop-CD, Troj/Omdork-E

C&C servers:
www.togolaga.com
Registrant Name: wang feng
Registrant Organization: wang feng
Registrant Street: beijingshi
Registrant City: beijingshi
Registrant State/Province: beijing
Registrant Postal Code: 100000
Registrant Country: CN
Registrant Phone: +86.01090888962
Registrant Fax: +86.01090888962
Registrant Email: battuya_2002@yahoo.com

More than 100 million users in 150 countries rely on Sophos as the best protection against complex threats and data loss. Sophos is committed to providing complete security solutions that are simple to deploy, manage, and use that deliver the industry’s lowest total cost of ownership. Sophos offers award winning encryption, endpoint security, web, email, mobile, server and network security backed by SophosLabs—a global network of threat intelligence centers. Read more at www.sophos.com/products.