White Paper

Bitdefender®

APT28 Under the Scope

A Journey into Exfiltrating Intelligence and Government Information

Contents

Authors:

- Răzvan BENCHEA – Team leader of Malware Research
- Cristina VATAMANU – Senior Malware Researcher
- Alexandru MAXIMCIUC – Senior Malware Researcher
- Victor LUNCAŞU – Junior Malware Researcher

The discovery of Stuxnet in the nuclear processing plant in Natanz, Iran laid the ground for a new family of cyber-attacks: advanced persistent threats. Although the term has since become highly popular, state-sponsored cyber-intelligence operations have been carried out since long before the advent of Stuxnet or Flamer; less known advanced persistent threats such as APT28 (or Sofacy) have been covertly running in Europe since 2007.

Preface

When it was initially coined, the term "Advanced Persistent Threat" was used to define an attack that, unlike regular commercial-grade malware, would focus on a particular target, its network topology and defenses. The purpose of this type of attack is exfiltration of sensitive data over a long period of time or silently crippling their industrial processes. In recent years, APTs have started to define persistent attacks launched by foreign state adversaries that no security company or victim government would explicitly name for fear of economic or political repercussions. Another aspect that makes calling on attacking states is the difficulty in attribution: the Internet has no physical boundaries and the technical skills of the attacking parties can make it look as the attack was launched by a rival state.

To connect the dots between an identified attack and the state actor(s) behind it, companies like Bitdefender look for solid evidence inside the APT code or in the used communication infrastructure. The following report is a technical investigation of some particularities in the APT28 payload implementation that allowed us to link the threat to its operators.

Targeted victims

Our recent investigation into the Sofacy operations revealed that the cyber-group is extremely active and focused on specific regions. The primary targets of APT28 are potential victims in several countries such as Ukraine, Spain, Russia, Romania, the United States and Canada.

A particular interest was found for Ukraine. Between Tuesday, 10th of February and Saturday, 14th of February 2015, the APT28 team scanned 8,536,272 IPs for possible vulnerabilities.

Coincidentally, during this period, in Minsk, the political leaders of Belarus, Russia, Germany, France and Ukraine were participating in a summit, discussing the ceasefire in the Donbass region in the east of Ukraine.

After February 14th, the APT28 operators shifted their attention to Spain.

The map of affected countries is illustrated in Figure 1.

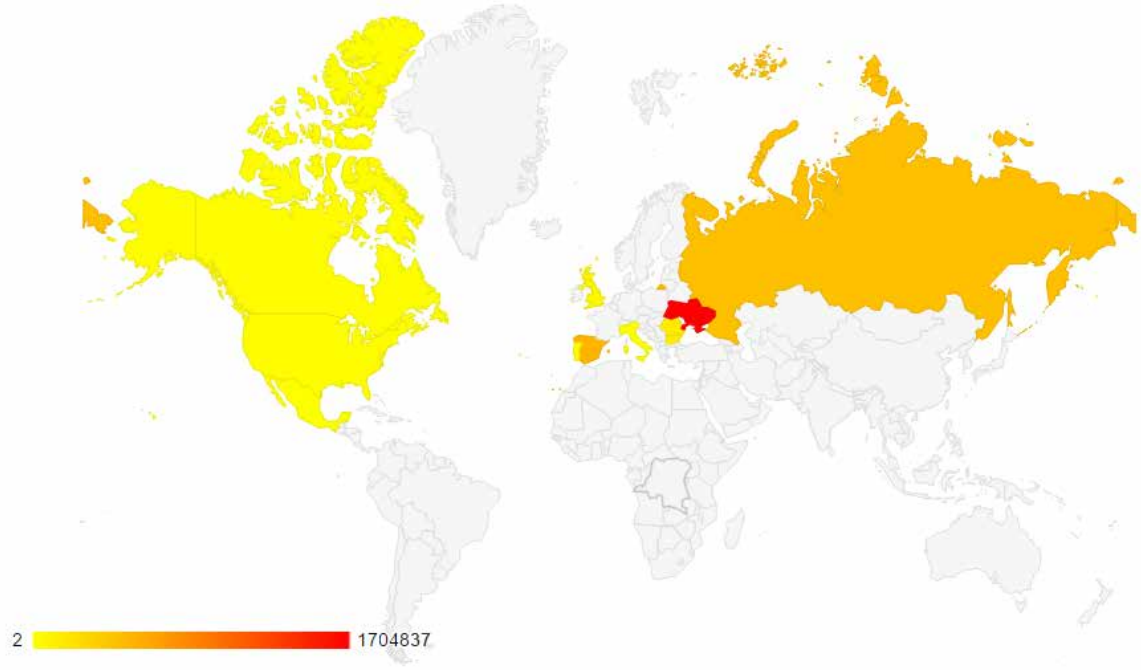


Figure 1

It is currently unknown what criteria the APT28 operators used to select targets, but our research identified that they are picked from a list of vulnerable IP addresses prepared beforehand. The same research shows all targets belong to different categories: political (**[removed]**), e-crime services, telecommunication services or aerospace industry.

More information about the victims can be found in **Appendix 1**.

Attribution

We have reasons to believe that the operators of the APT28 network are either Russian citizens or citizens of a neighboring country that speak Russian. Our assumption is supported by different markers identified during analysis.

When we first analyzed the pool of files we had collected on APT28, we counted the number of binaries compiled between 08:00 and 18:00 on working days (Monday to Friday) for every time zone. It was revealed that UTC+4 stands out, as 88% of files are compiled during this interval. Out of the other countries sharing the same time zone (Russia, Georgia, Azerbaijan), Russia is the only country that possesses the necessary skills and resources to pull off this kind of attack.

In the graphic below (Figure 2) the samples are grouped by the hour they were compiled (UTC+4). It can be seen that most were compiled between 08:00 and 18:00.

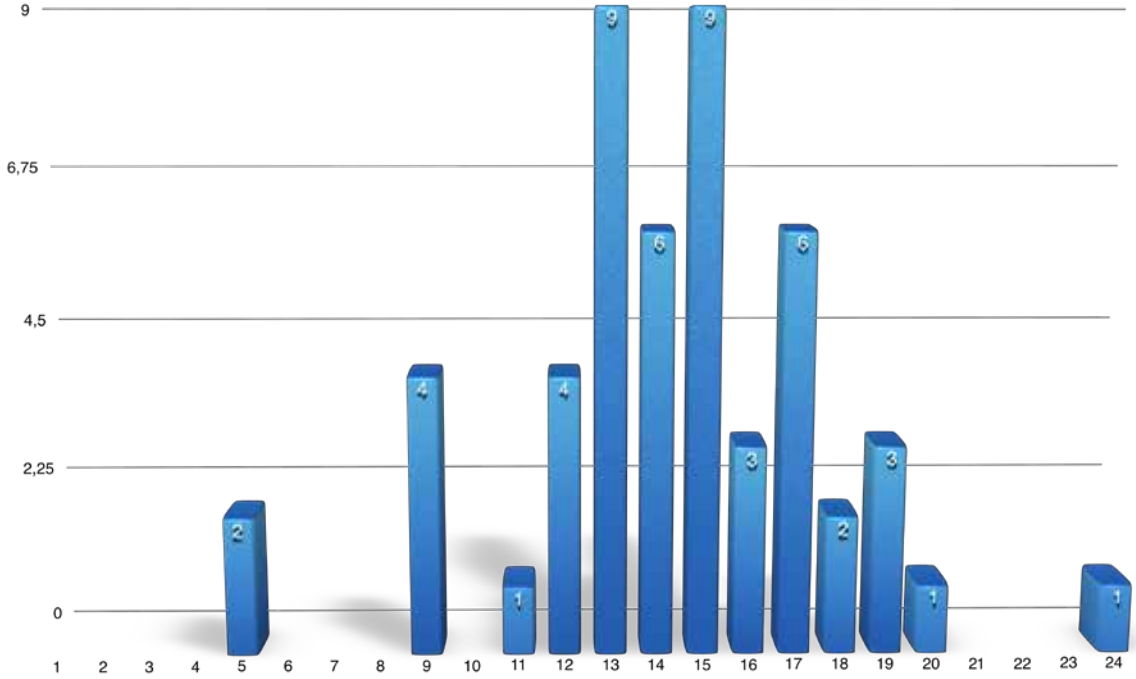


Figure 2

Another clue that supports the assumption that the authors are from a Russian-speaking country is found in a hacking tool designed to grant system privileges. This tool was found along with files related to APT28 files. The header of the file called xp.exe (78450806E56B1F224D00455EFCD04CE3) features a hardcoded path to a debug file (C:\Users\Пользователь\Desktop\cve-2014-4076\cve-19abdba\Debug\CVE-2014-4076.pdb). The string Пользователь translates to users in Russian.

We have reasons to believe that this file is part of the same APT28 attack since it had its modification/creation time modified to 14/04/2008, 16:00, just like most of the other APT28 files used in the attack. All files used in the attack were compiled after 2013. Changing the date was just a measure to throw off any victim who would perform a shallow search for new files, should they have any doubts about the security of their system.

Scanning for new targets

One of the main purposes of the server we analyzed is to automate the process of finding new victims that can be marked as "vulnerable." This is done by exhaustively scanning and probing a pre-determined pool of IP addresses contained in a range. The server-side application has four main components, as follows:

a python script (**gen_ip.py**) that generates random IPs from a certain class

a database (**shodb**) for centralizing all the data

the scan bots that probe and screen potential victims

a Django (**sho**) application for bot administration and reporting.

Rather than taking a "shotgun" approach, the APT28 victims are hand-picked. The script has a hard-coded list of IP classes; it iterates through these subnets and generates a list of random IPs within the range. If the operators scanned all IPs within the range, it would increase the chances of the attacked party noticing their systems are actively probed from the outside. After the random generation process completes, these IPs are added to the database. Each IP address is then given a priority level.

At the time of writing, the database contains the 58,624 IPs related to the incident described in Appendix 1. All these IP addresses have the same priority level, set to 1.

The scanning bots are individual systems, all situated in different networks. We presume this was a necessary approach to make the scanning process look organic. If a single IP or a set of IPs from the same network would scan entire subnets, it would raise suspicions or trip intrusion detection systems and their activity would likely be interrupted.

To get their scan jobs, each bot would authenticate to the server via cookies. Upon successfully authenticating, the scan bot receives a list of maximum 16 IPs per request. For each IP, it will scan certain ports using the nmap tool (Appendix 1). If it finds open ports, it then contacts the server and the information retrieved by nmap is saved in the database. Subsequently, the respective IP address is marked as vulnerable. The process is described in the figure below (Figure 3):

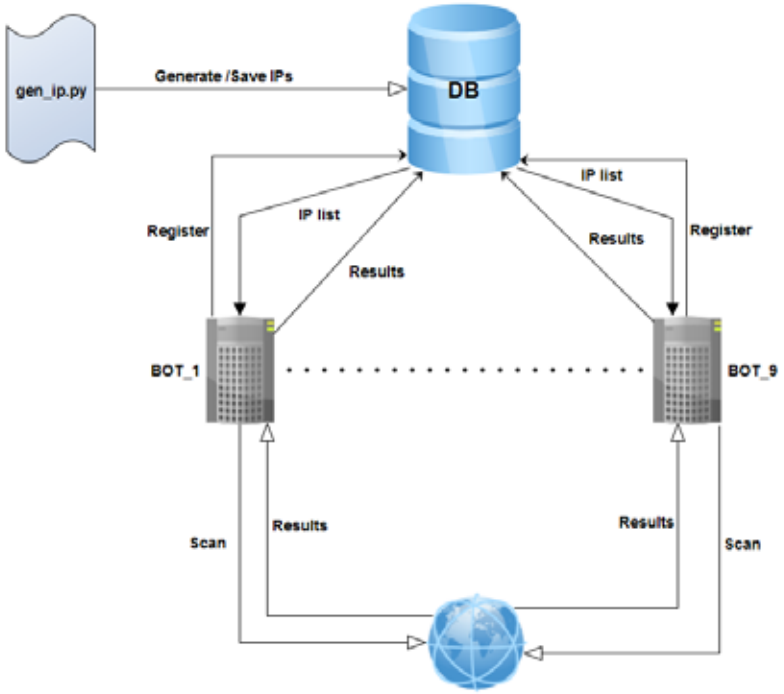


Figure 3

BOT UID	TARG NUM	VALID NUM	IP	LAST ACT
345051441193	8241	0	192 215	2015-03-17 05:46:35.012289+00:00
345051780961	5636	0	5. 36	2015-03-04 09:47:35.428945+00:00
345051783097	0	0	87 .155	2015-02-14 13:26:34.879942+00:00
161343679352	10335	0	5. 160	2015-04-18 02:36:10.382767+00:00
161343570082	9494	0	5. 150	2015-04-18 02:31:43.480646+00:00
345051780666	4122	0	87 12	2015-03-18 07:40:52.701857+00:00
161340745510	5685	0	5. 223	2015-04-18 02:53:33.532651+00:00
207376762666	10879	0	23. 98	2015-06-08 21:28:13.256460+00:00
345051770354	4232	0	87. .18	2015-06-06 04:02:03.173137+00:00

Scanned num:58624

Figure 4

activity.

w (Figure 4):

The scan bots are geographically distributed as follows: 3 in the United States, 3 in United Kingdom and 3 in Bulgaria (Figure 5):

Figure 5

More information is provided in Appendix 2.

Attack flow

The APT28 group relies on three distinct attack vectors to infect their targets: spear phishing e-mails with crafted Word and Excel documents attached, phishing websites hosted on typosquatted domains and malicious iFrames leading to Java and Flash zero-day exploits.

The client usually gets infected by accessing an URL hosting an exploit kit. Upon successful exploitation, a first stage dropper (in our case the name is **runrun.exe**) is written to the disk. Its main purpose is to drop a file (**api-ms-win-downlevel-profile-l1-1-0.dll**) and execute it using **rundll32.exe**. This, in turn, contacts the C&C server and downloads the second stage component.

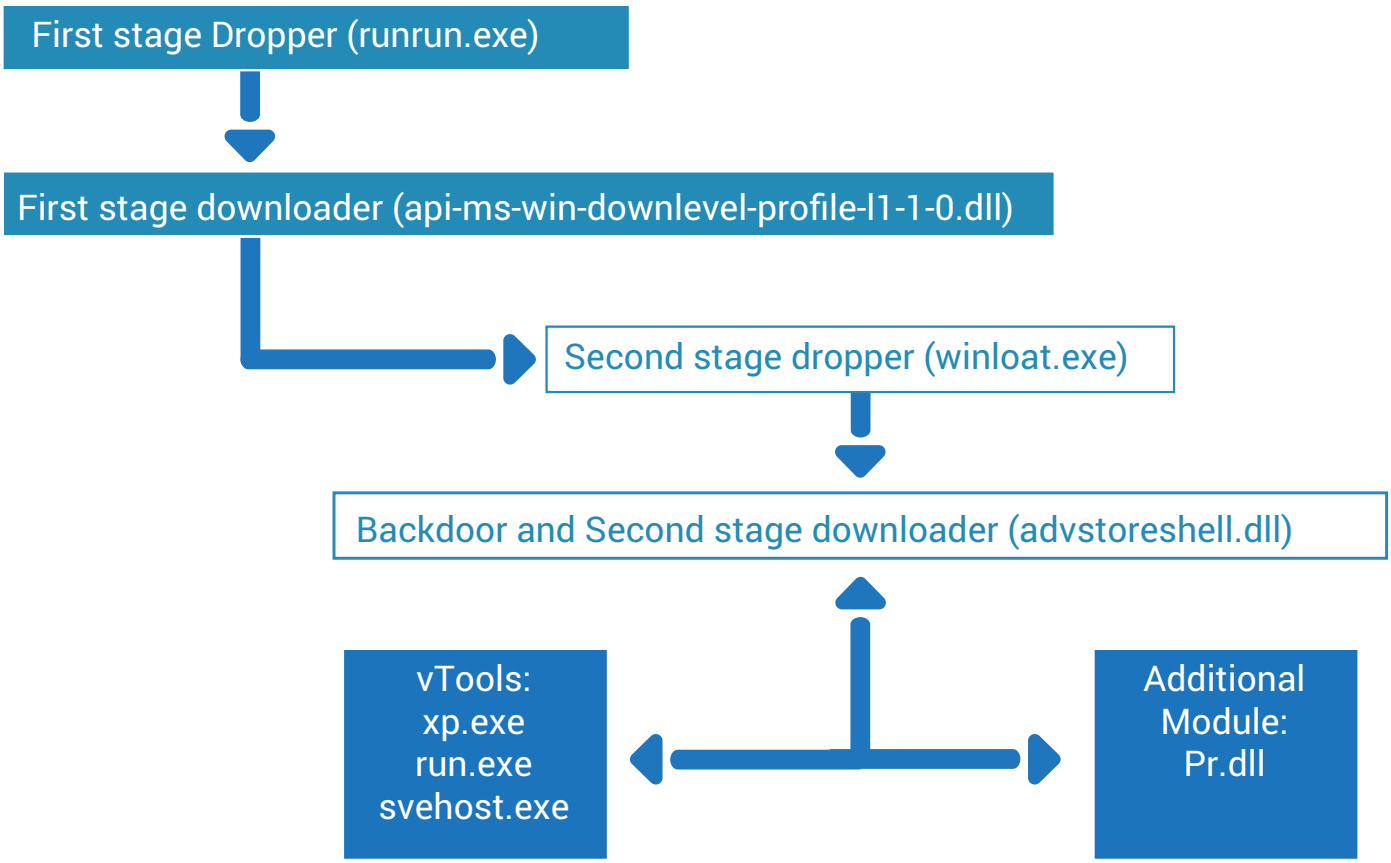
The second component is installed using the same method as described above. First, a dropper is executed (**winloat.exe**) which writes one of the key components of the attack (**advstoreshell.dll**) to the disk, along with a configuration file (**msd**). The configuration file contains essential information such as a list of three servers that the backdoor will try to contact (**win*********ore.net ,micro********er.com** and **1***.net**), the interval between requests and a flag that indicates whether key logging should be activated or not.

At this point, the attacker takes control of the machine and deploys different tools and components to achieve his goal. In the case we

analyzed, the attacker downloaded 3 hacking tools:

1. A tool to dump passwords from logged-in users (**`run.exe`**)
2. A tool that exploits a privilege escalation vulnerability (CVE-2014-4076) to gain system privileges (**`xp.exe`**)
3. A tool that acts as a proxy and allows the attacker to contact the system even if it is behind a router (**`svehost.exe`**)

An additional component is also downloaded (**`pr.dll`**) that is a modular component used to upload stolen data to the C&C.



The attacked flow is described in the figure below (Figure 6):

Figure 6

More information about the tools used is provided in Appendix 3. The first stage component and the second stage component are further discussed in Appendix 4 and Appendix 5, respectively. Finally, the additional module (pr.dll) is analyzed in Appendix 6.

Right after deploying their payloads, the attacking party changed the time stamps of all the files downloaded by the second stage dropper to 14/04/2008, 16:00. This technique allowed the attackers to hide the files in case victims would suspect the attack and attempt to list new files written on the disk.

The table below shows the compilation date and the file creation time for each of the files involved in the attack.

File Name	Compilation Date	Creation Time
%allusersappdata%\xp.exe	12/03/2015 12:54:57	14/04/2008 16:00
%allusersappdata%\run.exe	05/04/2013 12:51:41	14/04/2008 16:00
%allusersappdata%\svehost.exe	22/04/2015 11:49:54	14/04/2008 16:00
%localappdata%\Microsoft Help\advstorshell.dll	30/04/2015 13:13:13	14/04/2008 16:00
%allusersappdata%\msd		14/04/2008 16:00
%allusersappdata%\Pr.dll	13/05/2015 22:05:57	14/04/2008 16:00
%temp%\winloat.exe	30/04/2015 14:24:54	13/05/2015 17:23
%temp%\api-ms-win-downlevel-profile-l1-1-0.dll	08/05/2015 13:39:32	13/05/2015 16:22
%temp%\runrun.exe	08/05/2015 13:45:31	13/05/2015 16:22

Table 1

The latest creation date is 13/05/2015, which hints at the date the attack happened. Given that it took almost an hour from the moment the first downloader got written to the disk to the arrival of the second stage downloader, this process was likely carried out manually by a human operator. An important observation is that all of the components, except one, had been compiled before the attack. "%allusersappdata%\Pr.dll" is the only file that was compiled 5 hours after the compromise. This suggests this file was specifically built for the target.

Attacked victims

The victims we have identified belong to different industries and sectors. Our research shows that the attack targeted political figures, government institutions, telecommunication and e-crime services, as well as aerospace companies from Germany, Ukraine, Russia and Romania.

Some of the victims could be identified by analyzing the information stored on the central server we have investigated. The stored information contained traces of stolen information in the form of e-mails. Some were reported by our internal systems.

The e-mails we have identified on the central server revealed victims from the first two categories.

Political figures

On 13th May 2015 two **.pst** files named **'C:\ProgramData\backup.pst'** and **'C:\ProgramData\backup2.pst'** were copied to the server. The e-mails seem to be addressed to the **[redacted]**.

Aerospace industry

On July 9th 2015, fourteen archives were copied on the server. Among them were two **.**dbx (Outlook Express database) files. The e-mails seem related to **[redacted].**

On June 18th the file "F:\Outlook Express\AC_VTS_ĐÊÒ.dbx" was transferd to the central server. The emails belong to **[removed]**

Also on June 18th, the file "F:\Outlook Express\Kydla.dbx" was transfered to the server. The emails belong to **[redacted]**

All these victims seem related to the aerospace industry or aircraft research programs. The context also supports the Russian origin, given the huge media coverage of the Russian "smartplane" **PAK FA T-50 Fighter** and America's **F-35**. We presume the APT28 authors might have attempted to explore new technologies being developed in the aerospace industry for integration.

Our internal services also flagged a number of computers as under attack, such as those belonging to **[redacted]**. In Romania, these flagged computers are part of government infrastructures or are closely related to the Romanian government.

Appendix 1 (Target campaigns)

Analyzing the APT28 command and control server identified two ongoing attempts to find new victims.

The process consists of scanning the ports of a large IP list, extracted from a database. The port scanning process is carried out via the free tool "nmap". If one of the ports is found open, the IP is marked as "vulnerable" and the corresponding information is saved. The command used for probing is illustrated in figure below:

```
nmap
            -T5
            -p 21,22,23,25,80,110,143,443,465,993,995,8080,7071,3389,5900
            -sV
            -O
            --version-light
            --script=banner
            --script=http-headers
            -oX out_name
            -iL targets_file
```

Figure 7

Where:

-T5: scan will map a host in very little time, provided you are on a very fast network or don't mind losing some information along the way.

-p: scan specific ports

-sV: detect remote services (server / daemon) version numbers

-O: identify a remote host apps and OS

--version-light: enable light mode that makes version scanning much faster, but it is slightly less likely to identify services

--script=banner: shows ftp banner responses

--script=http-headers: performs a HEAD request for the root folder ("/") of a web server and displays the HTTP headers returned.

-**oX:** logs the results of the scan in XML form into the file specified as an argument

-iL: reads target specifications from a given configuration file rather than from the command line; the file should contain a list of host or network expressions separated by spaces, tabs or newlines.

The first attempt we have identified started on 10th of February 2015 when a list of 8.762.102 IPs were scanned. Out of this large list, 1.712.363 IPs were marked as "vulnerable." The probing process lasted until February 14th. Most of the IPs were from Ukraine (8.536.272 scanned, 1.704.837 found vulnerable).

The number of scanned IPs is exhaustive, almost equal to the total number of IPs allocated to Ukraine. Searching through the major events regarding Ukraine in that time frame, we realized that the probing took place during the summit in Minsk regarding the ceasefire in the Donbass region of Eastern Ukraine. On February 11, the leaders of Belarus, Russia, Germany, France and Ukraine met to negotiate the new set of peace measures that were announced on February 12 and signed on February 15.

The targeted countries in these campaigns are illustrated in table below (Table 2):

Country	Number of vulnerable IPs
Ukraine	1704837
Russian Federation	4666
Romania	1287
Bulgaria	1272
United States	150
Canada	149
Italy	2

Table 2

On February 14, a backup of the database was saved in a file (file.csv) and the attacker's focus moved from Ukraine to Spain.

On February 16, a new attempt to find new victims was made. The new database was loaded with 58,624 new IP addresses, out of which 6,146 were found vulnerable after probing. The targeted countries in this case are illustrated in Table 3:

Country	Number of vulnerable IPs
Spain	5511
United Kingdom	584
Portugal	47
United States	2
Mexico	2

Table 3

Since in this second attempt the set of IPs is rather small, we presume that this attack was highly targeted to specific institutions. By analyzing the target IP ranges, we were able to identify some of the victims.

The decision logic related to which victims should be infected is unknown to us at the time of writing, but we presume the vulnerable targets are next engaged by a human attacker rather than in an automated way.

Appendix 2 - The probing process

The screening process for finding new possible victims is carried out via multiple bots managed by a central command and control server. The operational flow can be broken down as follows:

1. Generating new victims

Potential victims are identified by port-scanning a list of IPs belonging to different subnets. This list is generated by a script, **gen_ip.py**, which iterates through IP classes and randomly picks IPs for port-probing. At the time of writing, the script has 11 IP classes hardcoded, which leads us to believe that victim organizations are picked manually.

The script generates random integer numbers that represent IPs belonging to these classes. The fact that it doesn't iterate consecutively through all the IPs shows that the attackers have taken all precautions not to trigger alarms in the victims' intrusion detection system - scanning all the IPs from a subnet may raise suspicions.

All generated IPs are added to a database on the central server. Most of these IPs are from Spain and belong to political institutions (**[redacted]**), telecommunication companies and public affairs institutions.

2. Centralizing all information

The command and control server's most important asset is the **shodb database**. After restoring the database, the following relations were found (Figure 9):

```
                List of relations
 Schema |            Name             | Type  |  Owner
--------+-----------------------------+-------+----------
public | sho_bot                      | table | sho
public | sho_bot_plugins              | table | sho
public | sho_code                     | table | sho
public | sho_cookie                   | table | sho
public | sho_ip                       | table | sho
public | sho_log                      | table | sho
public | sho_plugin                   | table | sho
public | sho_setting                  | table | sho
```

Figure 9

- **The sho_bot** table contains information regarding the active bots (Table 5)

id	uid	code_id	info	last_activity
8	345051441193	1		2015-02-12 16:09:53.698382+02
9	345051780961	1		2015-02-12 16:09:55.111999+02
7	345051783097	1		2015-02-12 16:09:59.12588+02
5	161343679352	1		2015-02-12 16:10:01.145779+02
6	161343570082	1		2015-02-12 16:10:01.382175+02
4	345051780666	1		2015-02-12 16:10:01.405667+02
3	161340745510	1		2015-02-12 16:10:02.940085+02

id	uid	code_id	info	last_activity
1	207376762666	1		2015-02-12 16:10:09.403215+02
2	345051770354	1		2015-02-12 16:10:09.97157+02

Table 4

There are 9 bots, each one identified through a cookie (**`uid`**). This cookie is used by the bot to register on the main server. If the server has a reference to that cookie, the process continues. The priority (**`code_id`**) is set to 1 for all the bots. The **`info`** table column is empty for all the entries.

- **`sho_cookie`** table contains information about the bots' activity (Table 6)

id	cookie	active	bot_id	last_activity	ip_addr	get_num
24	5PZLO3CDVTIN3VXCKTKEACIOC8960CDI	t	2	2015-02-12 16:09:05.202448	87.***.***.18	3
12	LZJMU3UUCWTR3B3MDQZUHHP12T9RF2Y1	t	3	2015-02-10 19:36:45.144918	5.***.***.223	3
22	5FIRR9LY5PAWAJE0ROJ0WJT2UE2XTYRD	t	5	2015-02-10 22:02:36.856878	5.***.***.160	3
26	VL1JCPSHSJGENP71CCLF9AOLUYXJN8IJ	t	8	2015-02-12 16:08:57.869532	192.*.**.215	3
25	6DFLVNT0E6GFQGIOWLL2BNE378KYF5OG	t	1	2015-02-12 16:08:59.608116	23.**.**.98	3
23	R814WUWHBFEJZP8K4RMFCUTP49U90IQO	t	6	2015-02-10 22:03:17.29154	5.***.***.150	3
7	345051770354	t	2	2015-06-06 07:02:03.173137	87.***.***.18	8285348
9	207376762666	t	1	2015-06-09 00:28:13.25646	23.**.**.98	7046287
20	9DPR3K6NI77WU48S1ECAH61YYUZE2QAJ	t	7	2015-02-12 16:09:00.011016	87.***.***.155	3
31	STGKT18882NZTA24YP9FCFNPE6EBCFBO	t	5	2015-04-18 05:36:10.383518	5.***.***.160	3
33	UB7M7VLR5I8S4V8H4CCKNEWG9JHD89QD	t	3	2015-04-18 05:53:33.533398	5.***.***.223	3
19	7JU8XX4IEKOIAYVBVM27IYS6G39ZINYO	t	9	2015-02-12 16:08:54.475175	5.**.***.36	3
35	UO2D3O5H91BMZZYNN7T8Q5ZA9BE0XIKM	t	1	2015-06-09 00:28:13.257237	23.**.**.98	3
21	S7JWM4N6LFBX18NFL21J15JYLWFBQYTZ	t	4	2015-02-12 16:09:05.202402	87.***.***.12	3

id	cookie	active	bot_id	last_activity	ip_addr	get_num
2	345051441193	t	8	2015-03-17 07:46:35.012289	192.*.**.215	3073574
32	CLQS319XK8UBN6AMNPZOG89J2YHNUWB4	t	6	2015-04-18 05:31:43.481397	5.***.***.150	3
8	161340745510	t	3	2015-04-18 05:53:33.532651	5.***.***.223	5445010
4	345051780666	t	4	2015-03-18 09:40:52.701857	87.***.***.12	2344761
3	345051783097	t	7	2015-02-14 15:26:34.879942	87.***.***.155	208586
30	W301ZX83BLGN0BAY6XU8QANKYOJYHKPT	t	4	2015-03-18 09:40:52.702595	87.***.***.12	3
5	161343679352	t	5	2015-04-18 05:36:10.382767	5.***.***.160	5651159
28	HPKFFT7ACLL89YLIXCYD9WFSZTLOZI3A	t	8	2015-03-17 07:46:35.013037	192.*.**.215	3
29	XWUHUJ3P6SEV02UN9O4C188MQGBTM2TL	t	7	2015-02-14 15:26:34.880691	87.***.***.155	3
27	7CH9P1PGXHPWH9J1ENU8XPAL27RC0GCL	t	9	2015-03-04 11:47:35.429737	5.**.***.36	3
34	7SEE8AJRY56OOCAHURYT14FSYIRJ69VX	t	2	2015-06-06 07:02:03.173882	87.***.***.18	3
6	161343570082	t	6	2015-04-18 05:31:43.480646	5.***.***.150	6785223
1	345051780961	t	9	2015-03-04 11:47:35.428945	5.**.***.36	1664623

Table 6

The **cookie** column represents the authentication code, **last_activity** is the last date when the bot has reached the server, **ip_addr** is IP of the bot, and **get_num** column represents the number of connections done by the bot with the central server.

This research also shows that the bots are duplicated and that there are two types of cookies: one using letters and numbers and one using only numbers. The activity of the first type is very poor (3 connection for each cookie), which leads us into believing that, over time, there were two versions of bots and plugins running in parallel. This can be explained also by the fact that the first type of cookie doesn't appear in the **sho_bot** table.

- **sho_ip** table contains the list of IPs generated by the **gen_ip.py** script

- **sho_plugin** table contains code, not data. It stores the script **plugin-nmap-0.43.py.** This script's main role is to fetch the targeted IPs (up to 16 IPs per request) and execute the scanning command (Figure 10)

```
########CALCULATE RESULT#########################################################################################

    if targets:

        out_name = targets_file +'.xml'
        print 'Scanning target', targets_file

        proc = subprocess.Popen('nmap -T5 \
                                 -p 21,22,23,25,80,110,143,443,465,993,995,8080,7071,3389,5900 \
                                 -sV -O --version-light --script=banner --script=http-headers \
                                 -oX ' + out_name + \
                                 ' -iL ' +targets_file,
                                 shell=True,stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
        out, err = proc.communicate()
        #print out, err
        print 'End scanning  target', targets_file
```

Figure 10

After parsing the nmap results, the data is sent back to the server and saved to the database in the **sho_ip** table.

- **sho_code** table contains code, not data. It stores the **scan-bot-03.py** script. This script describes the behavior of the system. The bot connects to the server and registers itself using a cookie. Once authenticated, the bot requests the last version of the code from the **sho_code** table (update mechanism) and the last version of the plugin (**sho_plugin**) that has to be executed. Once updated, it starts a list of threads that will use the latest plugin to scan the targeted IPs.

- **sho_log** table is empty

- **sho_setting** table is empty

A Django application is also running on the server. It serves as an administrative and reporting interface for the bots and targets. The figure below (figure 11) illustrates the **admin** panel. This interface allows a human operator to view statistics, add users, targets or bots or delete them.

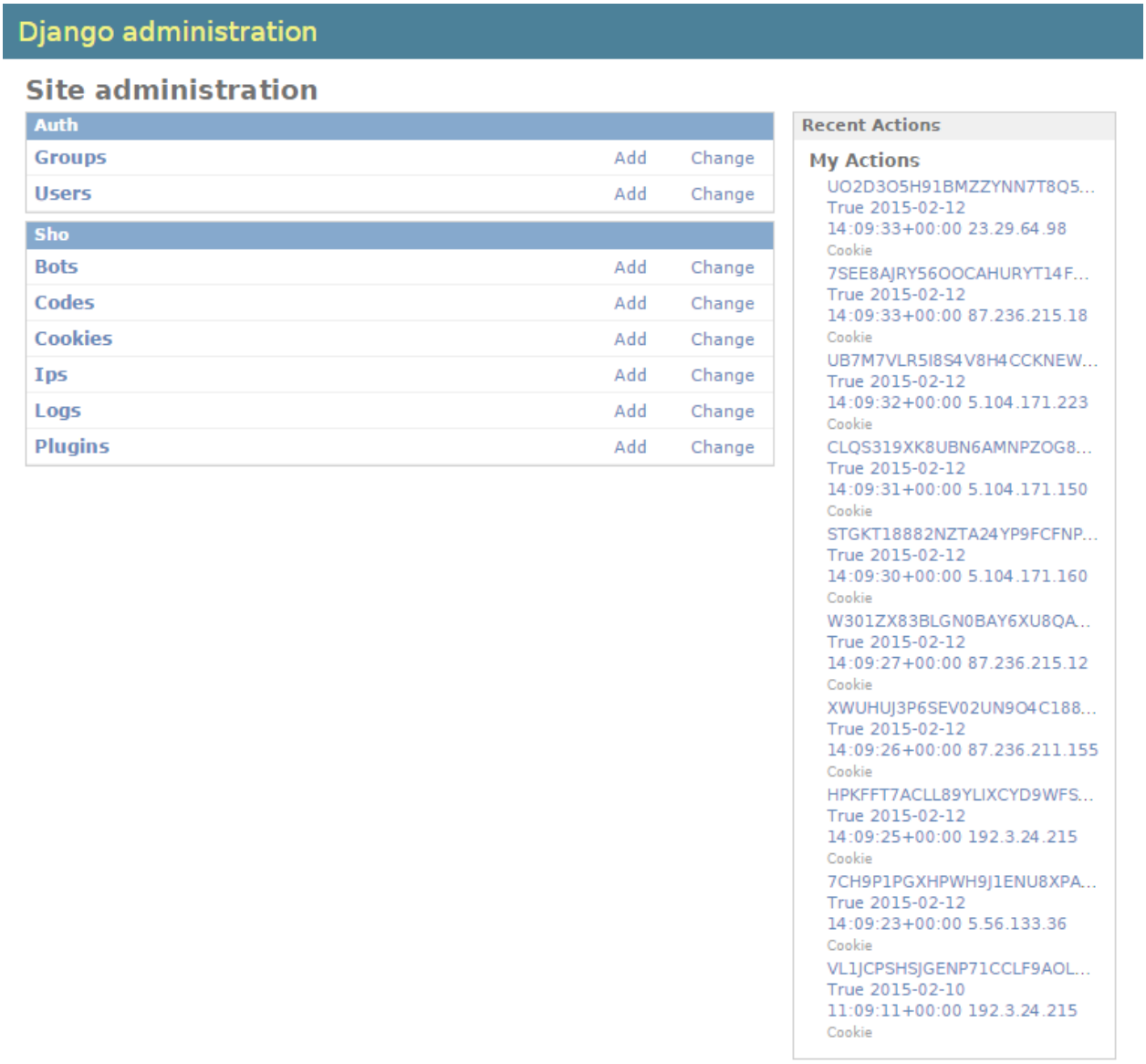

Django administration
Site administration
Auth
Groups Add Change
Users Add Change
Sho
Bots Add Change
Codes Add Change
Cookies Add Change
Ips Add Change
Logs Add Change
Plugins Add Change
Recent Actions
My Actions
UO2D3O5H91BMZZYNN7T8Q5...
True 2015-02-12
14:09:33+00:00 23.29.64.98
Cookie
7SEE8AJRY56OOCAHURYT14F...
True 2015-02-12
14:09:33+00:00 87.236.215.18
Cookie
UB7M7VLR5I8S4V8H4CCKNEW...
True 2015-02-12
14:09:32+00:00 5.104.171.223
Cookie
CLQS319XK8UBN6AMNPZOG8...
True 2015-02-12
14:09:31+00:00 5.104.171.150
Cookie
STGKT18882NZTA24YP9FCFNP...
True 2015-02-12
14:09:30+00:00 5.104.171.160
Cookie
W301ZX83BLGN0BAY6XU8QA...
True 2015-02-12
14:09:27+00:00 87.236.215.12
Cookie
XWUHUJ3P6SEV02UN9O4C188...
True 2015-02-12
14:09:26+00:00 87.236.211.155
Cookie
HPKFFT7ACLL89YLIXCYD9WFS...
True 2015-02-12
14:09:25+00:00 192.3.24.215
Cookie
7CH9P1PGXHPWH9J1ENU8XPA...
True 2015-02-12
14:09:23+00:00 5.56.133.36
Cookie
VL1JCPSHSJGENP71CCLF9AOL...
True 2015-02-10
11:09:11+00:00 192.3.24.215
Cookie

Figure 11

The interface is very simple, representing just a wrapper over the database (figure 12: add user (up) add targets (below)):

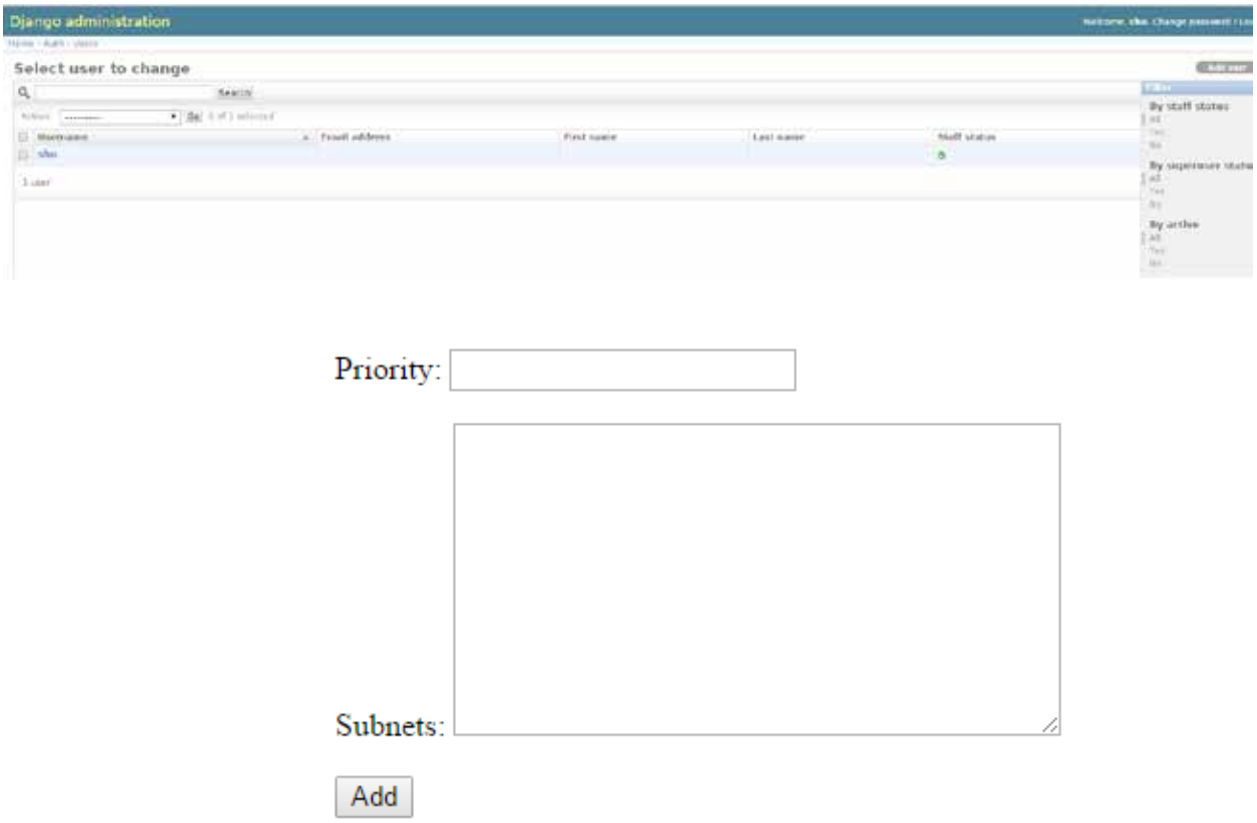


Figure 12

1. *The bots*

The machines used to scan the potential victims are in different networks. This helps the attackers conceal that scans originating from these IPs are part of the same operation. The table below shows the IPs of the bots found in the database at the time of writing (Table 7):

BOT IP	Country
192.*.**.215	US
5.**.***.36	US
87.***.***.155	UK
5.***.***.160	Bulgaria
5.***.***.150	Bulgaria
87.***.***.12	UK
5.***.***.223	Bulgaria
23.**.**.98	US
87.***.***.18	UK

Table 7

Appendix 3 - APT28 related tools

Proxy tool (svehost.exe)

This executable file can be either used with arguments passed via the command line or without. When executed without arguments, it attempts to contact the IP address: 176.**.***.10 on port 443.

It can also be started with the following command line:

svehost.exe start <ipaddress> <port>

The file uses an old version of the OpenSSL library (OpenSSL 1.0.1e). In fact, the reason this file is 1038 kilobytes in size is because of the inclusion of this library.

The main purpose of the tool is to allow an attacker to contact systems behind a router, which otherwise would not be accessible from outside the network.

Privilege escalation tool (xp.exe)

The tool is built around a vulnerability discovered in 2014 (CVE-2014-4076) and works by sending a specially crafted package to the \\.\\ TCP device through the use of DeviceIoControl function. The vulnerability was fixed in late 2014.

The tool works by receiving one executable file as argument. The tool will then run this executable file with system privileges.

The file is compiled using the debug configuration. Because of this, a path to a pdb file is also hardcoded into the file. The path references C:\Users\Пользователь\Desktop\cve-2014-4076\cve-19abdba\Debug\CVE-2014-4076.pdb

The string Пользователь means users in Russian. This is one of the reasons we suspect the authors are Russian speakers.

Password dumping tool (run.exe)

This seems to be based on the source of a public tool (**`mimikatz`**) for dumping passwords out of WDigest through LSASS. More information about the tool can be found by visiting the author's page: http://blog.gentilkiwi.com/mimikatz.

The file was compiled on 05/05/2013 and does not contain any version info. This suggests the authors used a custom build.

The tool receives a file as argument. The retrieved password will then be dumped into the file passed as argument.

Appendix 4 (First stage component)

This is the first component that gets on the computer after infection. Its purpose is to contact the C&C server and ask for further instructions.

The first stage payload is made of two items: a dropper and a first stage backdoor. The dropper, which in our case is named **runrun.exe,** has embedded the file **api-ms-win-downlevel-profile-l1-1-0.dll** in the data section. The dropper uses many custom encryption algorithms to protect itself from reverse-engineering.

The following algorithm (Algorithm 1) is used to decrypt its APIs:

Algorithm1:

```
b[idxbuff+1] = b[idxbuff+1] ^ b[idxbuff] ^ key[(idxbuff+1)%keylen]
```

The payload (**api-ms-win-downlevel-profile-l1-1-0.dll**) is encrypted using RTL and a custom encryption algorithm that uses a 10 byte key (Algorithm 2):

Algorithm2:

```
(unsigned char encrypt1(int pos

}

;xor = 0

(++for(int i=0;i<key_length;i

;(xor ^= key[i] + (unsigned char)(i*pos

return xorl

{
```

The dropped file is a downloader that contacts the C&C to get the second stage component. It contacts the domain: www.msc****vw.com and the IP address: 91.***.**.249.

The requests are performed via GET over HTTP. They are randomly generated. Each request has 1 to 5 groups of 1 to 6 randomly generated chars and numbers. Each group is separated by a slash. An example of a request is given below:

```
/ue/VHghm/ihXAIK/qpi/1c9.xml/?XK1=VrLYQndXGXwzURh9RBE=
```

The xml extension present above is actually picked from 4 available extensions (**xml, pdf, html, zip**).

The final argument is an encrypted key, probably used by the downloader to authenticate itself to the server.

This file works as a rudimentary backdoor that has the following commands available:

- Download
- Execute
- Delete files

Since it took more than an hour for the second stage component to be downloaded to the infected system, we presume this is downloaded manually.

Appendix 5 (Second Stage Component)

The purpose of the second stage component is to open a backdoor to allow the attacker to assess the target and download additional components.

a. Dropper:

As the First Stage Component, the important file is being installed using a dropper. (**winloat.exe**)

The main steps followed by the dropper are:

1. Decrypt its overlay (containing the backdoor: **advstoreshell.dll** and the configuration file: **msd**)
2. Write payload on disk
3. Call the InitW method of **advstoreshell.dll**

b. Configuration File:

To encrypt/decrypt the configuration file and the APIs used by **advstoreshell.dll** the component uses a custom stream cypher with key length of 6. The function given below is used to retrieve the corresponding byte used to XOR at an index in the buffer.

Algorithm 3:

```
unsigned char decrypt(unsigned char key[6], uint32& buffer_pos)
{
            ixk =  buffer_pos % 6 +2;
            ixb = (unsigned char)(ixk);
            buffer_pos++;

            v_10 = key[(ixk -2)%6]^byte((uint32(ixb)*uint32(key[ixk-2])) >> 7)
            v_11 = key[(ixk+1)%6] ^ ixb;
            xc = (key [ixk%6] & v_11) + v10;
            return xc
}
```

The configuration file (**msd**) file is encrypted using the algorithm described above and is 122 bytes long. It contains the following information:

Timeout1	Timeout2	Computer Name	Two domain names	Campaign id	?	?	Keylogging flag	Main domain name

Where:

- Timeout1 = 60000 ; represents the number of miliseconds until timeout is reached when contacting the C&C
- Timeout2 = 60000 ; represents the number of miliseconds until timeout is reached when using contacting the C&C using encryption
- Computer name = DAE13WS01204030501 ; The name of the infected computer
- DomainName1= micro**********ter.info
- DomainName2=dri********te.info
- Campaign id = rhbp ; we are still investigating the meaning of this field, but it is very likely a campaign identifier
- ? = 1000 ; we are still investigating the role of this field, although preliminary information leads us to believe that it is used as a counter
- ? = 600000; we are still investigating the meaning of this field, but it is very likely another timeout value
- Keylogging flag = 1; this indicates whether the backdoor should intercept keystrokes and send them to the C&C server.
- Main domain name = 1***.net ; this is the first domain that the malware tries to connect

c. Backdoor Component:

The most important component of APT28 is the backdoor, in this case, the file called **advstoreshell.dll**. This file reads the configuration file (**msd**) and contacts the C&C servers indicated in the configuration file. It can also read the configuration file from the registry. The configuration stored in the registry is probably obtained via update from one of the C&C servers.

On the system we have analyzed, there were two configurations:

1. The first one being the **msd** file that was located in the same path with **advstoreshell.dll**

2. The second one was found in registry (**HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Path**). It contains almost the same data as the **msd** file except for the first two domain names that are changed to win[redacted]ore.net and micr[redacted]er.com.

This was the only component that was set to run at startup. This was achieved by creating the value **AdvancedStorageShell** in **HKCU\Software\Microsoft\Windows\CurrentVersion\Run** with the contents:

"RUNDLL32.EXE "C:\Documents and Settings\<user-name>\Local Settings\Application Data\Microsoft Help\advstorshell.dll", InitW"

The backdoor is highly advanced and supports 26 commands. We analyzed some of the supported commands, while the analysis for the rest is still in progress.

5	find file
6	get file content and timestamps
9	enumerate registry keys and values
10	set registry value
11	delete registry value
13	start process using CreateProcess
14	kill process by id
16	start process using cmd /c
17	list drives and types
19	read file from offset
21	get system info (run systeminfo in cmd)
25	create thread

When running a command using the cmd /c option, the output is saved in a file named tmp.dat

Communication between this component and the C&C server is carried out on port HTTP (443). If this is not possible, it attempts to connect on port 80. In some cases, the communication is encrypted using 3DES and RSA.

One important function of this backdoor is that it can load external components. It can achieve this goal by calling the **`init`** function of any **`dll`**. This is how the **`s`** plugin is loaded.

Appendix 6 - Additional module

The final component installed by APT28 is a module named **pr.dll**. This file was compiled 5 hours after the system was compromised. This discrepancy in time prompts us that the file is individually crafted for each target system.

The file seems to be a modular framework that can accommodate different modules. It stores its configuration in an encrypted format, and saves it in the Registry key **"HKU\S-1-5-19_Classes\Software\Microsoft\MediaPlayer\{some_clsid}\chnnl"** .

Once executed, the malware checks for the presence of a mutex (XSQWERSystemCriticalSection_for_1232321) to make sure that another instance of itself is not already running. After that, it contacts the server (the IP: 198.[redacted].74 or the domain name sec[redacted]win.com). During the initial communication, the backdoor sends an encrypted buffer to the C&C server that contains a RC4 key and the IDs of the plugins it has installed. The buffer is shown below:

03 33 02 03 33 23 03 21 23 06 21 23 3A 66 95 64 4B 56 FF 22 4D 81 4F

We can identify plugins with the following ids from the buffer above: 0x3302, 0x3303, 0x2103, 0x2106

An interesting aspect of the **pr.dll** is that it creates two pipes that are presumably used to collect data sent by other processes.

The **pr.dll** seems to be a Windows version of a Linux file found on the server. The server hosts 4 elf files, two built for the x86 architecture (**075b6695ab63f36af65f7ffd45cccd39**, **f3bf929a35c3f198226b88537d9ccb1a**) and two for the x64 architecture (**2683624eacc490238e98c449bddbb573**, **5bf524a4860f3c33e3ad77b6b625db37**). The elf file uses SQLite3 rather than Registry to store its configuration. The database named My_BD holds two values chnnl and prms.

The files attempt to contact the server by constructing random requests. Each request is made of one of the verbs (watch/, search/, results/, search/, close) and two to nine arguments (selected from the list: text=, from=, itwm=, ags=, oe=, aq=, btnG=, oprnd=, utm=, channel=). The values of the arguments are randomly generated.

An example of a request is given below:

h[tt]p://198..***.74/watch/?aq=JTfJRp-s&oprnd=Dwtee&itwm=niKMuGE9Mp9Md9vHdggZMS16YlSTx&btnG=t&oprnd=FbLtw&AVVAT=m8I2tN***

The elf files contain the following modules:
1. AgentModule
2. KernelProvider
3. AgentKernel
4. ChannelController
5. Cryptor
6. LocalStorage (sqlite3)
7. ReserverApi
8. AgentChannel
9. HttpChannel
10. FSModule
11. RemoteShell
12. RemoteKeylogger

As stated above, there are two versions for each architecture. The only difference between them is that one version does not include the last two modules.

About the Authors

Răzvan Benchea has been working as a malware researcher for more than eight years. He is an associate professor at the Alexandru Ioan Cuza university of Iaşi where he teaches operating systems and assembly language. Răzvan's areas of expertise includes APT forensics and reverse engineering. His academic accomplishments include a PhD in machine learning algorithms for malware detection.

Alexandru Maximciuc spent the past 10 years doing antimalware research at Bitdefender. He is an expert in reverse engineering and cryptanalysis. Alexandru has played a key role in the analysis and mapping of the PushDO botnet and is in charge with the automation of sinkholing operations.

Cristina Vatamanu is a malware researcher with a focus on botnets. With more than six years of experience in reverse engineering, exploit analysis and cryptography, Cristina has brought her contribution to the threat intelligence scene at conferences such as AVAR and Virus Bulletin.

Victor Luncasu is a junior malware researcher focused on algorithms and data structures. His experience includes reverse engineering and payload analysis as well as custom communication protocols between bots and command & control servers.

Bitdefender is a global security technology company that delivers solutions in more than 100 countries through a network of value-added alliances, distributors and reseller partners. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a leading security provider in virtualization and cloud technologies. Through R&D, alliances and partnership teams, Bitdefender has elevated the highest standards of security excellence in both its number-one-ranked technology and its strategic alliances with the world's leading virtualization and cloud technology providers. More information is available at http://www.bitdefender.com/

FOR MORE INFORMATION VISIT: enterprise.bitdefender.com

BD-Business-Dec.17.2015-TK#: 70585