The Code Red Message in a Bottle
Jeffrey A. Tricoli[1]

Introduction:

On July 12th 2001, computers worldwide began seeing signs of what would arguably become one of the most significant security events of the past few years.[1] The original Code Red Worm (CRv1) had a rather quiet beginning, but Intrusion Detection Systems (IDS) across the globe were soon inundated with the following indication of its presence in their logs.

```
       256.36.243.21 - - [23/Jul/2001:11:16:24 -0400] "GET
/default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858
%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8
b00%u531b%u53ff%u0078%u0000%u00=a
HTTP/1.0" 404 207 "-" "-"[2]
```

The preceding signature illustrates the way Code Red used a buffer overflow in the Internet server application program interface (ISAPI) idq.dll extension[3] to compromise a web server running Microsoft Internet Information Service (IIS). Once infected with the original Code Red Worm, a compromised machine would start scanning for other vulnerable servers in a pseudo random manner, thereby continuing the cycle.

Their have been several versions of the Code Red Worm, most of which improved on the techniques of the preceding version. It is largely due to the improved scanning techniques in each additionally iteration that accounts for the speed in which these new versions spread, and the damage they caused. In fact, it wasn't until Code Red Version 2a[4] started scanning on July 19th, that the exponential growth of infected machines became apparent.

Purpose Of This Paper:

There are several lessons to be drawn from the Code Red incident, and this paper will focus on those I believe are the most important. These include the following areas:

- The Need For Faster Identification
- The Need For More Coordinated Analysis
- The Need For More Clear And Timely Warnings
- Identifying The Contributing Factors

Identification:

[1] Opinions or points of view expressed in this document are those of the author and do not necessarily reflect the official position of any other individual or entity.

The earliest known sightings of the Code Red Worm can be traced back to July 12th, 2001. However, the real alarm concerning CRv1 wasn't raised until several days later. I believe that this can be attributed to several factors including; a shortage of trained incident handlers and sensors, limited information sharing, and overall poor security practices.

Problem

While it is true that the number of security professionals is increasing everyday, the fact remains that the demand greatly outpaces the supply in this area. Additionally, those analysts that are qualified to understand what is occurring often do not have access to the information needed to make an accurate assessment. In the case of CRv1, all of these factors, in addition to the slow initial propagation rate, lead to a situation where the threat went largely unnoticed for several days.

Solution

Incident Handlers and Sensors:

Over the past year, several new innovative models for identifying threats have evolved. These include, the Internet Storm Center (www.incidents.org), www.dshield.org, and www.mynetwatchman.com among others. The idea behind these models is similar to that of weather forecasting with numerous sensors out on the web collecting information from which analysts can see trends before they become major threats. So far, this model has provided promising results; however, while these models are good for prediction, they lack a good mitigation strategy once something is discovered.

The previously mentioned models rely on both information from sensors and trained individuals to analyze that data. As previously stated, there are not nearly enough trained incident handlers to analyze all of the information coming in, and the situation is only worsened by the relatively small sampling of data being analyzed. Both of these areas need improvement, because as one area grows, so will the need for the other. As more sensors are deployed, the potential for identifying attacks increases, but the job of the handler becomes that much more challenging due to the amount of information needed to be processed and analyzed.

Information Sharing:

During the beginning stages when CRv1 was spreading, several groups started analyzing it without any knowledge of each other. While any new piece of malicious code stimulates a response on several fronts, the lack of some centralized location for sharing reliable information added to already conflicting reports. There are several reasons for this, the least of which is that some of these groups didn't even know that the other existed, and in those instances where they did know they existed, there were no guidelines for how analysis was to be performed or information shared. A quick look at the archives of www.securityfocus.com during this time period illustrates this fragmented approach.

Security Practices:

The other major hurdle in quickly identifying CRv1 is the overall poor security practices of the vast majority of Internet users. The fact that so many machines were vulnerable to this worm even after the vulnerability and patch had been known for over a month illustrates the pervasive lack of security awareness among Internet users. If even a small percentage of these systems had even modest security defenses in place, such as an intrusion detection system, the preliminary stages of CRv1 would have been detected much sooner.

Analysis:

As pointed out in the previous section, several groups were attempting analysis on CR simultaneously; however, very little coordination was taking place. The first comprehensive analysis was completed by eEye digital, on July 17th[5].

Problem

Whenever an incident such as CR occurs, it is important to perform analysis in a timely and accurate manner. It is also important to identify the strengths and weaknesses that certain groups and individuals possess to aid in analysis. Often, the rush to be the first group to analyze a new piece of malicious code can introduce flawed findings. This problem is likely to get worse as more complex code is being released and analysts are forced to make conclusions based on only limited amounts of real analysis. This is further worsened by the lack of standards for analysis, and reporting guidelines.

Solution

Standards:

In every respected scientific field there are specific rules that need to be followed in order for research to be accepted as analytically sound. Unfortunately, this scientific method has not been formalized in the relatively new field of "malware" forensics.

The only work to date in this area has focused on performing analysis without actually creating guidelines for others to follow[6]. It is important to make a distinction here between computer forensics, which is a field that has been established for some time, and "malware" or malicious code forensics. Computer forensics has been defined as "…involve[ing] the preservation, identification, extraction and documentation of computer evidence stored in the form of magnetically encoded information (data)."[7] Malicious code forensics on the other hand, is more focused on examining the evidence that is collected through the computer forensic stage, specifically on malicious code that may or may not have source code available.

Unlike the gathering of evidence itself, which has very strict guidelines, the actual examination of any unidentified code is still more of an art than a science. It is my belief that much of the contradiction surrounding what CR could and could not do was due in part to the lack of guidelines for examining malicious code. Clearly, this is an area where common standards for examination will aid the security community as a whole when the next piece of malicious code surfaces.

Reporting Guidelines:
Almost as important as the standards by which code should be examined, are the guidelines for the reporting of results.

Currently, when a piece of code is analyzed, the results contain different information depending on who actually performed the analysis. Some organizations don't even include the same information from one analysis to the next depending on the person writing the report. Clearly, the security community would benefit from having a framework for what information should be included in a malicious code analysis, similar to how the Open Systems Interconnection (OSI) model provides a framework for how protocols such as TCP/IP allow computers to communicate regardless of which operating system they use.

A perfect example of where having such guidelines would have been beneficial was when the different versions of CR started propagating in the wild. Without any guidelines for reporting findings, conflicting reports began surfacing over such facts as which version of CR did certain actions and what trigger dates were coded into which worm.

Better Warnings:

On July 30th, a group of private and public security experts meet in Washington, D.C. to warn the public about the CR outbreak.[8] This large-scale coordinated effort was the first time many of these organizations had come together to address a common concern. The message was simple, "there was a dangerous new worm on the loose that people needed to protect against."

Problem
As with any message, it is important to be aware of whom your audience is. Unfortunately, when it comes to warning users about computers, the intended audience is not always easy to reach. In the case of CR, the message needed to be broad enough to reach the highly technical user, such as a system administrator, as well as the average home user. The challenge then, is to craft a message that can at the same time reach every audience in a manner that will compel them to act without scaring them or insulting their intelligence.

Unfortunately this is not always the easiest thing to accomplish when trying to use the media as the means for expressing this message. For example, during the height of the CR outbreak I received a frantic message on my answering machine from my mother asking for help in applying the patch to stop CR. It took me almost a half hour of conversation on the phone to explain to her that she was running Windows 98 and therefore not vulnerable. Now, my mother is not a newcomer to computers, but clearly the message lacked enough focus, which lead to certain groups of individuals being confused.

Solution

Partnership:

The major lesson to be learned in this area is that while CR was bad, it could have been much worse. Soon after CR was released, at least two theories came out (Warhol Worm and the Flash Worm) describing how a similar outbreak could infect all vulnerable computers in under 15 minutes.[9] While these theories may sound fanciful, one need only look at the speed by which the Nimda Worm hit on September 18th to see how this possibility could become a reality.

Additionally, CR showed us that there is no one organization that has all the answers. It takes many groups working simultaneously, both in crafting the public message and working behind the scenes, to actually implement any changes and form a mitigation strategy.

Focus On One Audience At A Time:

It is certainly difficult to craft a message, such as the warning for CR, for such a large audience. In addition, it is important to remember that the original message needed to be crafted and disseminated before CRv1 was to start scanning again on the 1st of August. However, even with the large scale warning that went out on July 30th, there were still several thousand computers scanning, apparently infected with the CR worm.

So what went wrong? Why didn't people running the IIS software patch their systems? I believe that while extensive efforts were put into making the message as clear as possible, large segments of the Internet community either didn't care, or they didn't understand. While there is very little that can be done for those who don't care, those that do, need to have a more focused message. Something of a tiered approach to warning may be needed, where there are different versions of the same information depending on which audience is being targeted.

Crafting the correct message for the correct audience is a formidable challenge regardless of the organization. However, with such ideas as the Warhol Worm and the Flash Worm, it is only a matter of time before such methods are used in spreading the exploit for the next vulnerability. In the face of such a threat, there is nothing any one organization can do individually, it becomes vital for the security community to share their knowledge and warn the correct audience as quickly as possible.

In addition to warnings going out quickly, any information provided needs to be useful for the audience it is going too. For example, there is very little benefit in giving an average user the technical details behind a vulnerability. However, the same information is vital for the system administrator in charge of maintaining a server. It is up to the responsible warning entities to find this balance and learn from the successes and failures of the CR experience for the future.

Contributing Factors:

The Computer Emergence Response Team at Carnegie Mellon (CERT/CC) recently reported that they still receive reports of the Melissa Virus, which is over two years old![10] The hole that CR exploited had a patch available for download for over a month! The list of factors that contributed to CR having the impact that it did is quite long, but it is important to highlight some of the more significant problems so the community can learn from these mistakes.

Problem

The Internet was originally designed as a way to ensure the flow of information over a network in the event that a major catastrophe disabled significant portions of the infrastructure. However, it was not developed to ensure the confidentiality or integrity of that information. Since that time, the Internet has grown exponentially, but so to have the security concerns surrounding the products that take advantage of this new medium.

While there is no "one" factor that lead to CR being as successful as it was. However, there are some key areas that need to be highlighted. These areas include; increased sophistication of attacks, poorly designed hardware and software, increasing number of broadband connections, and the difficulty in fixing known problems.

Attack Sophistication:

The term "hacking" and "hacker" have been around since the beginning of computing and haven't always carried the negative connotation it does today. The term "hacker" originally referred to someone who was extremely knowledgeable about computers and could stretch a program to its limits through their mastery of the material. What a "hacker" could do was often considered extremely difficult due to the level of understanding that was needed to accomplish a successful "hack."

Unfortunately, as attacks have progressed and moved more into the main stream, the amount of skill needed to accomplish previously difficult tasks has decreased and created a new bread of hacker known as a "script kiddie". These script kiddies prefer ready-made scripts, or code, to accomplish what once was accessible only to the most sophisticated hackers. This trend has created a situation where the tools for attack are now readily available, thereby increasing the pool of potential "hackers."

In the case of CR, an attacker could have used the detailed information available about the buffer overflow in the IIS software to aid in the creation of the worm. Additionally, even if the author of the original version of the CR worm knew about the buffer overflow before it was reported, some of the subsequent versions were most likely created by script kiddies who simply modified the existing CR worm to fit their needs.

Poorly Designed Products:
Their have been coding practices that have been developed as far back as the 60's to guard against buffer overflows such as the one used in the CR worm. It is true that anything created by humans is bound to contain errors; however, this problem is only worsened by the rush to market in the computer industry as a whole.

Far too often, computer products are shipped to consumers with a default configuration that contains several security vulnerabilities. Users who are often untrained in security matters are left to figure out how to react to the overwhelming amount of information coming at them everyday encouraging them to update their systems and download some patch.

This problem continues to worsen as companies continually repeat the same problems over and over again in their rush to be the first to market. Additionally, vendors are increasingly incorporating more "features" for users that can expose sensitive information if not secured properly. Clearly, if there is any lesson to learn from CR, it is how fragile the Internet truly is to hackers exploiting default configurations of software.

Vulnerable User Base:
Going along with poorly designed products, are the networks that tie them together. CR exposed how truly devastating a piece of malicious code can be when introduced into a network that is relatively homogenous and has little to no way of identifying and stopping malicious traffic in a timely manner. In the case of CR, because the IIS software is so popular, and the role it plays in sharing information to a potentially hostile Internet, any vulnerability in that software can have widespread impact on the Internet as a whole.

With the rapid increase in the number of homes connecting to the Internet via high-speed access, attackers have a much larger base of computers to choose from when selecting a target. Because these high speed connections offer "always on" service, they stay connected to the Internet as long as the computer stays on, regardless of whether someone is at the computer or not. Individuals are often being attacked and they don't even know it. This may be one reason why many users were unaware that their computers were infected with the CR worm.

Difficult Fix:
It is not surprising that serious warnings are often lost in the background noise of the everyday computer user. Security often becomes a case of a self fulfilling prophecy, where security professionals are constantly raising alarms, to the point where, when a real threat comes along, it runs the risk of being perceived as just another "warning."

Even in those cases where the alarm has been raised above the background noise, implementing a fix can be difficult. System administrators are constantly being inundated with requests for mission critical things and security is often only one item on an otherwise crowded plate. It is also often the case that the sheer number of vulnerabilities is too much for an administrator to take care of with the limited amount of resources at their disposal.

Ultimately, the difficulty goes beyond only taking care of the systems under your control. When an incident like CR comes along, it appears that the entire Internet is attacking you. There is only so much hardening a system administrator can do when she is being attacked by thousands of hosts every day. It ultimately comes down to computer owners being conscientious about the health of their own system, thereby ensuring the health of the whole network.

Solutions

Each of the following solutions has been touched on in the preceding section, but it is important to pull them together and reiterate them here. As with most major problems facing society today, there is no one right answer, rather, it is a combination of several efforts that are needed to address the problems illustrated by CR.

Increased Awareness Among Vendors:
Software and hardware vendors need to include security as part of the foundation of their products rather than something to be added at the end of the development cycle. Vendors should also increase the amount of effort they put into stress testing their products prior to shipping. This includes creating a more secure “out-of-the-box” experience, so fewer of the vulnerabilities are included in a system or product when it ships.

Education And Awareness:
The idea of creating a more knowledgeable user base has been around for a long time, however it has never been more needed than it is now. The need for improved education and awareness also falls on the security professionals, who must learn how to focus on those issues most relevant to their systems users.

Increasing education is similar to the “80/20” analogy where 80% of the problems are caused by only 20% of the vulnerabilities. If users can become knowledgeable about those 20%, then the whole Internet would be safer.

Buying Power:
This is something of catchall category. Because market forces drive vendors, there needs to be a mechanism in place to ensure that security is a profitable venture for them. One such idea is the creation of something of an Underwriters Laboratory (UL) group that offers a third party subjective assessment as to the security of a product[11]. In this way, a company could have their products tested and approved and then use that as an incentive for people to buy their product.

Theoretically, having such a third party group verify a product could act as an additional requirement for vendors to meet before a customer purchases their goods. Such a group could rate the inherent security and soundness of a product. However, this only works until a users takes a piece of hardware or software home and changes the security installation parameters to suit their needs, potentially negating any benefit added. For example, a company making an operating system could have a requirement that the telnet service is turned off by default, however, that wouldn't preclude a customer from turning that service back on if they needed it.

Unfortunately, this idea would have some serious hurdles to overcome because companies often dislike new regulations on how they do business.

Conclusion:

Similar to the Melissa Virus, some versions of the CR Worm will most likely continue to infect systems for some time. CR was just the beginning of the new types of problems we will have to face in the near future. CR illustrated how quickly a threat to the Internet can spread across the globe and how difficult it can be to halt. It is important that the lessons highlighted throughout this practical and from individuals own experiences of this event, be used as another building block in an organizations overall defense in depth.

[1] Moore, David. "CAIDA Analysis of Code-Red." CAIDA. 4 Oct. 2001. URL:
http://www.caida.org/analysis/security/code-red/#ida

[2] Irwin, Vicki. "Handler's Diary-July 2001." SANS Institute. July 2001. URL:
http://www.incidents.org/diary/july2001.php

[3] Mitre. "CAN-2001-0500 (under review)." Common Vulnerabilities and Exposures. URL:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0500

[4] iDefense. "Code Red FAQ v1.0." iDefense. 10 Aug. 2001. URL
http://www.idefense.com/Intell/CI081001.html

[5] Maiffret, Marc; Permeh, Ryan. ".ida 'Code Red' Worm." EEye Digital Security. 17 July 2001. URL:
http://www.eeye.com/html/Research/Advisories/AL20010717.html

[6] Zeltser, Lenny. "Reverse Engineering Malware." GCIH Practical. May 2001. URL:
http://www.zeltser.com/sans/gcih-practical/revmalw.pdf

[7] Lunn, Dorthy. "Computer Forensics-An Overview." SANS Institute. 20 Feb. 2001. URL:
http://www.sans.org/infosecFAQ/incident/forensics.htm

[8] Becker, David. "Officials sound Code Red alarm." C|net news.com. 30 July 2001. URL:
http://news.cnet.com/news/0-1003-200-6718987.html

[9] Grim, Gary; Jonkman, Roelof; Staniford, Stuart. "Flash Worms: Thirty Seconds to Infect the Internet." Silicon Defense. 16 Aug. 2001. URL:
http://www.silicondefense.com/flash/
Weaver, Nicholas. "Warhol Worms: The Potential for Very Fast Internet Plagues." 15 Aug. 2001. URL:
http://www.cs.berkeley.edu/~nweaver/warhol.html

[10] Carpenter, Jeffrey. "Computer Security Issues that Affect Federal, State, and Local Governments and the Code Red Worm." CERT/CC. 29 Aug. 2001. URL:
http://www.cert.org/congressional_testimony/Carpenter_testimony_Aug29.html

[11] The Center For Internet Security. URL:
http://www.cisecurity.org/