Exploring Botnet Evolution via Multidimensional Models and Visualisation

William Dash[1] and Matthew J. Craven[2]

[1] University of Bristol, Bristol BS8 1TH, UK.
williamdash@dashsw.co.uk

[2] Centre for Mathematical Sciences, Plymouth University, Plymouth PL4 8AA, UK.
matthew.craven@plymouth.ac.uk

Abstract. A botnet is a program designed to perform a specific task using multiple computers connected in a network. In this paper we will focus on botnets being used to distribute malicious programs. In the real world, botnets have been shown to exhibit more aggressive and sophisticated behaviour than traditional malware. Botnets are used to infect computer networks and hence their success depends on the properties of the networks. We observe the behaviour of mathematical models used to describe botnets when botnet parameters are varied to understand if such variation is beneficial to their spread. We also introduce novel models for depicting botnet behaviour using master equations. These models, unlike previous ones, address nodes of distinct categories in a network as a sequence of probability distributions rather than a value at each time interval. We also contribute visualisations for these models. This paper is a substantial expansion of unpublished work the first author performed while on a Nuffield student research placement, with the second author the project supervisor.

Keywords: Botnet, differential equation, master equation, visualisation, complex systems security, security in P2P (peer to peer) systems.

1 Introduction

Despite the primary use of a botnet being a means of distributing malicious software, they were initially created to distribute computationally intensive tasks among a variety of devices, as in parallel processing. However, due to their ability to control large amounts of computer resources they have since become desirable in the distribution of malicious software. This makes botnets good for deploying software requiring large amounts of resources to be effective; an example of this is Distributed Denial of Service (DDoS).

As a result of the diverse capabilities of botnets and subtle, but aggressive, virus distribution they pose a large threat to modern cybersecurity. An example of such a case is the TDL-4 botnet [9]. As a rootkit, this modifies the master boot record of each infected node so that it is always loaded at startup. Such behaviour makes the botnet more difficult to eradicate than previous TDL generations. Another example is the Carna botnet [2]. Although Carna was used to collect data and not intended for malicious activities it became widespread. Comprised of around 420,000 nodes, it collected worldwide data regarding the geographical distribution of the usage of (it was claimed, all) IPv4 addresses.

Thus, as expressed in [1], modelling botnet behaviour and spread is crucial to preserve security. One method of modelling such spread as a threat is by using epidemiological models [1]. However, such models are often only comprised of the susceptible, infected, and recovered states (*SIR models*), due to the nature of diseases they are used to model. In contrast, botnets have a distinct lifecycle, which is described in [1] as follows: the payload (also know as the *worm*) is constructed by the botmaster and distributed across a network which proceeds to infect the maximum number of nodes possible. Each infected node receives commands from the command and control server of the payload and thus may begin or stop performing malicious tasks assigned by the botmaster at any time. After the malicious activities of a node are discovered by its true user, these activities may be terminated and the node has “recovered”. However, depending on the botnet, nodes may be re-infected after recovery. Botnet size is often measured by the number of constituent nodes, from thousands to millions [7, 8].

In terms of effectiveness and efficiency, several authors have proposed botnet modelling techniques. The work of [10] uses the CodeRed1v2 worm as a case study to model stochastic botnet behaviour. The work of [4] considers time zones in global botnet behaviour, and [12] considers interaction and co-operation between two botnets (and thus, many). Finally [11] considers statistical spread models of network subgraphs showing, by a search for subgraph isomorphisms in networks undergoing simulated network attacks, whether such subgraphs are likely caused by an initial botnet outbreak. Our work shall not consider these factors, as we wish our approach to be straightforward, focusing on interactions of nodes in distinct states within networks. The objective of this work is to firstly observe the behaviour of existing botnet models and then combine them with alternative epidemiological models. From this we derive more accurate and interesting probabilistic models describing botnet behaviour. Throughout, we detail model simulations and visualisations of botnet model results.

1.1 Modelling Contributions of this Paper

3D and 6D probabilistic models: Section 3 introduces a 3D probabilistic model based on the system of ODEs of [1]. Section 4 extends this, adding additional node states. This probabilistic approach is practical as it only considers integer numbers of items in each model state (often difficult with a - continuous - ODE approach). The approach also provides more information than models which produce fixed values for the number of nodes in each state on each iteration. This approach may also identify realistic worst and best case scenarios for any particular population/setting rather than just an expected value.

Extension to 7D model and applications to GSM networks: In this paper Section 5 extends the work of Section 4 to a 7D botnet model in order to allow multiple worms to be distributed simultaneously by a botnet. It then goes on to show how our work, despite some limiting assumptions inherent in any model, may be applied to botnets propagating through GSM (Global System for Mobile Communications) networks based on the work from [6]. This approach of allowing a botnet to be able to distribute multiple worms in a population is advantageous as it has the models derived in Sections 3–4 as a special case. Therefore this model simulates a more diverse range of scenarios than those models.

2 Using Sets of First Order ODEs

To begin, we review the ODE model proposed in [1], including suggested extensions/modifications, as a modified epidemiological model applied to botnets.

2.1 Model Setup

The botnet model proposed by [1] was based upon sets of ODEs developed for epidemiology, and comprises of the following classifications for each node:

S: Nodes vulnerable to infection by the worm being transferred by the botnet;
S_d: Nodes susceptible to the worm, but which are disconnected from the network;
I: Nodes infected by the worm and are able to infect other nodes, but show no signs of infection;
I_d: Infected nodes which are disconnected from the network;
V: Infected nodes which are executing the malicious task provided by the worm;
V_d: Infected nodes which previously executed malicious tasks but are disconnected from the network;
R: Previously infected nodes that have now permanently recovered.

The model assumes eleven connections between the possible nodes states. These are: a susceptible node becoming infected, disconnecting from the network and possibly reconnecting; similarly, a dormant infected node becomes active, disconnects or reconnects; an active infected node becomes dormant, temporarily or permanently recovers, or disconnects; a disconnected active infected node becomes dormant. As expressed in [1], this extended model is suited to botnets that transmit worms which mutate upon transmission or that contain multiple worms. This is highlighted by the transition from state V to S. This model proposed in [1] has the following parameters:

N, μ: Total population size, switching rate between hidden and active
b: Worm transmission rate
g, ρ: Permanent, temporary recovery rate
p: Apportioning coefficient of infected (dormant) nodes
σ: Switching rate between online and offline states
q: Apportioning coefficient of nodes connected to the network

The work of [1] provided a flow diagram, describing the transitions between each node in the configuration described above. We omit this in the present work; however, we may describe this model as a system of ODEs:

$$
\begin{aligned}
\frac{dS}{dt} &= \frac{-b(I(t)+V(t))}{N}S(t) + \rho V(t) + \frac{\sigma}{1-q}S_d(t) - \frac{\sigma}{q}S(t) \quad ; \quad \frac{dR}{dt} = gV(t) \\
\frac{dI}{dt} &= \frac{b(I(t)+V(t))}{N}S(t) + \frac{\mu}{p}V(t) - \frac{\mu}{1-p}I(t) + \frac{\sigma}{1-q}I_d(t) - \frac{\sigma}{q}I(t) \qquad (1) \\
\frac{dS_d}{dt} &= \frac{\sigma}{q}S(t) - \frac{\sigma}{1-q}S_d(t) \quad ; \quad \frac{dI_d}{dt} = \frac{\sigma}{q}I(t) - \frac{\sigma}{1-q}I_d(t) + \frac{\sigma}{1-q}V_d(t) \\
\frac{dV}{dt} &= \frac{\mu}{1-p}I(t) - \left(\frac{\mu}{p} + g + \rho + \frac{\sigma}{q}\right)V(t) \quad ; \frac{dV_d}{dt} = \frac{\sigma}{q}V(t) - \frac{\sigma}{1-q}V_d(t)
\end{aligned}
$$

Although the input parameters are given above, we must consider the initial numbers of nodes in all classes. The work of [1] showed that $I(0) > 0$ (i.e., there are nodes able to infect the network). Further, from parameter experimentation, it is crucial that $S(0) \geq 0.9N$ in order for the botnet to be able to grow to a sufficiently large size. Also, [1] showed the need for $V(0) = R(0) = 0$, to allow us to view the entire life cycle of the botnet from its initial network penetration. This model assumes that all nodes in the population are online and connected to the network being infected at the start of the simulation, allowing the botnet to initially enter the population and ensuring it does not necessarily die out immediately. Hence the starting values $S_d(0) = 0$, $I_d(0) = 0$, $V_d(0) = 0$ have also been used in our model simulations. The simulation was coded in Python, with the GNUPlot package used to visualise the results in the next subsection.

2.2 Visualisation

In Figures 1–2 the green line represents the proportion of class S nodes, orange class S_d, dark blue class I, yellow class I_d, light blue class V, brown class V_d and red class R. A simulation output is shown in Figure 1. In this, the model used example parameters: transmission rate $b = 0.5$, recovery rate $g = 0.25$, hidden-active switching rate $\mu = 0.1$, apportioning coefficient $p = 0.1$, temporary recovery rate $\rho = 0.01$, proportion of online nodes $q = 0.9$, and online-offline switching rate $\sigma = 0.09$.

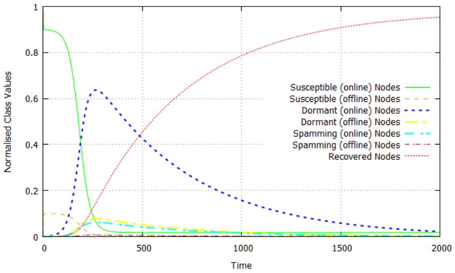


Fig. 1. Simulation results of the current model for 2000 iterations where $N = 100$.

Figure 1 shows a distinct peak where around 64% of the population are dormant infected nodes (class I). These nodes then tend to recover at a shallower rate than their infection. This is a combined result of the low values of parameters μ, ρ and g compared to b and the assumption that nodes can only recover once they have moved into class V. In addition, a single region of growth, followed by decay, for class I nodes is shown. This non-repetitive behaviour is caused by the small values of parameters ρ and σ compared to g, which causes most nodes to move from class V to class R, rather than iterating through any previous states.

To observe scenarios that the current model depicts we begin by varying some of the more influential parameters of this simulation. However, as stated in [1], it is clear that model behaviour is independent of population size, and so the most interesting variables concern the transition speeds (b, g, ρ, p and q) between each class. The functions used in our simulations will be examples only. At this stage, we focus on the parameters concerning the initial infection and recovery of nodes in the model. Thus we assume the botnet has a constant level of aggression (p is constant) and the variables b, g, ρ, q are functions of time, t.

First, we vary the parameter b using fractal Brownian motion (fBm) initialised with value noise with time as seed. This definition is more suitable than a simple constant as the propagation of botnets across a network depends on a number of clearly variable factors (e.g., network traffic). We initialise the fBm with value noise as opposed to (the more common) Perlin noise as we require a 1D noise function ($b(t)$ has one parameter). Also, to produce results more applicable to real-life botnets we, as suggested in [1], consider user response to the presence of a botnet. We model this by having a different proportion, q_v, of online class V nodes and assume the infection of a node is only detected when the node is active. As the number of class V nodes increases so do the number of users being informed, spreading the word and informing a given (for simplicity, fixed) number of other infected users how to recover nodes.

These people then recover nodes and exhibit the same behaviour as their predecessors. This shows, as in standard population growth models, exponential growth in the number of people recovering nodes. Assuming each user corresponds to a single node and that the first user attempt to recover nodes is by removal from the network, the proportion of infected (active) nodes that are not online is exponential in V_n (the normalised number of class V nodes) and the proportion of online node users, q_v, decays exponentially. We use the function

$$q_v(t) = \exp\left(-100V(t)/N\right) \tag{2}$$

to simulate this relationship. Figure 2a illustrates a simulation using (2). We extend this approach by assuming that progressively fewer users are able to temporarily and permanently repair (recover) nodes respectively. For this we use exponential functions to describe both relationships, meaning they exhibit similar behaviour to q_v. However, in order to represent the difference in difficulty of temporarily and permanently recovering a node we give each function distinct coefficients. We formalise this by the (example) equations

$$\rho(t) = \exp\left(V(t)/N - 1\right) \text{ and } g(t) = 0.5\exp\left(V(t)/N - 1\right), \tag{3}$$

and implement equations (2)–(3) in Figure 2b. Both results indicate a decrease in the peak number of infected nodes. Observe that Figure 2a also contains fewer class S nodes (in green) when in stable equilibrium than Figure 2b. This means the ability to recover an infected node, even temporarily, is more advantageous in reducing overall infection of the network than just disconnecting the node.

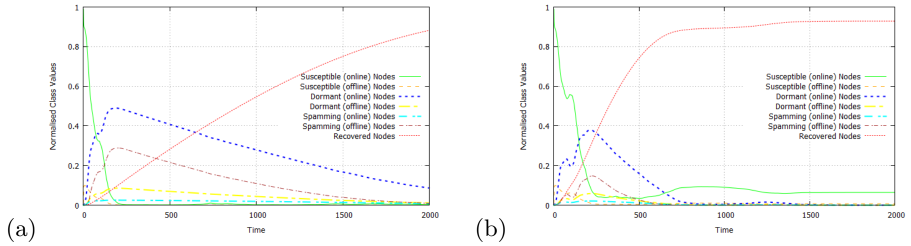


Fig. 2. The result of 2000 model iterations with b being fBm using value noise. On the left equation (2) is used, and on the right equations (2)–(3) are used.

Of course, this model may be considered constrained by its fixed population size as it assumes no node is destroyed and that the botnet does not expand to other networks or populations. The model is also restricted by its lack of consideration of individual nodes; treating the total population as a single entity and consequently allowing for non-integer numbers of nodes in all classes.

3 A 3D Botnet Model

We now apply a simplified version of the Section 2 model to multidimensional probabilistic modelling.

3.1 Multidimensional Modelling

So far we have used ODEs from a single type of biological model. However, producing a more realistic and accurate botnet behaviour model requires alternative approaches. One such approach, [5], constructs a multidimensional probability distribution of all possible combinations of node states a population may contain at any time. However, the model of [5], in the form of a master equation, was designed to consider epidemics conforming to a standard SIR progression, meaning each node is in one of those three states. Hence this model is not directly applicable to botnets due to its limited number of states. However, by extending the number of model states, and so the number of dimensions in the probability distribution, we produce a distinct type of botnet model to that of Section 2.

3.2 Model Explanation

This model is expressed and then simulated using a master equation. Under master equation notation, the number of equations to be evaluated is a function of population size, N. That is, a master equation acts as a generalised equation for each combination of S, I and V values that may exist within a given population. In order to produce a suitable master equation, we must consider a reduced set of possible transitions that occur between node states defined in the model of Section 2. We also allow for a variable-sized population. To do so, we use the following assumptions analogous to [5]: each time a new node is added to the population it is susceptible to the botnet, and each node may die or be removed from the population regardless of type. Using these additional transitions, shown in the flow diagram (Figure 3), the following events may occur within the model. A node may be added to the population, a susceptible node becomes infected (dormant), an infected (dormant) node becomes active, an infected (active) node becomes dormant, an infected (active) node recovers from the infection; or finally, a node in either the S, I or V category dies.

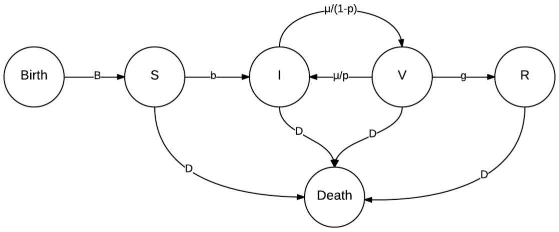


Fig. 3. Flow diagram showing the inter-class transfer rates for the current model.

Expressing transitions as equations requires two new parameters from [5]. The first is the rate, B, at which new population nodes are added, and the second is the rate, D, at which nodes die or are removed. Each such transition changes the probability that each S, I, V combination occurs in the population, and so the master equation includes terms describing each event. In reality if any such transition occurs to a node combination then a new node combination is produced, represented in our model by reduction of the likelihood of the original combination occurring and increase of the likelihood of the resulting combination occurring. Thus each S, I, V combination (Table 1) has pairs of terms, one representing the source decreasing the probability and the other its source of increase. Each term has a coefficient of the likelihood of its parent S, I, V combination occurring. Thus, when the likelihoods of new S, I, V combinations are calculated, the probability of each parent combination is considered (each iteration of the model depends upon the last). Each parent combination that results in a new combination, along with the causal event, are listed below.

A node is added to the population	S-1, I, V
A susceptible node becomes infected (dormant)	S+1, I-1, V
An infected (dormant) node becomes active	S, I+1, V-1
An infected (active) node becomes dormant	S, I-1, V+1
An infected (active) node recovers from the infection	S, I, V+1
A susceptible node is removed from the population	S+1, I, V
An infected (dormant) node is removed from the population	S, I+1, V
An infected (active) node is removed from the population	S, I, V+1

Table 1. Relative S, I and V combination of each event corresponding to a source of increase in probability for each unique combination of node types in a population.

The node combinations in Table 1 represent the original combination of nodes that each transition occurs to in order to produce the node combination: S, I, V. For example, if a node was added to the population, then the number of nodes would increase by 1. In addition, every time a new node is added it is added to the susceptible category. Thus the number of nodes in class S will increase by 1 and the other classes would remain unaffected. In order to produce the final combination of S, I, V we subtract 1 from the S term. This is so that we compensate for the addition of 1 to S caused by this transition, hence producing the original combination $S-1$, I, V. Using the flow rates from Section 2 and the new parameters from Section 3 we produce a master equation to describe the transitions (the inputs and outputs for each S, I, V combination) shown below.

$$\begin{aligned}\frac{dP_{S,I,V}}{dt} = &-\left(\frac{bS(I+V)}{N} + \frac{\mu}{1-p}I + \frac{\mu}{p}V + gV + BN + DS + DI + DV\right)P_{S,I,V}\\ &+\frac{b(S+1)(I-1+V)}{N}P_{S+1,I-1,V} + \frac{\mu}{1-p}(I+1)P_{S,I+1,V-1}\\ &+\frac{\mu}{p}(V+1)P_{S,I-1,V+1} + g(V+1)P_{S,I,V+1} + B(N-1)P_{S-1,I,V}\\ &+D(S+1)P_{S+1,I,V} + D(I+1)P_{S,I+1,V} + D(V+1)P_{S,I,V+1}\end{aligned} \tag{4}$$

When simulating the current model we set $P_{45,5,0}$ to one and the probability of all other S, I, and V combinations to zero. This was so that the number of infected nodes in the population is sufficiently large that the botnet is able to grow to a reasonable size and that the constraints explained in Section 2 are satisfied. In addition, as the current model allows for dynamically sized populations, the additional constraints: $S + I + V \leq N$ and $S,\ I,\ V \geq 0$ will be applied to all corresponding simulations. This is so that the population size in the simulation is bounded above and below, making it easier to simulate. Therefore the parameter N will now denote the maximum population size in all further simulations. A simulation output of the current model is shown in Figures 4a–4c, with transmission rate $b = 0.5$, recovery rate $g = 0.1$, hidden-active switching rate $\mu = 0.1$, apportioning coefficient $p = 0.5$, population increase rate $B = 0.0005$ and population decrease rate $D = 0.0005$.

3.3 Visualisation

The current model produces a 3D probability distribution, and would require a 3D output device in order to display all data produced in its raw form. We thus reduce the 3D distribution by effectively removing V combinations from the data. This is done by summing the probabilities assigned to positions with the same S and I combination but different V combinations. We repeat this process with the remaining combinations of node classes to produce three separate 2D probability distributions (effectively projections). Results are shown for susceptible and dormant infected nodes (Figure 4a), susceptible and active infected nodes (Figure 4b), and dormant infected and active infected nodes (Figure 4c).

Although the distributions of Figures 4a–4c contain a significant number of entries with negligible probability it does not contain any zero elements. In

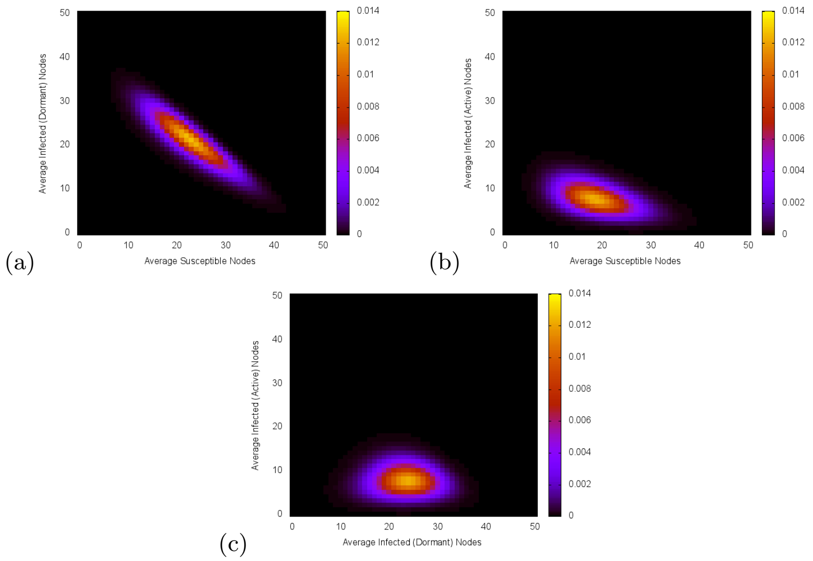


Fig. 4. The resulting distributions of the current model on iteration 2670 where $N = 50$ for each S and I combination (top left), each S and V combination (top right), and each I and V combination (bottom). The heat map corresponds to the probability of a given combination occurring in the population.

addition, each of the node types in this distribution has a distinct concentration. Specifically, in Figures 4a and 4b the susceptible nodes have a standard deviation of approximately 5.21. In Figures 4a and 4c the infected (dormant) nodes have a standard deviation of around 4.61. Also, the infected (active) nodes in Figures 4b and 4c have a standard deviation of around 2.71. The higher variation of nodes in class S, in comparison to classes I and V, is likely to be a result of the initial conditions and population increase rate used in the simulation. This model shows a very different approach to botnet modelling than the previous model, as it is probabilistic. Consequently, modelling real life botnets with this model may require more discipline. In addition, the current model is somewhat restricted by its simplicity and may be extended to include more node states.

4 A 6D Model

We now extend the model from Section 3 to allow nodes to temporarily connect and disconnect from the network/population being targeted by the botnet.

4.1 Model Explanation

The new model implements the more diverse behaviour exhibited by the model from Section 2 by making use of additional classifications of nodes. However, in order to do so we must change the number of dimensions of the probability distribution that we produce (since each node type is represented across an axis perpendicular to all others). As a result we produce a 6D master equation as there are six different node states being modelled. Making identical assumptions about the behaviour of offline states to the model of Section 2, we describe how the states in the current model transition to and from one another (Figure 5). Just as when deriving the model of Section 3 we need to consider all possible events or transitions that may occur within the model and their corresponding relative S, S_d, I, I_d, V and V_d combinations (Table 2).

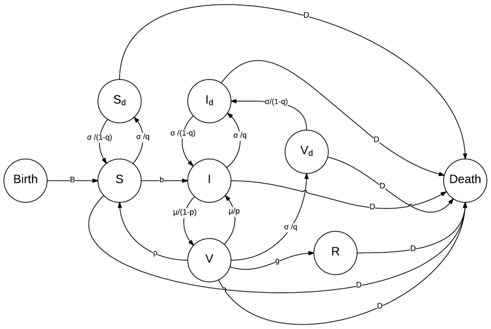


Fig. 5. Flow diagram describing transitions between node states for the current model.

Using the flow rates for each of the events (defined in Sections 2–3) we formalise the model as a 6D master equation (5), describing the inputs and outputs for each state combination. In this equation we use the notation $P_{S+1,S_d-1,*}$, for example, to mean that the states S and S_d have changed and all other states stay the same, and $P_* = P_{S,S_d,I,I_d,V,V_d}$.

$$
\begin{aligned}
\frac{dP_*}{dt} = &-\begin{pmatrix} \frac{\sigma}{1-q}S_d + \frac{\sigma}{q}S + \frac{bS(I+V)}{N} + \frac{\sigma}{1-q}I_d + \frac{\sigma}{q}I \\ +\frac{\mu}{1-p}I + \frac{\mu}{p}V + \rho V + \frac{\sigma}{q}V + \frac{\sigma}{1-q}V_d + gV \\ +DS + DS_d + DI + DI_d + DV + DV_d + BN \end{pmatrix} P_* \\
&+\frac{\sigma}{1-q}(S_d+1)P_{S-1,S_d+1,*} + \frac{\sigma}{q}(S+1)P_{S+1,S_d-1,*} \\
&+\frac{b(S+1)(I-1+V)}{N}P_{S+1,I-1,*} + \frac{\sigma}{1-q}(I_d+1)P_{I-1,I_d+1,*} \\
&+\frac{\sigma}{q}(I+1)P_{I+1,I_d-1,*} + \frac{\mu}{1-p}(I+1)P_{I+1,V-1,*} + \frac{\mu}{p}(V+1)P_{I-1,V+1,*} \\
&+\rho(V+1)P_{S-1,V+1,*} + \frac{\sigma}{q}(V+1)P_{V+1,V_d-1,*} + \frac{\sigma(V_d+1)}{1-q}P_{I_d-1,V_d+1,*} \\
&+g(V+1)P_{V+1,*} + D(S+1)P_{S+1,*} + D(S_d+1)P_{S_d+1,*} \\
&+D(I+1)P_{I+1,*} + D(I_d+1)P_{I_d+1,*} + D(V+1)P_{V+1,*} \\
&+D(V_d+1)P_{V_d+1,*} + B(N-1)P_{S-1,*}
\end{aligned}
\tag{5}
$$

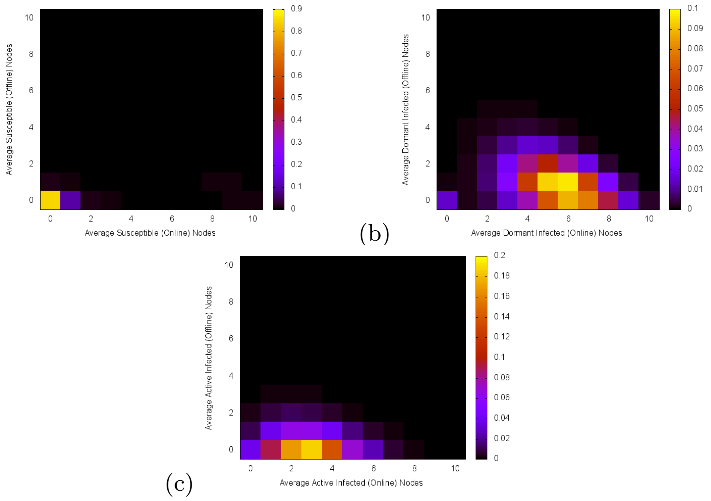
Using previous constraints gives $P_{9,0,1,0,0,0}$=1 and a zero probability of all other combinations initally. As in Section 3 we apply $S+S_d+I+I_d+V+V_d \leq N$ and S, S_d, I, I_d, V, $V_d \geq 0$ to bound the numbers of nodes in each class. A simulation output is shown in Figure 6a, with rates $b = 0.5$, $g = 0.1$, $\mu = 0.1$, $B = 0.0005$, $D = 0.0005$, $\rho = 0.01$ and $\sigma = 0.09$ (c.f. Section 2.1). The apportioning coefficient was $p = 0.5$ and proportion of online nodes was $q = 0.9$.

Node added to population	$S-1$, S_d, I, I_d, V, V_d
Susceptible node switches to being in an offline state	$S+1$, S_d-1, I, I_d, V, V_d
Offline susceptible node switches to online	$S-1$, S_d+1, I, I_d, V, V_d
Susceptible node becomes infected (dormant)	$S+1$, S_d, $I-1$, I_d, V, V_d
Offline infected (dormant) node switches to being online	S, S_d, $I-1$, I_d+1, V, V_d
Online infected (dormant) node switches to being offline	S, S_d, $I+1$, I_d-1, V, V_d
Infected (dormant) node becomes active	S, S_d, $I+1$, I_d, $V-1$, V_d
Infected (active) node becomes dormant	S, S_d, $I-1$, I_d, $V+1$, V_d
Infected (active) node temp. recovers from botnet payload	$S-1$, S_d, I, I_d, $V+1$, V_d
Infected (active) node switches to being offline	S, S_d, I, I_d, $V+1$, V_d-1
Offline inf. (active) node becomes offline inf. (dormant)	S, S_d, I, I_d-1, V, V_d+1
Infected (active) node permanently recovers from infection	S, S_d, I, I_d, V+1, V_d
Susceptible node is removed from the population	$S+1$, S_d, I, I_d, V, V_d
Offline susceptible node removed from population	S, S_d+1, I, I_d, V, V_d
Infected (dormant) node removed from population	S, S_d, $I+1$, I_d, V, V_d
Offline infected (dormant) node removed from population	S, S_d, I, I_d+1, V, V_d
Infected (active) node removed from population	S, S_d, I, I_d, $V+1$, V_d
Offline infected (active) node removed from population	S, S_d, I, I_d, V, V_d+1

Table 2. Relative S, S_d, I, I_d, V and V_d combinations for each event corresponding to a source of increase in probability of each unique combination of nodes in a population.

4.2 Visualisation

To visualise model simulations we found it best to view them as a combination of two-dimensional distributions, as in Section 3. However, this gives $\binom{6}{2} = 15$ distinct 2D distributions. So we decided to only view the distributions containing an online and offline state pair, producing only three images, but at the same time allowing us to view each node state. Figures 6a–6c show these visualisations.



(a)

(b)

(c)

Fig. 6. The probability distribution on iteration 2670 of the current model ($N = 10$) of each S and S_d combination (top left), each I and I_d combination (top right), and each V and V_d combination (bottom).

Here the number of class I nodes has a larger standard deviation (1.71) than the number of class V nodes (1.47). But, unlike the results of Section 3, the number of class S and S_d nodes have much smaller standard deviations than any other classes (1.13 and 0.22 respectively). The extended number of node states makes this model more general than that of Section 3. The current model also uses a discrete approach so that only integer numbers of nodes can exist in certain states (as per Section 3). The combination of these two characteristics is absent in the other models derived in this paper, meaning the current model may more realistically model botnet behaviour. However, the data produced may be more difficult to visualise and interpret than previous model data.

5 Extensions

Although the work of Sections 3 and 4 introduced new models, they only focussed on the behaviour of a botnet transferring a single worm propagating through a network. This section extends the model of the last section so that it will be capable of modelling botnets that transmit multiple worms through a population.

In order to model multiple worms in a network there are several assumptions that have been made on how the botnet assigns worms to infected nodes. So far we have considered each node in the population to be only distinguishable by their state within each model; hence, each I state node is indistinguishable to the Botmaster. In this extension we assume that all infected nodes will be assigned a worm independently of one another. In addition, this extension also preserves the previous assumption that only class I nodes may be told to execute malicious activities by the C & C server. So, in this model each infected (active) state will be unable to directly transition to one another. As a result of the above assumptions, this model will not contain state V as used in previous models. Instead we use V_1, V_2, $\ldots$, V_n to denote all the possible infected node states considered in this extended model. As a result the parameters p, ρ and g are no longer used and instead each V_i state has its own associated p_i, ρ_i and g_i values. Using the new states and parameters, we may construct Figure 7 (and so a 7D master equation, omitted) for the special case $n = 2$ of this extended model.

This model demonstrates one of the many ways in which the models from Sections 3–4 may be generalised to produce more diverse botnet models. However, the models introduced in this paper are only suitable for modelling perfect populations in which network parameters remain constant. In the next subsection we detail an implementation of the above extended model to depict a botnet propagating across a GSM network.

5.1 Simulation setup

We now use this model to simulate a botnet that operates on mobile phones. In particular this botnet will use multiple worms, will be transferred over WiFi and has the objective of disruption of the GSM network to which the infected phones are connected. The botnet uses the strategy described in [6], which is to send excessive numbers of requests to the home location register (HLR) in

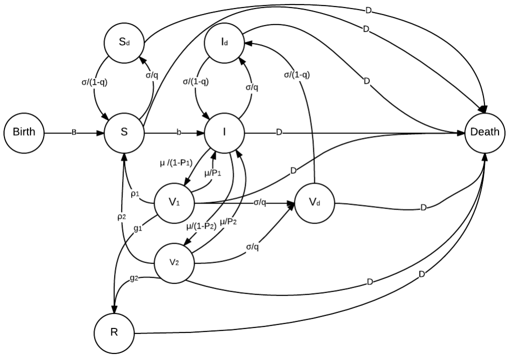


Fig. 7. Flow diagram describing transitions between node states for the current extended model in the special case when $n = 2$.

the network. The HLR within a GSM network is a database of the details of everyone authorised to use the network. All requests within the network need to interact with the HLR in order to be processed, making it a clear attack target.

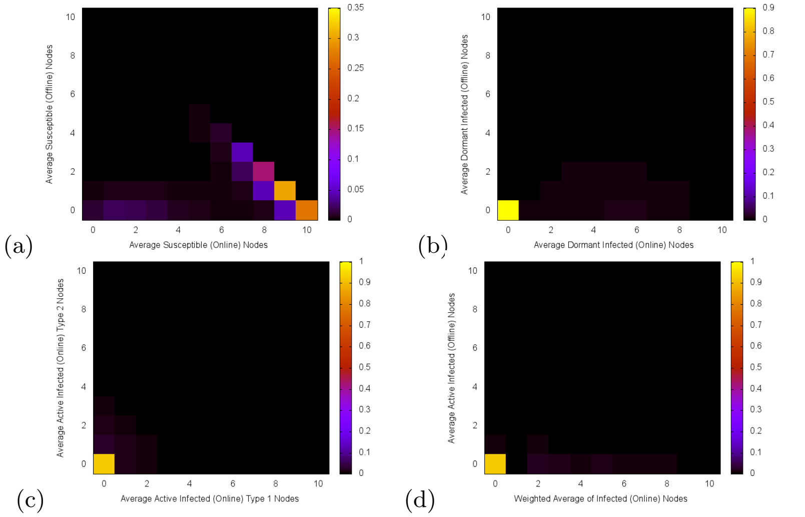
Here, the V_1 state represents a worm that excessively issues `insert_call_forwarding` requests to the HLR. This was chosen as, according to [6], this is the most effective request for attacking an HLR. However [6] also shows this request has a low occurrence in a typical GSM network, making any infected devices easily identifiable. To complement this, the V_2 state represents a worm that excessively issues `update_location` requests to the HLR. The results of [6] indicate this request is less strenuous on the HLR. However, [6] also indicates that in a typical GSM network the number of `insert_call_forwarding` requests issued is approximately one seventh of the number of `update_location` requests. So, we reasonably assume that seven times more malicious devices issuing `insert_call_forwarding` requests are identified and recovered (temporarily or permanently), as they are more conspicuous, than devices issuing `update_location` requests ($\rho_1 = 7\rho_2$ and $g_1 = 7g_2$). We also assume it is easier to temporarily recover an infected device than to recover it permanently (giving constraints $\rho_1 > g_1$ and $\rho_2 > g_2$). As a result, the values $\rho_1 = 0.7$, $\rho_2 = 0.1$, $g_1 = 0.35$ and $g_2 = 0.05$ were used.

The objective of this botnet is to exceed maximum total HLR throughput on the targeted network. Here this is equivalent to maximising the number of nodes in classes V_1 and V_2. However, the commands being issued and hence the strain put on the HLR by nodes in classes V_1 and V_2 are different. Interpreting Figure 5 in [6], nodes in class V_1 are approximately 1.5 times more strenuous on the HLR than nodes in class V_2. Therefore the objective of the botnet is to maximise $1.5V_1+V_2$ on each iteration of the simulation. Hence a suitable objective function of this botnet is $V_{\text{obj}} = 1.5V_1 + V_2$. This function will be used to produced a probability distribution from the results of the current model in the visualisation subsection. By Figure 5 of [6], it is also reasonable to assume that in this case

$p_1 = 1.5p_2$. In this example we have $p_1 = 0.3$ and $p_2 = 0.2$. All the remaining parameters are identical to those of Section 4. For the simulation, using the same constraints as previously gives the starting condition $P_{9,0,1,0,0,0,0} = 1$ and the constraint $S + S_d + I + I_d + V_1 + V_2 + V_d \leq N$ with non-negative summands.

5.2 Visualisation

As in Section 4, we display several of the 2D projections of the resulting 7D distribution. We produce plots of S with S_d and I with I_d. We also present a plot of V_{obj} with V_d, to allow comparison to the results of the Section 4 model, and a plot of V_1 with V_2 to allow comparison of both worms being transferred. Figures 8a–8d show outputs for transmission rate $b = 0.5$, recovery rates $g_1 = 0.35$ and $g_2 = 0.05$, hidden-active switch rate $\mu = 0.1$, population increase/decrease rate $B = 0.0005$ ($D = 0.0005$), apportioning coefficients $p_1 = 0.3$ and $p_2 = 0.2$, temporary recovery rates $\rho_1 = 0.7$ and $\rho_2 = 0.1$, proportion of online nodes $q = 0.9$ and offline-online switch rate $\sigma = 0.09$.

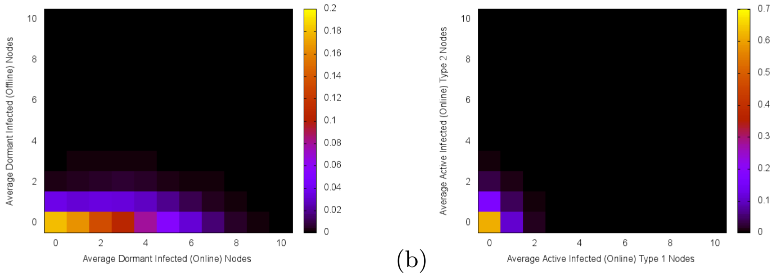


(a) (b) (c) (d)

Fig. 8. The probability distributions on iteration 2670 of the extended model ($N = 10$) of each S and S_d combination (top left), each I and I_d combination (top right), each V_1 and V_2 combination (bottom left) and each V_obj and V_d combination (bottom right).

These results show significantly larger variation in the number of class S and S_d nodes, with standard deviations 2.25 and 0.93 respectively, in comparison to the Section 4 results. This may result from the larger values of ρ_1 and ρ_2 used, as it causes more infected nodes to become susceptible again. Although Figures 8a–8d allow us to compare this model to that of Section 4, they show little information about the new states introduced. To address this we refer to Figures 9a–9b, which are distributions from the same simulation on iteration 540. These results indicate that the simulation parameters cause the infection to progress through the population faster than simulations of Sections 3 and 4. This is clear from the fact that on iteration 2670 the distribution for all classes except

S and S_d approaches zero. This indicates the population has reached a stable equilibrium by iteration 2670, in which infected nodes are no longer present. In addition, Figure 9b shows a higher expected number of class V_2 nodes (0.32) in comparison to those in class V_1 (0.20). This would suggest that the large values of ρ_1 and g_1 in comparison to ρ_2 and g_2 respectively are more influential in this simulation than the larger value of p_1 compared to p_2.



(a) (b)

Fig. 9. The probability distribution on iteration 540 of the current model ($N = 10$) of each I and I_d combination (left), and each V_1 and V_2 combination (right).

This implementation gives one illustration of how to make the models derived in this paper more applicable to real world scenarios. However, in order to asses how accurate the current model is, it is recognised that it needs to be compared to real world data.

6 Conclusion and Further Work

This work introduced the use and visualisation of 3D, 6D and 7D probabilistic master equations to depict a botnet lifecycle and to evaluate the likelihood of a given result occurring, given certain parameters. Section 3 highlights how the ability to recover infected nodes (temporarily or permanently) is far more advantageous than simply disconnecting them when attempting to reduce the damage caused by a botnet. The extensions of Section 5 also emphasise that the models introduced in this paper are constrained by their ideal nature. The work also shows how models may be tailored to more specific networks or scenarios.

To extend this work we will consider how the behaviour of the population changes when individual nodes or sections of a population have different properties to each other. This ability would account for the scenario in which offline botnets fail to receive updated instructions from the Botmaster and hence have different properties to the rest of the population when they come back online. We could also consider links between offline states other than to and from their corresponding online states. The work may also be extended by a comparison to publicly-available data obtained from real life botnet infections (e.g., [3]). Using a genetic algorithm (GA), for example, the model parameters given in this work could be adjusted to fit a specific real world data set such as this. A possible cost function for such a GA could be derived as follows.

Denote X as a state in the model, A as the set of all model states, $E[X]$ as the expected number of nodes in class X, and X_1 as the actual number of nodes in class X from the data set being used. Clearly for all $X \in A$ we wish to make $E[X] - X_1$ as close to zero as possible. This is equivalent to minimising $\sum\limits_{X \in A} [(E[X] - X_1)^2]$ (the squaring operation solves potential negativity issues). However, the number of nodes in class X varies with time, t, and so for all $t \in \mathbb{R}^+$ we wish to minimise $\sum\limits_{X \in A} [(E[X(t)] - X_1(t))^2]$. This is equivalent to wishing to minimise the integral $\displaystyle\int_0^\infty \sum_{X \in A} [(E[X(t)] - X_1(t))^2] \, \mathrm{dt}$.

Using this GA approach with suitable mutation and crossover operators may yield a suitable parameter fitting method. This may allow for an assessment of how the dynamic properties of the system vary in actuality and would assist in making this theoretical work even more representative of real life botnets.

Acknowledgements

The authors acknowledge the kindness of the Nuffield Foundation for making this work possible, and the generous research support of Plymouth University.

References

1. Ajelli, M., Lo Cigno, R., Montresor, A.: Compartmental Differential Equation Models of Botnets and Epidemic Malware (Extended Version), University of Trento report T.R. DISI-10-011, 2–3, 9 (2010).
2. Anon.: Internet Census 2012: Port Scanning /0 Using Insecure Embedded Devices (2013) (Carna Botnet): `http://census2012.sourceforge.net/paper.html`
3. CAIDA Datasets: `http://www.caida.org/research/security/#Datasets`.
4. Dagon, D., Zou, C., Lee, W. K.: Modeling Botnet Propagation Using Time Zones, in “Proc. 13th NDSS” (6), 2–13 (2006).
5. Keeling, M.: Population Dynamics MA4E7, Warwick University, 50 (2004), Available at `http://homepages.warwick.ac.uk/~masfz/Pop_Dyn/Handouts.pdf`.
6. Lin, M., Ongtang, M., Rao, V., Jaeger, T., McDaniel, P., La Porta, T., Traynor, P.: On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core, in “Proc. 16th ACM Conf. on Comp. and Comm. Security”, 223–234 (2009).
7. Nordlohne, C.: Measuring Botnet Prevalence: Malice Value, preprint (2015): `http://acdc-project.eu/wp-content/uploads/2015/05/malice-value2.pdf`.
8. Rajab, M., Zarfoss, J., Monrose, F., Terzis, A.: My Botnet is Bigger than Yours (Maybe, Better than Yours): Why Size Estimates Remain Challenging, in “Proc. 1st USENIX Workshop in Hot Topics in Understanding Botnets”, April 2007.
9. Rodionov, E., Matrosov, A.: The Evolution of TDL: Conquering x64, eSeT (2011): `https://www.welivesecurity.com/media_files/white-papers/The_Evolution_of_TDL.pdf`.
10. Rohloff, K., Başar, T.: Stochastic Behavior of Random Constant Scanning Worms, in “Proc. 14th Intl. Conf. in Comp. Comm. and Networks”, 339–334 (2005).
11. Rrushi, J., Mokhtari, E., Ghorbani, A.: Early Stage Botnet Detection and Containment via Mathematical Modeling and Prediction of Botnet Propagation Dynamics, University of New Brunswick Technical Report TR10-206 (2010).
12. Song, L. P., Jin, Z., Sun, G. Q.: Modeling and Analyzing of Botnet Interactions, Physica A 390, 347–358 (2011).