Design an Advanced Botnet to Monitor User Awareness on Harmful Malware Using VertexNet

Albert Sagala and Alexander Lumbantobing

***Abstract*—Malware creates a serious problem to the infected computer. People believe that installing antivirus can mitigate the computer from being infected. In fact, malware designer actively creates botnet which is concealed from the antivirus. Most people do not aware of being infected, that causes their PC harmful and can be controlled by C&C server. In this paper, we develop a botnet based on VertexNet bot loader, as a result it can be used as a tool to map the awareness of people from being infected.**

***Index Terms*—Botnet,Bot Loader,Malware,VertexNet.**

I. Introduction

A botnet is a network of compromised machines under the control of an attacker. The most prevalent peer-to-peer botnet in 2009 was Waledac Botnet [1]. The purpose of the actual botnet is to control and to monitor a large number of computers, so that it can steal the critical information or use the infected computers for illegal activity. The command is given by the botmaster to the bot through command and control (C&C), it performs various tasks, such as sending e-mail spam, adware, spyware, collecting up confidential information such as passwords or encryption keys, performing Distributed Denial of Service (DDoS) or just searching for further potential targets to be enrolled in the botnet [2].

Many people have been involved for crimes related to botnet usage. In recent years, the threat posed by botnets trends toward Internet applications and communications has been escalated. This is due to the fact that a botmaster controls a large number of bots that ranges from hundreds of thousands to millions. From a defender’s perspective, it is very important to understand the trends and practices of botnets.

Botmaster and bot operate on different communication protocols using various topologies: centralized, distributed hybrid, or randomized. According to their C&C architecture, botnets can be classified as IRC-based, HTTP-based, DNS-based, or Peer to Peer (P2P).

There are several approaches to study the botnet phenomenon: analyze its source code, study its control (particularly the activity of its C&C server(s)) and study its behavior by allowing a selected machine to become infected by an executable bot and analyze all possible scanning activities/actions triggered by the botnet [3].

In this paper, it is developed a botnet that can not be easily detected by the antivirus, so that we may use the botnet for user investigation on the awareness of the malware existence. The rest of the paper is organized as follow: Section 2 presents some of the most relevant related work on basic botnet characterization; Section 3 describes the botnet testing environment and botnet development that was used in this research; Section 4 presents result and discussion. Finally, Section 5 gives conclusions.

II. Related Work

Because botnets constitute a serious security problem, a lot of efforts have been invested towards understanding how they work, and to deploy effective counter-measures against them [4]. Every botnet has a basic property, like C&C structure (e.g., centralized, P2P, hybrid) and communication protocol (e.g., IRC, HTTP, FTP, DNS). Botnets’ traffic is different from the traffic of other types of malware, because the botnets’ traffic includes C&C communication channels traffic [5]. Communication protocol is a system rules that allow botnets and botmaster to communicate between them to transmit information [6].

Most current botnets have got centralized C&C architecture, enabling botmaster can control all bots easily, but if the C&C is down, all bots will also be down. Nowadays, peer-to-peer (P2P) structured botnets have gradually emerged as a new advanced form of botnets. P2P botnets are more resilient to defenses and countermeasures than traditional centralized botnets. In P2P, when C&C is down, botmaster can quickly transform a regular bot into C&C server, to take control of all bots. In other words, in P2P a regular bot can be transformed to a botmaster. As shown in Fig. 1, we may find the structure difference between centralized and P2P architecture. Every member of the P2P network can act as a server and/or client. The advantage of P2P structure is the disappearance of the central C&C as a single-point-of-failure, so when a C&C server is down, a bot can be assigned to be C&C server quickly, and therefore why this kind of botnet is a lot harder to takedown.

Botnets have been the subject of research for some years to reveal study and mitigate its evolving. To harden their infrastructure for delivering commands, the botmaster has begun to employ several different techniques, including the use of encryption and new protocols for communication [7].

A. Sagala and A.Lumbantobing are with the Faculty of Informatics and Electrical, Del Institute of Technology, Toba Samosir, Indonesia (phone: +62632331234; e-mail:albert@del.ac.id,alex.lumbantobing@del.ac.id).

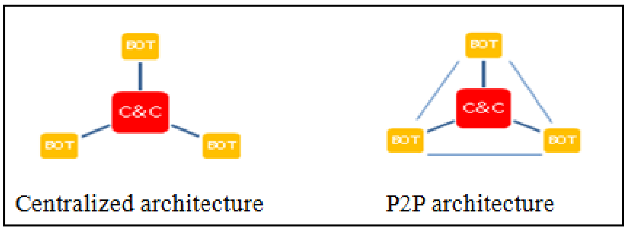


Fig. 1 Centralized Botnet Architecture vs. P2P Botnet Architecture

Recently, botmaster has developed an advanced peer-to-peer botnet, called Hybrid P2P Botnet to make their botnet more robust [8]. The Waledac creators routinely take advantage of various real-world events and occasions, using them as social engineering plays to trick users into performing certain actions [9]. In [10], botnets usually use a unique encryption system to communicate with each other to prevent from being detected using an exchange channel. Some types of botnet use repeater to act as proxy to increase the anonymous botnet's traffic, and make it to more difficult to analyze.

III. Testing Environment and Botnet development

To develop the botnet, the first thing to be discussed is what the C&C structure and communication protocol will be used. This is the first step to study more closely about botnet, and the researchers choose to deploy a simple botnet that can simulate how the connection between botnet and botmaster established. We choose to deploy a botnet that has traditional (centralized) C&C structure and use HTTP-based communication protocol. In our laboratory, there are three points that will be focused to develop the botnet: the bot loader, victim-botmaster topology, and bot spreading method. Bot loader is used to create a complete simple botnet quickly. A test topology must be described to make sure that the scenario is close enough to the real world case. A best spreading method must be chosen to ensure the botnet is infected widely. In short, the description of the points is shown in Figure 2.

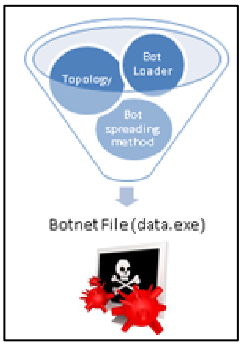


Figure 2 Testing Environment and Botnet Development

A. *Bot loader*

To create a botnet, we can complete building botnet using programming language (like C or Visual Basic) or using bot loader that can generate botnet in an easy way. We choose to create botnet using bot loader. There are many bot loaders tested in advance to create a simple botnet. For this research, we test ten (10) bot loaders: (1) Rebel Botnet, (2) uBot Botnet, (3) VertexNet Botnet, (4) Umbra Loader, (5) AryaN Botnet, (6) Open Project Botnet, (7) Di PHP Botnet, (8) Cythisia Botnet, (9) N0ise Botnet, and (10) vOlk Botnet. Shortly, it is tested the function of all loaders. Table I shows the comparison between the bot loaders.

Based on Table I, we can say that VertexNet is the best loader in the list; all basic loader functions are supported. The "*support*" in information's table means that loaders has the feature and works well at our test lab.

TABLE I.

Bot Loaders Benchmark Information

Bot Loaders	Rebel	U Bot	Vertex Net	Umbra	AryaN	Open Project	Di PHP	Cythisia	N0ise	vOlk
User Friendly Interface	v	x	v	v	v	x	v	v	v	x
Easy to Use	v	x	v	v	v	x	v	v	v	x
Startup	x	v	v	x	x	x	x	x	x	v
Upload	v	v	v	v	v	v	v	v	v	v
Download	v	v	v	v	v	v	v	v	v	v
DDoS	v	x	v	x	x	x	x	v	v	x

Legend:

v	= Supported
x	= Not supported

VertexNet Botnet is chosen because the interface of the control panel is simple and the feature is powerful enough to compete with other premium bot loaders. VertexNet is a HTTP botnet coded in 2011 by DarkCoderSc from France (also creator of the DarkComet RAT). This is a bot loader with great features and it is user friendly, easy to setup and use, and perfect for beginners.

There are many features of VertexNet Botnet Loader [11]. In this paper, to ensure the communication is established between botnet and botmaster, we use the simple command in VertexNet loader. That command is Message Box. Message Box will send a simple message box interface to the victim computer. To use this function, botmaster must set value for parameters.

This is the syntax to use Message Box command.
Syntax command: **msg::*@Param1,@Param2,@Param3***

Explanation:

```
@Param1 - String message title: The title for
messageboxinterface.
@Param2 - String message core: The text for
messagebox interface.
@Param3 - String message icon
{NONE,WARN,ERROR,INFO,QUEST}: The icon of messagebox
interface.
```

The example for the condition where we can use this command is such as **msg: CSRC IT Del,Helloworld,INFO.**

That command will send message box interface to victim, with title message (CSRC IT Del), textmessage (Hello world), and the icon message is information interface (INFO).

B. Victim-botmaster topology

After choosing the loader to deploy the botnet, the next task is to choose the test topology. A test topology must be described to make sure that the scenario is close enough to the real world case. For this purpose, we create a topology that consists of two different networks to demonstrate cyber war using botnets attacks.

Botmaster Network is a network environment where a botmaster will create, control, and monitor each botnet that has been spread. Through Command and Control Server (C&C server), botmaster will create a botnet that will be deployed to infect the victim's computer. Through repeater, botmaster will try to hide the identity of the C&C server and also acts as a medium for the botnet to perform the self-update. All botnet will be distributed globally with several techniques, such as mail spam. In Fig. 3, we can see the closer scenario to demonstrate the real world case botnet infection.

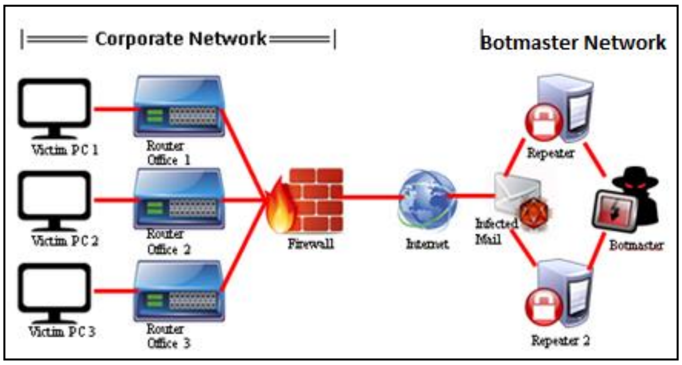


Fig. 3 Simulation Topology

Corporate Network is a network environment of a company that is being targeted by a botmaster. The network is connected directly to the Internet using a firewall, as network security protection. LAN on the company is divided into three subnets: Office 1 Subnet, Office 2 Subnet, and Office 3 Subnet.

C. Bot Spreading Methods

There are many methods that can be used to distribute botnets. In our work, it will spread botnets using email spam technique. Spamming is the use of electronic messaging systems to send unsolicited messages (spam), especially advertising, as well as sending messages repeatedly on the same site. Seeing huge potential in botnets, botmaster usually spreads their bots using spamming.

Fig. 4 illustrates how a botnet is created and used to send email spam. First, botnet operator (botmaster) attaches the bot to mail, and sends to many of mailing list addresses. Second, unaware users read the email and open the attachment. Then, the victim’s computer has been infected by the botnet. Third, botmaster sends a command to the victim’s computer, and then the botnet performs that command and sends the result to the C&C Server, and requests any task to perform again.

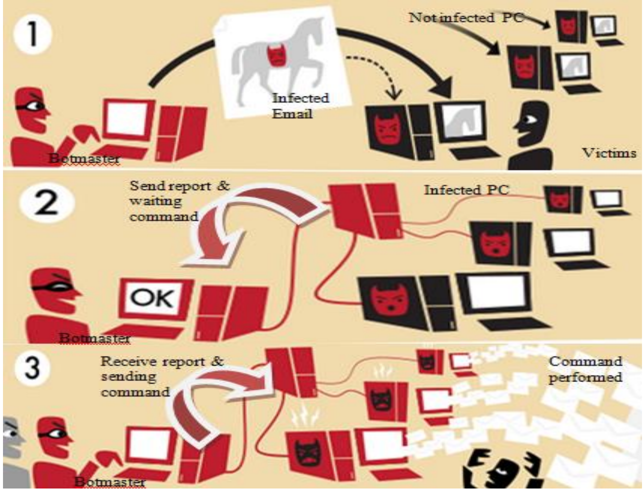


Fig. 4 Botnet Spam Email Attack

IV. RESULT AND DISCUSSION

A. ConnectionTesting

To be able to test whether the topology design successfully connects the victim with the botmaster using a botnet, it is necessary to test the connection in small environment. At this scenario, there will be a victim and a C&C server, using the different environment.

In our work, as shown in TABLE II and TABLE III, there are five (5) IP addresses that will be used by a C&C server to try to communicate with a victim, and four (4) IP addresses that will be used by a victim to try to communicate with a C&C server.

TABLE II. C&C SERVER INFORMATION

Network Connection	C&C Server Name & IP Address	
	C&C Server Name	***C&C Server IP address***
pamer2in.com/internet	C&C Server 0	174.120.70.155
wifi-id	C&C Server 1	192.168.2.33
internet tri - 3	C&C Server 2	192.168.43.101
Campus	C&C Server 3	172.21.41.248
Lab	C&C Server 4	192.168.10.102

TABLE III. PC VICTIM INFORMATION

Network Connection	Victim Name & IP Address	
	Victim Name	***Victim IP address***
pamer2in.com/internet	-	-
wifi-id	Victim 1	192.168.2.94
internet tri - 3	Victim 2	192.168.43.123
Campus	Victim 3	172.21.41.142
Lab	Victim 4	192.168.10.99

The Logical Network Topology is shown on Fig. 5. This is the existing topology that the researchers use in our experiment to test the spread of our botnet.

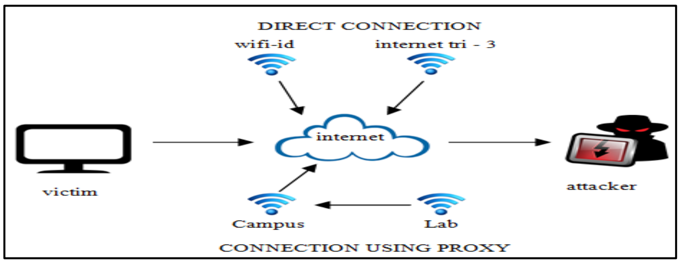


Fig. 5 Logical Network Topology Experiment

In Fig. 5, it is described as the followings:

1. Network “Lab” is a private NAT subnet from “Campus” network, which must use proxy to connect to the Internet. Network “Lab” cannot be accessed from “Campus” network.
2. Network “Campus” is a public Local Area Network, which must use proxy to connect to the Internet. Network “Campus” can be accessed from “Lab” network.
3. Network “internet tri - 3” and “wifi-id” are public NAT subnets, and have direct access to connect to the Internet. Neither “internet tri - 3” nor “wifi-id” network, can access each other.
4. Network “pamer2in.com” is an IP address of a public server, and has direct access to connect to the Internet. All connections that use direct access to connect to the Internet, can also access this IP address.

To test the connection in a small environment, we simplify the topology as shown in Fig. 6 and Fig. 7. In this topology, the victim was connected directly to the botmaster using network available in Table II. We may assume that the victim’s PC has been already infected because the victim opens the infected attachment from an unknown mail and C&C server is using the IP address available in Table III. Connection will be established when C&C server and victim can “ping” each other.

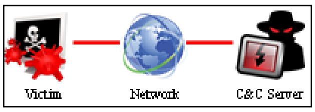


Fig. 6 Small Environment Connection Testing Topology

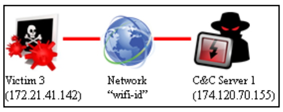


Fig. 7 Example of Small Environment

Based on the results of all connection testing scenarios between a C&C server and a victim in small environment, we obtained the data as shown in TABLE IV:

TABLE IV. CONNECTION TESTING SCENARIOS RESULT

	Victim 1	Victim 2	Victim 3	Victim 4
C&C Server 0	v	v	x	x
C&C Server 1	v	x	x	x
C&C Server 2	x	v	x	x
C&C Server 3	x	x	v	v
C&C Server 4	x	x	x	v

Legend:

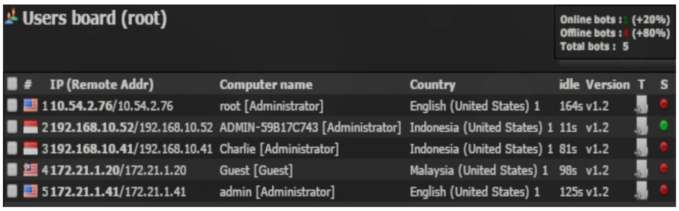
v = Connection established

x = Connection not established

Conclusions that can be obtained based on Table IV are:

1. If C&C server is on the Internet, and the victim has an Internet access, it will establish communication.
2. If C&C server or/and victim are on the intranet (local area network), communication may be established.
3. If C&C server is on the Internet, and the victim has no direct Internet access, it will never be established the communication.

Fig. 8 is the screenshot of the C&C Server interface hosted in public IP address (the Internet) using our testing domain (**pamer2in.com**). In the picture, we can see there are five (5) infected victims in the list, but only one is online.


Users board (root)

Online bots : 1 (+20%)
Offline bots : 4 (+80%)
Total bots : 5

#	IP (Remote Addr)	Computer name	Country	idle	Version	T	S
1	10.54.2.76/10.54.2.76	root [Administrator]	English (United States) 1	164s	v1.2		
2	192.168.10.52/192.168.10.52	ADMIN-59B17C743 [Administrator]	Indonesia (United States) 1	11s	v1.2		
3	192.168.10.41/192.168.10.41	Charlie [Administrator]	Indonesia (United States) 1	81s	v1.2		
4	172.21.1.20/172.21.1.20	Guest [Guest]	Malaysia (United States) 1	98s	v1.2		
5	172.21.1.41/172.21.1.41	admin [Administrator]	English (United States) 1	125s	v1.2		

Fig. 8 C&C Server on the Internet

B. *Connection Testing Simple Analysis*

In this session, it will studed botnet traffic in a small isolated local area network, using our lab network, victim (192.168.10.52) and C&C server (192.168.10.101), as displayed in Fig. 9. To determine and verify whether a connection has been established, we have to analyze the data traffic to be able to find evidence that the connection has successfully been sent between bot and botmaster.

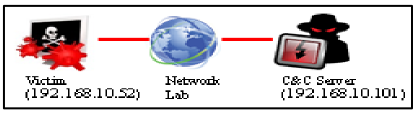
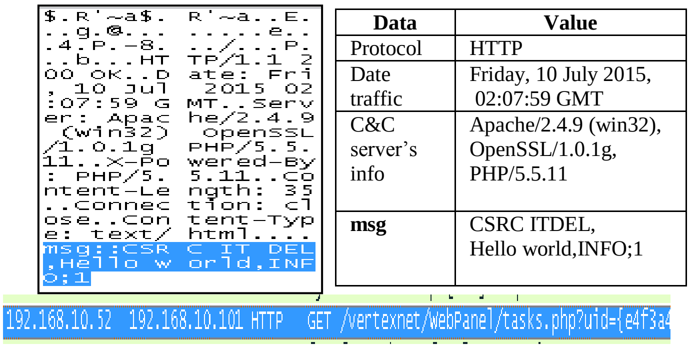


Fig. 9 Connection Testing Simple Analysis Environment

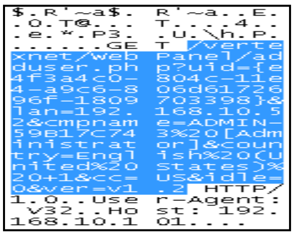

```
$.R'~a$.  R'~a..E.
..g.@...  ......e..
.4.P.-8.  ../...P.
..b...HT  TP/1.1 2
00 OK..D  ate: Fri
, 10 Jul   2015 02
:07:59 G  MT..Serv
er: Apac  he/2.4.9
 (win32)   OpenSSL
/1.0.1g   PHP/5.5.
11..X-Po  wered-By
: PHP/5.  5.11..Co
ntent-Le  ngth: 35
..Connec  tion: cl
ose..Con  tent-Typ
e: text/  html....
msg::CSR  C IT DEL
,Hello w  orld,INF
O;1
```

Data	Value
Protocol	HTTP
Date traffic	Friday, 10 July 2015, 02:07:59 GMT
C&C server’s info	Apache/2.4.9 (win32), OpenSSL/1.0.1g, PHP/5.5.11
msg	CSRC ITDEL, Hello world,INFO;1

```
192.168.10.52  192.168.10.101 HTTP   GET /vertexnet/WebPanel/tasks.php?uid={e4f3a4
```

Fig. 10 Bot Reporting and Request Command

As we can see in Fig.10, using Wireshark analytic, we observe the botnet report and request task to C&C server. Every botnet uses a “special ID” that makes botmaster can easily identify every victim. Then, the bot register itself to the list bot, using data like in Fig. 11. The information registered by bot included: a special ID, an IP address, a computer name, a user name, and a country location.

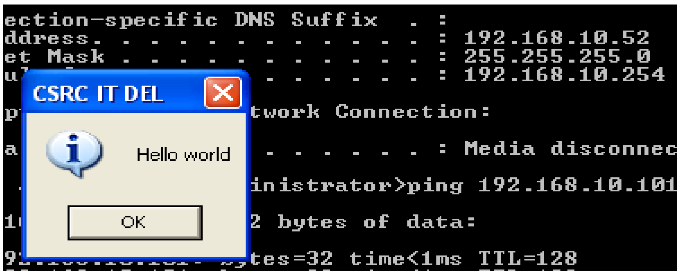


```
$.R'~a$.  R'~a..E.
.O.T@...  T.....4..
.e.*.P3.  .U.\h.P.
.......GE T /verte
xnet/web  Panel/ad
duser.ph  p?uid={e
4f3a4c0-  804c-11e
4-a9c6-8  06d61726
96f-1809  703398}&
lan=192.  168.10.5
2&cmpnam  e=ADMIN-
59B17C74  3%20[Adm
inistrat  or]&coun
try=Engl  ish%20(U
nited%20  States)%
20+1&cc=  US&idle=
0&ver=v1  .2 HTTP/
1.0..Use  r-Agent:
 V32..Ho  st: 192.
168.10.1  01....
```

Data	Value
uid	e4f3a4c0-804c-11e4-a9c6-806d6172696f-1809703398
lan	192.168.10.52
cmpname	ADMIN-59B17C743%20
username	Administrator
country	English%20 (United%20States)

Fig. 11 Victim Information

Once the bot has registered and been online, botmaster can send any commands to botnet from the features list of VertexNet Loader to perform. The simple command in VertexNet loader is Message Box. This is the appearance of message box interface in the victim's computer (Figure 12).



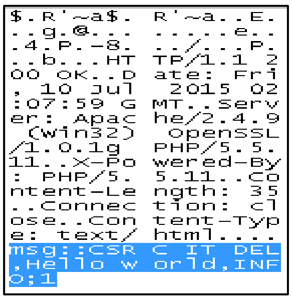
```
ection-specific DNS Suffix  . :
ddress. . . . . . . . . . . . : 192.168.10.52
et Mask . . . . . . . . . . . : 255.255.255.0
u                             : 192.168.10.254

p                 twork Connection:

a                 . . . . . . : Media disconnec

                  inistrator>ping 192.168.10.101

1                 2 bytes of data:

9                 ytes=32 time<1ms TTL=128
```

CSRC IT DEL

Hello world

Fig. 12 Message Box Command Appearance

Command was sent by botmaster using HTTP protocol on port 80. Fig. 13 is the traffic sniffing by Wireshark; we found that the C&C server (192.168.10.101) sent an HTTP protocol traffic to victim's PC (192.168.10.52). This command was performed by botnet because C&C server sent the traffic that will execute the command link file at C&C server. When the botnet receives the command link, it will execute automatically and report the result to botmaster. Request and response actions were performed in HTTP protocol in background mode.



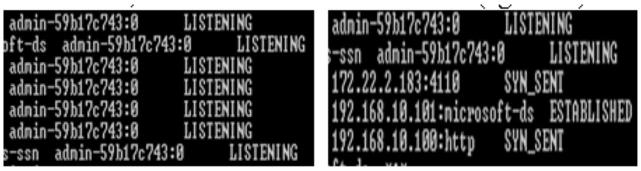
```
$.R'~a$.  R'~a..E.
..g.@...  ......e..
.4.P.-8.  ../...P.
..b...HT  TP/1.1 2
00 OK..D  ate: Fri
, 10 Jul   2015 02
:07:59 G  MT..Serv
er: Apac  he/2.4.9
 (Win32)   OpenSSL
/1.0.1g   PHP/5.5.
11..X-Po  wered-By
: PHP/5.  5.11..Co
ntent-Le  ngth: 35
..Connec  tion: cl
ose..Con  tent-Typ
e: text/  html....
msg::CSR  C IT DEL
,Hello w  orld,INF
O;1
```

Data	Value
Protocol	HTTP
Date traffic	Friday, 10 July 2015, 02:07:59 GMT
C&C server's info	Apache/2.4.9 (win32), OpenSSL/1.0.1g, PHP/5.5.11
msg	CSRC ITDEL, Hello world,INFO;1

```
192.168.10.101 192.168.10.52 HTTP   HTTP/1.1 200 OK  (text/html)
```

Fig. 13 Message Box Command Traffic

Another method to ensure that the connection is successfully established between the bot and botmaster is using Netstat tool. Netstat is Windows tool which display protocol statistics and current TCP/IP network connections. When we use command **netstat –a**, it will display all connections and listening ports on local computer. Thus, we compare the result of protocol statistics before and after being infected. Fig. 14 shows the difference before and after the botnet infected victim's PC. Before being infected, there is no connection to 192.168.10.101, but after got infected, there is an established connection.



```
 admin-59b17c743:0      LISTENING
oft-ds  admin-59b17c743:0      LISTENING
 admin-59b17c743:0      LISTENING
 admin-59b17c743:0      LISTENING
 admin-59b17c743:0      LISTENING
 admin-59b17c743:0      LISTENING
s-ssn  admin-59b17c743:0      LISTENING
```

Before infection

```
admin-59b17c743:0      LISTENING
s-ssn  admin-59b17c743:0      LISTENING
172.22.2.183:4110      SYN_SENT
192.168.10.101:microsoft-ds  ESTABLISHED
192.168.10.100:http    SYN_SENT
```

After infection

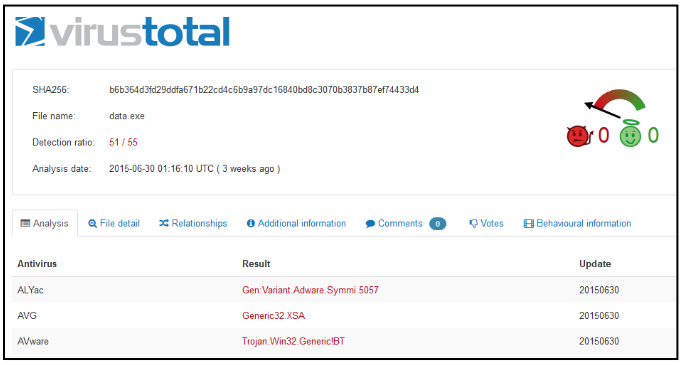
Fig. 14 Before and After Infection

C. Antivirus Detection Testing

Before we spread the botnet globally to test and survey on a company's security concerns, the botnet must tested with antivirus detection to ensure the survivability of the botnet. To fulfill this objective, we design an advanced botnet so that antivirus will not detect the file as a malware. In our work, we use online tools such as Virus Total (www.virustotal.com); this website can detect suspicious file uploaded.

Using VertexNet Loader, it is created botnet with filename **data.exe**. This botnet was checked by Virus Total, and the result can be shown at https://www.virustotal.com /en/file/b6b364d3fd29ddfa671b22cd4c6b9a97dc16840bd8c30 70b3837b87ef74433d4/analysis/1435626970/.

Fig. 15 is the analysis of the result. We may see that 51 of 55 (92,7%) antivirus detect the file as malware. We may conclude that the botnet will not survive when we broadcast it using social engineering technique.



SHA256: b6b364d3fd29ddfa671b22cd4c6b9a97dc16840bd8c3070b3837b87ef74433d4
File name: data.exe
Detection ratio: 51 / 55
Analysis date: 2015-06-30 01:16:10 UTC (3 weeks ago)

Antivirus	Result	Update
ALYac	Gen:Variant.Adware.Symmi.5057	20150630
AVG	Generic32.XSA	20150630
AVware	Trojan.Win32.Generic!BT	20150630

Fig. 15 Analysis of Botnet File (data.exe)

To solve this problem, we need to modify the botnet in order to hide from antivirus detection. we may use many tools to do that, but in our work, we use binder/joiner, and crypters.

1. Binder/Joiner

Binder/Joiner is a small program that allows you to easily join (bind) two or more files (no matter what their type is) into one single executable file. That executable (the one into which the files are included) is a simple compiled program that, when opened, it will automatically launch the included files one by one.

In our work, it is used putty as a chosen simulation tools to join botnet file (Fig. 16), and test with ten (10) joiner tools.

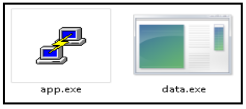


Fig. 16 Joining Putty and Botnet

TABLE V (analysis date: June 29, 2015) is the result of file detection as a malware. Using Deception v4 tools, only 34 AV detect the file as a malware.

TABLE V. JOINING DETECTION RESULT

No.	Joiner Tool	Detection Ratio
1	Deception v4	34/55 = 61.8%
2	Hell Packer 2.0	34/55 = 61.8%
3	Super Binder	36/55 = 65.4%
4	Micro Joiner 1.5	37/55 = 67.2%
5	Naked Bind 1.0	38/55 = 69.0%
6	File Joiner 1.5	39/55 = 70.9%
7	Crypor Crypter FUD	39/55 = 70.9%
8	Fearz Packer v0.3 Private	45/55 = 81.8%
9	Moruk Binder v1.0	45/55 = 81.8%
10	Pretator v1.6	45/55 = 81.8%

We may conclude, after using Binder/Joiner tools, the detection ratio decreases. This joining process is a first step to maximize the survivability of botnet when spreading to the real case (globally to test and survey on a company's security concerns). It is still difficult to make botnet more stealth to evade detects by antivirus. Joining/binding technique is not strong enough to maximize the survivability of botnet.

2. Crypters

Crypters can pack the file in a way that the actual bytes are not readable. People use a crypter to protect a file that they release from crackers and reverse engineering, or make files that are detected for some reasons fully undetectable (FUD). False positives can be defeated on this way and the file will get protected with a polymorphic and strong encryption.

In this project, we use five crypters to crypt botnet file. The crypters are: (1) Aegis Crypter6.3, (2) Boson Crypter 3.3 Free Version, (3) Free Crypter 27-06, (4) Kazy Crypter1.3, and (5) Swayz Cryptor. We crypt the botnet three times. Table VI shows the result of crypt detection ratio based of Virus Total databases.

TABLE VI. CRYPT DETECTION RESULT

Crypters	**Detection Ratio**			**File Size (KB)**		
	1st crypt	**2nd crypt**	**3rd crypt**	**1st crypt**	**2nd crypt**	**3rd crypt**
Aegis	26 / 55	26 / 55	27 / 55	615	1073	1542
Boson	0 / 55	0 / 55	0 / 55	288	575	1149
F.Crypter	5 / 55	0 / 55	0 / 55	768,18	768,20	768,22
Kazy	11 / 55	18 / 55	22 / 55	156	184	208
Swayz	13 / 55	13 / 55	12 / 55	1038	2289	4552

In Table VI, we can confirm that the detection ratio is decreased when botnet is protected by crypters, except for Kazy and Aegis. At this point, we have the best crypter that can solve our problem about evading detection of antivirus. Boson Crypter and Free Crypter is the best crypter at this test lab. Free Crypter is recommended because it completely evades all antivirus detection with small bits added.

V. CONCLUSIONS

To deploy a botnet, it is necessary to choose a botnet loader, topology, and spreading method. After doing some test cases of connection testing, joining testing, and crypter testing, it was deployed a new kind of botnet that can 100% evade all of the 55 antiviruses registered at virustotal.com databases.

The malware that was developed in the paper can evade the antivirus so that it can be used to monitor the awareness of employee in the company or organization. The result of monitoring was a report result from the mapping of the people aware from being infected, for example, if the result found that 90% of employees are infected, it is an indication of a very dangerous level of the awareness. Then, the Top Level Management may take action based on that monitoring report result.

REFERENCES

[1] B. Stock, J. Gobel, M. Engelberth, F.C. Freiling, and T. Holz, “Walowdac-analysis of a peer-to-peer botnet,” *2009 European Conference on Computer Network Defense (EC2ND),* IEEE, pp. 13-20, November, 2009.

[2] R. Sharp, *An Introduction to Malware,* Technical University of Denmark, 2013.

[3] P. Correia, E. Rocha, A. Nogueira, and P. Salvador, “Statistical characterization of the Botnets C&C traffic,” *Procedia Technology*, 1, 158-166, 2012.

[4] J. Calvet, C.R. Davis, CJ.M. Fernandez, J.Y. Marion, P.L. St-Onge, W. Guizani, and A. Somayaji, “The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet,” *Proc. of the 26th Annual Computer Security Applications Conference,* pp. 141-150, ACM, Dec., 2010.

[5] B. AsSadhan and J.M. Moura, “An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic,” *Journal of Advanced Research*, vol. 5(4), pp. 435-448, 2014.

[6] P. Wang, L. Wu, B. Aslam, and C.C. Zou, “ A systematic study on peer-to-peer botnets,” *Proc. of 18th Internatonal Conference on Computer Communications and Networks, ICCCN 2009* ,pp. 1-8, IEEE, Aug., 2009.

[7] L. Trolle Borup, “Peer-to-peer botnets: A case study on Waledac,” *Doctoral dissertation, Technical University of Denmark, DTU*, DK-2800 Kgs. Lyngby, Denmark, 2009.

[8] P. Wang, S. Sparks, and C.C. Zou, “An advanced hybrid peer-to-peer botnet,” *IEEE Trans. on Dependable and Secure Computing,* vol. 7(2), pp. 113-127, 2010.

[9] J. Baltazar, J. Costoya, and R. Flores, “Infiltrating WALEDAC Botnet’s Covert Operations: Effective Social Engineering, Encrypted HTTP2P Communications, and Fast-Fluxing Networks,” *TREND MICRO*, 2009.

[10] W. Tarng, L.Z. Den, K.L. Ou, and M. Chen, “The analysis and identification of P2P botnet’s traffic flows,” *International Journal of Communication Networks and Information Security (IJCNIS),* vol. 3, No. 2, Aug., pp 138-148, 2011.

[11] VertexNet Tutorial and Guide (http://www.hackforums.net/showthread.php?tid=4181421) visited June 20, 2015.

Albert Sagala is a Senior Lecturer in the Electrical Study Program, Del Institute of Technology. His main research focuses on Network Computer, Computer Network Security, ICS and SCADA Security. He was graduated from Engineering Physics, Bandung Institute of Technology. He got a master degree from Bandung Institute of Technology on Instrumentation and Control field. He is an active researcher, also member of APTIKOM.

Alexander Lumbantobing is a Research Assistant at Cyber Security Research Centre. He is an active person, a leader of Indonesian Backtrack Team-Toba Samosir Regency. He was graduated from the Computer Engineering Study Program, Del Institute of Technology.