## A virus and a worm: lessons learned from SirCam and Code Red in a university environment.

Marc Mazuhelli
August 2001

### Introduction

Viruses and worms are two types of malware that we heard a lot from in the summer of 2001. Two specimens, one from each of these forms of malware, were released a few days apart in July 2001, keeping security personnel busy and generating a lot of coverage in the press.

In this text we will cover impacts felt and lessons learned from these two incidents in the university environment where the author recently started working as a computer security analyst.

### What makes university environments special?

Unlike companies and e-commerce sites, universities have been present on the Internet since the very beginning, when other users could be trusted. This is unfortunately not the case anymore, and networks that are not sufficiently protected rapidly become targets for hackers.

In particular, networks of many universities are among those that don't offer sufficient security. The reasons for this are cultural, monetary and organizational [1].

Universities are well known for the free exchange of ideas and community sharing. These notions are incompatible with the behavior that we must now adopt regarding safe computer practices, which is: trust no one!

Also, universities often operate very elaborate networks with thousands of workstations and servers hooked up by high-bandwidth links to the Internet. But in many cases, they don't have sufficient physical or human resources to adequately protect this equipment with all the necessary firewalls, anti-virus software or intrusion detection systems.

Finally, many universities are based on decentralized operations. We often find a centralized group maintaining common services like the network and central servers, and also sometimes individual workstations used for administrative purposes. But computational resources used for research and academic purposes are often managed by different employees whose superiors are not part of the central IS department. This results in a lack of cohesiveness that greatly complicates the work of the security team.

### Definitions

What is a virus? What is a worm? What makes them different?

A virus is a piece of malicious code that cannot live on its own; it has to attach itself to a program, a file or a disk. It will propagate itself to other programs, files or disks, but only after a manual operation from the unsuspecting victim. This operation can take multiple forms, but it's often the opening of a file attached to an e-mail message.

A worm will also propagate itself, but it can do it with no human intervention as it exploits a vulnerability of the attacked system, for example a buffer overflow vulnerability. Also, the worm doesn't need to attach to something else; it is self-contained.

### The SirCam virus

The SirCam virus was discovered on July 17, 2001. Like many recent viruses, it spreads itself with a file attached to an e-mail message. The text of the message tries to convince the reader to open the attached file with a text like: "Hi! How are you? I send you this file in order to have your advice. See you later. Thanks". If the file is opened, a document is selected at random from the "My Documents" folder of the hard disk, and the virus is prepended to this file. This newly infected file will be sent out to spread the virus, so there is a possibility that confidential information will be disclosed.

The virus spreads when the newly infected file is sent to all addresses in the Windows address book and all e-mail addresses found in temporary internet cached pages; as a consequence all the mailto: links of recently visited Web pages will receive the virus. It also spreads to other computers on the local LAN through unprotected network shares. Also, under certain conditions, the virus can completely fill or even erase the C: drive.

The name of the attached file has double-extensions (for example budget.xls.exe or proposition.doc.pif). The second extension is always .BAT, .COM, .EXE, .LNK or .PIF.

Also, the subject of the message is the same as the name of the randomly selected file (without the two extensions) so simple filtering on the subject of the message is impossible.

### The impact of SirCam in our university environment

As soon as I received a call concerning SirCam, I researched the usual sources [8,12] to find out if it was a real menace. It turned out to be the case, so I prepared an e-mail message that was sent to all employees of the university. It was already too late for some as they had already opened the attached file. These people were told to update their anti-virus definitions and to scan their disk.

Unfortunately, the site license that we had acquired two years earlier had expired in June 2001 and it had been replaced only in July (when many employees were on vacation), and by a different product. The result was that many user workstations still had the old anti-virus software installed, which could not be updated with new definitions anymore. In these cases, we had to either take the time to install the new anti-virus software which we had just acquired, or use Symantec's FixSirc tool [13], which is a stand-alone tool. All this took a lot of manual operations and a lot of time.

I soon realized that all employees were not in the distribution list that I used; for example, temporary employees hadn't received my message. I also knew that students were absent from the mailing list I used to send my message, but unfortunately I had no easy way to reach all students at once. So, many people who received copies of the virus did not get the information in time.

Our institutional mail servers currently have limited filtering capabilities. They can filter messages based on the Subject: header, but not on the name or the type of an attached file. Whereas the ILOVEYOU, AnnaKournikova, Homepage and other viruses all sent e-mails with identical subjects that could easily be filtered, we had no way of doing this with SirCam as the e-mails that are sent always have a different Subject: line. The result is that messages kept coming in at an alarming rate, even weeks after the initial release of the virus. Employees reported receiving many tens of copies on the worst days, and at more than 200K a piece, many inboxes went over quota resulting in major inconveniences for people who hadn't even executed the virus themselves!

**Lessons learned from SirCam**

Our users have to be educated not to open files attached to e-mail messages before checking their legitimacy, even though they come from people they know. Many people executed the SirCam virus specifically because it came from someone they knew well and even though the text of the message was in English and we mainly write to each other in French (which is the main language in the province of Québec where we are located).

We have to rapidly install the new anti-virus software for which we recently acquired a site license on all our users' workstations. Some still have the old anti-virus software (for which virus definitions can't be updated anymore); others have no anti-virus software at all. And we have to make sure that virus definitions are rigorously kept up to date.

Also, we absolutely have to find a way to filter messages other than by the Subject line on our centralized e-mail servers. Filtering by the type of attached files would be nice. A commercial anti-virus also capable of recognizing known viruses would be even better.

Finally, we have to find ways to easily and rapidly reach all persons concerned by security incidents; the distribution list used in this case was incomplete.

**The Code Red worm**

The Code Red worm was discovered on July 11, 2001. It uses a buffer overflow vulnerability in Index Server 2.0 and Indexing Service which are full-text search and indexing engines used with Windows NT 4.0 (Internet Information Server version 4) and Windows 2000 (IIS 5), respectively. This vulnerability was discovered on June 18th by eEye Digital Security [5]. The CERT published an advisory on the following day [2]. Microsoft immediately released a patch for the vulnerability [9].

Unlike most incidents of this nature where many months pass between the publication of the vulnerability and malicious code that exploits it, in this case it took only a few weeks! Using this vulnerability, arbitrary code can be executed in the Local System security context. This essentially can give the attacker complete control of the victim system.

The worm replicates itself with a simple HTTP request beginning with /default.ida to a vulnerable server. This request will call the idq.dll dynamic library since a default mapping exists to map .ida and .idq file types to this library even though index services are not used.

The first version of the worm defaced Web pages on the attacked server with pages that said: "Welcome to http://www.worm.com ! Hacked By Chinese!". Then the worm probed other semi-random addresses for a Web server responding to port 80 and the same request was sent to responding servers.

Until the 19th of the month, the worm was in "propagation" mode, after which it switched to Distributed Denial Of Service (DDOS) mode on the IP address that used to be www1.whitehouse.gov. Changing the IP address of this server before July 19th successfully dodged this DDOS attack.

SecurityFocus published a very detailed report describing the Code Red worm [11]. CNET also closely followed Code Red's progression [3].

On July 30th, a rare event took place. Officials from the American government, Microsoft and other computer security experts organized a press conference to urge IIS users to install the Microsoft patch on their servers before round 2 of Code Red which was expected on August 1st.

Many system administrators who still had not installed the patch did so then, but not all of them since Digital Island reported that more than 150000 machines were infected by round 2 [4].

Code Red and its first minor variant were fortunately not too bad; no files were modified on disk as the worm was completely memory-resident. Installation of Microsoft's patch and a system reboot was all that were necessary to get rid of the worm.

But this changed on August 4th, when a new version named Code Red II (because this string of characters appears in the worm's code) was released. It uses the same vulnerability as Code Red to spread, but this is the only thing they have in common. Code Red II is a lot more dangerous as it leaves a back door permitting remote access to the infected server. Even though recipes exist to undo the actions of Code Red II, there's no way to know if other back doors or other dangerous activities (like stealing of the password file) occurred between the infection and its removal.

Code Red II also spreads much more aggressively. It concentrates on addresses that are "closer" to the infected machine by using IP addresses where the first number or the first two numbers are the same 7 out of 8 times. This created a lot of problems for broadband Internet service providers. Some providers temporarily blocked access to port 80 on their networks in the hope that it would lessen the slowdowns they were experiencing. In doing so, access to legitimate, patched (or non-IIS) servers was also blocked.

**The impact of Code Red in our university environment**

On Thursday, July 19th, we experienced a complete failure of our internal network for about 5 hours during the afternoon. Our network team found out that the ARP table (which stores mappings between IP and MAC addresses) of our main router was overflowing because it couldn't handle all the traffic it was receiving. Before the exact cause could be identified, the problem disappeared.

The next day, when I read about Code Red, we concluded that it was probably responsible for our network failure. Unfortunately, we didn't have the necessary tools (like an intrusion detection system) to confirm this. Then on July 21st, I received an e-mail from SecurityFocus informing me that 10 of our machines were probably infected with Code Red. There were servers in this list that I didn't even know existed, so I certainly didn't know who administered them!

After some research and many phone calls, I finally traced all the persons responsible for these servers, and I made sure they were patched (and even closed in some cases). I then wondered how I could quickly find out if there were more vulnerable (unpatched) servers on campus. No inventory of all running servers was available, and since many employees were on vacation, this wasn't easy.

I used a Web site called Netcraft [10] which keeps track of Web sites and can list which Web server software is running and under which host operating system. Around 40 Web servers from our campus were listed, of which 13 were supposedly running IIS 4 or 5. But the list was not complete; some servers I knew about weren't mentioned.

So I fired up nmap [7] on my Linux box and scanned our whole class B address range. More than 300 Web servers were found! Many of those were used to administer routers, switches, printers and other peripherals. But almost 100 servers were identified as potentially running Windows by nmap's "remote OS detection" option. Unfortunately, nmap doesn't always have enough information to identify the remote OS reliably.

By then a few days had passed and I learned that eEye had released a tool to identify vulnerable servers [6]. We used this tool to clearly identify around a dozen other IIS servers that were vulnerable, and they were all patched or closed before August 1st.

### Lessons learned from Code Red

We have to reduce the number of Web servers running on campus by consolidating many servers on a small number of well managed servers. We have to eliminate Web servers that are run by end-users on their individual workstations because these users rarely have sufficient network security and system administration notions.

Once legitimate servers have been identified, an inventory of these servers with at least 2 persons responsible for their management has to be constructed and kept up to date. Instead of dozens of people on campus constantly watching for new vulnerabilities and patches, the security team can do this work as long as it knows where to distribute the information.

We have to implement audit procedures so that the security team can verify that critical security patches have been installed.

We should implement filtering at our network border so that requests coming from the outside world can only reach legitimate servers. This implies that a central authority has the power to decide what is legitimate and what is not. For this to work, clear rules have to be written and approved by higher management. In a university environment where everybody is pretty much used to do whatever they want, forcing everyone to ask permission to a central authority before implementing a new service will not be easy as it will require a change of attitude.

We should periodically scan our internal network to find newly opened ports on legitimate servers (which could signal the installation of a back door) and to find new servers that have not been approved.

### Conclusion

We have listed the impacts of the spreading of the SirCam virus and the Code Red worm in a university environment as well as the lessons that we learned from these two incidents. In both cases, technical tools can be implemented to alleviate the negative impacts of these kinds of incidents (installation of anti-virus software on individual workstations and on mail servers, installation of relevant patches, etc.). But all these steps will not help much without proper user awareness training. Users must learn that e-mail attachments are dangerous and that running software has to be kept up to date by installing the manufacturer's patches.

### References

[1] Associated Press. "Colleges serve as hacker training grounds." June 1, 2001. URL: http://news.cnet.com/news/0-1003-202-6157694.html (August 15, 2001).

[2] CERT. "CERT Advisory CA-2001-13 Buffer Overflow In IIS Indexing Service DLL." July 30, 2001. URL: http://www.cert.org/advisories/CA-2001-13.html (August 15, 2001)

[3] CNET. "Code Red: The worm returns." August 10, 2001. URL: http://news.cnet.com/news/0-1003-201-6741564-0.html (August 15, 2001)

[4] Digital Island. "Code Red Status." August 8, 2001. URL: http://www.digitalisland.com/codered (August 15, 2001)

[5] eEye. "Advisory AD20010618." June 18, 2001. URL: http://www.eeye.com/html/Research/Advisories/AD20010618.html (August 15, 2001)

[6] eEye. "Code Red Scanner from eEye Digital Security." URL: http://www.eeye.com/html/Research/Tools/codered.html (August 15, 2001)

[7] Insecure.org. "Nmap." June 21, 2001. URL: http://www.insecure.org/nmap (August 15, 2001)

[8] McAfee. "W32/SirCam@MM Help Center." URL: http://www.mcafee.com/anti-virus/viruses/sircam/default.asp?cid=2360 (August 15, 2001)

[9] Microsoft. "Microsoft Security Bulletin MS01-033." June 18, 2001. URL: http://www.microsoft.com/technet/security/bulletin/MS01-033.asp (August 15, 2001)

[10] Netcraft. URL: http://www.netcraft.com (August 15, 2001)

[11] SecurityFocus. "SecurityFocus Code Red Information Headquarters." URL: http://aris.securityfocus.com/alerts/codered (August 15, 2001)

[12] Symantec. "SARC Write-up - W32.Sircam.Worm@mm." August 14, 2001. URL: http://www.sarc.com/avcenter/venc/data/w32.sircam.worm@mm.html August 15, 2001

[13] Symantec. "W32.SirCam.Worm@MM Removal Tool." August 10, 2001. URL: http://www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.removal.tool.html (August 15, 2001)