A spread model of flash worms

Yury Bulygin
Security Evaluation Center of Excellence, Intel Corporation
JF4-318, 2111 NE 25th Ave, Hillsboro, OR 97124-5861, U.S.A.
yuriy.bulygin@intel.com

Abstract

In this work we introduce a mathematical model for epidemics of worms using hit-list spreading technique. It provides an estimation of time required for flash worms to infect the whole vulnerable population. The estimated infection time shows that even heavy network worm can potentially infect large-scale vulnerable population within few seconds. Primarily the work is based on results of the work *Top Speed of Flash Worms by S. Staniford et al.*. We also generalize infection doubling technique used to increase a resilience of epidemics of flash worms.

It took the whole day for Code Red I v2 to spread among over 350,000 Internet hosts. Slammer worm infected more than 90 percent of up to 100,000 vulnerable hosts within 10 minutes (*Inside the Slammer Worm by D. Moore et al.*), Witty worm infected almost all of its 12,000 victims in 45 minutes (*The Spread of the Witty Worm by C. Shannon and D. Moore*).

Index Terms

Internet worms, spyware, Warhol, flash worms, botnet, hit-list, conqueror worm, perfect spread, Witty, Slammer, Code Red, Sasser.

I. Introduction

During past few years there were a very few global worm epidemics. Thousands of hosts are usually infected creating local gainful "explosions". This follows the predicted tendency of growth of cyber crime. Spreading mostly among large corporate networks infecting relatively small amount of computers pursues the purpose of creating botnets. Botnets are a perfect base for attacking Internet sites to put them out of service, stealing or guessing passwords, web money credentials, credit card numbers and sending dozens of spam to install spyware or perform fishing attacks. Please refer to an overview of botnets in [1]. A botnet contained over 100,000 infected hosts has been recently shut down [2]. This seems to be the highest level of predicted criminalization of cyber society [3]. But in the real world we suffer from both global natural and human induced disasters all over the world and the impact of these disasters increases eventually. But however it may seem sometimes that global computer worm outbreaks are in the past we must be prepared to global 'explosions' of network worms, rapid, mostly meaningless, resulting in a huge number of victims.

We can consider the following reasons why global outbreaks are unprofitable nowadays. First, a global epidemic requires more time and resources for preparation - searching for an exploitable vulnerability in widely used software. But a Witty worm came very close to 'zero-day' worm epidemics in much smaller population. Second, cyber criminals pursue their own selfish ends creating rather large but not global botnets and using them for getting a profit. And third, past Internet worms searched randomly for addresses of victims, namely used random-scanning technique for spreading which is likely the easiest one to implement. This technique results in rather slow spread due to large number of infection failures and causing an increasing number of network failures when infection grows. A slow speed of spreading allows signature-based systems to success in worm containment and network failures caused by aggressive random scanning introduce exponential decreasing into the epidemics.

The first two reasons are beyond the scope of this work and attention is drawn to the techniques that can be used by worms for spreading. A new *hit-list spreading* technique was introduced in [4], [5]. Is means

The work done while author was at Technological Research Dept., Kaspersky Lab and Moscow Institute of Physics and Technology, Moscow, Russia

propagation over prepared list of vulnerable hosts and delivering this list from generation to generation of the worm. Worms using hit-list spreading were named as *Warhol worms*. The work [5] generalizes the hit-list spreading technique for the Internet-scale vulnerable populations introducing *flash worms*. The work also shows that it takes flash worms significantly faster to spread over vulnerable population than worms using random scanning technique. It was also shown in [6] that imperfect hit-lists due to high rate of actually invulnerable hosts in the hit-lists significantly increase a failure rate of a total number of infected hosts. The authors suggested solutions to decrease an impact of imperfect hit-lists but they considerably complicate creating such a worm.

In this work we provide a lower bound of a total infection time of the whole hit-list of vulnerable population and analyze some of its properties introducing a *perfect hit-list spreading model*. We use a term *the conqueror worm*[1] when consider a flash worm using introduced spreading model. It shown in the work that the conqueror versions of even large Internet worms as Sasser could potentially infect Internet-scale population within several seconds.

Also in this work we generalize a solution suggested in [6] which reduces an impact of imperfect hit-list on spreading of flash worms by doubling the infection and consider a new solution allowing to avoid failing. The solution combines random scanning for addresses and hit-list spreading which allows to achieve linear infection failure rate $\tau(\sigma)$ depending on a fraction of invulnerable hosts in a hit-list σ.

II. Epidemic model of flash worms

A. Perfect spreading over hit-list

To estimate the total infection time we'll consider a theoretical approximation of resilient worm spreading over hit-list, a *perfect hit-list spreading model*. The model which can not be strictly achieved but nothing prevents any real worm to approximate to it as close as possible. The model is based on the following assumpions:

1) A worm instance is *single-threaded*; infects vulnerable hosts subsequently.
2) Spread tree is **k**-way.
   An each instance infects at most **k** vulnerable hosts. This property is the same as the limited number of contacts that infected person has in classic epidemic models.
3) $\forall i, 1 \leq i \leq D : t_i - t_{i-1} = \overline{\delta t} = Const$
   A time of a single infection is averaged over the spread tree.
4) A "child" worm instance starts infection simultaneously with a "parent" worm instance starting infection of the next "child".
5) A worm transmitts $\frac{1}{k}$ fraction of the hit-list to each "child" instance during infection; this information is necessary for further spreading over hit-list of vulnerable population.

Some of the above assumptions significantly simplify simulation of the epidemic spread. In this model we show that even a single-threaded flash worm infecting one vulnerable host at a time cause exponential growth of the epidemics. Let's describe the model in more details.

To infect each victim an instance of the worm should seed random number generator, randomly choose an address from the portion of a hit-list the worm instance has, initialize a connection with the vulnerable host and send its payload to the vulnerable host exploiting its vulnerability. After finishing infection (Figure 1(a)) the instance of the worm starts infecting the next victim i.e. its next child in a spread tree. At the same time a just infected child in its turn starts infecting its own vulnerable children propagating deeper over the spread tree (Figure 1(b)). Thus all infection consists of successive synchronous steps where duration of each infection step equals to time required for infecting of the next vulnerable node by any worm instance. Number of steps required to infect any node within the perfect spread tree is shown inside each node circle on Figure 1.

B. Hit-list infection time

Now we can estimate a total time necessary for the worm to infect all vulnerable addresses in a hit-list. This is the duration between a time when the worm start spreading and a time when epidemics reaches its highest level.

[1] A term 'The Conqueror Worm' is taken from 'The Conqueror Worm' poem by Edgar Allan Poe (http://en.wikipedia.org/wiki/The_Conqueror_Worm).

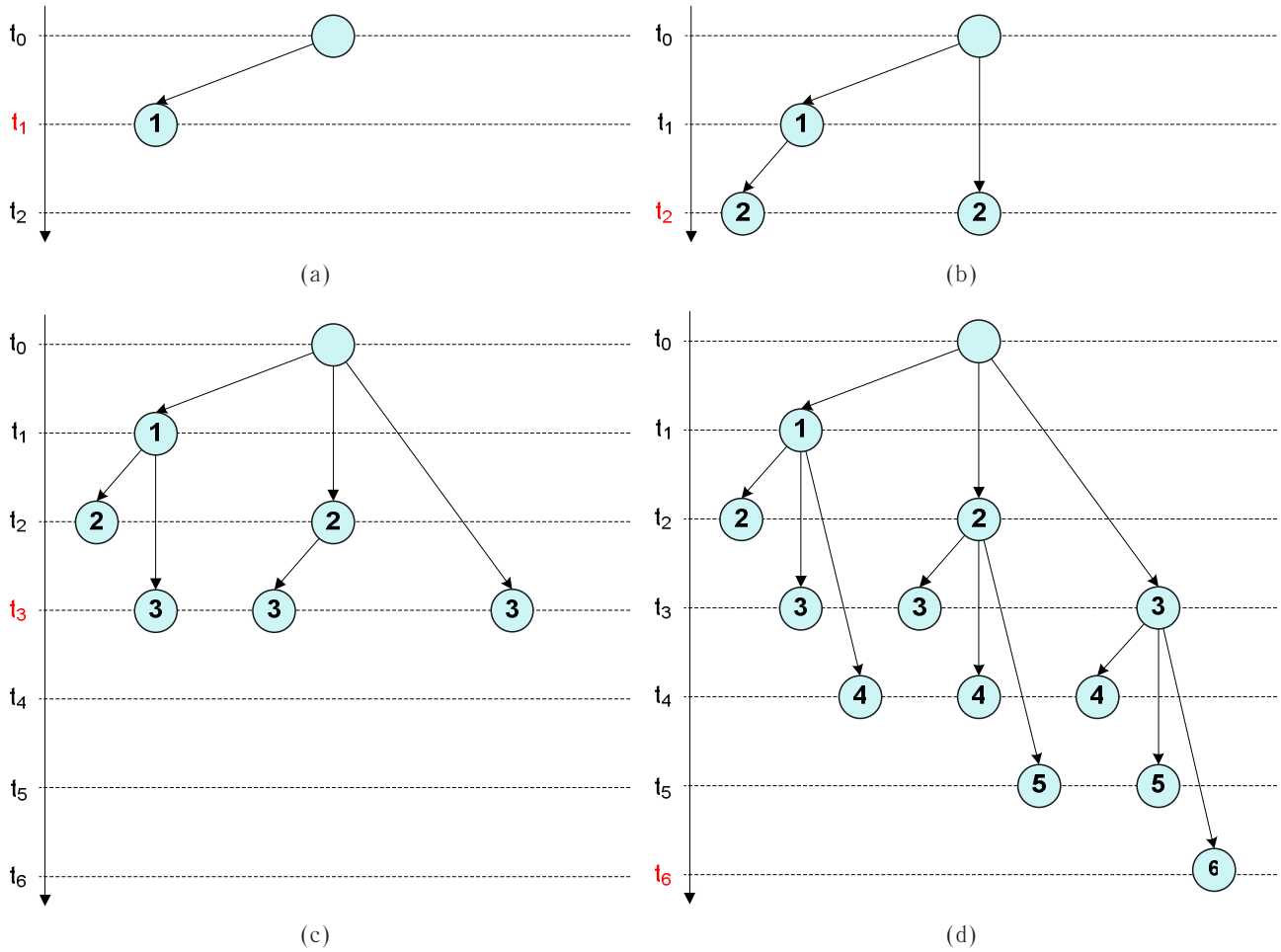

Fig. 1. Perfect hit-list spreading model: (a) The 1st perfect spread step infecting one node; (b) The 2nd perfect spread step infecting two nodes; (c) The 3rd perfect spread step; (d) The last 6th perfect spread step finishing infection of the whole tree.

In general, the number of discrete spread steps required to reach all k-way tree nodes in perfect spread model is given by the equation:

$$\Lambda(k, N) = k \times D(k, N), \tag{1}$$

where $D(k, N)$ is the depth of k-way tree and N is the total number of nodes.

Proof: The rightmost node of the $1st$ level is reached within k steps. The rightmost node of the $2nd$ level is reached within $k+k$ steps since the rightmost node of $1st$ level reaches its leftmost child within $k+1$ steps, the next child within $k+2$ steps and the last child within $k+k$ steps. This becomes obvious from Figure 1. In general, the rightmost node of the $i-th$ level is reached in $\underbrace{k+k+...+k}_{D}$ steps which is $k \times D$. ■

The total number of nodes N is:

$$N = 1 + k + ... + k^D = \frac{k^{D+1} - 1}{k - 1}$$

The depth $D(k, N)$ of a k-way tree consisting of N nodes is therefore given by the equation:

$$D(k, N) = \log_k [N(k-1) + 1] - 1 = \log_k \left[\frac{(N-1)(k-1)}{k} + 1 \right]$$

And the total number of infection steps required to infect the whole hit-list is given by the equation below and is represented by Figure 2. It shows that the minimum number of steps required to infect a hit-list of

somewhat close to the total number of hosts available in Internet is slightly larger than 40 and is achieved by the worm spreading across 3-way tree constructed from the hit-list.

$$\Lambda(k,N) = k \times \log_k \left[\frac{(N-1)(k-1)}{k} + 1 \right] \tag{2}$$

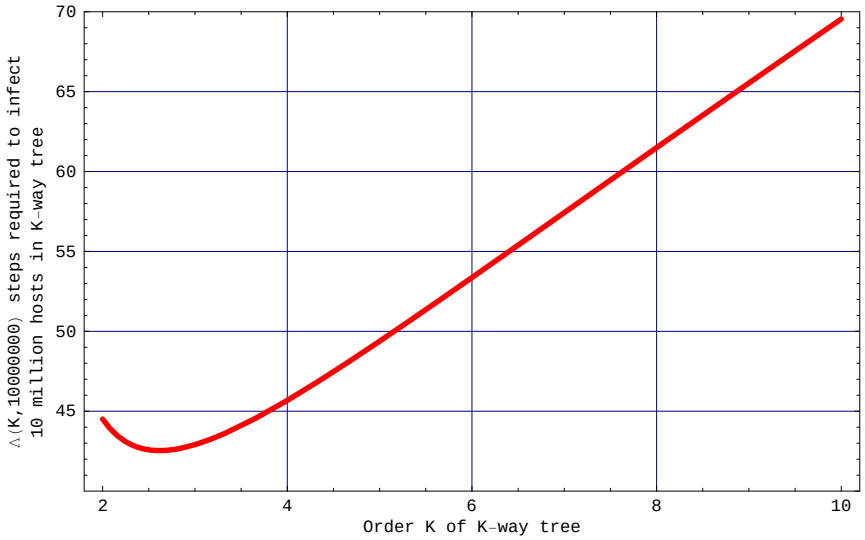

Fig. 2. Number of steps $\Lambda(k)$ required to infect 10 million vulnerable computers (close to a total number of Internet hosts) according to perfect spread model

To estimate the whole hit-list infection time we need to estimate the duration of each infection step which is the ratio of the total size of the worm $\overline{w}$ consisting of a constant payload W and a fraction of the hit-list $4A$ averaged over the whole spread tree to average Internet speed of infected host which is about `1 Mbps` according to the analysis of the spread of the Witty worm [7]. The averaged size of the worm thus is governed by:

$$\overline{w} = \frac{\sum_{i=0}^{D} k^i \left(\frac{4A}{k^i} + W \right)}{\sum_{i=0}^{D} k^i} = W + \frac{4A(D+1)(k-1)}{k^{D+1}-1} \tag{3}$$

This equation represents averaged worm instance size over the spread tree implying that each worm instance random scans hit-list for next k victims and divides its hit-list fraction into k equal parts before infecting its children. It gives $84\,bytes$ of the averaged size of a 10 million hit-list over 3-way tree. But if consider resilient spreading with full infection doubling of $(i+1)-th$ level nodes (Figures 3 and 4) then $i-th$ level nodes must have the addresses of the other subtrees to double infections. In this case more complex averaging of a hit-list size should be considered and the average instance hit-list size will be larger.

Thus the averaged size of a heavy $15\,kB$ Sasser-like network worm [8] spreading over a hit-list with 10 million addresses is $15\,kB + 84\,bytes = 15084\,bytes$ The average duration of a single infection is given by $15084\,B/(1\,Mbps/8) = 0.12\,sec$ Thus the total time required to infect all 10 million vulnerable hosts in Internet-scale hit-list is the product of the average duration of infection $0.12\,sec$ and a number of steps required to spread across 14-level deep 3-way tree of 10 million hosts which is 43 (according to Figure 2) and is bounded by $5.16\,sec$. A Code Red $4\,kB$ [9] worm could potentially infect the same vulnerable population within $1.42\,sec$ and a lightweight SQL Slammer worm which had only $376\,bytes$ of payload [10],[11] within even $0.17\,sec$

III. Imperfect hit-lists

Hit-list spreading technique was first introduced in [5] and it can potentially allow an attacker to create a worm producing rapid Internet-scale epidemic. A total infection time is measured in minutes rather than in hours or days of previously released worms. It was further simulated in [6] that a flash UDP worm infects 1

million vulnerable hosts within slightly more than a second and a flash TCP worm within slightly more than 3 seconds. But even with such a high infection speed a worm doesn't reach a fraction of vulnerable hosts in the hit-list. This depends on a fraction of false positives, i.e. the actually invulnerable hosts in the hit-list. It was simulated in [6] that almost all hosts in 20-level binary tree remain uninfected even if 20% of hosts in a hit-list are invulnerable. A technique that each node doubles infection of sibling's children was suggested by the authors. It allows reducing infection failure rate at the same invulnerability rate. But almost all hosts are not reached by the worm at more than 40% of invulnerable hosts.

Infection failure rate τ grows as the $D-th$ order of invulnerability rate σ where D is the depth of the propagation tree and therefore is increased significantly in deep trees. This is due to that each whole subtree remains uninfected when an intermediate root node of the subtree is actually invulnerable and left uninfected in the propagation tree during worm spreading.

A. Full infection doubling

The main reason why hit-list spreading worms fail on imperfect hit-lists is the precomputing of a propagation tree before spreading and strictly following the tree when spreading. Each vulnerable address has its subtree in the propagation tree and when a particular instance reaches a host with this particular address, it can spread strictly over the subtree of this address. Fixed propagation tree means that each node can have only one particular address from the hit-list. If worm fails to infect this node due to its invulnerability the whole subtree remains uninfected. S. Staniford *et al.* [6] suggested a solution to break fixed propagating across the tree by doubling up the worm spread path by infecting children of the sibling node after infection of its own children. In this case any node is not reached by the worm if both its parent and a sibling of the parent are uninfected.

Following [6] we consider a more general solution in which all nodes of $i-th$ level infect children of each node of the same level. This is clearly shown by Figure 3.

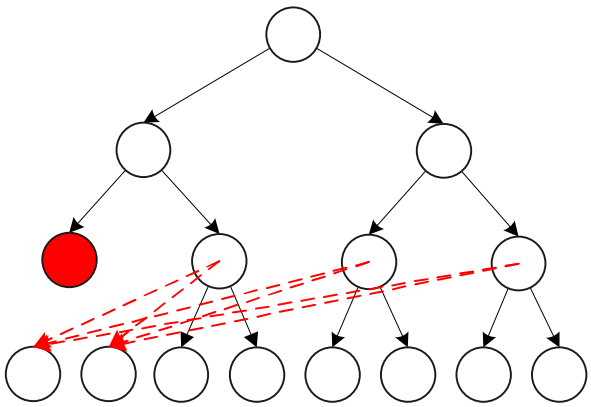

Fig. 3. All nodes infect children of each node of the same level

The probability $\tau_i(\sigma)$ that a particular address is uninfected depending on invulnerability rate satisfies the equation (1).

$$1-\tau_i = (1-\sigma)(1-\sigma^{2^{i-1}})(1-\sigma^{2^{(i-1)-1}})...(1-\sigma^{2^0}) \times 1 \tag{4}$$

An averaged infection failure rate over all layers of k-way tree of depth $D \equiv i_{max}$ and consisting of N nodes is therefore calculated:

$$\tau(\sigma) = \frac{1}{N}\left(\sigma + \sum_{i=0}^{D} k^i \left[1-(1-\sigma)\prod_{k=1}^{i}(1-\sigma^{2^{k-1}})\right]\right) \tag{5}$$

Figure 4 shows that even with the described above resilience technique imperfect hit-lists cause high infection failure rate. Obviously the technique complicates the worm and it is likely to increase the size of the worm.

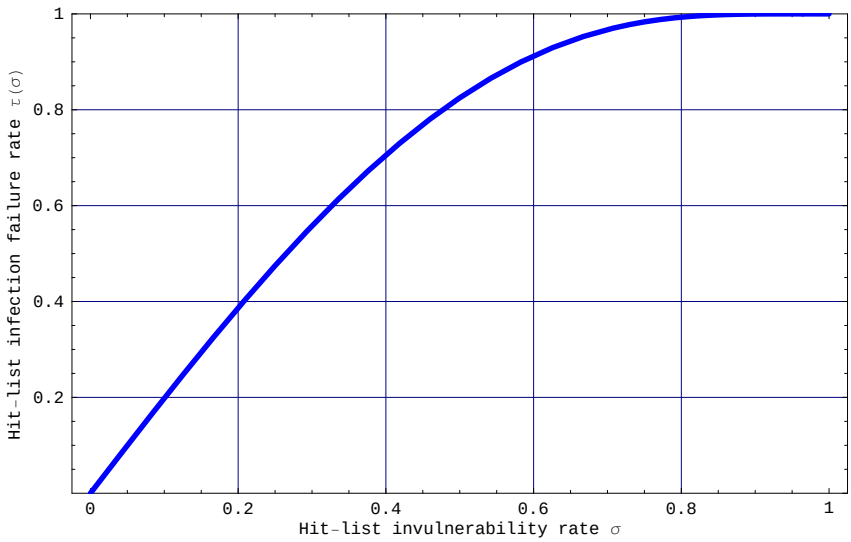

Fig. 4. Averaged infection failure rate in a 20-level binary tree with infection by all upper-level nodes

Obviously this gives a significantly better picture of a total failure infection rate which reaches almost 100% of failures only at 80% of invulnerable addresses in a hit-list. Despite rather satisfactory result the disadvantages of such resilience technique are the significantly complicated spread logic, increased size of a hit-list fraction transmitted from instance to instance (please refer to the next section) and it still leaves the worm with a large number of infection failures.

B. Run-time spread tree

It's worth considering random scanning worms which are not affected by imperfect hit-lists since they do not spread over the fixed spread tree. Each address is chosen randomly which can be considered as creating spread tree at run-time. Each next victim is chosen from the whole address space which results in an increasing number of vain re-infections and therefore small spreading speed unlike fast spreading across a hit-list.

The suggestion to avoid failing due to imperfect hit-lists as in random scanning and retain a high spread speed of hit-list technique is to use *run-time spread tree*. This spreading technique implies that each worm instance randomly chooses the next address to be infected from the prepared hit-list hence a spread tree is created from the hit-list at worm spread-time. If the chosen address is actually invulnerable and the worm instance fails to infect it, the address is wiped out the hit-list and randomly chooses the next address from the remainder of the hit-list.

Obviously each worm instance must check first if the next randomly chosen address is vulnerable, wait for the result and only after that infect it. In this infection technique the infection failure rate $\tau(\sigma)$ depends linearly on the invulnerability rate σ. Before infecting k children in k-way spread tree a worm instance divides its portion of a hit-list (further *instance hit-list*) into k fractions. Choosing the next address randomly from the hit-list was mentioned in [5] but spreading over precomputed spread tree was considered in [6].

IV. Conclusion

The reappearance of fast network worms can become a great threat to the whole Internet and for quickly growing 3G mobile networks. This work introduces a spread model based on hit-list which can help worm containment policies and systems to estimate a time required for fast hit-list worms to spread over the vulnerable population of Internet hosts. Any containment system should consider the results obtained for hit-list infection time when this worm capability becomes crucial. A "good worm" concept is very questionable but if it find its application in worm containment and vulnerability patching then it's worth considering the conqueror worm as a model for developing a "good worm".

Acknowledgment

The author would like to thank Yury Mashevsky and Eugene Kaspersky (Kaspersky Lab), Dr. Valentin Aphanasiev (Laboratory of Data Analysis, Error Correction Codes and Cryptology of Institute of Information Transmission Problems (IPPI) of Russian Academy of Sciences) for helpful comments and discussions and Dr. Fernando C. Colon Osorio (Wireless Systems Security Research Laboratory) for final comments on this paper.

References

[1] M. Overton, "Bots and Botnets: Risks, Issues and Prevention," in *Proceedings of 15th International Virus Bulletin Conference 2005*, Dublin, Ireland, Sep. 2005, http://arachnid.homeip.net/papers/VB2005-Bots_and_Botnets-1.0.2.pdf

[2] Worm Blog by J. Nazario, http://www.wormblog.com

[3] Kaspersky Lab White Paper, "The Changing Threat: From Pranksters to Professionals", 2005, http://www.kasperskyusa.com/promotions/wp_index.php

[4] N. Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues," http://www.cs.berkeley.edu/~nweaver/warhol.html

[5] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in Your Spare Time," in *Proceedings of the 11th USENIX Security Symposium (Security'02)*, San Francisco, CA, pp. 149-167, Aug. 2002, http://www.icir.org/vern/papers/cdc-usenix-sec02/

[6] S. Staniford, V. Paxson, and N. Weaver, "Top Speed of Flash Worms," in *Proceedings of the 2nd ACM Workshop on Rapid Malcode (WORM)*, 2004, http://www.caida.org/outreach/papers/2004/topspeedworms/topspeed-worm04.pdf

[7] C. Shannon, D. Moore, "The Spread of the Witty Worm," *IEEE Security & Privacy*, vol. 2, no. 4., Jul./Aug. 2004, http://www.caida.org/outreach/papers/2004/witty/mal.xml

[8] *Viruslist.com*, http://www.viruslist.com/en/analysis

[9] D. Moore, C. Shannon, and J. Brown, "Code-Red: a case study on the spread and victims of an Internet worm," in *Proceedings of the 2nd ACM Internet Measurement Workshop 2002*, Marseille, France, pp. 273-284., Nov. 2002, http://www.caida.org/outreach/papers/2002/codered/codered.pdf

[10] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer Worm," *IEEE Security & Privacy*, vol. 1, no. 4, Jul./Aug. 2003, http://www.computer.org/security/v1n4/j4wea.htm

[11] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford and N. Weaver, "The Spread of the Sapphire/Slammer Worm," http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html

[12] T. Vogt, "Simulating and optimising worm propagation algorithms," Sep. 2003 (Updated on Feb. 2004), http://web.lemuria.org/security/WormPropagation.pdf