Modular worms

Viktor S. Grichenko

Ural State University, Yekaterinburg, Russia,
gritzko@ural.ru

Abstract. CodeRed and Nimda internet worms clearly demonstrated today's internet vulnerability. Technologies currently used by worm writers allow serious paradigm shift: modular, reusable and upgradeable worm design. Wide acception of these principles will dramatically increase malware penetration ability and staying power.
Clear understanding of these possibilities will help to make adequate solutions on software design.
Keywords: internet worms, malware, self-propagating software

1 Introduction

> "MessageLabs, which now scans over three million emails per day on the internet, intercepted an average of 3.3 email viruses per minute throughout the year, or one every 18 seconds. In 1999, the rate was one an hour; in 2000 one every three minutes." (from *vnumet.com* article)

Computer viruses and worms remain significant and constant problem since wide deployment of PCs. Technical infrastructure becomes more complex and more networked thus increasing own vulnerability. During last two years several highly virulent worms of more or less complex structure appeared making us to think hard about new dangers.

One of those is modular worm design allowing worm to propagate through diverse communication channels, to update itself and to evolve in the wild, actively fighting for survival.

Today worm writers do employ more powerful paradigm of software deployment than software developers and users (just try to compare expenses, not businesses). This work is just one step towards adoption of this paradigm for peaceful goals.

2 Definitions

Definition 1. *A computer WORM is a self-contained program (or set of programs), that is able to spread functional copies of itself or its segments to other computer systems (usually via network connections). (from comp.virus FAQ)*

Definition 2. *A program that makes copies of itself, for example from one disk drive to another, or by copying itself using email or some other transport mechanism. It may do damage and compromise the security of the computer. It may arrive in the form of a joke program or software of some sort. (from Symantec glossary)*

Definition 3. *A worm is a program that can run by itself and can propagate a fully working version of itself to other machines. (Bob Page; from report on the Morris Worm)*

This paper uses following definitions (in the order of refinement) :

Definition 4. *Worm is self-replicating program traveling from host to host via some (network) communication space.*

Definition 5. *Multicomponent worm is a worm consisting of parts what are able to function separately (example: I-Worm.Badtrans has separate body and trojan horse binaries, I-Worm.Klez infects host by Win32.Klez virus)*

Definition 6. *Plugin-enabled worm is a worm able to dynamically download and run optional pieces of code called plugins. (I-Worm.Hybris)*

Definition 7. *Modular worm is a worm consisting of body and some theoretically unlimited number of optional parts (such as different exploits and payloads) that could be transferred from worm to worm. (Was not ever seen in the wild)*

Definition 8. *Fully decentralized (or fully symmetric) worm is a worm which uses no hard-coded, pre-selected internet resource in his propagation mechanism or payload (Simply decentralized could use in payload something like DNS name to DDoS. If the worm downloads own code from the preselected location or sends IP of newly-infected host to some e-mail address then it isn't decentralized at all).*

2.1 Current plugin-enabled worms

First and famous Morris worm carried two raw bodies: object files for VAX and Sun platforms (array for 20 filenames was allocated in the code but only two cells were actually used). Exploit (one of three) gains control on victim host, downloads both raw bodies, then compiles and runs appropriate one. Morris worm also used several scan methods to find new victims. So, since day one, active worms were multiexploit, multiscanner, multiplatform and not-so-monolithic.

According to Kaspersky Labs, several worms previously seen in the wild were able to use dynamically downloadable plugins. Those known to me are I-Worm.Music, I-Worm.Unis and I-Worm.Leave (last one also uses special internal script language). The real quality of implementation was achieved by I-Worm.Hybris [1] [1]. It's a fully functional plugin-enabled mail worm using strong encryption.

[1] Remark: I've got it once by e-mail. But it was under Solaris and I have no habit to launch unchecked e-mail attachments. Any of two is enough. By the way, one my

Tendency seems obvious: wide adoption of modular architecture by worm writers is a matter of near future.

This paper is an attempt to discover possible consequences of the long expected paradigm shift [6].

3 Modular worm: perspectives

3.1 Production cycle

- Modular worm could be developed and tested in parallel by large team of developers.
- MW could be maintained in ready for use state for long time while obsolete exploits are constantly replaced by fresh ones.
- MW can be effectively reused because it's core algorithms are independent of current technical details and could be easily adopted to correspond any technological changes.

3.2 Staying power

Once have reached significant population size MW could be supported by new plugins so worm does "dig in and hold on" in the cyberspace. Worm also could behave in a fashion similar to some diseases hiding in "refugia" [4] for bad times and using it as a launch pad for new attacks being armed with new exploits. In case of Internet worms "refugia" means array of poor-maintained computers with Internet connection.

Empiric rule of post-growth phase of CAIDA CodeRed population dynamics graph[7] is polynom-like decrease, approximately $\frac{100\%}{4\times months}$. Expected population for today is about several thousand copies, that matches TRIUMF[8] statistics.

CodeRed did no attempt to hide himself. On the contrary it intensively scanned Internet that may lead to notable performance degradation of the server.

SirCam and Magistr (including clones) are living in the wild for about an year.

So, hypothetic modular worm could remain in the Internet for months, waiting for new tools and tasks.

3.3 Local usage

The primary goal of previous worm writers was wide worm propagation. So, the attack was mostly undirected. Existence of large exploit library allows "dense local" opposite to "sparse wide" coverage. I.e. worm may be used against particular part of net being kind of automated hacker.

---

friend had to completely reinstall server infected by Nimda. Another friend spent a night curing mission-critical server infected by CodeRed v2 (this accidentally led to some service permanent crash); during recover administrator password was changed and forgotten so local representative had to travel from Boston to Detroit to crack the server locally.

3.4 Digital convergence

Digital convergence concept assumes digitalization of all telecom, media and home devices and formation of unite networked and web-enabled system for both home and mobile use. These devices will be made by methods of mass production. In absence of technical support they may become perfect target for automated exploitation.

3.5 Information warfare

Possibility of rapid collective development, local usage and staying power allow to use MW as most modern non-lethal weapon. It's possible usage mostly limited to "enemies" having different IT landscape (to prevent symmetric counterattack). It may be used anonymously against state, several states or particular company.

4 Formal model

4.1 Communication spaces

Worm travels from host to host via some communication space.

Nimda (monolithic but multiexploit worm) travels from host to host in three ways:

(E) by e-mail, like classic mail worm
(I) in CodeRed fashion, infecting random IPs where vulnerable program is found (unpatched IIS)
(V) infecting visitors of infected web site via MS IE

I'll consider these to be three different communication spaces E, I and V although all three has a common basement (mostly, the Internet). It's convenient to present them so because, for example, e-mail network E is small-world and scale free unlike (theoretically) fully connected IP address space I. Graph of web site visits V has its own features.

Additionally, these types of contacts are protected in different ways so better consider them to be different graphs (layers, spaces) which vertex sets are based on the same Big Set of IP devices $\mathfrak{B}$.

So, we present Internet as set of graphs. Communication space formed by protocol p_i is $S_i = \langle P_i, L_i \rangle$ where:

P_i is set of communication points $(h_j, p_k), h_j \in \mathfrak{B}, p_k \in Wi$

L_i is set of unidirectional links between communication points

W_i is set of software able to communicate by protocol $\mathbb{P}_i$

4.2 Active vs. passive

Definition 9. *"Active worm is a worm able to spread in completely autonomous manner" [3].*

I'll give own definitions considering some particular *spread method*.

Definition 10. *Active method is a method independent of human choice (opposite to passive)*

Definition 11. *Attack method is a method independent of human interaction (opposite to trap)*

Example 1. In this classification theoretical CodeRed-like hyperrapid Warhol-Worm from the outstanding work of N.Weaver [3] is an active attacking worm. Nimda is active, both attacking (CodeRed-like IIS scan'n'infection) and trapping (via mail read and web page load). Grandfather of all mail worms ILoveYou is passive and trapping.

Note: trapping methods are mostly used vs. clients (PCs), while attacking against servers. It's a consequence of typical network behavior: server accepts connections from any (or many) sources while client requests information randomly (not forced from outside) from preselected trusted locations be it WWW servers or ICQ contact persons.

4.3 Body, scanners, probes, exploits, payloads: our little zoo

Body is the only required part of MW.
Scanner is part of worm obtaining new host addresses (http/ftp/other URLs, IP addresses, e-mail addresses, ICQ numbers and so on). About great significance of scanners see [3].
Probe is part of worm which discovers availability and type of vulnerable software on the remote victim host.
Exploit is exploit or trojan or alike. It's a code which delivers body to victim host and runs it.
Payload is payload.
Combinations like "probing scanners" also may be mentioned to classify piece of code which performs several tasks (imagine monster e-mail parser detecting victim e-mail addresses, and software they are running; also intermediate mail servers and their software).

4.4 Program of MW

One thread of worm activity could be described as set of sequences alike $(s_i, p_j, p_k, e_l, e_m)$, where s_x is a scanner, p_x is a probe and e_x is an exploit. Usually we can present overall worm activity as pseudocode program similar to the following:

001 local IIS detection, IIS webpages trap patch (CAN-2001-0154 vulnerability)
001 MAPI e-mail address scan, mail exploit for CAN-2001-0154
001 local and mapped network drives scan, infection by CAN-2001-0154 traps
200 "better-near" random scan, IIS probe, CAN-2001-0333 and CAN-2000-0884 vulnerabilities exploitation, CodeRed II and sadmind/IIS backdoors exploitation

(It's Nimda; numbers mean allocated threads)
The question: buggy scanner mentioned in nth line of worm program can devaluate all subsequent lines. 26th line may introduce more effect than all previous altogether. How the worm could manage mass of own code? Two trivial methods are possible: moving successfull lines upwards or stochastic mutations (GA/EC).

4.5 Collaboration of MWs

Except option (exploit, scanner, etc) exchange worms may engage such forms of collaboration as: gender reproduction (genome exchange), owner command distribution, different worm copy specialization (example: explorers, data miners, data processors, data storages). Possibilities are endless.

5 Topological issues of multiexploit worm propagation

5.1 Types of spaces

Worm may propagate via different communication spaces: e-mail, IPv4 search, ICQ contacts, space of website visits, Windows shares and others. Here I'll provide some classification of spaces.

Open spaces are present when agent (worm) may immediately reach any communication point (CoreRed way). N.Weaver[3] shows that worm propagating in open space of IPv4 addresses may infect majority of susceptible hosts in minutes. (About open space graph topology see **??**.)

Mass service spaces are spaces of contacts between *servers* and their numerous *clients*. Each client may contact with multiple servers, so worm, infecting servers from clients and clients from servers (Nimda) may infect much of both. Topology of space graph is kind of bipartite.

Social contacts. E-mail, ICQ, AOL Messenger, chats: all these communications are reasoned by social contacts. Social networks is topic of much interest of researchers, I'll briefly state that they are supposed to be small-world, scale free graphs [9].

Neighbourhoods space is archipelago of separated local communities. Morris worm *.rhosts* exploitation and today's common Windows share exploitation and infection are both of this type.

Interservice communications are links between servers (from mass service model). In actual Internet it is hyperlinks between sites. Wide adoption of dynamic XML content and service-oriented architectures will introduce a whole new kind of interservice communications. It is straightforward to suppose that interservice links will be established in process of growth and preferential attachment. It is known [14–17] that processes of this kind lead to scale-free small world topologies perfect for infection spread [5, 18–21]. Especially strong effect may be introduced by interservice communications immune to firewalls.

5.2 Locality: optimization vs growth

According to [12] there are two basic cases for virus propagation: spatial and space-free topologies (simplest case of second is random graph). Second case may be also called "small world". CodeRed v2 was choosing victim randomly. So it was travelling back and forth over planet inducing unnecessary traffic. This leads to an advantage of small world topology but ignores spatial preferences.
Modular worm establishes own planet-wide network of code distribution, so it is extremely vulnerable to spatial issues. Induced traffic may devaluate all camouflage efforts.
"CodeRed II answer" on this question seems complete: it's "better near" scan when $\frac{1}{2}$ of time worm scans /16 network where it resides, $\frac{1}{4}$ of time it scans /8 neighbourhood and at remaining time it chooses random victim. This method may be optimized by better distance metrics but generally it's enough to get both advantages of small world and locality.

6 Merging worms

6.1 Worm model

Simple worm propagation model was constructed to explore dependency of worm propagation on exploit effectivity and topology of available links. Briefly, model consists of hosts embedded in 2D space. Links between hosts are distributed according to the rule of preferential attachment: probability of receiving new connection is proportional to number of existing ones. So, topology is scale-free [2]. Distribution of link lengths (2D, euclidean) is power-law. Some spatial correlation of host defence quality was introduced by self-similar distortion[3].

[2] As [18] states, epidemic threshold for scale-free networks is much smaller than for homogeneous ones.

[3] Algorithm: take grid with the same defence quality q at each node, change it in quadrants to keep grid average, then take each quadrant...

Our computational experiments are in agreement with 10 years old "rule of the thumb" by J.O. Kephart [12]: random graphs with average node degree of 5 and more show virus population dynamics similar to that observed for the homogeneous topology (i.e. on fully connected graph).
One more well-known dependency was discovered: worm's coverage depends on critical value $p_s \times d_{avg}$, where p_s is percentage of worm successes and d_{avg} is average node degree. Skipping clustering issues we may call it *reproduction rate*.

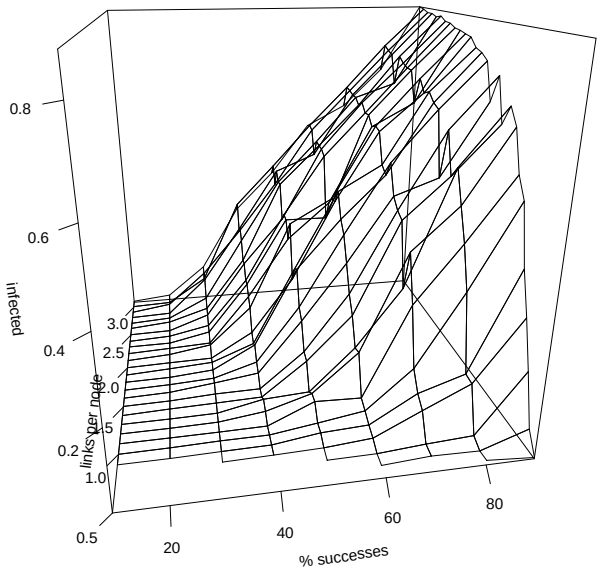

Fig. 1. Worm propagation model results. Axes are: average percentage of expoit successes (p_s), links per host ($\frac{1}{2}d_{avg}$) and percentage of infected hosts (i). Obvious: $i \to p_s$, as d_{avg} grows (i.e. worm touches almost every host when reproduction rate is above 1.0)

6.2 Effects of merge

Merge of several worms may produce different effects. Main of them are:

- increase of exploiting efficiency (p_s grows)
- adoption of additional communication links (d_{avg} grows)
- adoption of different communication spaces (besides d_{avg} it may deliver more convenient and more safe propagation channels)

Note 1. Exploit and spread method merge obviously has its saturation limit well described in terms of correlation. For example, integration of e-mail and ICQ spread methods most probably will not deliver significant advantage in d_{avg} because these types of links are highly correlated. Exploitation of two vulnerabilities usually fixed by the same "service pack" is also has low expected output. But even extra exploits may introduce large advantage of possible evolutionary maneuver for mutating worm.

6.3 Nimda as merged worm

Nimda (originally named as "Concept Virus (CV) v.5") seems ideal in this aspect: open space propagation to infect servers, two-way mass service infection, e-mail to propagate via social contact space and trivial but existing neighbourhood (Windows shares) infection.
But why did Nimda gain so little prevalence? Population peak measured by CAIDA was about 140.000 hosts. One possible reason is Nimda's rule not to infect IISes from host infected by readme.exe (i.e. via mail or webpage) [25]. So, mail-propagating Nimda wasn't visible via port 80 monitoring. At the same time, accsoftware distribution and maintainance is emerging. Like many powers before it was first used for destruction. But nothing prevents its adoption for our needs.

7 Future work

Pay additional attention to digital convergence impact on infrastructure.
Explore possible topologies of self-update networks (both worm and counterworm ones) [24].

8 Conclusions

In this paper the attempt was made to analyze near practical perspectives of self-propagating software concept. Today it is mostly used for malicious and destructive goals. Because of its effectiveness and power the most promising defense against it is to adopt it.

References

1. http://www.viruslist.com/eng/index.html?tnews=1001&id=785 (Hybris worm description)
2. http://www.cert.org/advisories/CA-2001-26.html (Nimda worm description)
3. Nicholas Weaver: Potential strategies for high speed active worms: a worst case analysis *
4. G. Abramson, V. M. Kenkre: Spatio-temporal patterns in hantavirus infection arXiv:physics/0202035

5. C.P. Warren, L.M. Sander, I.M. Sokolov: Firewalls, disorder and percolation in epidemics.
6. Jose Nazario and others: The future of Internet worms
7. http://www.caida.org/dynamic/analysis/security/code-red/ Dynamic Graphs of Code Red worm by CAIDA
8. http://andrew.triumf.ca/codered/ CodeRed scan rates at TRIUMF (Canada's National Laboratory for Particle and Nuclear Physics)
9. Holger Ebel, Lutz-Ingo Mielsch and Stefan Bornholdt: Scale-free topology of e-mail networks
10. Martin J. Fischer, Thomas B. Fowler: Fractals, heavy-tails, and the Internet
11. Trang Dang Dinh, Sandor Molnar, Atilla Vidacs: Investigation of fractal properties in data traffic
12. Jeffrey O. Kephart: How topology affects population dynamics *
13. M.E.J. Newman: Small worlds. The structure of social networks. http://www.santafe.edu/sfi/publications/
14. Albert-Laszlo Barabasi, Reka Albert: Emergence of scaling in random networks (Science, www.sciencemag.org, Vol 286, Oct 15 99)
15. Reka Albert, Hawoong Jeong, A-L Barabasi: Error and attack tolerance of complex networks (Nature, www.nature.com, Vol 406, Jul 27 2000)
16. M.E.J. Newman: Clustering and preferential attachment in random networks arXiv:cond-mat/0104209
17. C.F. Moukarzel and M. Argollo de Menezes: Shortest paths on systems with power-law distributed long-range connections arXiv:cond-mat/0201083
18. R. Pastor-Satorras and A. Vespignani: Epidemic dynamics in finite size scale-free networks (arXiv:cond-mat/0202298)
19. Cristopher Moore and M.E.J. Newman: Exact solution of site and bond percolation on small-world networks http://www.santafe.edu/sfi/publications/
20. M.E.J. Newman, I. Jensen, R.M. Ziff: Percolation and epidemics in a two-dimensional small world http://www.santafe.edu/sfi/publications/
21. M.E.J. Newman: Exact solutions of epidemic models on networks arXiv:cond-mat/0201433
22. Java 2 Enterprise Edition homepage: http://java.sun.com/j2ee
23. FAQ on apt and RPM: http://bazar.conectiva.com.br/ godoy/apt-howto/
24. N.J. Gunther: Hypernets - good (g)news for Gnutella (arXiv:cs.PF/0202019)
25. SANS Institute: "Nimda Worm/Virus report" http://www.incidents.org
26. M.Girvan, D.S. Callaway, M.E.J. Newman, S.H. Strogatz: A simple model of epidemics with pathogen mutation http://www.santafe.edu/sfi/publications/