Issues in modeling user behavior in computer virus propagation

Stefano Zanero

Post-doc Researcher
Dipartimento di Elettronica e Informazione
Politecnico di Milano
zanero@elet.polimi.it

WTCV, Nancy, France, 05/05/2006

Outline

- Introduction: the Vierika case
- A look at the code
- The court's questions
- Modeling the behavior of the Vierika worm
- Simulation results
- Conclusions and future work

Introduction: the Vierika case

- March 20, 2001: "Vierika" worm released. Non-notable, but italian self-confessed author was convicted.
- During the trial, we were called upon as expert witnesses for a technical analysis of this worm
- Key point: to analyze the destructive potential of the Vierika worm
- We cannot discuss in detail some aspects, due to the case still being considered for repeal, but we can report on our technical analysis, on the techniques and theories we applied, and on the forensic methodology we adhered to
- In particular, in this workshop we will focus on the issues in modeling this particular worm specimen

A look at the code

- Let's have a look at the code of the worm
- Vierika is written in Visual Basic Script (powerful Windows scripting language)
- Vierika has a curious, two-stage propagation mechanism:
  1. An e-mail attachment (`Vierika.JPG.vbs`), which sets "home page" to `http://web.tiscalinet.it/krivojrog/vierika/Vindex.html`, and lowers "Security Settings" of the "Internet Zone" to "low" level
  2. The above named web page, which contains the instructions to create and propagate a copy of the above-described attachment

The questions to be answered (in brief)

- The target platform: police investigators got it wrong ("the totality of the computers being sold today"): just Windows + Outlook + IE machines
- Lowering "Security Settings" is akin to "unauthorized access to computer systems" ?
- Does the worm execute "without the user consent" ?
- Does the fact that the worm mass-mail itself "disclose" confidential data ?
- Is the worm **dangerous**?

Modeling Vierika's propagation: motivations

- A successful pathogen tries not to destroy the hosts it uses for propagating (seen as early as the first Internet Worm analyses [1], but particularly demonstrated by the Witty worm). Vierika is no exception.
- As studied in [2, 3, 4, 5] the structural threat to the Internet stability is directly proportional to the worm propagation speed and to its ability of saturating network resources
- Difference between the behavior of mass mailers (such as Vierika), TCP worms (such as Code Red, [5]), "flash" [6] or UDP worms (e.g. Slammer, [7]).
- Estimating dangerousness means modeling worm propagation

Overview of propagation modeling

- Evolution of models:
  - Biological models (see, e.g. [8])
  - Traditional viruses: in [9] a traditional SIS model is transferred onto a directed random graph
  - Mail worm propagation: Zou et al. [10]
  - TCP worms: Random Constant Spread (RCS) model [6] developed by Staniford, Paxson and Weaver. In [4] a discrete time model for worms, but limited benefit vs. heavy computation
  - UDP worms: must account for bandwidth bottlenecks, compartment-based model [2]
- Some problems still unresolved (e.g. multimodal worms, new location-bound worms, etc.)

Basics of e-mail worm propagation

- Most famous model of e-mail propagation in [10]:
  - E-mail modeled as an undirected graph of relationship between people. Node degree generated with a power-law probability function, and a small world topology.
  - Each user "opens" an incoming virus attachment with a fixed probability P_i, a function of the user but constant in time.
  - E-mail checking time T_i is modeled as either an exponentially or Erlang distributed random variable.
  - $T = E[T_i]$, and $P = E[P_i]$ are assumed to be independently distributed gaussians.
- Interesting observations:
  - since user e-mail checking time is much larger than the average e-mail transmission time, the latter can be disregarded
  - the overall spread rate of viruses gets higher as the variability of users' e-mail checking times increases, and depends mostly on $T = E[T_i]$

Modeling Vierika's propagation: issues

- Issues in Zou's model
  - Fixed open probability
  - small world assumption averages out the effect of the existence of interest groups and organizations
  - power law distribution of the node degree is based on experimental observations on mailing lists, not on real address books
- More importantly: the two-stage propagation mechanism of Vierika needs some further elaborations: we must take into account both the e-mail check time and a "web access time", since the worm propagation routine is activated only when the user launches his browser to the home page

Estimating simulation parameters (1)

- A user is a tuple characterized by the following variables:

  Check Probability (P_c) : the probability that a user checks his e-mail

  Open Probability (P_o) : the probability that a user carelessly opens an infected attachment

  Home Probability (P_h) : the probability that a user opens his browser on the homepage

  Contact List Size (C) : the size of the address book of the user

- Greenfield Online: USA, March 2000, 1.000 respondents [11]: 27% unaware that attachments can be malicious, 19% had opened a malware attachment at least once. Of these, 45% 1 to 2 times, 34% 3 to 4 times, 10% 5 to 6 times, 11% an astounding 7 or more times.
- No reliable stats on checking time: we compared various studies [12, 13, 14, 15] and created the following Table

Estimating simulation parameters (2)

Percentage of users	Mean Checking Time ($E[T_i]$)
10%	$E[T_i] \simeq 24$ hours
20%	$E[T_i] \simeq 12$ hours
45%	$E[T_i] \simeq 60$ minutes
25%	$E[T_i] \simeq 10$ minutes or less

Table: Composition of a reasonable synthetic user population for simulating year 2001

- No such data is available for web surfing
- We assumed that there's a relationship between P_c, P_o, P_h and C (skilled users vs. casual users)

Generating a synthetic population

- In order to generate the i-th syntetic user, we first generate a random number x_i following a normal distribution: $X \sim N(\mu_X, \sigma_X^2)$. We then generate the user tuple as follows:
  - $P_{c,i}$ as a random number, normally distributed following $N(\alpha_c x_i, \sigma_c^2)$
  - $P_{o,i}$ as a random number, normally distributed following $N(\alpha_o / x_i, \sigma_o^2)$
  - $P_{h,i}$ as a random number, normally distributed following $N(\alpha_h x_i, \sigma_h^2)$
  - C_i as a random number, distributed following a Pareto distribution $Par(\alpha, x_i)$
- Explanation: a user with a "high" pivot will, on average, check his e-mail more often, open attachments more carefully, and open his browser more often; and vice-versa.
  $\alpha_c, \sigma_c, \alpha_o, \sigma_o, \alpha_h, \sigma_h, \alpha$ are all fixed parameters of the simulation.

Simulation tool and parameters

- Simulation tool written in C++ code, uses a Mersenne-Twister random number generator
- Considering the limited diffusion of Vierika (i.e. 24 hours), the tool must reproduce in detail the spreading process, using an high granularity model to perform sensitivity tests
- Effects of immunization, antiviruses and so on can be ignored
- A birth-death model, with 3 states: *Contacted*, *First Stage* and *Second Stage*, simulated step-by-step in discrete time

C	100	α	1.0
μ_X	10.0	σ_X	5.0
α_o	7.0	σ_o	0.5
α_c	3.0	σ_c	5.0
α_h	5.0	σ_h	7.0

Table: Parameters used for simulations

Simulation results (1)

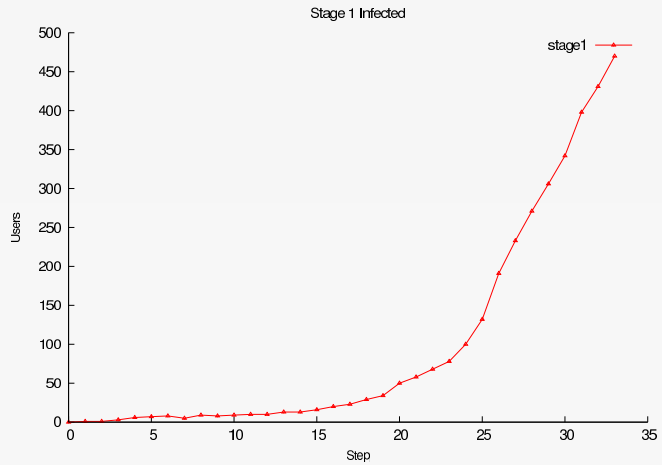

Figure: Number of users who opened the attachment

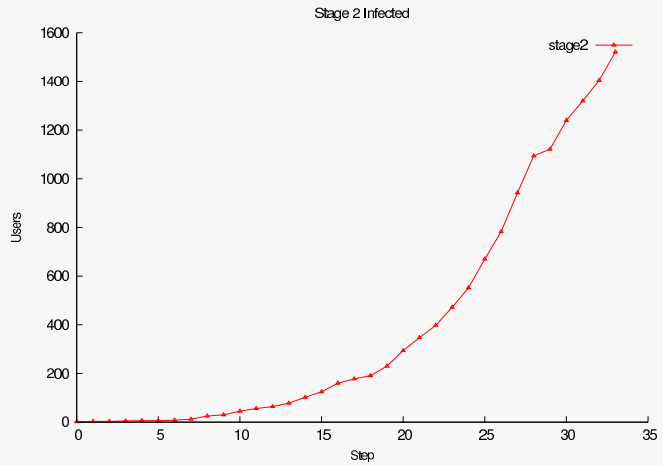

Figure: Number of users who triggered the second level infection

Simulation results (2)

- Users in *Second Stage* can re-open the webpage many times, they won't infect additional users but they will increase sent mail and webpage hit counters
- Infection exhibits a slow, but still exponential, progression, as expected
- We have a fixed point: 1500 page hits in slightly less than 24 hours. 16 hours were approximately needed, on average, to reach the limit.
- Average after 24 hours: 360.000 users *contacted*, 620 /textitFirst Stage infected, 2.000 /textitSecond Stage infected. Average of 450.000 mails sent.
- 1/3 of the simulations did not create an epidemy
  - 80% of these sent less than 300 emails in total before dying
- This is a good, per excess simulation of the early propagation of Vierika

Sensitivity

- Since model is very empirical and makes guesses about parameter values, sensitivity tests are needed
- Varying α_c with fixed σ_c does not impact epidemic threshold or average number of infected users, but changes the speed of the infection (in a linear way, around our set of parameters)
- Varying α_o with fixed σ_o does not impact epidemic threshold or average number of infected users, but changes dramatically the speed of the infection, increases the number of emails sent, and generically makes the virus more efficient
- Varying the initial population impacts the number of successful outbreak in the expected way. We know Vierika begun its spreading from alt.sex.binaries, but we cannot estimate this on the birth/death model
- Generation of a huge cluster on the graph causes invariably the outbreak to happen: topology needs to be evaluated properly, but no data really available

So, was that the end-of-the-Civilization-as-we-know-it worm?

- **NO WAY !**, and it couldn't really be
  - TCP-based worms: bound by network latency
  - E-mail worms: this + bottleneck of mail server availability
  - UDP worms: purely bandwidth-limited, saturate connections and create widespread network outages [2]
- Therefore Vierika belongs to the lamest type of worm, and is even more limited by the web page availability and the web browsing habits!

Worm Type	Example	Peak diffusion	Time to peak	Damages
Mail + Web	Vierika	TBD	days	TBD
Mass-mailer	Melissa	100.000	2 days	None
Mass-mailer	LoveLetter	1.000.000	4 hours	Mail servers overload
TCP-based	Code Red	359.000	14 hours	Firewall overload
UDP-based	SQL Slammer	75.000	10 minutes	Random network outages
UDP-based	Blaster	8.000.000	n.a.	Widespread network outages

Conclusions and future works

- We have demonstrated, by means of simulation, that the impact of a mail worm like Vierika with a dual component structure is significantly lower than the impact of a traditional e-mail based worm
- These considerations likely helped to make the point with the judge, who fined the author but did not sentence him to prison... (under repeal)
- The webpage mechanism looks like a "shutdown button" to me...
- To refine the model a number of parameters are missing! How do we retrieve them?
- Small numbers are difficult to deal with, even if simulating worm behavior: sensitivity on parameters too high
- Nowadays heterogeneity of browsers and mailers is higher: a worm based on the "two stage" mechanism would be far less harmful, if focused on just one platform.

References I

[1] E. H. Spafford.

Crisis and aftermath.

Communications of the ACM, 32(6):678–687, 1989.

[2] Giuseppe Serazzi and Stefano Zanero.

Computer virus propagation models.

In Maria Carla Calzarossa and Erol Gelenbe, editors, *Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures - MASCOTS 2003*. LNCS Springer-Verlag, 2004.

[3] Yang Wang and Chenxi Wang.

Modelling the effects of timing parameters on virus propagation.

In *Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM'03)*, Oct 2003.

[4] Zesheng Chen, Lixin Gao, and Kevin Kwiat.

Modeling the spread of active worms.

In *Proceedings of IEEE INFOCOM 2003*, 2003.

References II

[5] David Moore, Colleen Shannon, and Jeffery Brown.

Code-red: a case study on the spread and victims of an internet worm.

In *Proceedings of the ACM SIGCOMM/USENIX Internet Measurement Workshop*, Nov 2002.

[6] Stuart Staniford, Vern Paxson, and Nicholas Weaver.

How to 0wn the internet in your spare time.

In *Proceedings of the 11th USENIX Security Symposium (Security '02)*, 2002.

[7] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver.

The spread of the sapphire/slammer worm.

`http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html`.

[8] Herbert W. Hethcote.

The mathematics of infectious diseases.

SIAM Review, 42(4):599–653, 2000.

References III

[9] J. O. Kephart and S. R. White.
Directed-graph epidemiological models of computer viruses.
In *IEEE Symposium on Security and Privacy*, pages 343–361, 1991.

[10] Cliff Changchun Zou, Don Towsley, and Weibo Gong.
Email virus propagation modeling and analysis.
Technical Report TR-CSE-03-04, University of Massachussets, Amherst.

[11] `http://www.gfi.com/news/en/langreenfield.htm`.

[12] `http://www.cio-asia.com/pcio.nsf/0/BF633D26EB9A9E0248256B3E0024C44F?OpenDocument`.

[13] `http://www.better-business.co.uk/09aboutus/statistics.shtm`.

[14] `http://www.analysphere.com/20Aug01/email.htm`.

[15] Brad Knowles and Nick Christenson.
The design and implementation of highly scalable email systems.
In *Proc. of the 14th System Administration Conference - LISA 2000*. USENIX, December 2000.