Pulse: A Class of Super-Worms against Network Infrastructure

A.G. Voyiatzis and D.N. Serpanos
Department of Electrical and Computer Engineering
University of Patras
GR-26500 Patras
Greece
E-mail: {*bogart,serpanos*}*@ee.upatras.gr*

Abstract

Super-worms constitute the most advanced and dangerous threat for networks and the whole Internet. Their goal is to infect the significant majority of Internet hosts in the minimum possible time, by using advanced techniques to partition the Internet address space and to coordinate the infection process.

In this paper, we present Pulse, a new class of super-worms, which target network systems and specifically routers, in contrast to conventional worms and super-worms which target network hosts. Pulse super-worms can be very effective and efficient, because they exploit one significant Internet vulnerability: the assumption of Internet's development model that all routers are trustworthy and can coordinate to defend against attacks from external enemies, who have been considered the only enemies traditionally. Pulse super-worms infect routers, thus creating internal enemies undefeatable using the existing security model. As we demonstrate, Pulse super-worms are more efficient than alternatives in infecting network systems and utilize available information for self-organizing their infection policy. Furthermore, we demonstrate through specific attack scenarios, that Pulse super-worms can be extremely effective for a wide range of attacks, especially in information warfare. Finally, we describe countermeasures which are necessary for a successful defense against Pulse super-worms.

1. Introduction

The Internet is a network interconnecting autonomous end-systems. Its structure is hierarchical, because it has emerged as a network interconnecting autonomous networks. Its infrastructure consists of an interconnected set of routers (switching devices), which route the data that flow among end-systems. In order to make routing decisions, routers maintain routing information locally; this information is maintained in the well-known *routing table*. Considering that routes are dynamic and that interconnected networks or end-systems may change their connectivity, the routing tables often need to be updated, in order to maintain efficient routes between interconnected networks and end-systems and to adapt dynamically to network topology changes. These changes are made by specific routing protocols that are included in the suite of Internet protocols, such as RIP, OSPF, IEGP, BGP and EGP.

The Internet has been developed as a network with distributed management, in order to operate in hostile environments, where portions of the network may "disappear". However, it is clear that its development model has been assuming that all enemies are external, i.e., no participating end-system or router is malicious. Thus, the protocols typically implement mechanisms that tolerate transient faults and misbehaviors due to human errors; they were never designed nor implemented to be robust in hostile environments, where there are internal enemies, such as routers that misbehave intentionally. One simple demonstration of this fact is the ARP protocol, which is used for IP to Ethernet address translation. Whenever a host A needs to know the Ethernet address of a host B, it issues an ARP request containing B's IP address. ARP assumes that only the authorative host will reply with the Ethernet address of the specific IP address. So, the first reply received by A will be treated as an authorative reply and no further replies will be processed, even if received. This characteristic of ARP is a security weakness, which is commonly exploited to eavesdrop network packets in switched Ethernet segments. For example, in the previous scenario, a malicious host C may try to reply immediately to the ARP requests of host A giving hist own Ethernet address rather than B's. If A receives the reply by C first, it will direct its traffic to C, ignoring all other ARP replies that will be sent to A from other hosts in the network.

Network engineers have been constantly developing

0-7695-1921-0/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

countermeasures for such threats and attacks. Despite the good defense characteristics of these countermeasures, these efforts are far from complete, because the security problems lie in the assumptions and design of the protocol and not in its implementation. The security problems have increased actually, due to the development of even more sophisticated attacks using viruses and worms, which exploit many of the security weaknesses of most existing Internet protocols. In light of these fundamental Internet weaknesses, the security problem of the Internet has become significantly more acute, because of its use as a network infrastructure for commercial transactions and e-business. The Internet has become the critical infrastructure for several organizations. In such a critical environment, there are many threats from people with malicious intentions, ranging from sabotage and industrial espionage to information warfare and cyber-terrorism.

In current attacks on the Internet, the targets of malicious attackers have been the end-systems, servers or clients, that are attached to the network. However, the active Internet infrastructure, i.e. the routers and the servers for network services, like the DNS and the Time Protocol (NTP), can be an attractive target to launch information warfare. These components are crucial for operation of the Internet: once they are put out of order, the Internet may collapse and disconnect all interconnected networks. Such network separation can have disastrous effects to the operations of many organizations and even nations.

In this paper, we present a new threat, in the form of super-worms that target the Internet infrastructure rather than end-systems. Such a super-worm can infect routers and propagate through them to other network infrastructure and thus, act as an internal enemy of the network. Although such a worm has not appeared and there is no known implementation until today, the possibility of its appearance requires the adoption of countermeasures, because, as we show, such an attack can be very effective: even a few compromised routers can result in the total disruption of communication over the Internet.

The paper is organized as follows. Section 2 provides the necessary definitions and describes the anatomy and structure of worms and super-worms. Section 3 introduces the class of Pulse super-worms and describes their infection and propagation method, their effectiveness, through description of possible attack scenarios and analysis of their traceability and infection rate. Section 4 describes possible countermeasures to prevent Pulse super-worm attacks or recover from them successfully.

2. Worms and Super-worms

Viruses and worms have emerged lately as the dominant security problem of networked hosts. Worms especially, have become a significant threat, more dangerous than viruses. In the context of this paper, we consider a worm to be a special type of malicious software similar to a virus, but with the difference that it does not attach to programs, but simply uses system resources, multiplies and spreads through the network [12]. Worms are not new; the well-known example of the *Morris worm* (or *Internet worm*), which appeared in November 1988 [9] constitutes the first known appearance of a worm more than a decade ago. The explosive growth of the Internet and the significant security vulnerabilities in the dominant operating systems have increased their effectiveness, because authors of worms use the network resources to spread automatically.

The code of a worm program can be considered as a composition of three main functional modules (segments):

1. **Infection code:** it installs the worm on a target machine and hides it from the user(s);

2. **Propagation code:** it spreads the worm to other machines, typically using network resources such as e-mail systems, disks shared through the network, peer-to-peer applications, etc.;

3. **Action code:** it designates the action of the worm on the infected system; this can be the most disastrous part of a worm, because it may perform a malicious action on the infected machine, such as exposing user private data, sending unwanted e-mail messages, creating unintended network traffic, etc.

A characteristic example of a worm is the *sadmind* worm [2], which targets machines running the Solaris operating system. The *infection code* of the worm takes advantage of a remote buffer overflow vulnerability to gain administrative access on the target machine. It then hides its *propagation code* in the `/dev` directory. This directory is commonly used to reference the various hardware devices and thus, no ordinary users check its contents under typical conditions. The hidden propagation code targets new machines. The *action code* actually performs three distinct actions: first, a service is installed on the system, which allows remote access to the `root` account (the most privileged account) of the system. Second, it creates or modifies the `.rhosts` file of `root` account, in order to allow access to the system from remote users through the typical remote services (`rsh, rlogin, rcp`). Third, the action code starts to attack Microsoft Windows machines with the installed IIS server (web server software), in order to alter their web content. When the worm compromises 2,000 IIS servers, the web content on the infected system is also compromised.

Recently, a new class of worms, called *super-worms*, has emerged as a theoretical threat model [10] [13]. These worms have advanced infection policies, targeting to infect the largest possible fraction of Internet hosts (end-systems) in the minimum possible time. To achieve this they use multiple techniques to propagate, exploiting as many vulnerabilities as possible. For example, the well-known Nimda worm [3] used five different methods to propagate, ranging from web server vulnerabilities to back-doors of the older Code Red II [4] and *sadmind* worm. Based on their infection rate, the known super-worms have been named *Warhol*, *Flash* and *Curious Yellow* [10] [13]. Warhol is expected to infect the whole Internet in time of the order of 15 minutes, Flash in the order of 30 seconds and Curious Yellow in approximately 15 seconds; the last two worms require a time-consuming pre-scan phase, before starting to spread.

In order to achieve a successful and fast Internet-scale attack, super-worms need to solve a fundamental problem: the partitioning of the address space. This problem is fundamental to achieve a high infection rate, because, from the attacker's point of view, it is desirable to minimize duplication of work: different worm processes should not try to infect the same sequence of hosts or even worse some already infected hosts. For this purpose, authors of worms partition the Internet's address space with various methods, such as decentralized partitioning [10], or co-ordinated partitioning [13].

Conventional worms and super-worms focus on the infection of end-systems. They exploit network vulnerabilities, but use the network resources only to route their processes appropriately. In the following, we describe a new class of super-worms which target to infect routers and network resources, in general, in contrast to end-systems.

3. The Pulse Super-worms

Pulse[1] constitutes a new class of super-worms, which target network routers and, in general, network resources rather than end hosts and user data as conventional super-worms do. Furthermore, Pulse super-worms exploit the available and highly optimized information contained in routing tables, in order to effectively partition Internet's address space.

In the following, we describe the infection method, propagation method and possible (malicious) actions of Pulse super-worms and analyze their traceability characteristics as well as their performance in terms of infection rate.

3.1. Infection

Conventional routers and other network systems that constitute the network infrastructure are rather complex, because, in addition to traditional routing, they typically perform many value-added functions, such as content filtering, management of bandwidth utilization, maintenance of per TCP port statistics and firewalling. In order to perform these value-added functions, routers are able to execute software and often provide remote services for monitoring their operations, e.g. a web server or an SSH server. It is a reasonable expectation that this complex software has vulnerabilities that allow unauthorized access to a router's internal information and services. For example, a list of more than 150 known vulnerabilities for products of a well-known vendor appears in MITRE's CVE [7]; many of these vulnerabilities allow access to internal information and services.

Pulse can utilize such vulnerabilities to infect a router. Once infected, the router can be instructed to execute the code of the worm along with its regular tasks, thus allowing the worm to run and propagate. Although a vulnerability may not provide full administrative privileges on the router, some extra code in the worm can easily enable it to gain full administrative privileges, if necessary.

The size of the infection code heavily depends on the exploited vulnerability. In a minimum, it can be in the range of a few bytes, as in the case were a carefully crafted HTTP request can result in executing code with full administrative privileges.

3.2. Propagation and address space partitioning

Pulse super-worms can utilize the routing tables of the infected router, in order to efficiently propagate along the Internet. Routing tables provide rich and highly optimized information. Actually, the routing tables contain the optimal propagation strategy to reach the whole Internet infrastructure, assuming that a worm's action code does not destroy or alter their contents. A Pulse super-worm simply needs to follow the routes indicated by the routing tables, from router to router, to reach in the minimum possible time the most distant attached network or end-system. Furthermore, the routing tables contain also the downstream neighbouring attached networks; thus, the worm can use this information to propagate to the internal network structure of an attached network. Taking advantage of the routing information of routers, the Pulse super-worm, clearly, does not need to solve the difficult problem of conventional worms to partition the address space of the Internet.

The size of the propagation code is dependent on the propagation strategy and the designated action. For example, an instance of the worm may be spawned for each available route. In any case, an enumeration loop is needed, in order to process each route; the code for this loop may not exceed a few dozens of bytes, taking into account the necessary code for accessing each respective network interface.

[1] The name Pulse originates from the EMP that is produced after a nuclear bomb explosion and practically destroys every electromagnetic device in range; the Internet was designed to operate even after a nuclear explosion.

3.3. Action

Any worm or super-worm can include arbitrary action code. However, their actions are limited by the data they can access. As Pulse super-worms take over routers and network infrastructure, they have access to rich and significant information that can be exploited for damage not possible with other super-worms. In the remaining of this subsection, we describe some possible scenarios that a Pulse super-worm attack can implement. The list is not exhaustive; rather, it serves as a guideline indicating the significant strength of Pulse super-worms, which can be used for disastrous attacks.

One line of actions relates to the well-known Distributed Denial-of-Service (DDoS) attacks. While defense against DDoS attacks is an open research issue, novel and promising techniques appear in the literature, which utilize routers to trace back to the originator of an attack [1] [8]. These techniques assume a cooperative environment, where legitimate routers exchange vital information to traceback to the origin of the attack. A malicious user can use the Pulse super-worm to attack routers and cancel all these reporting mechanisms, thus destroying the protection mechanisms.

Having access to a router and its high speed network interfaces can result to more efficient DDoS attacks than the conventional DDoS attacks, which require the distributed coordination of the attack's processes. For example, a Pulse super-worm can alter a router's routing tables and direct all outgoing traffic to a specific host. Even worse, in an Internet-scale attack, all routers may direct their traffic to the root DNS servers, which are currently only thirteen (13). Recently, a significantly less sophisticated attack, than the one a Pulse super-worm can launch, brought down nine (9) of these root DNS servers [11]. If all Internet routers directed their traffic to these thirteen hosts, the DNS system would definitely collapse, due to overload, and the whole World-Wide Web (WWW) would stop functioning.

One could argue that methods for network data integrity and authenticity can provide a protection against Pulse super-worms. Several methods have been proposed to ensure network data integrity and authenticity, such as the DNSSEC extensions [5]. However, such methods are useless, when a router is under the control of a Pulse super-worm, because these methods are developed simply to ensure that data are not manipulated in transit. A Pulse super-worm can have access to the cryptographic keys stored in a router; thus, it can sign any data –making it authentic and valid– or replace cryptographic keys for other routers and thus, trick the router to accept false information as authentic.

Another line of actions can be implemented to steal "sensitive" data easily. Such actions would be of interest to attack Virtual Private Networks (VPN), Voice-over-IP (VoIP) services, etc. A Pulse super-worm can steal cryptographic keys for setting up VPN tunnels or direct traffic in a desired location for storage and later processing. In regard to VoIP services, a Pulse super-worm may direct such traffic to a system for recording and then forward the stream to the appropriate recipient. This action implements a successful man-in-the-middle attack; attacks on VoIP systems can actually be easier and more efficient than attacks against VPN's, because VoIP data streams are usually not encrypted for performance reasons.

The size of the action code is heavily dependent on the attack scenario. A router configuration alternation can be performed using command lines of a few bytes. A sophisticated, targeted attack will not need more than to execute a simple command and apply the new configuration. Such actions can be coded in a few dozens of bytes as well.

3.4. Traceability

Even when a worm successfully spreads over the Internet and appropriate countermeasures are developed, it is often necessary to trace back its origins, in order to identify the source of the problem and collect evidence for legal actions. Unfortunately, Pulse super-worms can be untraceable by network engineering means. This is possible, because routers do not hold extensive logging for user actions; instead, they concentrate on their routing-related functions. Furthermore, if the action code directs alteration of routing information, this cannot be traced, because routers do not maintain strict time and historical information for route updates. Finally, once the worm has spread enough, it is very difficult to decide which router is attacking which; all routers perform identical actions.

These characteristics of Pulse super-worms can be easily demonstrated with a simple example. Assume that a malicious user attached to network A exploits a vulnerability in a very remote network X to launch an attack from X to the whole Internet. The worm will start spreading from network X to the whole Internet. If no precise time is maintained at each router, e.g. using the NTP protocol, it may be impossible to even identify that the origin of the attack is network X, since each network operates in an autonomous fashion. Furthermore, even if the source of attack (network X) is identified, a capable attacker can cover his/her access to the router of network X, thus remaining untraceable.

3.5. Infection rate and spread policy

The Pulse super-worm can achieve infection rates similar to the Flash super-worm [10]. It can infect all Internet routers in some seconds to minutes at most, assuming that the payload of the worm will be very small, e.g. some few dozens of kilobytes. Given that routers are connected to high speed interfaces, in the order of hundreds of Mbps to Gbps, the propagation delay to transmit the worm will be less than 20 $msec$. Summing up the delay for privilege escalation and initial worm configuration, we calculate the delay for a single router infection to a few hundreds of milliseconds, at most.

In contrast to the Flash super-worm, a Pulse super-worm does not need to perform any pre-infection computations to obtain a hit-list; all the necessary information is obtained from the routing tables. Furthermore, this information is always updated and optimized as long as the routers maintain their operational status.

For an Internet-wide spread, one needs to estimate the number of available routers; in most attack scenarios it is not necessary –and is probably undesirable– to infect every single router. In any case, the required number of infections will be in the order of a few thousands routers. In order to launch a successful attack, the Pulse super-worm needs only the information of the routing tables, which contain all the neighbouring routers: these neighbors are the next hops for infection. The infection process can advance in a store-and-forward fashion: first, the Pulse super-worm infects a router (store) and then it starts infecting its neighbours (forward). For this, it is necessary to select a direction on each router, either upstream or downstream, in order to avoid multiple and unnecessary infections of already infected routers.

Overall, a Pulse worm can be as small as a few hundreds of bytes or a couple of kilobytes, which is a rather small footprint to transmit over the high-speed links. For a 2 Mbps link (a rather conservative example), it would take less than a second to infect and install the worm on a neighboring router.

4. Defenses

Defense against security threats such as the described Pulse super-worms requires not only technical support but increased awareness and avoidance of potential vulnerabilities.

As soon as the Internet was designated as critical infrastructure, many efforts have been made to increase security awareness of users. Network administrators are characterized by increased security awareness, but even in their methods and practices there is still a window of opportunity for dedicated, malicious attackers as experience has shown. We have witnessed many cases where network administrators use insecure protocols like telnet and tftp, or they use call-back modems installed on routers for easier maintenance. Such practices allow a dedicated and well-funded attacker to exploit vulnerabilities in a router's software and launch attacks, such as one with a Pulse super-worm. Once the worm succeeds to infect one router, the worm can start to spread all over the Internet automatically. Thus, it is necessary to improve the security culture of network administrators, so that they do not use insecure methods and practices to manage their networks. The description of Pulse indicates that insecure management of one network attached to the Internet compromises the security of the Internet overall and not only the security of the network.

In order to support effective defenses, router vendors must develop bypass channels, through alternative means, to distribute patches and software fixes for their systems. Once a Pulse super-worm is out in the wild, its action code may disrupt communications over conventional channels with vendor's distribution points over the Internet. These bypass channels should be as fast as possible, in order to enable router disinfection in a minimized amount of time.

A *Cyber-Center for Disease Control* for network infrastructure, as proposed by Staniford, Paxson and Weaver [10], would clearly be a useful development. Monitoring tools of the Internet's stability are necessary to identify and respond to abnormal operations, such as incomprehensible routing table updates and traffic flow fluctuations. While an Intrusion Detection System (IDS) could face this problem well, human experience is priceless. Actually, network instabilities are not a well-understood process [6] and only an experienced network administrator equipped with the necessary network monitoring tools can identify transient instabilities from a worm outbreak.

5. Discussion and conclusions

Pulse super-worms constitute a new model of worm attack to the Internet. They differ from conventional worms and super-worms in that they target network infrastructure rather than hosts (end-systems). As we have demonstrated, Pulse super-worms are very efficient, in terms of infection and spreading speed. Furthermore, they can be very effective, because they get access to rich and important information contained in routing tables. A network router infected with a Pulse super-worm can be very dangerous for the whole Internet, as the described scenarios have demonstrated.

We presented defense measures to eliminate the probability of a Pulse super-worm outbreak. However, our proposals are clearly incomplete, because the scenarios for exploitation of router vulnerabilities are limitless. Actually, we believe that critical business and government operations should not be attached to the Internet, directly or indirectly, as long as information warfare is an existent threat. The Internet was not designed to work as an environment with hostile insiders, but rather to work as an environment of mutual, trustworthy collaboration and cooperation among insiders, resistant to external enemies. The existence of the class of Pulse super-worms demonstrates that this development model was incomplete and thus, conventional Internet is open to a wide range of disastrous attacks by malicious users, who can successfully infect even one Internet router.

References

[1] S. Bellovin, M. Leech, and T. Taylor. ICMP Traceback Messages, October 2001. Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-itrace-02.txt (current February 2003).

[2] CERT/CC. CERT Advisory CA-2001-11 sadmind/IIS Worm, May 2001. http://www.cert.org/advisories/CA-2001-11.html (current February 2003).

[3] CERT/CC. CERT Advisory CA-2001-26 Nimda Worm, September 2001. http://www.cert.org/advisories/CA-2001-26.html (current February 2003).

[4] CERT/CC. CERT Incident Note IN-2001-09, August 2001. http://www.cert.org/incident_notes/IN-2001-09.html (current February 2003).

[5] D. Eastlake. RFC 2535: Domain Name System Security Extensions, March 1999. http://www.ietf.org/rfc/rfc2535.txt (current February 2003).

[6] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet Routers Instabilities. In *Proceedings of the IEEE INFOCOM 9́9*, New York, NY, March 1999.

[7] MITRE. Cve: Common vulnerabilities and exposures. http://www.cve.mitre.org/ (current February 2003).

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical Network Support for IP Traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, pages 295–306, Stocholm, Sweden, August 2000.

[9] E. H. Spafford. The Internet Worm Incident. In C. Ghezzi and J. A. McDermid, editors, *ESEC'89 2nd European Software Engineering Conference*, University of Warwick, Coventry, United Kingdom, 1989. Springer.

[10] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. In *Proceedings of the 7th USENIX Security Symposium*, Aug. 2000.

[11] P. Vixie, G. Sneeringer, and M. Schleifer. Events of 21-Oct-2002, November 24, 2002. http://f.root-servers.org/october21.txt (current February 2003).

[12] Webopedia.com. Virus definition. http://www.webopedia.com/TERM/v/virus.html (current February 2003).

[13] B. Wiley. Curious Yellow: The First Coordinated Worm Design. http://blanu.net/curious_yellow.html (current February 2003).