Challenges Of Modeling BotNets For Military And Security Simulations

Sheila B. Banks, Ph.D.
Calculated Insight
Orlando, Fl 32828
(407) 353-0566
sbanks@calculated-insight.com

Martin R. Stytz, Ph.D.
Institute for Defense Analyses
Washington, DC
(407) 497-4407, (703) 338-2997
mstytz@ida.org , mstytz@att.net , mstytz@gmail.com

Abstract. Simulation environments serve many purposes, but they are only as good as their content. One of the most challenging and pressing areas that call for improved content is the simulation of bot armies (botnets) and their effects upon networks and computer systems. Botnets are a new type of malware, a type that is more powerful and potentially dangerous than any other type of malware. A botnet's power derives from several capabilities including the following: 1) the botnet's capability to be controlled and directed throughout all phases of its activity, 2) a command and control structure that grows increasingly sophisticated, and 3) the ability of a bot's software to be updated at any time by the owner of the bot (a person commonly called a bot master or bot herder.) Not only is a bot army powerful and agile in its technical capabilities, a bot army can be extremely large, can be comprised of tens of thousands, if not millions, of compromised computers that can surreptitiously communicate with each other and their command and control centers. In sum, these capabilities allow a bot army to execute technically sophisticated, difficult to trace, tactically agile, massive, coordinated attacks. Clearly, botnets pose a significant threat to all computing and network systems. To improve our understanding of their operation and potential, we believe that it is necessary to develop computer security simulations that accurately portray bot army activities, with the goal of including bot army simulations within military simulation environments. In this paper, we investigate issues that arise when simulating bot armies.

1. INTRODUCTION

Bot armies are a new type of malware that are more powerful and possibly dangerous than any other type of malware. Their power and threat derive from the fact that bot armies, unlike other forms of malware, can be controlled and directed throughout all phases of an attack using a command and control structure that is increasingly sophisticated and allows the bot's software to be updated at any time by the owner of the bot (commonly called a bot master or bot herder.) A bot army is composed of tens of thousands, if not millions, of compromised computers that can surreptitiously communicate with each other and their command and control centers; allowing them to execute massive, coordinated attacks upon Internet resources and upon any equipment attached to the Internet. The deployment and operation of bot armies are aided by the security vulnerabilities that exist in contemporary software; vulnerabilities that are likely to increase in number commensurately with the increase in the size of software products. The operation of bot armies is also aided by several freely available software technologies that support covert communication within the bot army and between the bot master and the bot army.

To advance the state of the art and of the practice of military and security simulation environments, the simulation community must come to grips with the challenges posed by botnets. Botnet challenges arise from their inherent flexibility as well as from the rapid development of botnet technologies. The development of botnet simulation capabilities requires advances in two main thrust areas: improving our understanding of bot army technologies and capabilities as well as the development of standards and technologies that support the simulation of bot army operations under a variety of conditions and their full panoply of capabilities. In addition to the challenges posed by botnet simulation, there are also the challenges posed by the integration of bot army simulations into larger interactive and constructive simulation environments. To date, little work has been reported in the open literature concerning these issues. In this paper, we will delve into these and subsidiary issues to better illuminate the challenges we must address as well as outline what we believe to be worthwhile areas of botnet research and standards development, areas that will yield improved bot army simulations as well as more realistic and useful simulation environments. The importance of the need for standardizing and improving botnet simulation stems not only from their potential use in military operations but also the affect they can have upon support functions, such as logistics and medical support, that are also critical to the efficient operation of a military or security operation.

In this paper, we discuss the need for bot army simulation environments along with the need and benefits from their incorporation into military simulation environments. The next presents background material and a discussion of related topics. Section Three contains a discussion of the challenges that we anticipate in developing standards and our suggested foundation for the standards. Section Four contains the conclusion and suggestions for further work.

2. BACKGROUND

"Botnets", or "bot armies" [1-35], are large groups of remotely controlled malicious software. Botnets, remotely controlled and operated by botmasters or botherders, can launch massive denial of service attacks, multiple penetration attacks, or any other malicious network activity on a massive scale. In a "botnet" or "bot army", computers can be used to spread spam, launch denial-of-service attacks against Web sites, conduct fraudulent activities, and prevent authorized network traffic from traversing the network. Botnets are remotely controlled and operated by botmasters (also called botherders). While bot army activity has, so far, been limited to criminal activity, their potential for causing large-scale damage to the entire internet is incalculable.

Bots and bot armies, as shown in Figure 1, arose almost as soon as internet chat was developed and have been developing in their capabilities ever since. No one technology is responsible for the rise of bot armies as a threat, rather it is the development of several technologies that permits bots to pose the threat. At its most basic, a bot requires a command and control (C2) channel, malware, and a distribution technology. The simplest, and earliest, bots used simple internet relay chat (IRC) for C2, malware in the form of a packet generator (to conduct a denial of service attack), no host for distribution of additional software for the bot, and a C2 node at a fixed IP address for C2. However, bot technology has accelerated in its development in the last few years and bots have become increasingly malicious. The modern era of bot army activity was initiated in February 2000, when a Canadian hacker commanded his bot army to attack CNN.com, Amazon.com, eBay.com, Dell Computer (at dell.com), and other sites with a huge volume of traffic, a traffic volume that was sufficient to take the targeted computer systems off-line. Bot armies are effective for two reasons: they can execute multiple overt actions against targets and can, alternatively, provide multiple coordinated and covert listening points within targeted networks and computer systems. Bot software exhibits three main characteristics at different points in its operation. These characteristics are those of a virus, a worm, and a Trojan. From the point of view of a botherder, virus technology is just a means that can be exploited to plant the initial infecting bot software into a computer. Also for the botherder, worm technology is just a means for allowing the bot software to move through the internet. Finally, the botherder uses Trojan technology for the host so that it can disguise itself by behaving like a program that purports to do one thing while, in fact, doing additional nefarious activities.

The general pattern of botnet creation requires a few basic steps: 1) malware creation, 2) command and control creation, 3) malware propagation, 4) malware infestation, 5) command and control setup, 6) further malware download, and 7) malware check-in for further instructions via the command and control setup. To activate a botnet, a malware author needs to gain access to the Internet in a manner that allows him/her/them to hide their identity, access the Internet from a wide variety of Internet Protocol (IP) addresses, and acquire as much total bandwidth as possible. In order to facilitate initial contact with the bot after it has infected a computer, the malware author typically encodes an initial contact domain name into the malware binary. In preparation for contact by the bots as they become active after infection, the bot master prepares a command and control computer, or set of computers operating off of a variety of Internet Protocol (IP) addresses.

Infestations can be accomplished using a number of techniques; for example, the bot may have been inserted into the person's computer by being wrapped in a file or e-mail attachment that looks innocent. The bot software may also have infested the computer because there was some hidden code on a website that the user visited, which downloaded it to their machine. Once infestation is complete, the bot checks in to receive instructions. The instructions generally direct the bot to search out additional hosts to infect, to locate and exfiltrate information of interest to the botmaster, or to participate in a coordinated attack on computer targets. While the bot army is in operation, the botherder has two main tasks: assigning tasks to the army (via the command and control nodes) and developing new software for the bots.

Currently, the key to botnet defense lies in the detection of the subtle indicators of infection and detecting bot command and control activity. Detecting an individual bot is difficult; therefore, armies are usually detected by their command and control activity. Command and control is a challenge for botherders because the connection is both their means for control and is the easiest way for them to be caught. Botherders solve the problem by directing the bots to connect to specific command and control machines. This approach, while easy to implement, is also easy to detect and defeat. As a result, botherders continue exploring ways to improve command and control of their bots.

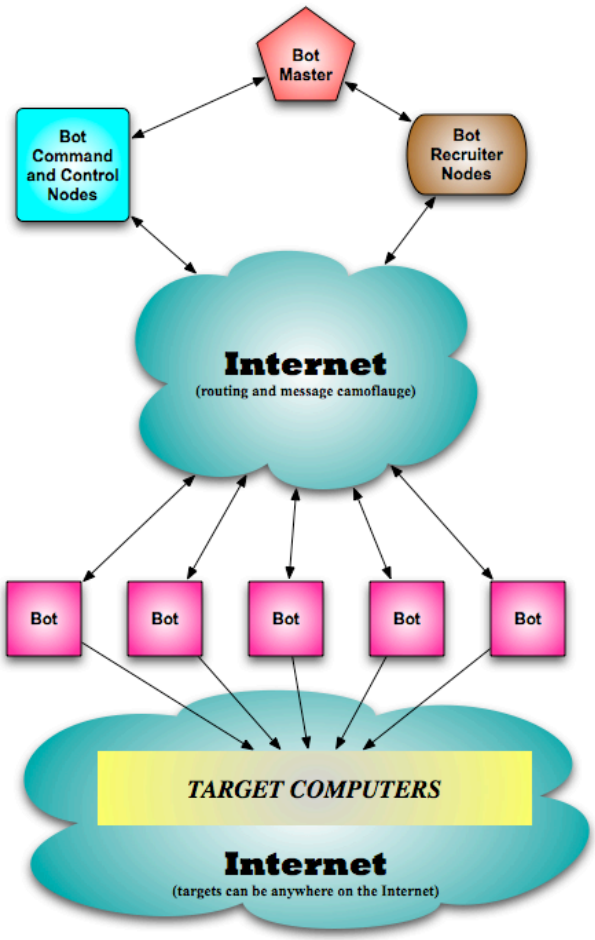

Figure 1: Typical Generalized Bot Army Configuration

Botnets are capable of migrating through a network and the internet. Their progression largely is constrained by the types of operating systems and computer systems defenses that are in place and the malware that was implanted within the hardware or software during manufacture (if any). An approach for simulating the complexities of botnets and their infestation is discussed in the next section.

3. CHALLENGES TO DEVELOPING MODELING STANDARDS

Developing standards for botnet simulation is complex for a variety of reasons. In addition to the wide variety of botnets and their manner of propagation, there is also the challenge posed by modeling the amount of time and patterns of their infestation. However, we need not start without a basis; there is a broad body of work in the field of epidemiology that can be drawn upon for modeling purposes [36-47]. The general transfer diagram used to portray disease transmission and outcomes is presented in Figure 2. The transfer diagram portrays, in an abstract format, the potential sources, infestation pathways, and outcomes for fatal disease transmission. There is a large body of work that has been developed to describe and model the transmission and infestation vectors in the model for various diseases, a much larger body of work than we can discuss here in reasonable detail. We believe that this model and body of work can be used as a basis for describing bot army infestation and propagation. (The actual model used for a given disease is modified from this general model based upon the type of infection, transfer modality, and potential for re-infection.)

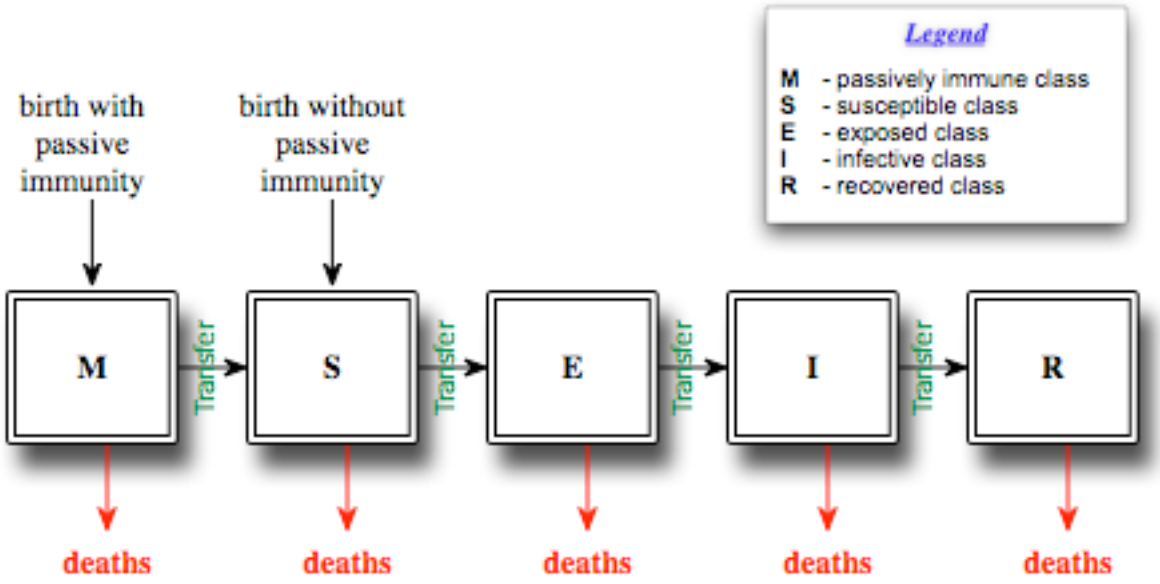

Figure 2: General Disease Transfer Diagram

To preserve commonality with preceding epidemiology research, we suggest using the same symbology for each stage of transmission, but just change their meaning. Typically, M is the class of babies born with passive immunity (due to the mother), in our formulation **M** is the class of computers (hardware or software) who are not infected with malware that can be exploited to enable bot infestation. S is usually employed to represent the class of newborns that have lost passive immunity or newborns that never had any immunity, with the transfer from the M to S class modeled by the rate at which passive immunity disappears from newborns. In our formulation, the class **S** is used to represent the class of computers (hardware or software) that are infected during manufacture with malware that can be exploited to enable bot infestation. The class E is the set of individuals who have been exposed to the infection but do not show signs of infection. In our formulation, the class **E** is the set of computers that have been infected, are not transmitting the infection, and in whom the infection has not been detected. The class I is typically comprised of the individuals in whom the latency period for the infection has passed, who can transmit the infection, and who exhibit signs of infection. In our formulation, the class **I** is the set of computers that have been infected, are transmitting the infection, and in whom the infection has not been detected (the equivalent of people that exhibit signs of infection.) The class R is typically the set of individuals for whom the infection period has ended and who have acquired permanent infection-acquired immunity. In our formulation, the class **R** is the set of computers that have been infected, whose infection has been detected, and that have had their bot removed. While we have defined the classes of susceptibility for botnet infection, we need to examine each class in somewhat more detail in order to present the basis for the development of a complete model.

Clearly, in our proposed model the class S is not derivative from the class M, and these two classes are parallel initial states, with both states contributing to the class E. However, since there are many types of bot armies, the model must account for the possibility that a computer that is predisposed to falling victim to a bot infection may not become infected because it is not exposed to the required malware or a computer may become infected by several bots simultaneously but none of the bots are the bots that the computer was pre-disposed to be infected by due to its implanted malware. For any given type of bot, the classes M and S are disjoint, but for the set of all bots there can be a significant overlap between the two classes. Therefore, for a given type of bot, there is a different transition probability from the class M and the class S to the class E. The class E, while being the class of infected computers, is comprised of two subclasses: 1) the subclass of infected computers that provide command and control for the botnet, called E_C and 2) the subclass of infected computers that are the bots, called E_B. The class I is comprised of the subclass of computers in the class E that are actively attempting to infect additional computers and place them into the botnet: either as a command and control member or a plain bot. Because there are two subclasses in class E, there are four transfer equations/probabilities to transition from class E to I; $E_C \Rightarrow$ command and control, $E_C \Rightarrow$ bot, $E_B \Rightarrow$ command and control, and $E_B \Rightarrow$ bot. These probabilities represent the probability that members of the class will be attempting to spread the infection, not the probability of detection for the class. As regards detection, each subclass in classes E and I have their own detection probabilities, and those probabilities are used to determine the transition rate from each of the subclasses to class R. The probabilities of detection for each subclass are also related to the volume of data transmitted, frequency of transmission, the activity of each subclass of bot within its host computer, and the bot's defenses. Note that since there is no "natural" immunity conferred on a computer after having been cleansed of a bot infection, it is possible for a previously infected computer to be infected by the same bot again.

This probability is portrayed by a transition probability from state R back to one of the two subclasses in state I.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have discussed the challenge posed by botnets. One of the most challenging and pressing areas that call for improved content is the simulation of bot armies (botnets) and their effects upon networks and computer systems. Botnets are a new type of malware, a type that is more powerful and dangerous than any other type of malware. In order to advance the state of the art for botnet understanding, improved modeling and simulation can be invaluable tools. However, if these tools are to provide their maximum benefit, we require standard models for their operation; models that capture all aspects of their behavior and that are flexible enough to portray every type of bot and the variations in their operation. Because botnets have the entire internet as their domain of operation, modeling them has posed a challenge, which has hindered the development of standards for modeling botnet propagation and operation. In response to these challenges we propose drawing upon the epidemiological literature. This field of research has had to address many of the same challenges posed by botnets, such as worldwide dispersion of infection sources, rapid transmission, dormant infections, different types of resistance to infection, opportunity for re-infection, and other factors. Their model provides a solid foundation for botnet modeling efforts. Using the epidemiological model as a basis, we proposed a model for botnet infection and transmission that can be used as a foundation for development of a comprehensive standard for botnet operation.

Our future work in the area of botnet operation modeling and simulation will concentrate on refining the model that we proposed. In addition to developing models for the transition probabilities, we will also address the operation of the botnets in finer detail, their relationship to firewalls and other defenses against malware, and the modeling challenges posed by the different types of botnets. We believe that there is much research remaining to be done, but that we have a solid foundation for our own further research on botnets.

REFERENCES

Malware and Botnets

1. Binkley, J.R. and Singh, S. (2006) "An Algorithm for Anomaly-Based Botnet Detection," *Usenix: Steps to Reducing Unwanted Traffic on the Internet (SRUTI) '06*, San Jose, CA, http://www.usenix.org/events/sruti06/tech/full_papers/binkley/binkley.pdf
2. Butler, J. and Silberman, P. (2006) "RAIDE: Rootkit Analysis Identification Elimination," *Blackhat Europe 2006*, Amsterdam, The Netherlands, February-March, 2006.
3. Cohen, F. (1987) "Computer Viruses," *Computers & Security*, vol. 6, no. 1, pp. 22-35.
4. Conti, G. (2006) "Hacking and Innovation," *Communications of the ACM*," vol. 49, no. 6, pp 33-36, June.
5. Curve (2003) "Just What is a Botnet?" *Dalnetizen*, January, http://zine.dal.net/previousissues/issue22/botnet.php
6. Dagon, David; Takar, Amar; Gu, Guofei; Qin, Xinzhou; and Lee, Wenke. (2004) 'Worm population control through periodic response." Technical report, Georgia Institute of Technology, June.
7. Farrow, C. and Manzuik, S. (2006) "Injecting Trojans via Patch Management Software and Other Evil Deeds," *Blackhat Europe 2006*, Amsterdam, The Netherlands, February-March, 2006.
8. Heasman, J. (2006) "Implementing and Detecting an ACPI BIOS Rootkit," *Blackhat Federal 2006*, Washington, DC, January.
9. Hoffman, B. (2006) "Analysis of Web Application Worms and Viruses," *Blackhat Federal 2006*, Washington, DC, January.
10. Hoglund, G. and Butler, J. (2005) *Rootkits: Subverting the Windows Kernel*, Addison-Wesley, Boston.
11. Ianelli, N. and Hackworth, A. (2005) *Botnets as a Vehicle for Online Crime*, Cert Coordination Center, http://www.cert.org/archive/pdf/Botnets.pdf
12. Kaspersky Labs (2006) *Malware Evolution*. January-March, http://www.viruslist.com/en/analysis?pubid=184012401, April.
13. Kaspersky Labs (2005) *Malware Evolution*. January-March, http://www.viruslist.com/en/analysis?pubid=162454316 , April.
14. Kienzle, Darrell M. and Elder, Matthew C. (2003) "Recent worms: A survey and trends," *WORM'03: Proceedings of the 2003 ACM workshop on Rapid Malcode*, NY, NY. pp. 1-10.
15. Killourhy, Kevin; Maxion, Roy; and Tan, Kymie. (2004) "A Defense-Centric Taxonomy Based On Attack Manifestations," *International Conference on Dependable Systems and Networks (ICDS'04)*.
16. Mohay, G.; Anderson, A.; Collie, B.; DeVel, O.; and McKemmish, R. (2003) *Computer and Intrusion Forensics*, Artech House: Boston, MA.
17. Moore. D. (2002) "Code-red: A case study on the spread and victims of an Internet worm." http://www.icir.org/vern/imw-2002/imw2002-papers/209.ps.gz.
18. Moore, D.; Paxson, V.; Savage, S.; Shannon, C.; Staniford, S.; and Weaver, N. (2003) "Inside the slammer worm." *IEEE Magazine on Security and Privacy*, vol. 1, no. 4, July.
19. Moore, D.; Shannon, C.; Voelker, G. M.; and Savage, S. (2003) "Internet quarantine: Requirements for containing self-propagating code. *Proceedings of the IEEE INFOCOM 2003*, March.

20. Murdoch, S. and Danezis, G. (2005) "Low-Cost Traffic Analysis Of Tor." In *Proceedings of the IEEE Symposium on Security and Privacy*.

21. Naraine, R. (2005) "Where are Rootkits Coming From?," *eWeek*, December, http://www.eweek.com/article2/0,1895,1897728,00.asp

22. Naraine, R. (2006) "VM Rootkits: The Next Big Threat?," *eWeek.com*, March 10, http://www.eweek.com/article2/0,1895,1936666,00.asp

23. Naraine, R. (2006) "'Blue Pill' Prototype Creates 100% Undetectable Malware," *eWeek.com*, http://www.eweek.com/article2/0,1895,1983037,00.asp

24. Ollmann, G. (2006) "Stopping Automated Application Attack Tools," *Blackhat Europe 2006*, Amsterdam, The Netherlands, February-March, 2006.

25. Overlier, L. and Syverson, P. (2006) "Playing Server Hide and Seek," *Blackhat Federal 2006*, Washington, DC, January.

26. Pfleeger, C.P. and Pfleeger, S.L. (2006) *Security in Computing, 4th ed.*, Prentice-Hall, Upper Saddle River: NJ.

27. Ramachandran, A.; Feamster, N.; and dagon, D. (2006) "Revealing Botnet Membership Using DNSBL Counter-Intelligence," *Usenix: Steps to Reducing Unwanted Traffic on the Internet (SRUTI) '06*, San Jose, CA, http://www.usenix.org/events/sruti06/tech/full_papers/ramachandran/ramachandran_html/

28. Realtime Community, "Botnet Threats," http://www.realtime-websecurity.com/061205_sullivan.asp

29. Ripeanu, M.; Foster, I.; and Iamnitchi, A. (2002) "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal*, vol. 6, no. 1.

30. Rutkowska, J. (2004) "Red Pill... or how to Detect VMM Using (Almost) One CPU Instruction," *Invisible Things*, http://www.spidynamics.com/spilabs/education/articles/ Internet-attacks.html

31. Rutkowska, Joanna. (2006) "Rootkit Hunting vs Compromise Detection," *Blackhat Federal 2006*, Washington, DC, January.

32. Rutkowska, J. (2005) Rootkits vs Stealth by Design Malware," *BlackHat Europe*, Amsterdam, March.

33. Shannon, C. and Moore, D. (2004) "The spread of the witty worm," *Security & Privacy Magazine*, vol. 2, no. 4, pp. 46-50.

34. Skoudis, E. (2004) *Malware: Fighting Malicious Code*, Prentice Hall, NJ.

35. Spitzer, L. (2003) *Honeypots: Tracking Hackers*, Addison-Wesley: Boston: MA.

Epidimiology

36. Hethcote, H.W. (2000) "The Mathematics of Infectious Diseases," *SIAM Review*, vol. 42, no. 4, pp. 599-653.

37. Hyman, J.M. and Li, J. (2006) "Differential Susceptibility and Infectivity Epidemic Models," *Mathematical Biosciences and Engineering*," vol. 3, no. 1, January, pp. 89-100.

38. Allen, L.J.S. (1994) "Some Discrete Time SI, SIR, and SIS Epidemic Models, *Mathematical Biosciences*, vol. 124, pp. 83-105.

39. Allen, E.J. "Jump Diffusion Model for the Global Spread of an Amphibian Disease," *International Journal of Numerical Analysis and Modeling*, vol. 1, no. 2, pp. 173-187.

40. Filiol, E.; Helenius, M.; and Zanero, S. (2006) "Open Problems in Computer Virology," *Journal of Computer Virology*, vol. 1, pp. 55-66.

41. Anderson, R.M. and May, R.M. eds (1991) *Infectious Diseases of Humans: Dynamics and Control*, Oxford University Press, Oxford, UK.

42. Bailey, N.T.J. (1975) *The Mathematical Theory of Infectious Diseases*, 2nd ed, Haufner, NY.

43. Brauer, F. (1990) "Models for the Spread of Universally Fatal Diseases," *Journal of Mathematical Biology*, vol. 28, pp. 451-462.

44. Busenberg, S.N. and Hadeler, K.P. (1990) "Demography and Epidemics," *Mathematics of Biosciences*, vol. 101, pp. 41-62.

45. Cliff, A.D. (1996) "Incorporating Spatial Components into Models of of Epidemic Spread," *Epidemic Models: Their Structure and Relation to Data*, Mollison, (ed), Cambridge University, UK.

46. Metz, J.A.J. and vanden Bosch, F. (1996) "Velocities of Epidemic Spread," in *Epidemic Models: Their Structure and Relation to Data*, D. Mollison (ed), Cambridge University Press, UK, pp. 150-186.

47. Mollison, D. (1996) *Epidemic Models: Their Structure and Relation to Data*, D. Mollison (ed), Cambridge University Press, UK,