

Worms - survey and propagation

Pankaj Kohli

MS by Research - Computer Science and Engineering
International Institute of Information Technology
Hyderabad, India
E-Mail: pankaj_kohli@research.iiit.net

Abstract. Cellular automata has been used to model network flow dynamics and associated attacks [2]. This paper presents a model for internet worm propagation for random scanning worms using a probabilistic cellular automata. We then compare the results produced by this model with those given by the actual data for the Code Red v2 worm. Experimental results show that our model can effectively capture the propagation of random scanning worms.

1 Introduction

1.1 Worms

A worm is a self-replicating computer program, similar to a computer virus but unlike a virus which attaches it to, and becomes part of, another executable program, a worm is self-contained and does not need to be a part of another program to propagate itself. Whereas a virus attaches itself to a host program, a worm spreads by exploiting security flaws in the services on a network. The term ‘worm’ was first used for a computer program when, in 1978, two researchers, John F Shoch and John A Hupp, of Xerox PARC, wrote a piece of self replicating code to find idle machines on the network and assign them tasks, sharing the processing load. The first worm to appear on a worldwide network was the *Christmas Tree* worm, spreading across both IBM’s own international network (*known as VNET*) and BITNET sites in December 1987. The first worm that appeared on the Internet was the famous *Morris Worm*, written by Robert Morris, a graduate student at Cornell University, on November 2, 1988. The worm exploited a buffer overflow in Sendmail and Finger daemons. Although not written to cause any damage, a bug in the *Morris worm* allowed the worm to reinfect the same server several times, consuming CPU resources with each new copy of the worm, and causing the world’s first Denial of Service (*DOS*) attack.

1.2 Classification

For a worm to infect a target, it must first locate a target. A worm may use any of the following strategies to locate its target.

Random Scanning A random scanning worm generates a random IP address using a pseudorandom number generator. Thus every host on the network is equally likely to be scanned. Random scanning was used by worms such as *Code Red v2* and *Slammer*.

Localised Scanning A worm employing localised scanning preferentially scans for vulnerable hosts on the local subnet. An intuition for such a strategy is that vulnerable hosts are clustered, and localised scanning can rapidly compromise all the vulnerable hosts on the same subnet.

Sequential Scanning A worm using sequential scanning scans IP addresses sequentially. After the worm compromises a vulnerable host, it checks the host next to this vulnerable host. *Blaster* worm employed sequential scanning.

Topological Scanning The worm relies on the local information contained in the compromised hosts to locate new targets. Local information includes /etc/hosts file, email addresses etc. Topological scanning was used by *Morris* worm.

Hitlist Scanning The worm writer gathers a list of potentially vulnerable hosts beforehand, which are targeted first when the worm is released. This speeds up the spread of the worm at an initial stage. Hitlist scanning was used by *Slammer* worm.

1.3 Cellular Automata

Cellular Automata are simple models of interacting automata for modelling self reproduction in machines. They have been adapted to model complex systems built from interactions of multiple components. Formally a cellular automaton can be defined as follows.

Definition 1. A k -dimensional cellular automaton consists of

- A set of cells C .
- A lattice $\Lambda \subseteq \mathbf{Z}^k$
- A 1:1 mapping $\eta : C \rightarrow \Lambda$ assigning a position to each cell.
- A state space Q and a function $\varphi : Q^r \rightarrow Q$; $r \leq k$

The function φ is called the update rule of the cellular automaton. The parameter r defines the neighbourhood of the cellular automaton. An element $q \in Q$ is called a configuration of the cellular automaton. If $Q_0 \in Q$ is the starting configuration, then the evolution of a cellular automaton A proceeds by repeated application of the update rule. Thus the evolution of A is the sequence $\langle Q_0, \varphi(Q_0), \varphi^2(Q_0), \dots \rangle$. Each configuration of the sequence is called a generation of the cellular automaton. A cellular automaton in which the rules are probabilistic, rather than deterministic, is known as a *Probabilistic Cellular Automaton*.

2 Worm propagation model using cellular automata

To model propagation of random scanning worms, we use a one-dimensional probabilistic cellular automaton with N states, N being the number of vulnerable hosts. The neighbourhood of any cell is N . Here we assume that every machine on the internet is reachable from every other machine, which may not be the actual case, but it helps in the simplification of the model. The state space of the cellular automaton is $Q = \{V, I, NV\}$ where the states V, I , and NV denote the vulnerable, infected and non-vulnerable machines. Only the following transitions are allowed:

- $\mathbf{V} \longrightarrow \mathbf{I}$ This transition represents a machine is being infected by the worm.
- $\mathbf{V} \longrightarrow \mathbf{NV}$ This transition represents a vulnerable machine is being patched, so it becomes non-vulnerable.
- $\mathbf{I} \longrightarrow \mathbf{V}$ This transition represents death of a machine, so it again becomes vulnerable.
- $\mathbf{I} \longrightarrow \mathbf{NV}$ This transition represents an infected machine is being patched, so it becomes non-vulnerable.

Let q_k^i be the state of cell k at generation i , p be the patching rate and d be the death rate. We define update rules of the above cellular automaton as follows:

If $q_k^i = I$ then

$$q_k^{i+1} = \begin{cases} V & \text{with probability } d, \\ NV & \text{with probability } p. \end{cases}$$

If $q_k^i = V$ then

$$q_k^{i+1} = \begin{cases} I & \text{with probability as calculated below,} \\ NV & \text{with probability } p. \end{cases}$$

If $q_k^i = NV$ then

$$q_k^{i+1} = \{ NV \text{ with probability } 1. \}$$

Theorem 2. *At generation $i + 1$ of the cellular automaton, the total number of cells in state I is*

$$n_{i+1} = (1 - d - p) n_i + \left[(1 - p)^i N - n_i \right] \left[1 - \left(1 - \frac{1}{2^{32}} \right)^{sn_i} \right] \quad (1)$$

where s is the scanning rate of the worm.

Proof. At generation $i + 1$ of the cellular automaton, the total number of scans is sn_i where n_i is the number of infected machines or the number of cells in state I at generation i .

$$\Pr[\text{a machine is hit by one scan}] = \frac{1}{2^{32}}$$

$$\Pr[\text{a machine is not hit by one scan}] = \left(1 - \frac{1}{2^{32}}\right)$$

$$\Pr[\text{a vulnerable machine being infected by the } j^{\text{th}} \text{ scan}] =$$

$$\Pr[\text{machine is not scanned in first } (j - 1) \text{ scans}] \cdot \Pr[\text{machine is hit by the } j^{\text{th}} \text{ scan}]$$

$$= \left(1 - \frac{1}{2^{32}}\right)^{j-1} \cdot \frac{1}{2^{32}}$$

$$\Pr[V \longrightarrow I \text{ transition from generation } i \text{ to } i + 1] =$$

$$\Pr[\text{a vulnerable machine being infected in } sn_i \text{ scans}] =$$

$$\sum_{j=1}^{sn_i} \Pr[\text{machine is infected in } j^{\text{th}} \text{ scan}]$$

$$= \sum_{j=1}^{sn_i} \left[\left(1 - \frac{1}{2^{32}}\right)^{j-1} \cdot \left(\frac{1}{2^{32}}\right) \right]$$

$$= \frac{1}{2^{32}} \sum_{j=1}^{sn_i} \left(1 - \frac{1}{2^{32}}\right)^{j-1}$$

$$= \frac{1}{2^{32}} \left[\frac{1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_i}}{1 - \left(1 - \frac{1}{2^{32}}\right)} \right]$$

$$= 1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_i}$$

$$\text{Number of cells in state } V \text{ or } I = (1 - p)^i N$$

$$\text{Number of cells in state } V = (1 - p)^i N - n_i$$

Therefore, from generation i to $i + 1$,

$$\text{Number of } V \longrightarrow I \text{ transitions} = \left[(1 - p)^i N - n_i \right] \left[1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_i} \right]$$

$$\text{Number of } I \longrightarrow V \text{ transitions} = dn_i$$

$$\text{Number of } I \longrightarrow NV \text{ transitions} = pn_i$$

Therefore, number of machines that remain infected from generation i to $i + 1 = n_i - pn_i - dn_i = (1 - p - d) n_i$

Hence, at generation $i + 1$ number of cells in state I is given by

$$|I| = (1 - p - d) n_i + \left[(1 - p)^i N - n_i \right] \left[1 - \left(1 - \frac{1}{2^{32}}\right)^{sn_i} \right]$$

Number of cells in states V and NV are given by

$$|V| = (1 - p)^i N - n_i \tag{2}$$

$$|NV| = N - (1 - p)^i N \tag{3}$$

The above theorem can also be proved using induction, as done in [4].

3 Model Evaluation

Each generation i in the above model represents the state of the network at time tick i . To evaluate the model, *entropy* [2] can be used. Entropy is a measure of disorder of a system. Let A be a cellular automaton with state space $Q = \{q_0, q_1, \dots, q_n\}$. Let N_q^n be the number of cells with state $q \in Q$ in generation n . Then the entropy of generation n is given by

$$H(A, n) = \sum_{q \in Q} -\frac{N_q^n}{|C|} \log \left(\frac{N_q^n}{|C|} \right) \quad (4)$$

Here each generation of the cellular automaton represents one time unit. For the above model, a low value of entropy means that a large fraction of the total number of cells are in the same state. Thus, we can compute entropy at each generation and construct an entropy chart, which can be compared to the entropy chart constructed using actual data for Code Red v2 worm. Figure 1 shows the simulation of propagation of the worm using the above model. Figure 2 shows the propagation of Code Red v2 worm using actual data.

3.1 Simulation of Code Red v2 worm

On July 19th, 2001, the Code Red v2 worm infected more than 359000 computers in less than 14 hours. This worm spreads by probing random IP addresses and affecting all the hosts that are vulnerable to IIS exploit. CAIDA [6] collected real data to measure the spread of Code Red v2 worm. The data were collected from two locations: one /8 network at UCSD and two /16 networks at Lawrence Berkeley Laboratory (LBL). In these data, hosts were considered to be infected if they sent TCP SYN packets on port 80 to nonexistent hosts on these networks. For the simulation of the Code Red v2 worm, we assume that there are 500000 vulnerable machines on the Internet, the worm starts on a single machine, it performs 2 scans per second, and takes one second to infect a machine. Patching rate and death rate were taken as 0.000002 /second and 0.00002 /second respectively. The simulation performed is for the 115200 generations of cellular automata, which means for first 32 hours of worm propagation, if each generation is assumed as 1 second. Figure 1 show the results of the simulation of the worm using cellular automaton. The initial rise in the entropy is due to the increase in the number of infected hosts. Entropy at the peak level indicates the point at which the disorder is maximum. After this point, the entropy starts decreasing, as more and more number of cells change their state to I . Figure 2 shows the propagation of Code Red v2 worm using actual data. As more and

more number of hosts become infected, disorder increases, thereby causing an increase in the entropy. The peak indicates the point of maximum disorder. After this point, the entropy starts decreasing as more and more number of hosts enter into infected state.

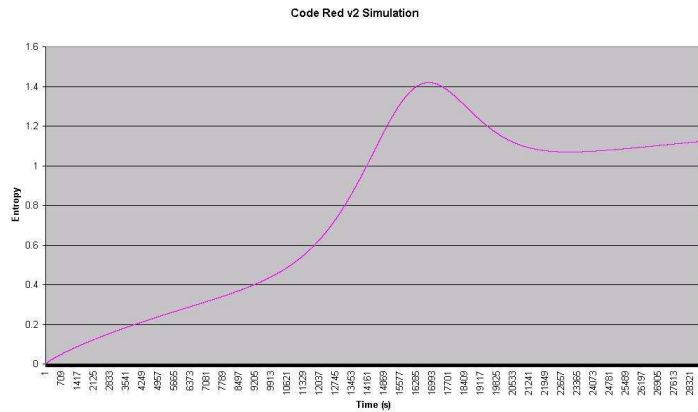


Fig. 1. Simulation of Code Red v2 worm (500000 vulnerable machines starting on a single machine, a scanning rate of 2 scans/sec, patching rate of 0.000002 /sec, death rate of 0.00002 /second, and a time period of 1 sec to complete infection)

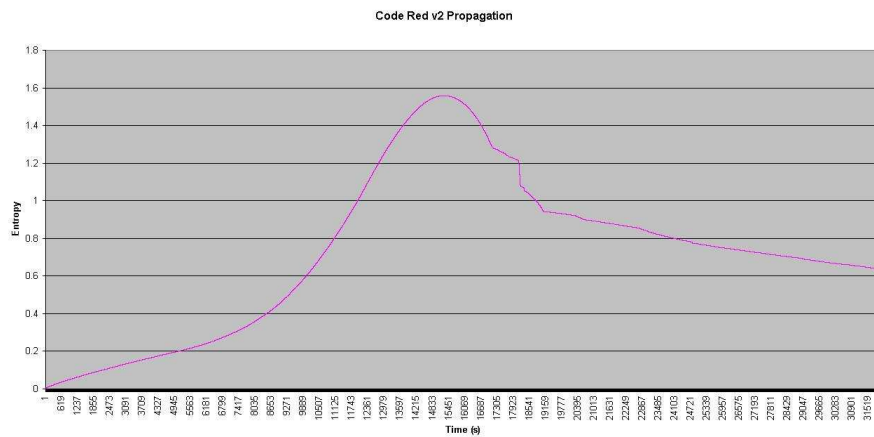


Fig. 2. Code Red v2 worm propagation using real data

4 Conclusion

Cellular automata has been used to model network dynamics and can be extended to model the propagation of random scanning worms. The deviations from actual results shown by the above model can be due to the following reasons.

- The above model does not take into consideration the size of the worm, which adversely affects the rate of propagation of a worm.
- The above model cannot differentiate between worms which use TCP and the ones which use UDP, both of which may have different propagation rates.
- The above model assumes that every machine on the Internet is reachable from every other machine, which is not a genuine assumption. The above model ignores NAT.
- Deviations from predicted results may also be due to experimental error.

References

1. Chowdhury D.; Guttal V.; Nishinari K.; Schadschneider A., A cellular-automata model of flow in ant trails: non-monotonic variation of speed with density, *Journal of Physics A: Mathematical and General*, Volume 35, Number 41, 2002, pp. L573-L577(1)
2. Brooks, R.; Orr, N.; Zachary, J.; Griffin, C., “An interacting automata model for network protection,” *Information Fusion*, 2002. Proceedings of the Fifth International Conference on , vol.2, no.pp. 1090- 1097 vol.2, 2002
3. Craig Fosnock, *Computer Worms: Past, Present and Future*, (2005).
4. Chen, Z., Gao, L., Kwiat, K.: Modeling the spread of active worms. Proceedings of IEEE INFOCOM 2003. (2003)
5. Stephan Wolfram, *Cellular Automata as Models of Complexity*, <http://www.stephenwolfram.com/publications/articles/ca/84-cellular/index.html>
6. The CAIDA Dataset on Code-Red Worms - July and August 2001, David Moore and Colleen Shannon, http://www.caida.org/data/passive/codered_worm_dataset.xml