

Detecting Potential Peer-to-Peer botnets using the payload of network packets

Karun Dambiec u4462988

Australian National University

COMP8760 Computer Science Project

Supervisor: Dr Warren (Huidong) Jin, NICTA/CSIRO



Introduction

- Increasing threats to the security of computer systems
- In 2003, total internet attacks increased by 67.5% compared to the previous year[1]
- Peer-to-Peer botnets are one of the threats to computer and network security.
- They are growing in size, difficult to detect and could severely affect an organisations ability to function.



Honeypots

- A honeypot is a system which collected any data that has been sent to it over a network, and it allows us to record information about unauthorised traffic or attempts to attack a network[2].
- SGNet is a worldwide honeypot project, consisting of a distributed network of approximately 25 honeypots with multiple participants. It uses a combination of technologies being ScriptGen, Argos and Nepenthes to allow a high level of interaction with its attackers, so it can collect more detailed information than low interaction honeypot projects[3].



Hypotheses

- There is some peer-to-peer bot which sends an encrypted packet to a system, to determine if it is already infected and a potential peer with the peer-to-peer bot before attempting to inject the system with it or attack the system.
- By finding packets which are not similar to the standard operating system implementation of ICMP, we assume we can detect a peer-to-peer bot before it manages to attack other systems in the network, by detecting its communication with peers and potential peers as soon as possible.



Experiments and Results

- Method for detecting peer-to-peer bots where peers are checked for before injection
- Method for detecting peer-to-peer bots where they communicate using ICMP packets



Method for detecting peer-to-peer bots where peers are checked for before injection

- Consists of a method for detecting encrypted contents of network packets
- Then finding sessions where encrypted packets are the first packets making up the session



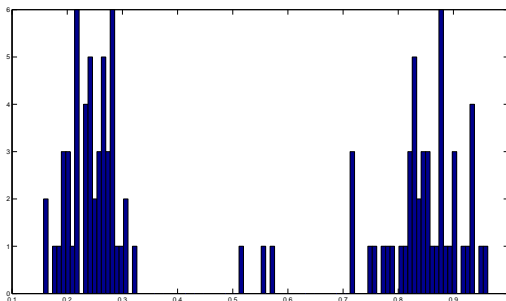
Method For Encryption Detection

- Data: Packet
- Result: Percentage of Plain Text Characters
- For each character c in packet
- If c is a letter between A to Z, a to z or a number 0 to 9.
- Increment number of plaintext characters
- Else If c is not a space
- Increment number of non plaintext characters
- Return: number of plaintext characters divided by number of non plaintext characters



Histogram of Percentage of Alphabetical Characters for Packets

We tested the method we proposed for detecting encrypted data using files that were created using PGP desktop and the plaintext versions. We then created a histogram as follows:



Confusion Matrix

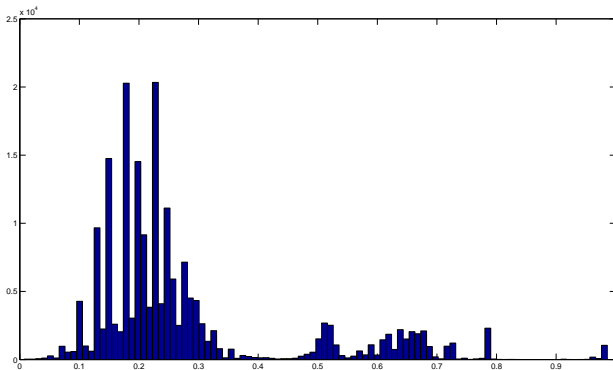
We then examined the data around the peaks of the histogram and selected 70% as the percentage required to be considered plaintext. Then we created a confusion matrix of the results:

	Encrypted	Plaintext
Encrypted	47	0
Plaintext	3	50

Given the test data we found this method to be 97% accurate.



Period of 15th - 21st August 2008



Histogram of Percentage of Alphabetical Characters for Packets for Period of 15th-21st August 2008

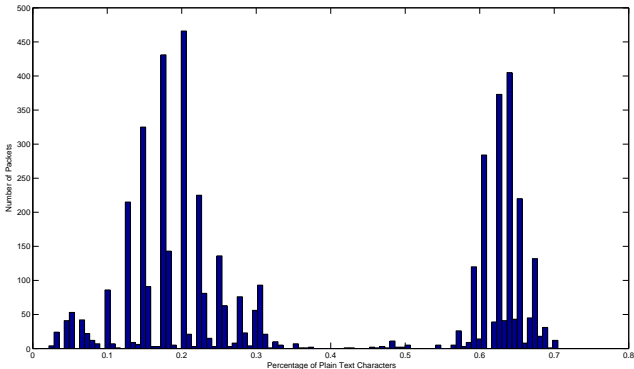


Period of 15th - 21st August 2008

- We examined ten packets around each of the peaks of the histogram to determine which peak was plaintext and encrypted data
 - Plaintext packets $> 60\%$
 - Possible Plaintext or Encrypted $> 50\%$ and $< 60\%$
 - Encrypted packets $< 50\%$
- Then we created a list of all sessions where plaintext packets occurred after encrypted. We found a total of 320 sessions.
- Subject to the usual problems associated with not being able to differentiate between encrypted and binary data, no obvious systematic effects on the results were observed.



Period of 15th - 21st August 2008



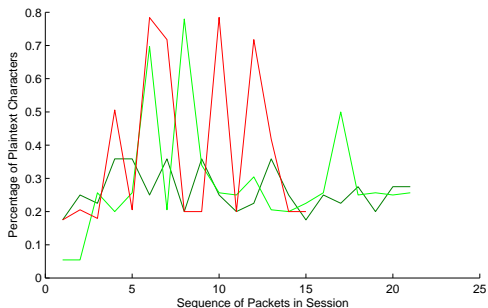
Percentage of Plain text characters for the first packet of each sessions during the period 15th - 21st August 2008

Beginning of Session 737250

```
E | 2Sp P$f P
E | @UpS P8$gPtO
E 4 2Sp P$g9Pj
E G 2Sp P$g9PT HEAD / HTTP/1.0
E G 2Sp P$g9PT HEAD / HTTP/1.0
```



We found ten sessions with similar behaviour to session 73250:



Combined Graph of Sessions

- This behaviour is indicative of a peer to peer botnet attempting to check for peers.
- As a result of current data mechanisms used by the SGNNet honeypot project, we are currently unable to confirm if it was a peer to peer bot, as it was not downloaded successfully by the honeypot project.



Method for detecting peer-to-peer bots where they communicate using ICMP packets

- ICMP is a protocol used for notification of events relating to the availability of a computer network and hosts on it[4].
- We used a random sample of fifty sessions, where encrypted packets occurred before plain text packets and we selected only ICMP packets from those sessions.
- We then captured the payload of ICMP ECHO packets for Windows XP, Mac OS X, and Linux



Payload of the Windows XP ICMP ECHO implementation

- ..k F... ..5...E. .jG.....: A.abcdef ghijklmn opqrstuv wabcdefg hi
- Next we will use levenshtein distance



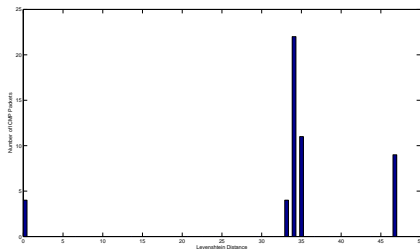
Levenshtein Distance

- Levenshtein distance allows us to determine the number of operations that is required to transform one sequence into another sequence[5].
- For the transformation of the word pear to beers, we calculate the Levenshtein distance as follows:
 - 1 pear → bear (We substitute b for p)
 - 2 bear → beer (We substitute e for a)
 - 3 beer → beers (We insert an s at the end)



Levenshtein Distance

- We then calculated the Levenshtein distance between each operating systems implementation and the data collected by the SGNet Honeypot project
- For the windows XP implementation we have the following



histogram:



Levenshtein Distance

- We used the method we developed previously in this work, and found that the ICMP packets which did not match the windows implementation were encrypted.
- As a result of the analysis using levenshtein distance, and the previous method proposed, we found these sessions could use ICMP with an encrypted payload to determine if a system is a peer.



Discussion

- Main difficulty is detection of peer-to-peer botnets
- As a result of encrypted network traffic, and decentralised control structure
- We examined two methods which we proposed to detect peer-to-peer botnets that have certain behaviours.
- We have progress towards our goal of detecting network traffic peer-to-peer botnets



Future Work

- Large scope for future work
- Improving and developing methods of detecting potential peer-to-peer botnets
- Methods to collect data regarding peer-to-peer botnets
- Automating selection of peaks that are used to determine the difference between encrypted and plaintext network packets
- Improvements to differentiate between binary data and encrypted data.
- Simulation of peer to peer botnets, or methods for capturing traffic between peer to peer botnets.



What I Learnt

- Gained skills in conducting research
- Improved Project and Time Management Skills



Conclusion

- Currently a long way to go in research in peer to peer bots, as they are relatively new with storm peer-to-peer botnet being introduced in 2007.
- We contributed a method of detecting communication between potential peer to peer bots, and a second method of detecting peer to peer bots using levenshtein distance where ICMP ECHO packets are not the standard operating system implementation.



Conclusion

- This work is only the beginning of research into the detection of peer to peer botnets, and the future development of proactive mechanisms to defend against this threat to computer and information security.
- We developed an understanding of what needs to be done for future experiments to improve the quality of data available regarding peer-to-peer botnets.



References



T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the dos and ddos problems," *ACM Comput. Surv.*, vol. 39, no. 1, p. 3, 2007.



I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *ACM-SE 45: Proceedings of the 45th annual southeast regional conference*, (New York, NY, USA), pp. 321–326, ACM, 2007.



C. Leita and M. Dacier, "Sgnet: A worldwide deployable framework to support the analysis of malware threat models," *edcc-7*, vol. 0, pp. 99–109, 2008.



J. Postel, "Rfc 792 - internet control message protocol," 1981.
from <http://tools.ietf.org/html/rfc792>.



Wikipedia, "Levenshtein distance," September 2008.
from http://en.wikipedia.org/wiki/Levenshtein_distance.



Acknowledgements

- NICTA for allowing me to visit from July 2008 to November 2008.
- SGNet Honeypot Project Participants
- Huidong (Warren) Jin for his supervision, and encouragement.

