# An Advanced Hybrid P2p Botnet 2.0

T. T. Lu, H.Y. Liao, M .F. Chen

***Abstract*****—**Recently, malware attacks have become more serious over the Internet by e-mail, denial of service (DoS) or distributed denial of service (DDoS). The Botnets have become a significant part of the Internet malware attacks. The traditional botnets include three parts – botmaster, command and control (C&C) servers and bots. The C&C servers receive commands from botmaster and control the distributions of computers remotely. Bots use DNS to find the positions of C&C server. In this paper, we propose an advanced hybrid peer-to-peer (P2P) botnet 2.0 (AHP2P botnet 2.0) using web 2.0 technology to hide the instructions from botmaster into social sites, which are regarded as C&C servers. Servent bots are regarded as sub-C&C servers to get the instructions from social sites. The AHP2P botnet 2.0 can evaluate the performance of servent bots, reduce DNS traffics from bots to C&C servers, and achieve harder detection bots actions than IRC-based botnets over the Internet.

***Keywords*****—**Peer-to-peer, Botnets, Botnet 2.0, Hybrid peer-to-peer

## I. Introduction

THE Internet is a communication infrastructure that interconnects the global community of end users and content servers. In recent years, malware attacks become more serious over the Internet by e-mail, denial of service (DoS) or distributed denial of service (DDoS). The Botnets become a significant part of the Internet malware attacks.

The traditional botnets consist of three parts -botmaster, command and control (C&C) servers and bots. (1) The botmaster sends commands to C&C servers and do malware attacks. (2) The C&C servers receive commands from botmaster and control the distributions of computers remotely. (3) Bots use DNS to find the positions of C&C server, and then communicate with C&C servers, periodically. However, the C&C server is easily detected or blocked by network manager or firewall in that the C&C server is a bottleneck in traditional botnets. Morales et al. [1] analyze the DNS traffic to find infected hosts when the infected hosts first join a botnet. Peer-to-peer (P2P) botnets consist of three parts – botmaster, servent bots and client bots. The P2P botnets are distinctive from traditional botnets in that there is no central C&C server for a P2P botnet.The current researchers have focused on monitoring and detecting the traffic of existing botnets. Jang et al. [2] and Grizzard et al. [3] monitored the Waldac and the Trojan.Peacomm botnet traffic to detect malicious peer-to-peer botnet, respectively. Chang et al. [4] discussed some mechanisms to detect the existing P2P botnets. Jian et al. [5] proposed a neighbour list selecting mechanism to decrease the connection time from control nodes to bots in P2P Botnet. Some of researchers presented specific peer-to-peer botnets (Wang, Sparks, Zou, [6]; Xie and Tan, [7]; Hung and Tan, [8]) that they are harder to be monitor than traditional botnets.

The social websites use Web 2.0 technology to interact and collaborate with each other in virtual community, such as blog, video sharing, instant messaging. Nguyen and Josef [9] named Bot 2.0 the bots that use Web 2.0 communication methods, such as when the attacker uses public blog service as an information temporary storage for C&C server.

Therefore, the motivation of this paper is to reduce the DNS traffics from bots to C&C servers and achieve harder detection than traditional botnets over the Internet. To achieve this, we use social websites regarded as C&C server to hide the encryption malware information and index factors to select the candidates of servent bots. Then, P2P botnet 2.0 mechanism is applied for botnet structure to achieve harder detection.

The paper is organized as follows: Section II. presents the proposed of hybrid P2P botnet 2.0 structure (AHP2P botnet 2.0). Simulation results are given in Section III.. Finally, the conclusions are drawn in Section IV..

## II. Hybrib P2P Botnet 2.0 Structure

The proposed advanced hybrid P2P botnet 2.0 structure (AHP2P botnet 2.0) is illustrated in Fig. 1. The AHP2P botnet 2.0 consists of three parts – botmaster, social websites, bot groups. Bot groups consist of servent bots and client bots. Malware Information from botmaster is embedded into social websites. The servent bots in bot groups will link with the social website to get the malware information from the social websites and send to client bots. Client bots attack target after they receive malware information from servent bot. The details of the process of embedding social website, the process of servent bots, and the process of client bots are described separately as follows.

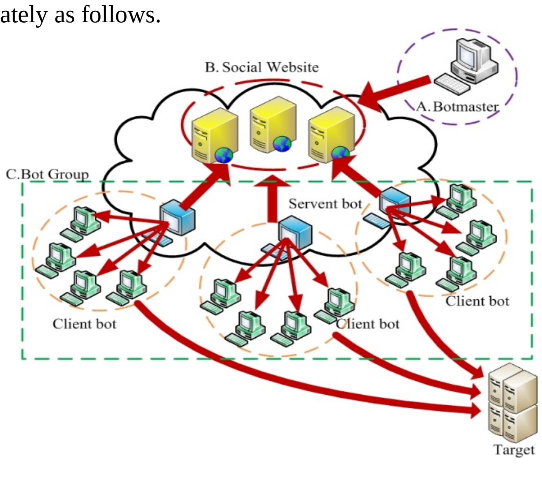

Fig. 1 The proposed hybrid P2P botnet 2.0 structure

T. T. Lu is with the Department of Computer Science & Information Engineering, University of Ching Yun, Chung-Li, Taoyuan County 320,Taiwan ROC (phone: 886-3-4581196 ext. 7700; fax: 886-3-2503013; e-mail: ttlu@ cyu.edu.tw).

H. Y. Liao is with the Department of Institute of Computer, Communication, and System Engineering. University of Ching Yun, Chung-Li, Taoyuan County 320,Taiwan ROC (e-mail: m9852003@cyu.edu.com)

M. F. Chen is with the Department of Computer Science & Information Engineering, University of Ching Yun, Chung-Li, Taoyuan County 320,Taiwan ROC (e-mail: m9913001@cyu.edu.com).

### *A. The Process of Embedding Social Website*

The encryption malware information is embedded into blog service after the malware information is encrypted by Message-Digest algorithm 5 (MD5). The malware information includes an attack version, malware contents, attack time and peer list about IP address of servent bots. Attack version is used to check whether the malware attack has been executed or not. Malware contents include target's MAC and IP address, etc. Peer list addresses the IP address of existence servent bots

### *B. Selection Servent Bots and Client Bots*

Hardware configuration, online time, and social websites visiting times are important index factors to select the servent bots from infected host *i*.

$$IF_{i=} WF_{CPU} + WF_{FREQ} + WF_{MEM} + WF_{OT+} WF_{VT'} \qquad (1)$$

where IFi is index factor for host i, WFCPU is number of process weighting value, WFFREQ is CPU frequency weighting value, WFMEM is a weighting value for memory size, WFOT is a weighting value for online time, WFVT is a weighting value for visiting social websites times

$$\begin{cases} F_i = 1 \quad IF_i \geq T_0, \, host\ i\ is\ servent \\ F_i = 0 \quad IF_i < T_0, \, host\ i\ is\ client \end{cases} \qquad (2)$$

Initial flag $F_i$ is null for each infected host *i*. If the index factor $IF_i$ is larger than threshold $T_0$, then the host *i* set as a servent bot, $F_i$=1. Otherwise, the host set as a client bot, $F_i$=0. The next sections describe the servent bots and client bots process.

### *C. The Process of Servent Bots*

The steps of servent bots process include register mode and attack mode are shown in Fig. 2. Step 1 to step 3 is register mode and step 4 to step 5 is attack mode that all of the steps are described as follows:

● Register mode

Step 1: Check initial flag $F_i$. If $F_i$=1 go to step 2. Otherwise, $F_i$=null return to section B.;

Step 2: Link to social website to paste the peer list about encryption IP address information;

Step 3: Save MAC and IP address about client bot and go to step 4 if client bot register is succeed;

● Attack mode

Step 4: Link to social website to get encryption malware information from botmaster;

Step 5: Check attack version and send malware contents to client bots in bot group if attack version hasn't been executed; Otherwise, the servent bot goes to step6;

Step 6: Wait for the next malware commands from botmaster and go to step 4;

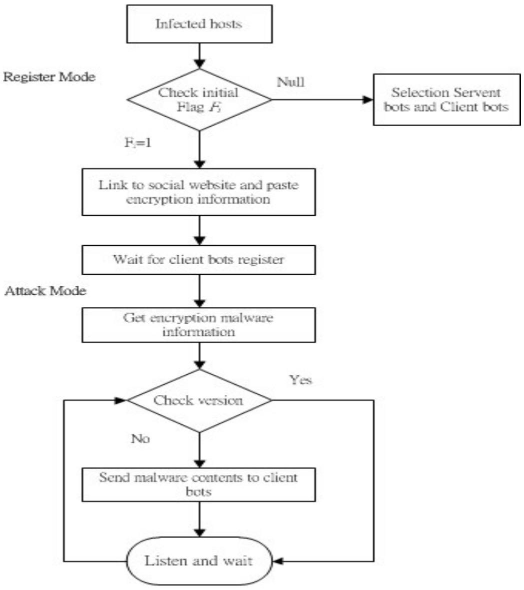
Infected hosts
Register Mode
Check initial Flag $F_i$
Null
Selection Servent bots and Client bots
$F_i$=1
Link to social website and paste encryption information
Wait for client bots register
Attack Mode
Get encryption malware information
Yes
Check version
No
Send malware contents to client bots
Listen and wait

Fig. 2 Steps of servent bots process: register mode and attack mode

### *D. The Process of Client Bots*

The steps of client bots process include register mode and attack mode as shown in Fig. 3. Step 1 to step 4 is register mode and step 5 is attack mode. All of the steps are described as follows:

● Register mode

Step 1: Check initial flag $F_i$. If $F_i$=0 go to step 2. Otherwise, $F_i$=null return to section B.;

Step 2: Link to social website to get the peer list about IP address of servent bots from encryption information;

Step 3: Round-trip time (RTT) represents the distances between client bot and servent bots. RTT mechanism is applied to get the minimum distance of servent bot from peer list;

$$RTT = min\{RTT_{1,} RTT_{2,} \mathbf{I} \ RTT_{n=}\} \qquad (3)$$

Step 4: Each client bot will register to a servent bot from step 3 and get one ID's number;

$$\begin{cases} ID_i \leq T_1, \, register\ succeed \\ ID_i > T_1, \, register\ \ fail \end{cases} \qquad (4)$$

The client bot *i* joins into the bot group *i* and go to step 5 if register succeeds. Otherwise, the client bot returns to step 3 and gets the next IP address of servent bot.

● Attack mode

Step 5: Wait malware commands from servent bot;

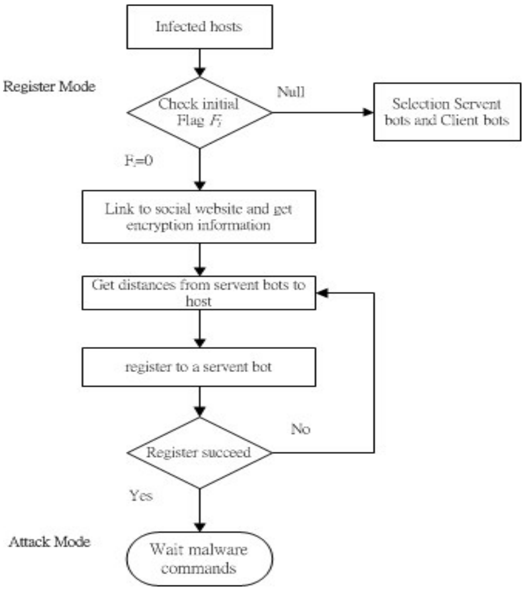
Infected hosts
Register Mode
Check initial Flag $F_i$
Null
Selection Servent bots and Client bots
$F_i$=0
Link to social website and get encryption information
Get distances from servent bots to host
register to a servent bot
No
Register succeed
Yes
Attack Mode
Wait malware commands

Fig. 3 Steps of client bots process: register mode and attack mode

## III. Experiment Results

Index factor *IF* is used to evaluate the performance for each infected host. If the number of process is larger than 2, $WF_{CPU}$=3. If CPU frequency is larger than 3.0 GHz, $WF_{FREQ}$=1. If memory size is larger than 2 G, $WF_{MEM}$=2. If online time exceed 24 hours, $WF_{OT}$=3. If visiting social websites times exceed 2, $WF_{VT}$=4. The threshold $T_0$ = 10 and $T_1$ = 300 are chosen for the simulation test. Table 1 shows the percentage of CPU used and P2P traffic in each servent bot to control numbers of client bots. Table 2 compares the percentage of CPU used for AHP2P botnet 2.0 and Hung and Tan, 2009. Simulation results show that AHP2P botnet 2.0 has better performance than Hung and Tan for the percentage of CPU used over 50 client bots. Table 3 compares the numbers of DNS queries in 24 hours for AHP2P botnet 2.0, Waledac. 5, and Waledac. D.gen. In our simulation results, AHP2P botnet 2.0 has the lowest DNS queries than Waledac. 5 and Waledac. D. gen. AHP2P botnet 2.0 actually achieves harder detection than traditional botnets.

TABLE I
AHP2P Botnet 2.0 Simulation Results

| | AHP2P botnet 2.0 | |
|---|---|---|
| Numbers of client bots / servent bot | Percentage of CPU used | P2P traffic ( byte /sec) |
| 50 | 0.4% | 10.5 K |
| 100 | 0.77% | 17 K |
| 150 | 1.15% | 27.5 K |

TABLE II
Comparison the Percentage of CPU Used for AHP2P Botnet 2.0 and Hung and Tan.

| | AHP2P botnet 2.0 | Hung and Tan |
|---|---|---|
| Numbers of client bots / servent bot | Percentage of CPU used | Percentage of CPU used |
| 50 | 0.4% | 5% |
| 64 | 0.5% | 50% |
| 100 | 0.77% | - |
| 150 | 1.15% | - |

TABLE III
Numbers of Packet in DNS Queries

| | AHP2P botnet 2.0 | Waledac.5 | Waledac.D.gen |
|---|---|---|---|
| DNS Packets/24ur | 4 | 360 | 792 |

## IV. Conclusion

This work presents an advanced hybrid peer-to-peer (P2P) botnet 2.0 mechanism using web 2.0 technology to instruct social sites. The approach is particularly suitable for hiding the encryption malware information. Simulation results show that the proposed method reduces DNS traffics from bots to C&C servers and achieves harder detection than traditional botnets

## References

[1] J. A. Morales, S. Xu, and R. Sandhu, “Analyzing DNS Activities of Bot Processes, ” *in Proc. IEEE Int. Conf. Malicious and Unwanted Software*., pp. 98-103, Oct. 2009.

[2] D. I. Jang, M. Kim, H. C. Jung, and B. N. Noh, “Analysis of HTTP2P Botnet: Case Study Waledac,” *in Proc. IEEE Int. Conf. Communications*, pp. 409-412, Dec. 2009.

[3] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, , “Peer-to-peer botnets: Overview and case study”, *in Proce. Hot Topics in Understanding Botnets*, pp. 1-8, April 2007.

[4] S. Chang, L. Zhang, Y. Guan, and T. E. Daniels,“A Framework for P2P Botnets”, *in Proc. IEEE Int. Conf. Communications and Mobile Computing*, pp 594-599, Jan. 2009.

[5] G. Jian, Y. Yang, K. F. Zheng, and Z. M Hu, “Research of An Innovative P2P-Based Botnet”, *in Proc. IEEE Int. Conf. Machine Vision and Human-machine Interface*, pp 214-218, April 2010.

[6] P. Wang, S. Sparks, and C. C Zou, “An Advanced Hybrid Peer-to-Peer Botnet” *IEEE Trans. Dependable and Computing.*, vol. 7, no. 2, pp. 113-127. April-Jun 2010.

[7] J. Xie, and L. Tan, “Detection Methods Research of Half Distributed P2P Botnet”, *Application Research of Computers*, vol. 26, no. 10, pp. 3925-3928. Oct. 2009.

[8] P. Hung, and L. Tan, “Design and Implementation of Partially Decentralized P2P Botnet Control”, *Journal of Computer Applications*, vol. 29, no. 9, 2009, pp. 2446-2449.

[9] H. Nguyen, P. Josef, 2010, “Protecting Web 2.0 Services from Botnet Exploitations”, *in Proc. IEEE Int. Conf. Cybercrime and Trustworthy Computing Workshop.*, pp.18-28. July 2010.

## Acknowledgment

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 99-2221-E-231-031