Improving Simulation of Botnet Infection and Propagation

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

The simulation of cyber warfare and cyber activities, especially the activities of bot armies (botnets) and their effects upon networks, computers, users, and society, are an important simulation challenge. The importance of improving botnet simulation stems from their potential use in military operations and in other security-oriented simulations. Botnets are malware that can be remotely controlled at all times, uses increasingly sophisticated command and control structure, and can be upgraded at any time by the controller. A bot army is powerful and agile in its technical capabilities and can be extremely large, comprising tens of thousands or millions of computers. Botnets are a threat to all computing and networked systems. To improve our understanding of botnet operation and combat future hostile uses, bot army simulations that can be inserted into military simulation environments are needed.

Developing botnet simulation capabilities requires advances in two areas: improved understanding of bot army technologies and development of standards and models that support the simulation of bot army operations. Additional challenges are posed by integrating bot army simulations into interactive and constructive simulation environments. To date, little work has been reported in the open literature concerning these issues. In the paper, we address these and related issues to highlight the challenges of botnet research and standards development.

In this paper, we discuss the need for botnet simulations, describe a model for botnet operation, and discuss the need and benefits realized by their incorporation into broader simulation environments. Section One presents an introduction to bot armies and malware, the expected benefits, and the motivation for our research and for research on bot armies. Section Two presents background material on bot armies and malware and a discussion of related topics. Section Three presents the characteristics of our botnet model and its uses. Section Four contains the conclusion and suggestions for further research.

ABOUT THE AUTHORS

SHEILA B. BANKS is the President of Calculated Insight. Dr. Banks received her Bachelor of Science from the University of Miami, Coral Gables, FL in 1984 and a Bachelor of Science in Electrical Engineering from North Carolina State University, Raleigh, NC in 1986. Also from North Carolina State University, Raleigh, NC, she received a Master of Science in Electrical and Computer Engineering in 1987 and her Doctor of Philosophy in Computer Engineering (Artificial Intelligence) from Clemson University, Clemson, SC in 1995.

MARTIN R. STYTZ is a retired Lieutenant Colonel in the U.S. Air Force. He received a Bachelor of Science degree from the U.S. Air Force Academy in 1975, a Master of Arts degree from Central Missouri State University in 1979, a Master of Science degree from the University of Michigan in 1983. Stytz received his Ph.D. in Computer Science and Engineering from the University of Michigan in 1989. He is a member of the ACM, the IEEE, the IEEE Computer Society, USENIX, AAAI, and the Society for Computer Simulation.

Improving Simulation of Botnet Infection and Propagation

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

1. INTRODUCTION

Bot armies are a new type of malware that are more powerful and possibly dangerous than any other type of malware [48]. 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 address 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 models 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, modeling, 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 and the benefits of botnet modeling and the incorporation of these models into military simulation environments. The next section presents background material and related topics. Section Three contains a discussion of our botnet model, suggested uses for the model, and a suggested foundation for standards. Section Four contains the conclusion and suggestions for further work.

2. BACKGROUND

"Botnets", or "bot armies" [1-35, 66], 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 army 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 Internet protocol (IP) address for C2.

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 technical means for moving the bot software 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, performing additional nefarious activities.

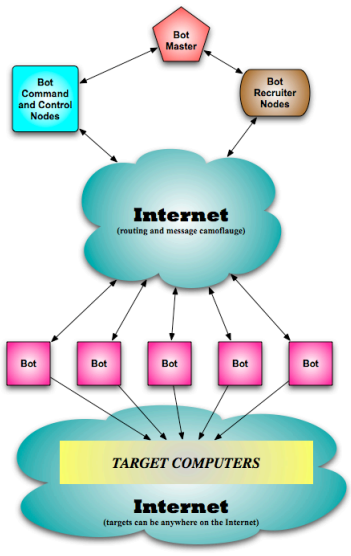

Figure 1: Typical Bot Army Configuration

Figure 2 illustrates the general pattern of botnet creation. Botnet creation consists of 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 by each bot, and 7) bot check-in with its C2 node for further instructions distributed via the bot master's command and control setup.

To activate a botnet, a bot 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 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 bot malware author typically encodes an initial contact domain name into the bot software 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 IP addresses. Once infection of a computer is accomplished, the bot uses the host computer's IP information to acquire additional malware and to obtain its operational instructions. Obviously, the bot can acquire additional malware and instructions at any other time in the future as well. The command and control computer(s) in the typical bot army are of one of two types: a high-bandwidth compromised machine, or a high-capacity co-located computer. The command and control computer or computer suite is set up to run an Internet Relay Chat (IRC) service to provide command and control of the bots by the bot master. There are numerous intermediate steps and processes that a bot herder, and the bot army, must continually perform in order to expand the bot army, insure its survival, and secure its communications. Establishing and managing a bot army is a complex undertaking, requiring considerable time and resources.

Widespread bot infestations can be achieved by combining a number of simple, apparently innocent techniques. For example, the bot may have been inserted into a computer by being wrapped in a file or e-mail attachment that looks innocent. The bot software may also infest a computer using some hidden code on a website, which was downloaded to the computer, that the computer's user visited. 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 bot herder has two main tasks: assigning tasks to the army (via the command and control nodes) and developing new software for the bots.

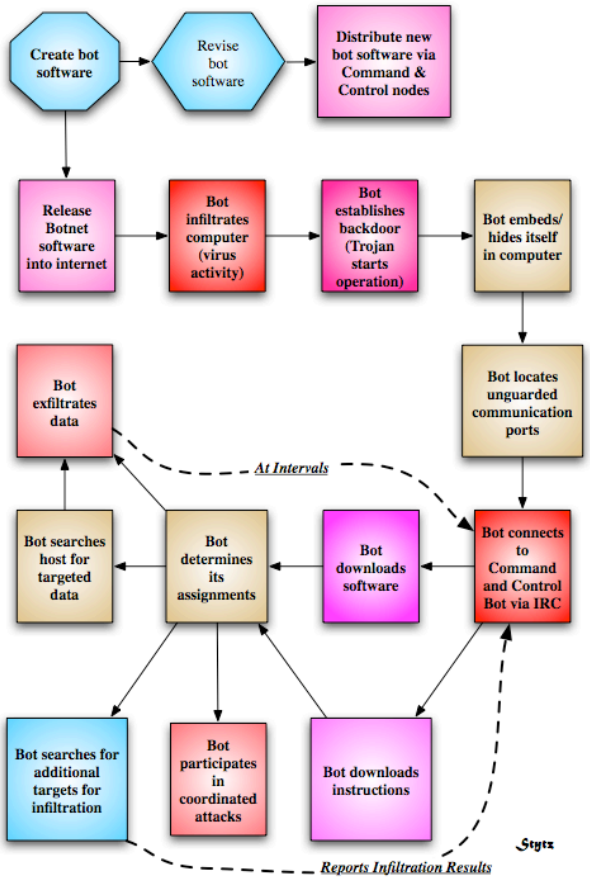

Figure 2: Botnet Development and Expansion Life cycle

Currently, the centerpiece of botnet defense lies in the detection of the subtle indicators of infection and detecting bot command and control activity. Detecting an individual bot on a single computer is difficult; therefore, bot armies are usually detected by their command and control activity and not their malicious activity. Command and control is a challenge for bot herders because the C2 connection is both their means for control and is the easiest way for the bot herder to be caught. Bot herders address the problem in part 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, bot herders have developed new ways to achieve command and control of their bots; techniques that are effective even after bots lie dormant, after bots migrate to different computers, or even after a user tries to cleanse a computer of a bot infestation. As indicated, bots are now capable of migrating through a network and the Internet, which further increases the difficulties that must be overcome when attempting to remove a bot infestation and aids a bot master in maintaining their army's size and potency. Bot movement through the internet is somewhat constrained by the types of operating systems and computer system defenses that are in place and can be aided by malware that was implanted within the hardware or software during manufacture (if any.) An approach for modeling the complexities of botnets and their infestation is discussed in the next section.

3. DEVELOPING AND EMPLOYING MODELING STANDARDS

The first step toward achieving a botnet simulation capability is developing a set of requirements for the cyber warfare simulation system. We determined a set of requirements after examining several training systems, other reported approaches to developing malware models [56-63, 67] and derived from our experience assembling other simulation systems. Because a botnet simulator is a type of cyber warfare simulator, the same requirements apply.

In no particular order of importance, we determined that the requirements for a cyber warfare (and botnet) simulation capability are the following: 1) Simulate the effects of malware without introducing the actual malware infection into the system, 2) Provide machine and human-detectable attack signatures as well as information and taskload exposure that are identical to

those in the real-world, 3) Execute known/observed attacks as well as hypothetical attacks, 4) Provide execution and replay of cyber attacks at any speed and allow defenders to try alternative defensive maneuvers at any point in the cyber attack, 5) Simulate attacks that are scalable to any size/scope of attack, able to be executed within a distributed virtual environment as well as within a constructive environment, 6) Produce effects upon human beings and their decision processes similar to the those produced in the real-world by a similar attack (confusion, task overload, and high degree of uncertainty), and 7) Support multiple, parallel, simultaneous attacks against one or more targets that the defense must protect. In sum, the requirements call for building the cyber warfare equivalent of the Air Force Red Flag capability, and for the same reason, acquisition of realistic experience.

Realistic experience in conducting cyber warfare is needed because cyber warfare will be confusing and large-scale conflicts in cyberspace can create serious information, financial, and infrastructure disruption and damage because the target of a cyber attack is information. The distinctions between cyber attacks are based upon whether the information targeted by the attack is to be taken, corrupted, blocked, or replaced. In a cyber attack, events will unfold rapidly and mistakes by defenders and attackers will be inevitable unless the key decision-makers (who are usually non-technical) and key attack analysts (who are technical) have experience in managing major cyber warfare. A simulation system, or better a distributed simulation comprised of multiple networked cyber warfare simulation systems, is the ideal and safest means for providing the required cyber warfare management expertise. However, to develop the simulation system, we must be able to model cyber operations. Because botnets are among the most dangerous forms of malware, we decided to begin our modeling efforts by addressing the botnet modeling challenge.

Developing botnet simulations 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 an infestation persists in a computer and the patterns of infestation displayed by a particular bot's software. However, we need not start without a foundation; there is a broad body of work in the field of epidemiology that can be drawn upon to model infection transmission and infection [36-47] as well as the work reported by Tarakanov and Dasgupta [50-55] concerning immunity. We apply their work to model cyber attacks, specifically botnet army propagation.

In the medical literature, infection transmission is usually portrayed using the general disease transmission and outcome diagram presented in Figure 3. The transfer diagram portrays, in an abstract format, the potential sources, infestation pathways, and outcomes for fatal disease transmission. The research describes and models disease transmission and disease infestation vectors for various diseases; there is 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, potential for re-infection and combinations of these three factors.)

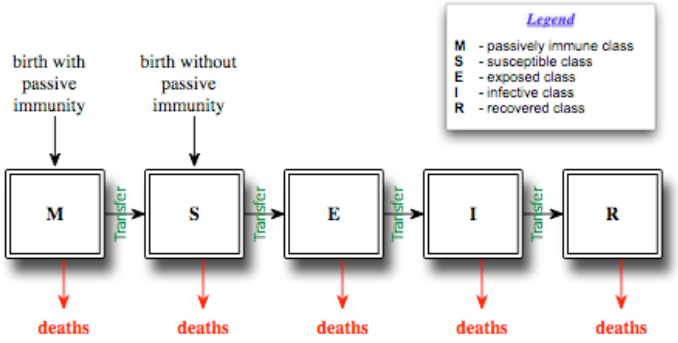

Figure 3: General Model of Disease Transmission/Infection in Public Health

To fully exploit existing epidemiology research, we use the same symbology shown in Figure 2 for each stage of bot infestation, but that the meaning of the symbols should be changed to suit the botnet modeling challenge. In medicine, typically, M is the class (or portion) of the population born with passive immunity (due to the mother.) In our formulation **M** is the class of computers (hardware or software) that are not infected with malware during manufacture or development that can, nevertheless, be exploited to enable bot infestation. In medicine, S is employed to represent the class of the population that has lost its maternally acquired passive immunity plus the portion of the population that never had immunity, with the

transfer from the class M to class S determined by the rate at which passive immunity disappears. In medicine, S represents the susceptible population. 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 (hence these computers are more susceptible than expected.)

In medicine, the class E is the set of individuals who have been exposed to the infection but do not show signs of infection. In our model, the class **E** is the set of computers that have been attacked, that have been infected, and in whom the infection has not been detected. Note the size of the set (**M** + **S**) is not equal to the entire set of computers that have been constructed, and the probability of a computer changing state from class **M** to **E** is not the same as the probability of a computer changing state from class **S** to **E**. In medicine, the class I is typically comprised of the set of individuals in whom the latency (or dormancy) 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, are performing the task(s) that the bot was intended to perform, that show (or contain) signs of bot infection, and in whom the infection has not been detected. The members of the set **I** are the most active bots in the bot army and the most likely to be detected; for a given bot the number of computers in the class **I** is less than or equal to the number of computers in the class **E**, with the difference being the number of bots that are lying dormant.

In medicine, the class R is the set of individuals for whom the infection period has ended and who have acquired permanent infection-acquired immunity; they are sometimes referred to as the infection's reservoir. In our formulation, the class **R** is the set of computers that have been infected, whose infection has been detected, and whose bot software has been removed. However, these bots can, potentially become susceptible to the bot and become re-infected; to account for this possibility cleansed computers that have lost their immunity are placed back into the set **S** and are treated the same as any computer that lacks passive immunity from the bot infection. Figure 4 presents our model and the relationships between the classes of bot infection that we have defined.

Having a basic model for the classes of susceptibility for botnet infection, we had to examine each class in more detail in order to propose and justify a complete model, one that accounts for all of the significant steps in a bot army's lifecycle as illustrated in Figure 2. A more detailed analysis is currently required because, at this time, the worldwide penetration of computers by bots is estimated to be 90% [49], but this number only provides a useful upper bound for all bot infections and provides no insight into the infection rate for subsets of the worldwide set of computers, especially the computer systems of interest to military operations, whereas our objective is to develop comprehensive but closed-form models of bot army operation. The model outlined above is a first step in the development of the desired capability.

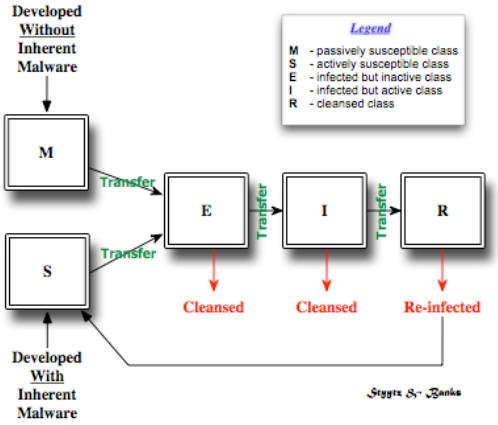

Figure 4: General Model of Bot Infection

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 membership in the class **E** (the class of currently infected computers) based upon the type of exposure to the bot infection. However, since there are many types of bot armies, the model must account for the possibility that a computer that is simulated as predisposed to being vulnerable to a particular bot infection may not become infected because it is not exposed to the required malware. This may occur because the current bot that infects the computer prevents further infection, or because a computer may become infected by several bots simultaneously. Note that for any single type of bot, while the classes **M** and **S** are disjoint, for the set of all types of bots there can be a significant overlap between the two classes. Therefore, for each different type of bot, there is a different transition probability from the class **M** to the class **E** and from the class **S** to the class **E**.

The class **E**, the class of infected computers, is comprised of two subclasses: 1) the subclass of infected computers that can provide command and control for the botnet, called $\mathbf{E}_C$ and 2) the subclass of infected computers that are designed to be merely the bots, called $\mathbf{E}_B$, but have not yet been activated. The class **I** is drawn from the class of computers in the class $\mathbf{E}_B$. The members of the class **I** are the bots performing botnet tasks and/or 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 that affect the transition from class **E** to **I** that must be considered: $\mathbf{E}_C \Rightarrow$ command and control, $\mathbf{E}_C \Rightarrow$ bot, $\mathbf{E}_B \Rightarrow$ command and control, and $\mathbf{E}_B \Rightarrow$ bot. Historically, the class **E** is a "safer" class than the class **I** (bots in class **E** have a lower probability of detection than those in class **I**), and the transition $\mathbf{E}_B \Rightarrow$ bot is the transition most likely to be detected. However, in this paper we are primarily interested in the probabilities regarding recruitment and alteration of class state as well as the probability that a member of a class will attempt to spread the infection, not the probability of detection for a particular class. Nevertheless, as regards detection, each subclass in classes **E** and **I** have their own detection probabilities, and those probabilities specify the transition rate from each of the subclasses to class **R**. The probabilities of detection for each subclass in **E** and **I** are related to the volume of data transmitted by the members of the subclass, 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 re-infected by the same bot again. This probability is portrayed by a transition probability from state **R** back to the state **S** and is represented by P_R.

There are at least two additional factors that must be addressed in order to model and simulate a bot infestation. The first factor is modeling the effect of types of exposure to bot software (botware) in order to determine the probability of being attacked through the network, USB drives, removable media (CD-ROMs, DVD-ROMs, etc,), or other media. The second factor that must be addressed is how to model the probability of an infection by a particular bot, P_I, since this number determines the number of computers in the class **I**.

The effect of types of exposure to bot software, called $P_{BotWare}$, is modeled by the probability of being successfully attacked through media and is a component of the variable called probability of an attack, P_P. P_P has other components in addition to $P_{BotWare}$. Other components of P_P include the probability that the hardware was infected during manufacture, P_{PH} , and the probability that the software was infected during manufacture, P_{PS}. It is known that these probabilities are independent; that is, $P_{BotWare}$ is not related to P_{AH}, which is not related to or dependent upon $P_{PS.}$ Rather, all probabilities operate independently, so $P_P = P_{BotWare} + P_{PS} + P_{PH}$. The computer security literature indicates that several infections due to manufacturing can be in place simultaneously in both the hardware and software. However, since we are not trying to determine the probability of a particular computer being infected, but instead the probability that a set of computers is infected by a specific bot, the mean infection rate of hardware and software due to manufacturing will be sufficient for our purposes. The infection rate for hardware is known to be non-zero, small, and less than entirely pervasive. For our purposes here, P_{PH} is assigned a reasonably small number. The infection rate for software is quite high and is due to two factors. The first factor is deliberate insertion of malware for later use. The second factor is the persistent problem of software errors, and is generally related to both the complexity of the software and the number of lines of code. Schneider [64] empirically determined that the expected number of software errors (B) in a software development project to is related to the following: 1) the overall reported months of programmer effort for the project (E), 2) the number of subprograms (n), and 3) the count of thousands of coded source statements (S). These estimators are the following two equations:

$$B \approx 7.6 E^{0,667} * S^{0.333} \quad (1)$$

and

$$B \approx n * \left(\left(S/n \right) / .045 \right)^{1.667} \quad (2)$$

In general, the error count in software is a relatively accurate estimate of the number of errors that can be exploited by malware [65]. Therefore, in our model, we use Schneider's model to compute B, as described in equations 1 and 2, which allows us to approximate the number of errors in software and determine a reasonable estimate for the number of deliberately inserted pieces of malware in software. The sum of these figures divided by the count of thousands of coded source statements (S) gives us our approximate value for P_{PS}; $P_{PS} = B + P_{SM} + P_{HM}$.

To keep the model manageable, extensible, and useful for modeling multiple bot army attacks, each type of botnet is modeled and simulated individually. P_I for a particular botnet is computed for the complete set of computers in the simulation, not for a single computer; thereby, allowing us to estimate the size of the infected set for a particular bot. As indicated by the preceding discussion, there are many factors that determine the probability of infection for a particular bot. A baseline model for bot infection can be built based upon the known progression and effect of a bot attack. The baseline model is the probability of an attack, P_A, times the probability of a successful penetration, P_P, times the probability of a successful exploitation, P_E.

$$\mathbf{P_I = P_A * P_P * P_E} \quad \mathbf{(3)}$$

$\mathbf{P_I}$ will, for a given computer population, determine the number of computers in the sets **E** and **I**.

However, there are multiple factors that affect formula (3) and that must be considered. One factor that affects the formula is that $\mathbf{P_A}$ and $\mathbf{P_P}$, which are applied to the size of the sets **M** and **S** for a given bot and gives rise to the set **E** for all computers for a particular bot, are dependant upon separate probabilities for attack and penetration for hardware, P_{AH} and P_{PH}, and software, P_{AS} and P_{PS}, for a given bot. The size of the sets **M** and **S** are related to both P_{AH} and P_{AS} because a computer must both be vulnerable to attack and in a position and a condition on the network to be attacked (an isolated, powered-down computer cannot be attacked in a meaningful manner) in order for it to be exposed to the bot infection. The different values for **M** and **S** are due to the presence of defenses against the bot attack that were inserted into the computer during construction or after deployment; therefore, for specific values of P_{AH} and P_{AS} we should expect the number of computers that transition from state **M** to **E** to be greater than the number of computers that transition from the state **S** to **E**. Additionally, the number of computers in the set **E** for a particular bot is dependant upon the values for P_{AH} and P_{AS} and the values for P_{PH} and P_{PS}. If the hardware and software for a particular computer were developed to insure that the hardware and software have no malware that enables a particular bot infection and the defenses against bot infection are strong, then the values for P_{PH} and P_{PS} are small because the computer can be considered to have "passive" immunity to the bot infection and must be infected via transmission of malware that can not exploit inside assistance and bot defenses must be overcome. The equation expresses this intuition concerning bot infestation. However, it is difficult to assemble a computer that is immune to all bot infections; therefore, the probability of attack upon a computer's hardware and software by a bot must be considered in relation to all bots and in relation to the possibility that a bot infestation can be designed both to exploit covert malware[1] and to conduct an un-aided penetration.

Because of the reported volume of malware attacks conducted via the Internet, we assume that P_{AS} has a value of 1.0. The value for P_{AH} is very small. We assume that P_E has a value very close to 1.0 because any competent bot developer will insure that the bot software executes within the target environment if the bot software successfully penetrates the defenses erected against bot infestation.

To properly portray bot activity within a simulation environment, the guiding principle must be to portray the bot activity without running the risk of introducing an active bot infection. Therefore, we need only assign the probability of infection (and re-infection) as well as the portrayal of the results of the infection. To portray the activity of a bot army within a simulation the simulationist must determine the human-visible effects and the machine-detectable effects that the bot can inflict, develop the software needed to produce the desired effects, and then determine P_I. Once P_I is determined, then the infected computers are determined at runtime during the simulation and the appropriate effects are portrayed. Standards are needed to aid in the determining reasonable values for all of the probabilities required by our model. Additional standards are required to address simulated bot communications and simulated bot activity within an infected computer. Development of standards to portray simulated bot activity within an infected computer is especially important; bots regularly inflict damage upon their computer hosts and disrupt information flow; we require standards that allow these effects to be presented in a consistent manner across a simulation without actually altering or destroying any information.

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 medical epidemiological literature.

The medical epidemiological field of research addresses many of the same challenges posed by botnets propagation modeling, such as worldwide dispersion of infection sources, rapid transmission, dormant infections, and 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

[1] Malware inserted into the software or hardware during development or manufacture.

that can be used as a foundation for development of a comprehensive standard for botnet operation.

The future of bot army technology will consist of improvements in attack technology; bots will become increasingly stealthy, mobile, and sophisticated; and bot armies will increase in size and capability. The current typical bot master is not highly skilled and lacks sophisticated programming skills; however, we should expect this bot master skill profile to change over time and that, in a few years, bot masters will be more skilled and their bot armies will be correspondingly more effective, powerful, and difficult to detect. Defenses will also improve, but the offense will retain its advantage over the defense over the next five years. It is clear that relying on detecting bot communication is not a viable long-term strategy for defeating them. All characteristics of bots and their communication should be used in the hunt for bot armies; hence, host performance assessment tools and network traffic monitors are needed in order to determine if a bot army has infested a network. Nevertheless, removal of C2 nodes for the bot army will remain the best approach for defeating and disarming bot armies.

Unfortunately, we have yet to experience the worst cyber activities that bot armies can commit. Currently, entry to systems by a bot is easy [49], more than 90% of attacks succeed because they are opportunistic and piggyback upon normal activity; most bots do not penetrate by targeting and "attacking" a specific computer. Few penetrations are detected while they are underway, most are detected only after the bots are implanted and the bot master makes a mistake that leads to detection of the bot. Furthermore, once a bot has infiltrated a system it is easy for it to gain additional privileges, which in turn allows the bot to gain access to ever more valuable data during the course of its infiltration. Regrettably, most defenses concentrate on protecting systems; whereas, the most valuable component of an organization is its data; defenses need to concentrate on protecting data as well as protecting against infiltration by bots. Furthermore, since bots can piggyback upon web viruses; a powerful entry vector for bots is still available, especially since web viruses have demonstrated the ability to place themselves into databases directly from the web page (allowing them to hide among dynamic content and placing them in a location that is not examined by traditional anti-virus tools.)

Our future work in the area of botnet operation modeling and simulation will concentrate on refining the model that we proposed, especially determining as basis for computing the probability of detection of a bot army infection and the probability of re-infection of a computer by a given bot after the computer has been cleansed of the bot. In addition to improving and refining the model, 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. Our goal will be to build a suite of models able to portray the operation of different classes of botnets. A further modeling question that we want to address is the development of a model for likelihood of infection that accounts more fully for the variables involved in bot army propagation, such as the likelihood that a software-based infection will hide its software off the hard drive so that the computer can be easily re-infected after cleansing and power-up. We also plan to begin to address the standards issues, especially the development of standards to simulate bot communications and standards for the simulation of bot activity within an infected computer. A final research area that we plan to address is the development of technologies to provide and to model bot immunity, which would extend the work of Tarakanov into the field 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. (2004) "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,

Bot Army

48. Evron, Gadi. (2008) "Battling Botnets and Online Mobs," *Georgetown Journal of International Affairs*, Winter/Spring, pp. 121-126.
49. Swatit (2006) "Bots, Drones, Zombies, Worms and Other Things That Go Bump In The Night.," http://swatit.org/bots/.
50. Tarakanov, A. O. (2008) "Immunocomputing for Intelligent Intrusion Detection," *IEEE Computational Intelligence Magazine*, May, pp. 22-30.
51. Dasgupta, D. Ed. (1999) *Artificial Immune Systems and Their Applications*. Springer, Berlin, 1999.
52. de Castro, L.N. and Timmis, J. (2002) *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, London.
53. Tarakanov, A.O.; Skormin, V.A.; and Sokolova, S.P. (2003) *Immunocomputing: Principles and Applications*. Springer, New York.
54. Tarakanov, A.O. and Nicosia, G. (2007) "Foundations of Immunocomputing", *Proc. 1st IEEE Symposium on Foundations of Computational Intelligence (FOCI'07)*, Honolulu, Hawaii, pp. 503–508.
55. Dasgupta, D. and Gonzalez, F. (2005) "Artificial Immune Systems In Intrusion Detection," *Enhancing Computer Security with Smart Technology*, V. Rao Vemuri Ed., Auerbach Publications, pp. 165–208.
56. Staniford, S. ; Paxson, V. ; and Weaver, N. (2002) "How to 0wn the Internet in Your Spare Time," *Proceedings of the 11th USENIX Security Symposium (Security '02)*.
57. Su, J.; Miklas, A. G.; Chan, K. K. W.; Po, K.; Akhavan, A.; Saroiu, S; d. Lara, E.; and Goel, A. (2006) "A Preliminary Investigation of Worm Infections In A Bluetooth Environment," *Proceedings of WORM'06*.
58. Thommes, R.W. and Coates, M.J. (2006) "Epidemiological Modelling of Peer-to-Peer Viruses and Pollution," *Proceedings of IEEE Infocom'06*.
59. Yan, G.; Cuellar, L.; Eidenbenz, S.; Flores, H. D.; Hengartner, N.; and Vu, V. (2007) "Bluetooth Worm Propagation: Mobility Pattern Matters!," *Proceedings of ACM ASIACCS'07*, March.
60. Yan, G. and Eidenbenz, S. (2006) "Bluetooth Worms: Models, Dynamics, and Defense Implications," *Proceedings of ACSAC'06*, December.
61. Yan, G. and Eidenbenz, S. (2007) "Modeling Propagation Dynamics of Bluetooth Worms," *Proceedings of ICDCS'07*, Toronto, Canada, June.
62. Zou, C. C.; Gong, W.; and Towsley, D. (2002) "Code Red Worm Propagation Modeling and Analysis," *Proceedings ACM CCS'02*, October.
63. Zou, C. C.; Towsley, D.; and Gong, W. (2003) "Email Worm Modeling And Defense," *Proceedings of the 13th International Conference on Computer Communications and Networks*.
64. Schneider, V. (1989) "Approximations for the Halstead Software Science Software Error Rate and Project Effort Estimators," *ACM SIGMETRICS Performance Evaluation Review*, vol. 16, no. 2-4, February, pp. 22-29.
65. Whitaker, J. (2002-2006) Personal communication.
66. Hoagland, J.; Ramzan, Z.; and Satish, S. (2008) "Bot Networks," in *Crimeware: Understanding New Attacks and Defenses*, Jakobsson, M. and Ramzan, Z. eds., Addison-Wesley, pp. 183-224.
67. Colizza, V.; Hu, S.; Myers, S.; Stamm, S.; Tsow, A.; and Vespignani, A. (2008) "Crimeware in Firmware," in *Crimeware: Understanding New Attacks and Defenses*, Jakobsson, M. and Ramzan, Z. eds., Addison-Wesley, pp. 103-127.