An epidemiological model of virus spread and cleanup

Matthew M. Williamson, Jasmin Léveillé
Information Infrastructure Laboratory
HP Laboratories Bristol

HPL-2003-39
February 27th , 2003*

E-mail: matthew.williamson@hp.com, jasmin.leveille@UMontreal.CA

Signature based anti-virus technologies are widely used to fight computer viruses. It is difficult to evaluate such systems because they work in the wild and few companies would be willing to turn them off to be part of a control group! This paper presents a new model of these technologies that can be used to predict and evaluate their effectiveness. The paper will demonstrate how the model can be used to understand the overall system dynamics, calculate expected costs of outbreaks, give insight into the relative importance of parts of the system and suggest ways to improve the technology. It is also used to evaluate new approaches to fighting viruses.

* Internal Accession Date Only

An epidemiological model of virus spread and cleanup

Matthew M. Williamson and Jasmin Léveillé
HP Labs Bristol, Filton Road, Stoke Gifford, BS34 8QZ, UK
matthew.williamson@hp.com, jasmin.leveille@UMontreal.CA

Abstract

Signature based anti-virus technologies are widely used to fight computer viruses. It is difficult to evaluate such systems because they work in the wild and few companies would be willing to turn them off to be part of a control group! This paper presents a new model of these technologies that can be used to predict and evaluate their effectiveness.

The paper will demonstrate how the model can be used to understand the overall system dynamics, calculate expected costs of outbreaks, give insight into the relative importance of parts of the system and suggest ways to improve the technology. It is also used to evaluate new approaches to fighting viruses.

1 Introduction

Computer security is an arms race between defenders and attackers. One area where this is particularly obvious is computer viruses[1]. Over the years computer viruses have changed and the predominant anti-virus technology (signature-based scanners) has adapted too (Grimes, 2001). It is a matter of debate (see Schmehl (2002b,a); Leyden (2002)) whether these technologies will continue to be able to adapt in the face of fast spreading viruses e.g. Nimda (CERT, 2001b), or more apocalyptic theoretical viruses (Staniford et al., 2002).

Unfortunately it is difficult to evaluate anti-virus systems, since they work in the wild and few companies would be willing to turn off their anti-virus software to be part of a control group! The alternative is modelling. This paper presents a new model of virus spread and cleanup that gives insights into the performance of present day anti-virus defences. The model provides a system view to test otherwise ad hoc intuitions, is used to expose weaknesses and dependencies between parameters and is used to evaluate new ideas. The model thus allows practical ideas to be tested as well as exposing where new technologies might be needed.

Models in general are best when validated against real data. Since in this case that would probably require releasing many test viruses into the wild and measuring prevalence and cleanup rates, strict validation is a non-starter. Calibrating the models using data from existing outbreaks would be weaker than validation but still useful. Unfortunately such data is rare and is often commercially sensitive. If more organisations collected and released suitably anonomised data on outbreaks it would greatly help in efforts to model and predict overall system properties.

The model described in this paper is in the style of an epidemiological model (Murray, 1993) used for many years to study the spread of biological disease. In recent years there had been a growing body of interest in using these models to gain insights into computer viruses. Kephart and White (1991) and Kephart et al. (1993) looked at the effect of network topology on the speed of virus propagation. Pastor-Satorras and Vespignani (2001) looked at virus spread on different network topologies, particularly scale free networks that are thought to model well some computer networks, e.g. email (Ebel et al., 2002), router topology (Faloutsos et al., 1999). Some computer scientists have also used these techniques to model the spread of viruses such as Code Red (Zou et al., 2002), immunisation strategies (Wang et al., 2000) and theoretical viruses (Staniford et al., 2002).

What is missing from all these models is any detailed model of the effect of anti-virus software on the propagation of the virus. The model presented in this paper

[1] While there are rigorous definitions of virus, worm, trojan etc., in this paper the word virus is used to mean any malicious mobile code

tackles this issue head on, being an explicit model of virus spread and traditional anti-virus technologies for cleanup.

The rest of the paper consists of a short introduction to epidemiological models, and a description of the model showing how it relates to reality. The model is then analysed in detail, providing intuition as to its operation and detailed insights into the various aspects of anti-virus technology. The model is used to evaluate a candidate new technology: Virus Throttling (Williamson, 2002; Twycross and Williamson, 2003). The paper concludes with some recommendations for improving anti-virus technology.

2 The model

The model uses the techniques of epidemiological models (Murray, 1993). The idea is to abstract away the particular details of an infection and express individuals as progressing through a set of states at different rates. For example, in the simplest epidemiological model, individuals transition from a Susceptible state to an Infectious one at a certain rate, and become Susceptible again at a different rate. This models systems where having the infection and being cured does not confer immunity. This model is called the SIS model, because individuals move between the S (Susceptible) and I (Infectious states). While there are more complex epidemiological models, none of them capture all the aspects of computer virus spread and cleanup.

The general process for a virus infection is as follows. First the virus is released into the wild by its creator. The virus spreads freely, infecting machines and delivering its payload. As some point the virus is noticed and an anti-virus company is alerted. The company then works to isolate the virus and generate a “signature” that can be used in scanning software to detect the presence of the virus. This process can take some time, during which the virus can spread unchallenged.

Once the signature has been developed, it needs to be distributed to the many millions of client machines. This is usually accomplished by the client machines regularly polling a central server for anti-virus updates.

Once the client machines have the signatures, one of two things happens. If the machine was not yet infected by the virus, that machine is made immune to the virus by possessing the signature (assuming that the anti-virus software is installed and working properly). If the machine was infected, then the virus can be detected and the user can set about cleaning their machine. Most vendors recommend that the computer is disconnected from the network if a virus is detected, so preventing further spread of the virus. Once the computer has been cleaned and the anti-virus signatures updated it can be safely reconnected to the network.

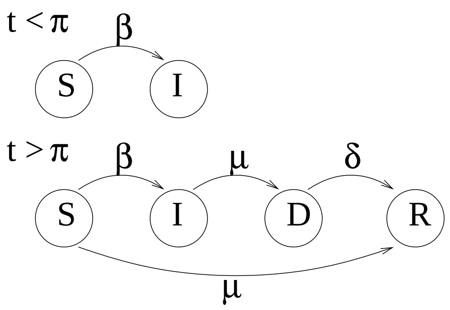

Figure 1: The PSIDR model. In the model, machines move between four states: Susceptible (S), Infectious (I), Detected (D) and Removed (R). Initially, the signature is not available (the time t is less than the signature delay time π), and the virus spreads unhindered, each Infectious machine infecting more Susceptible ones. After the signature is released ($t > \pi$) it is distributed at a rate μ, causing Susceptible machines to become immune or Removed, and machines with the virus to be Detected. Those machines that are Detected no longer spread the virus and are cleaned up at a rate δ.

The virus spread and cleanup can be modelled as shown in Figure 1. Machines are assumed to be in one of four states: Susceptible (S) meaning vulnerable to the virus, Infectious (I) meaning infected and actively spreading the virus, Detected (D), a state in which the virus has been detected and is prevented from spreading further, and Removed (R), which corresponds either to immunity from the virus, or from having been cleaned up after a virus infection.

Individuals progress through these states in two consecutive stages (which gives the model its name: Progressive Susceptible Infectious Detected Removed (PSIDR) model). Before the signature is available, the virus can spread among the Susceptible machines making them Infectious. The spread is modelled with the parameter β that indicates the number of infection attempts an infectious machine will make per timestep. The number of timesteps before the signature is released is modelled by the parameter π.

The virus signature is distributed to the client machines, modelled using the parameter μ, indicating the proportion of machines that will receive the virus update per time period. The virus update is applied to machines independent of their infection state (i.e. to S and I), and machines that are Susceptible become Removed or immune, while those that are Infectious will become Detected. The parameter δ models the process of cleaning up the Detected machines, and making them immune or Removed.

To simulate the model, time is divided into a number of discrete steps, and on each timestep the population of individuals in each state is altered according to the different rules.

The starting condition is that one individual is Infectious, and the rest are Susceptible. Before the signature ($t < \pi$), the virus spreads and infects Susceptible machines. The probability of a Susceptible machine becoming Infectious is

$$P(S \rightarrow I) = \beta I/N \tag{1}$$

where β is the spreading rate of the virus, I is the number of infectious machines and N is the total number of machines. This is equivalent to each Infectious machine infecting on average β others per timestep. β thus rolls together a number of real world factors, for example the number of connections made per second by a scanning worm, the sparseness of machines in IP address space, the chance of a random machine being vulnerable to the virus etc.. It does not model secondary effects such as reduced spreading due to network congestion when many machines are infected (see Zou et al. (2002) for a model that includes these factors).

After the virus signature is available ($t > \pi$), the virus continues to spread, the virus signature is distributed and machines are cleaned up. i.e.

$$\begin{aligned} P(S \rightarrow I) &= \beta I/N \\ P(I \rightarrow D) &= \mu \\ P(S \rightarrow R) &= \mu \\ P(D \rightarrow R) &= \delta \end{aligned}$$

One useful aspect of this model is that it allows the costs of an outbreak to be estimated. A real virus attack can cause damage in many different ways: machines can be infected and require cleanup, there may be loss or corruption of data, downtime of critical services, loss of network performance and even loss of business reputation. The challenge from a modelling perspective is to find costs that are a good approximation to these real costs. Two measures have been chosen for this paper: the outbreak size, and the outbreak duration. It is possible to calculate other costs from the model, these are reported elsewhere (Léveillé, 2002).

The outbreak size is defined as the total number of machines that become infected and have to be cleaned up. This is directly related to the work required to clean up after an outbreak. It is also a measure of the extent of data loss/corruption.

The duration of the attack is defined (for the model) as the time from the beginning of the outbreak to the time that a large proportion of the machines are free of the virus. This can be calculated as the time for (say) 95% of machines to be in the Removed state. Since machines enter this state by two routes (see Figure 1), this measure is really "time to safety", i.e. the time when 95% of machines are not vulnerable to the virus.

Even though this model has been explained for signature-based anti-virus technology, it also model well the generic process of fighting a virus: a delay while IT staff determine what is going wrong and decide what to do about it, a process to detect and stop further infection from infected machines, and a cleanup process. These processes can thus be modelled with the same model, perhaps using slightly different parameter values.

It also provides insight into variations with the usual process. For example, recent viruses (e.g. Bugbear (Symantec, 2002)) have attempted to switch off anti-virus software. In the model this corresponds to breaking the I → D path. Machines infected with Bugbear would then have to be detected and stopped by some other means. The model helps with understanding the system dynamics here: if the signature can be delivered quickly, many machines will go from S → R and the outbreak will be small. If fast signature delivery is not possible, the outbreak will probably be larger than if Bugbear had not switched off the software, but working on quickly detecting and stopping it (I → R) will improve the situation.

3 Initial results

The model was simulated on a network of 6250 nodes, connected so that every node can contact and infect any other. This is a good model for viruses that spread using IP addresses (e.g. Nimda (CERT, 2001b). Other types of viruses would require different types of networks, although the overall results are similar (Léveillé, 2002).

To improve the accuracy of the simulation, each timestep is split into n sub-timesteps, with the parameters β, μ etc. being divided by n. To decide the number of subtimesteps required, the model was simulated in the phase before the signature is available, and the results compared to an analytical solution (possible to calculate from Equation 1 for that phase). The value $n = 10$ was found to reduce noise in the simulation without greatly impacting performance (see Léveillé (2002) for more details).

The parameters are chosen as follows. The virus spreading rate (β) is the average number of infections per machine per timestep. This was arbitrarily set between 0.1 and 0.8. The distribution rate of the anti-virus signature (μ) was set to be slower than this (most anti-virus clients poll for updates once per day), in the range 0.01–0.1. The cleanup rate (δ) will be slower still, since it is often a manual process. The range for δ was 0.005–0.05. The values for the signature delay (π) were chosen to span the situation when all the machines in the network become infected, i.e. 0–20.

Figure 2 shows a time series plot for the model with a standard set of parameters and three different values for π. At first the virus spreads slowly, since only a small number of machines are infected and spreading, but over time the number of machines that are infected per timestep increases. When $t = \pi$, the response is initiated, as indicated by the vertical line in the plot. From that point onward, the virus signature is distributed, removing susceptible machines and detecting infectious ones. The number of detected machines thus increases. The cleanup process also starts at this point, but is slower than the anti-virus distribution, so the number of detected machines peaks and then declines as machines are cleaned up.

The effect of the time delay π is also evident in Figure 2. For small values of π the effect of the virus is low, while for higher values the virus effectively saturates the network, requiring that all the machines be cleaned up.

The trace for number of detected machines gives perhaps the best comparison with existing data on virus outbreaks e.g. CAIDA (2002). This is because incidence rates of viruses are generated from users detecting the virus. The model thus highlights that measures of incidence are sensitive to how widely the detectors are distributed.

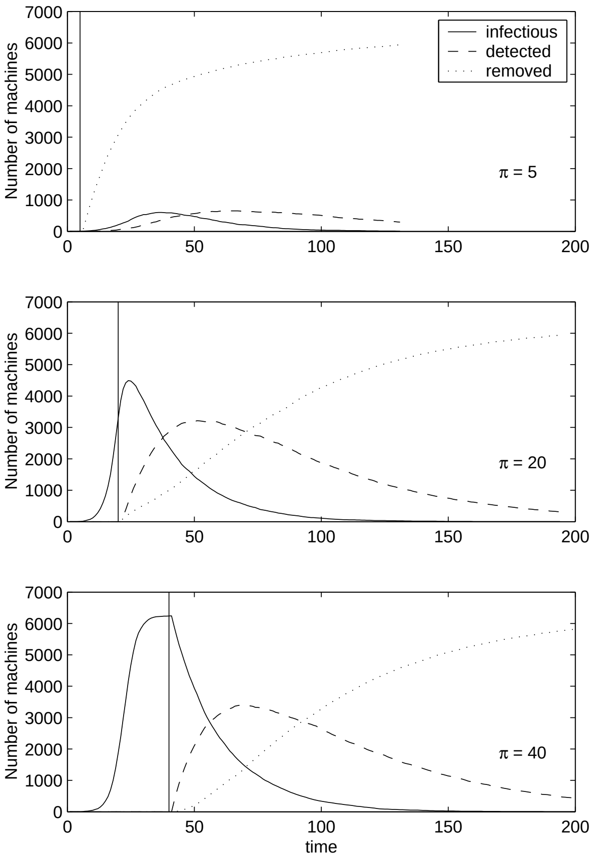

Figure 2: Time series traces for the model. The plots show the time progression of the machine states for different values of signature delay π. The onset of the signature is indicated by the vertical line. Before the signature is available, the virus spreads unhindered. After it is available, the signature is distributed making susceptible machines immune (dotted line) and infectious machines detected (dashed line). Over time the detected machines are cleaned up, so the dashed line decays, and the number of immune machines (dotted line) further increases. For these plots $\beta = 0.4, \mu = 0.05, \delta = 0.02$.

4 Results

The model has various parameters (π, β, μ, δ) that interact with one another in different ways. One way to evaluate these interactions is to simulate the model with a variety of parameter settings and calculate the predicted costs of the outbreak.

Figure 3 shows the effect of varying the virus spreading

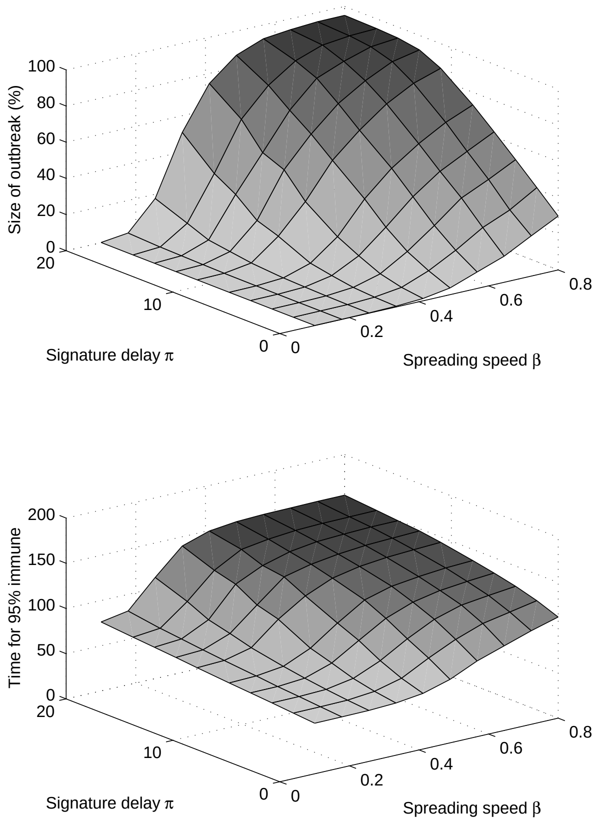

Figure 3: Outbreak costs as a function of delay π and virus spreading speed β, $\mu = 0.055, \delta = 0.0275$. The top plot shows the outbreak size, showing that high spreading rates and late signatures result in large outbreaks and vice versa. Even when the signature is available immediately there still can be an outbreak, particularly for large β. The lower plot shows the time for 95% machines to be immune and has a similar overall shape. Even quite small outbreaks can take some time to be over, because this cost includes the time for all machines to have the virus signature and be cleaned up.

rate β and the anti-virus signature delay π on the overall costs of the outbreak. Each point on the graph is an average of 200 runs of the model.

The top graph shows the outbreak size (number of infected machines) for different values of β and π. The graph is a surface, giving the cost as each parameter is varied independently. The general trend is not surprising. The lowest costs are for slow viruses and prompt signatures, and the highest for the opposite: fast viruses and slow signatures. The costs saturate because of the limited size of the network—all machines are becoming infected. The higher costs are all for faster spreading viruses, showing that the anti-viral system is weakest against these attacks.

The graph also shows more subtle points. For fast viruses, even if the signature is available immediately, there is still a significant outbreak. This is caused by the signature and the virus "racing" to find machines, with the virus managing to infect some machines because of its higher propagation speed. The graph also shows (particularly for slower viruses) that increased delays (larger π) result in increasingly larger outbreaks. This effect is not so clear for faster viruses because the network saturates. This occurs because the global spreading rate (new infections/time period) increases as the number of infectious machines increases. A small increase in the delay results in a larger increase in the outbreak size. This effect becomes less strong when the supply of vulnerable machines is reduced (the network saturates). This is why a prompt signature has proportionally more effect for slower viruses than faster ones.

The lower plot in Figure 3 shows the outbreak duration (time till 95% of machines have been cleaned up or are immune). This shows a similar shape to the upper graph. It is interesting that even small outbreaks take some time to clean up. This cost measures "time to safety" for a particular virus attack, so even if the attack is slow, the time to distribute the signatures to all the machines can be large.

In summary, these results show that the weakness of the anti-virus system is fast spreading viruses. Generating virus signatures quickly can significantly reduce costs, but for fast viruses this is not enough: the signature either needs to be be available before the virus starts (which is difficult), or needs to be distributed faster. Anti-viral mechanisms that work before the signature is available would help solve this problem, for example behaviour blocking (Messmer, 2002), although that approach appears to be plagued with practical problems (Lu, 2001). Virus throttling (Williamson, 2002; Twycross and Williamson, 2003) would also be appropriate; its effectiveness is analysed in Section 5.

One of the questions that can be addressed by this model is "What is more important, a prompt signature or a fast signature distribution system?". Figure 4 shows the costs from varying the delay π and distribution rate μ for a fixed virus spreading speed.

The top graph shows the size of the outbreak as before, showing that both factors are important. It is only with a prompt signature (low π) and a fast update (high μ) that the outbreak will be small. The lower graph shows the duration, which is quite different. The time is strongly related to distribution speed, with the effect of the delay being small and almost linear. Together with the previ-

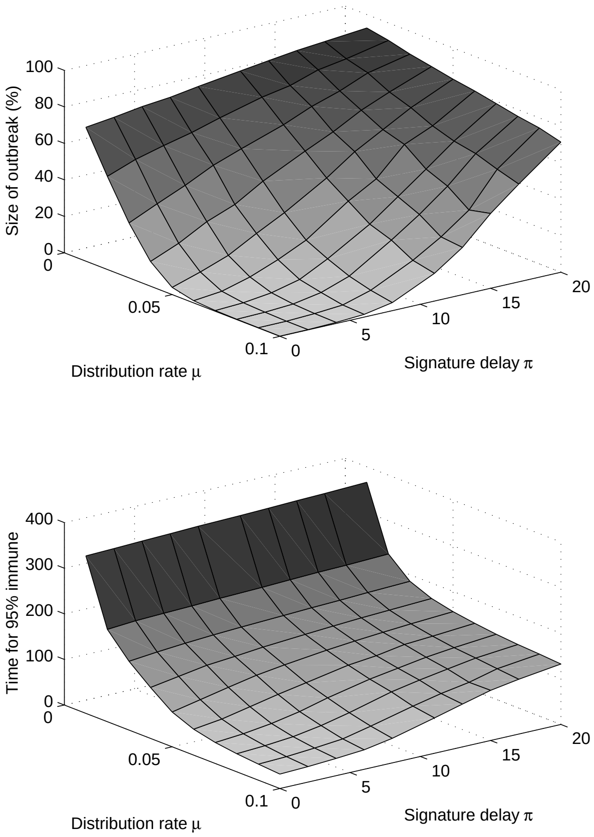

Figure 4: Outbreak costs as a function of signature delay π and distribution rate μ, $\beta = 0.45, \delta = 0.0275$. Note that the axis for μ runs with its higher values toward the centre of the plot. The top plot shows the outbreak size, showing that both a fast signature update and prompt signature is needed to reduce the size of the outbreak. The lower plot shows the outbreak duration, which is most strongly dependent on μ, with faster distribution rates resulting in much shorter outbreak times.

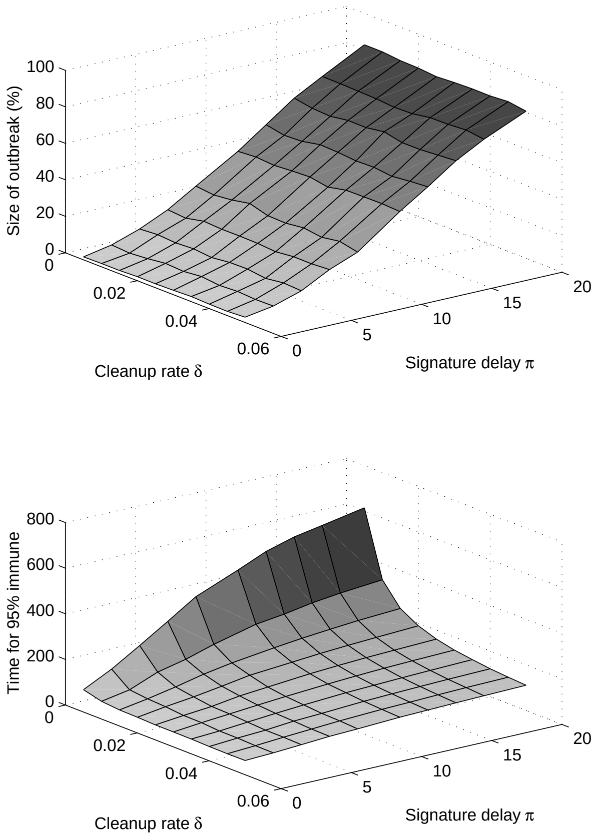

Figure 5: Outbreak costs as a function of delay π and cleanup rate δ, $\beta = 0.45, \mu = 0.055$. Higher values of δ are toward the centre of the plot. The top plot shows the outbreak size, showing that size is independent of the value of δ. The lower plot shows the duration, showing that for low values of cleanup rate, and for large outbreaks the time is increased.

ous graph this shows that increasing distribution speed both reduces outbreak size and the time to clean up.

The detection and clean up process is really a race between the virus and the anti-virus signature. The signature will eventually win the race, because it both combats the virus directly (I $\rightarrow$ D) as well as removing vulnerable machines for the virus to spread to (S $\rightarrow$ R). However, during the race the virus can spread more quickly and indeed will have a headstart due to the late virus signature. This is why both factors are important: a prompt signature will reduce the headstart, and fast distribution will gain ground on the virus.

This analysis may explain the success of web-based automatic virus signature updates, but suggests that increasing the rate of release of signatures (i.e. not waiting and bundling them together), and also increasing the rate with which machines poll for new updates might be helpful. If viruses continue to increase in propagation speed it might be necessary to "push" signatures to machines to ensure that they get the signature update as quickly as possible.

The final parameter is the cleanup rate δ. The variation of this with π is shown in Figure 5. The upper plot shows that the size of the outbreak is independent of δ, which is expected: the cleanup process occurs after the virus has been stopped and thus has no influence on the number of infected machines.

The effect of δ is clearer in the bottom plot, which shows the outbreak duration. The duration is driven by two factors, the speed that vulnerable machines are made immune (S $\rightarrow$ R, driven by μ) and the speed that detected machines can be cleaned up. If the cleanup rate is low, and the outbreak large (e.g. high β or late π) the time will be long, otherwise δ has little effect.

A high value for cleanup speed is thus most important for large outbreaks. Increasing δ means increased efficiency of cleanup, which could be obtained using automated solutions. One technological solution addressing this problem is Norman and Williamson (2003), a scanning system that breaks into systems using existing vulnerabilities in order to administer them. This solves the common problem where it is possible to detect vulnerable or infectious machines on the network (though scanning), but it is difficult to physically locate them for attention. This might be because the machines are not managed with a network management system, or because the mapping from network address (e.g. IP address) to machine location is not maintained or is dynamic through the use of DHCP. By using the network address of machines to initiate cleanup, the efficiency of cleanup is increased.

5 Applying the model: virus throttling

One of the advantages of using a model is that it is possible to evaluate the system properties of new ideas and technologies. This section considers the system-wide effect of virus throttling (Williamson, 2002; Twycross and Williamson, 2003).

Virus throttling is a technique to automatically contain the damage caused by fast spreading viruses. Rather than attempting to prevent a machine becoming infected (the role of most anti-virus software), the throttle prevents the further propagation of the virus from that infected machine. This has the effect of slowing the overall global spread of the virus (because fewer machines will be actively spreading it), and also to reduce load on network infrastructure (fewer spreaders mean less network traffic). The technology consists of a rate-limiter that effectively stops machines acting in a viral way (making many connections/sending many emails to many different machines in a short space of time) without affecting normal usage of the machine.

Since virus throttling only prevents further infection, its effectiveness will be determined by how widely it is deployed. Throttling can be incorporated into the model by dividing machines into two groups, throttled and unthrottled. If a throttled machine becomes infected, it does not spread the virus further, and immediately enters the Detected state. This is because the throttle stops the virus spreading and contacts the user when it notices a virus attempting to spread.

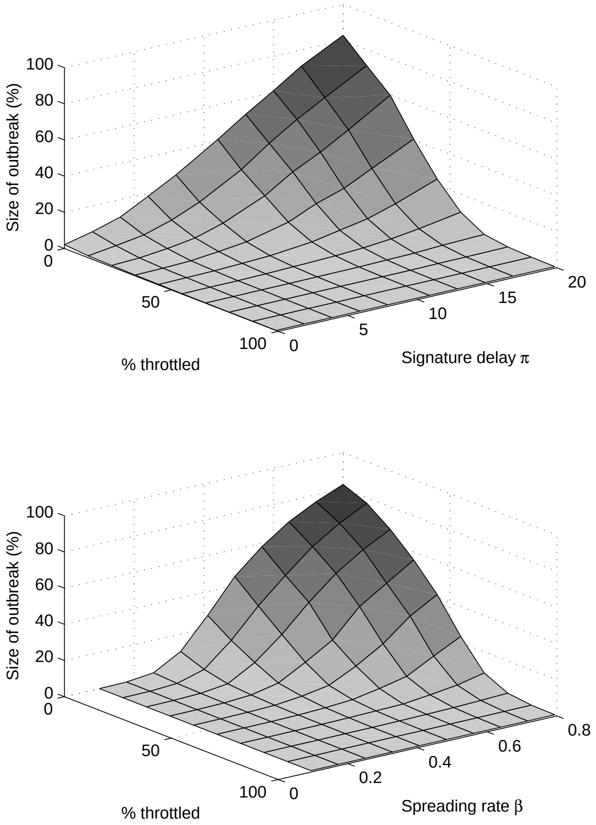

Figure 6: Outbreak costs for virus throttling. The upper plot shows outbreak size as a function of the % of throttled machines in the network, and the signature delay π, $\beta = 0.45, \mu = 0.055, \delta = 0.0275$. Throttling always improves the costs, but gives very low costs when 50–60% of machines are throttled. A similar situation is seen in the lower plot, showing variation with virus spreading rate β, $\pi = 10, \mu = 0.055, \delta = 0.0275$.

Figure 6 shows the behaviour of the overall system as a function of the percentage of machines with throttles and the signature delay π (top plot) and virus spreading speed β (lower plot). In both plots the line for % throttled $= 0$% shows how large the outbreak would be with no throttling. The duration shows a similar pattern and is not shown.

In the top plot the effect of the throttling is most strong when more than half of the machines have throttles, when even a late signature ($\pi = 20$) will result in a much smaller outbreak. For faster signatures the outbreak is small anyway. In the lower plot a similar shape is seen, with throttling above 50–60% giving a large reduction in the outbreak size, even for high spreading rates.

These plots give confidence that throttling would be effective against a wide range of virus spreading speeds. In addition it shows that even when only half of the machines have throttles, the effect of the throttling is strong. One of the aims of throttling is to buy time for slower and more definite response mechanisms such as signatures. The use of throttling to "hold off" viral attacks means that signatures can be safely developed later, without resulting in the large outbreaks discussed in section 4.

6 Further Work

The first area for further work is calibration. The results of the model would be stronger if the values for the parameters could be related to real units (days, weeks, etc.). It would be relatively straightforward to match the initial spreading part of the model to known virus behaviour. What would be more difficult is calibrating the signature and distribution parameters. Information from anti-virus companies (e.g. logs of signature update requests from clients) would be most useful in this respect.

There are number of ways that the model itself could be improved. The first way would be to look at the effect of network topology on the overall system behaviour. The results presented in this paper are all for completely connected machines. This is a reasonable model of a Code Red (CERT, 2001a) style virus spreading within a firewall, but is not so good for e.g. an email virus, which spreads over the graph of email addresses found on each infected machine. Computer networks are layered systems, with different topologies at the different layers. The topology that is important for the virus is the graph of the addresses that it uses to spread. For email viruses, it is the network formed by the email addresses on infected machines that is important, not the underlying mail delivery infrastructure. There has been a considerable amount of work on the effects of network topology on virus outbreaks (see Pastor-Satorras and Vespignani (2001); Newman (2002); Murray (1993)).

One interesting result from the work of Pastor-Satorras and Vespignani (2001) is that the flow of viruses over networks where different nodes have different connectivity is strongly dependent on the nodes with highest connectivity. In the networks which are thought to best model email networks (Ebel et al., 2002) these highly connected nodes are rare[2]. Theoretical work by Deszö and Barabási (2002); Pastor-Satorras and Vespignani (2002); Wang et al. (2000) suggests that targeting immunisation on these highly connected nodes is considerably more effective than random immunisation. Anti-virus signature updates are random in the sense that they do not target highly connected computers.

In practise it is difficult to determine *a-priori* what nodes are the most highly connected, except by traversing the same graph as the virus! While "good viruses" (e.g. the cheese worm (Symantec, 2001)) are rightly frowned on by the security community for a variety of reasons, it might be interesting to compare the efficiency of random anti-viral updates with viral transmission of the updates. If it proved to be the case that viral transmission was much more effective than random, it might encourage researchers to find ways to enable such a mechanism safely.

A second area for further work is better models for the signature distribution and cleanup processes. At present these are modelled with a single parameter, so that at each timestep, each individual has exactly the same chance of getting a virus signature/being cleaned. In reality this is more likely to be a distribution: some users update frequently, others rarely. Both of these processes could be better modelled by having each individual have a unique parameter drawn from a distribution (e.g. a normal distribution with mean and variance to capture behaviour and variations from it). Adding this would only add two parameters to the model (the variances) but would allow individual differences to be modelled.

7 Conclusion

This paper has presented a model of virus spread and cleanup that extends previous work by explicitly modelling the processes of fighting viruses using signature-based anti-virus software. The model has a small set of parameters and allows the cost of outbreaks to be evaluated in a variety of different ways.

The model is a useful tool, allowing a general feel for the overall virus/anti-virus system. Looking at the relative rates and ways that machines progress through the model gives a good high level understanding of the system dynamics. It also allows an understanding of the consequences of any exceptions to the system, for example the effect of viruses switching off anti-virus software.

[2] These networks are known as scale-free networks (Barabási and Albert, 1999).

The model can be used to evaluate weaknesses and determine dependencies between parameters. In the anti-virus system the two key parameters are the signature delay and the signature distribution rate. The model has confirmed intuitions that these two parameters are dependent on one another: small outbreaks only occur when the signature is available promptly and is distributed quickly. The weakness of the whole system is to fast spreading viruses.

Analysis of the model reveals more subtle effects. For example that the signature should be prompt because any increase in the delay will result in larger and larger outbreaks. However, even if a signature is available immediately there can still be an outbreak as the virus and the signature "race" to find machines. These results suggest that the overall anti-virus system could be improved if signatures were delivered more quickly, perhaps by increasing the rate with which clients poll central servers for updates.

Finally the model can be used to evaluate the systems effects of new ideas and technologies. This paper has presented some encouraging results on the properties of virus throttling (Williamson, 2002). Throttling, which prevents the onward propagation of viruses, appears to complement well signature-based approaches, particularly in areas where they are weak i.e. faster viruses.

To conclude, while this model would benefit from calibration against real world data, in its present form it has been shown to be useful in three areas: providing a systems-level view, exposing weaknesses and dependencies and evaluating new technologies. With more data this sort of model could provide valuable insight and prediction for the entire anti-virus industry.

References

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

CAIDA (2002). Caida analysis of code-red. Available from `http://www.caida.org/analysis/security/code-red/`.

CERT (2001a). CERT Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. Available at `http://www.cert.org/advisories/CA-2001-19.html`.

CERT (2001b). CERT Advisory CA-2001-26 Nimda Worm. Available at `http://www.cert.org/advisories/CA-2001-26.html`.

Deszö, Z. and Barabási, A.-L. (2002). Halting viruses in scale free networks. In cond-mat/0107420. Available from `http://www.arxiv.org/PS_cache/cond-mat/pdf/0107/0107420.pdf`.

Ebel, H., Mielsch, L.-I., and Bornholdt, S. (2002). Scale free topology of email networks. In cond-mat/0201476. Available from `http://www.arxiv.org/PS_cache/cond-mat/pdf/0201/0201476.pdf`.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. In *ACM SIGCOMM*, pages 251–262.

Grimes, R. A. (2001). *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly & Associates, Inc.

Kephart, J. O. and White, S. R. (1991). Directed graph epidemiological models of computer viruses. In *Proceedings IEEE Symposium on Security and Privacy*.

Kephart, J. O., White, S. R., and Chess, D. M. (1993). Computers and epidemiology. *IEEE Spectrum*, pages 20–26.

Léveillé, J. (2002). Epidemic spreading in technological networks. Master's thesis, University of Sussex. Available from `http://www.hpl.hp.com/techreports/2002/HPL-2002-287.html`.

Leyden, J. (2002). Viruses are dead. Long live viruses! The Register.

Lu, L. (2001). Reducing false positives in behaviour blocking. In *Proceedings of Virus Bulletin Conference*. Virus Bulletin Ltd.

Messmer, E. (2002). Behavior blocking repels new viruses. Network World Fusion News. Available from `http://www.nwfusion.com/news/2002/0128antivirus.html`.

Murray, J. D. (1993). *Mathematical Biology, (2nd, corrected edition)*. Springer Verlag, New York.

Newman, M. E. J. (2002). The spread of epidemic disease on networks. *Physical Review E*, page 016128.

Norman, A. and Williamson, M. M. (2003). Hitting back at code red. Technical report, Hewlett-Packard Labs. Available from `http://www.hpl.hp.com/techreports/`.

Pastor-Satorras, R. and Vespignani, A. (2001). Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200–3203.

Pastor-Satorras, R. and Vespignani, A. (2002). Immunization of complex networks. *Physical Review E*, 65:036104.

Schmehl, P. (2002a). Life after AV: If anti-virus is obsolete, what comes next? Security Focus Online. Available from `http://www.securityfocusonline.com/infocus/1604`.

Schmehl, P. (2002b). Past its prime: Is anti-virus scanning obsolete? Security Focus Online. Available from `http://online.securityfocus.com/infocus/1562`.

Staniford, S., Paxson, V., and Weaver, N. (2002). How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02)*. Available at `http://www.icir.org/vern/papers/cdc-usenix-sec02/`.

Symantec (2001). Linux.cheese.worm. Available from `http://www.symantec.com/avcenter/venc/data/linux.cheese.worm.html`.

Symantec (2002). W32.bugbear@mm. Available from `http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.%html`.

Twycross, J. and Williamson, M. M. (2003). Implementing and testing a virus throttle. In *Submitted to USENIX Security Symposium, 2003*. Available from `http://www.hpl.hp.com/techreports/`.

Wang, C., Knight, J. C., and Elder, M. C. (2000). On computer viral infection and the effect of immunization. In *Proceedings of ACSAC '00*.

Williamson, M. M. (2002). Throttling viruses: Restricting propagation to defeat malicous mobile code. In *Proceedings of ACSAC Security Conference*, pages 61–68, Las Vegas, Nevada. Available from `http://www.hpl.hp.com/techreports/2002/HPL-2002-172.html`.

Zou, C. C., Gong, W., and Towsley, D. (2002). Code red worm propagation modeling and analysis. In *Proceedings ACM CCS '02*, Washington, DC.