Note on public release

21 Sep 2012

I originally wrote document for internal discussions for security groups I was involved with (hence the “For Official Use Only” header). Given that the document is over 10 years old, and malware had made considerable advancements (e.g., stuxnet), I figure it is OK to release this publicly.

Todd Heberlein
LTH@NetSQ.com

Understanding Strategic Malicious Code Attacks: Some Initial Thoughts

Todd Heberlein
Net Squared, Inc.
todd@netsq.com

This paper provides some of our initial thoughts on strategic malicious code attacks. We examine three primary areas. First, we consider what makes an attack a strategic one. Next, we look at what damage can be done to computers, their operational capabilities, and our willingness to use them. Finally, we examine how such attacks can be delivered.

1 Strategic Malicious Code Attack

For the purposes of this paper, we define a strategic malicious code attack, or just "malcode attack", as follows:

> A malcode attack is the execution of code of an attackers' choosing on an appropriate portion of the Internet's computers to achieve strategic disruption of activities.

An "appropriate portion" may range from a significant portion of general Internet's computer to system to disrupt the national economy, to destruction of a smaller set of computers necessary for the proper operation of strategic assets. For example, we could easily envision a scenario where an attacker could destroy hardware on 90% of the world's computers. Such an attack would devastate the United States economy for years and seriously inhibit our ability to defend ourselves against physical attacks by nation states or terrorist groups. Alternatively, destroying a much smaller number of computers necessary for proper operations of some strategic asset could have a major impact on our nation. For example, taking out the Air Traffic Control System or computers throughout the nation's electrical grid could grind the economy to a halt for days. A well-timed attack could have strategic results as well. Imagine an attacker suddenly destroying most of the Pentagon's computer just as a major military operation begins (e.g., an invasion of Iraq).

So a strategic attack can involve huge numbers of computers across the Internet or just a high percentage of the computers necessary for a strategic asset (ATC, military command and control, etc.). The important point is that the result of the attack has strategic effects.

2 Wide Range of Disruption

What are the ranges of disruption that would characterize a malcode attack? For the most part, we believe that a malcode attack should have a significant time component; although, there will be exceptions. Many of the more notorious Internet-wide security incidents, including Morris, Melissa, Love Bug, and Code Red, were largely benign and relatively simple to clean up. Their time components were largely negligible, and they could hardly be considered strategic threats. Subsequent variants had more malicious effects, but because they followed in the footsteps of the earlier benign attacks, much of the Internet was immune to their attacks.

However, these later malicious worm variants, and the longer history of PC viruses, can show the potential time requirements and costs of a malcode attack. Here is just a sample set of actions a malcode attack may try:

Overwrite or reformat entire hard disks. While some worms that achieved wide-scale penetration deleted or modified some files, none have made a significant effort to systematically destroy all data on a computer system. The time for system administrators to reinstall and re-patch all operating systems and recover all data from backups for all the machines they are responsible could easily take days or weeks. And since many systems are not backed up on a daily basis, much data would be permanently lost.

Mangle the partition table on a hard disk. When trying to install multi-boot operating systems we have on more than one occasion mangled the partition table on the front of a disk so badly that no mainstream operating system could be installed. Windows 95, 98, and NT all failed. Several repartitioning tools in Redhat's Linux failed as well. On each occasion, only an old version of Solaris for x86 could fix the problem. Since few people have Solaris for x86 lying around, repairing these disks would be a problem, and with much of the Internet down, distributing new tools to fix the problem would be a problem.

Distributing sensitive information. One of the fundamental problems of digital data is that it can be copied an infinite number of times, so once it is released on the Internet there is no way to repeal the data. Malcode could peruse file systems looking for any or all potentially sensitive information, from medical information to romantic missives to financial information, and post them in public locations such as IRC channels, chat rooms, web sites, and mailing lists. Imagine the cost of having to halt all credit card transactions until virtually every card in the world is re-issued?

Flashing BIOS. Much of today's computer hardware, from motherboards to video cards, is field upgradeable. This is achieved by using erasable programmable read-only memory (EPROM) chips. The most well known EPROM is the motherboard's Basic Input/Output System (BIOS) chip, and the process of upgrading it is called "flashing the BIOS." Unfortunately, since the BIOS is required to even boot from a floppy disk, a bad BIOS flash could convert a PC into a paperweight[1]. In other words, a wide-scale malcode attack could create the equivalent affect of an Electro-Magnetic Pulse (EMP) attack on computers. The virus known as CIH and Chernobyl included this capability as part of its attack[2] [3]. Replacing BIOS chips or motherboards for much of the Internet could take months or years.

Damaging hardware. Malcode could damage computers hardware in other ways as well. The XFree86 configuration file warns users that they can permanently damage their monitors (due to "overdriving" the monitors) if they provide incorrect settings. The XFree86.org documentation includes a disclaimer that says in part "XFree86, allows the user to do damage to their hardware with software", and " if you think the Xserver is frying your screen, TURN THE COMPUTER OFF!!"[4] There are anecdotal stories of monitors actually smoking, so the fact that the first test of a new monitor configuration is called a "smoke test" is very appropriate. Some system performance testing software includes warnings that in some cases enough heat can be generated from the testing that computers can be damaged. For example, a Linux application called *cpuburn* is designed to "cause maximum heat output on P6 and P5 grade Intel-architecture chips", and it may cause "permanent damage to electronic components."[5]

---

[1] http://www.techtv.com/callforhelp/answerstips/story/0,24330,2487451,00.html
[2] http://www.cert.org/incident_notes/IN-99-03.html
[3] http://www.symantec.com/avcenter/venc/data/cih.html
[4] http://www.xfree86.org/4.2.0/chips8.html
[5] http://linuxquality.sunsite.dk/articles/testsuites/

We believe an exhaustive analysis of the ways a computer system, or a strategic asset through its computer systems, can be disrupted is critical to understanding the potential threats, because without understanding the threats we cannot prepare to meet them.

3 Distributing Malicious Code

In the beginning, we stated that a malcode attack required that an attacker run instructions of his choosing on the computer systems, but we did not say how those instructions should be delivered to the computer. Likewise, we did not specify when those instructions needed to be delivered to the computer. For maximum affect, we believe the execution of the malicious code needs to be synchronized, but the delivery of that code does not need to be synchronized.

When most people think of an Internet-wide attack, they think of fast spreading worms such as the Love Bug email worm, the Code Red server worm, or the very fast (and so far hypothetical) Warhol[6] and Flash worms[7]. The advantage to the attacker of this approach is that because of the speed of the attack, the Internet community has very little time to respond to such an attack. One disadvantage to the attacker is that the attack is also very noisy and easy to detect.

A more subtle attack could quietly infect many machines over a longer period of time. For example, the zombie programs used in distributed denial of service attacks (DDOS) are often seeded over a period of weeks or months. This long period of initial penetration is possible in part because the end system is not adversely affected by this part of the attack.

Other attack approaches can take advantage of the natural connectivity of hosts on the network. In "How to Own the Internet in Your Spare Time"[8] the authors describe how a flaw in peer-to-peer networks such as KaZaA could be used to subtlety propagate malcode to millions of machines within a month. Indeed, there have already been viruses that target KaZaA[9] [10] [11], and of course it does not help that KaZaA includes its own backdoor called AltNet[12].

Contrary to popular opinion, malcode does not need to include propagation code. Peer-to-Peer networks such as KaZaA can provide the propagation capability free of charge. Also, we have observed that interesting content is self-propagating. By that, we mean that users will happily propagate the content to coworkers, friends, and family. Several years ago I wrote a sensor to determine how many executable files were passed around by email. I was surprised by the amount code being passed between users, and the number one executable at that time was gerbil.exe. Apparently it is an animated scene of a gerbil in a microwave (see [13] for more examples of such executables, including the very popular frog in the blender). We have also received electronic gift cards that were essentially small executable programs.

Also, soon after 9-11, a number of editorial cartoons and PowerPoint slides designed to boost morale were circulating by email. One such cartoon we saw a number of times, including at our local auto repair shop, is the eagle in Figure 1 (used without permission). A well-crafted image that would be willing propagated by users through email, newsgroups, or even posted on

[6] http://www.cs.berkeley.edu/~nweaver/warhol.html
[7] http://www.silicondefense.com/flash/
[8] http://www.cs.berkeley.edu/~nweaver/cdc.web/
[9] http://securityresponse.symantec.com/avcenter/venc/data/w32.shermnar.worm.html
[10] http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.electron.html
[11] http://www.techtv.com/news/security/story/0,24195,3390306,00.html
[12] http://news.com.com/2100-1023-873181.html
[13] http://greenstuff.decoder.com.au/Fun/programs.htm#noadds

web sites, could exploit a flaw in JPEG decoder such as CVE-2000-0655[14] [15] to infect millions of computers.

Figure 1: Eagle Sharpening Talons

Compelling content, by its nature, is self-propagating. In some cases the content may be represented by executable code, in which case the attacker can easily embed malicious instructions that are set to go off at some future time. In other cases, flaws in content decoders such as HTML renderers[16] or JPEG decoders could be exploited to hide malicious code inside the content.

Malcode can also be delivered straight to the computer without the need for a propagation mechanisms of any type. For example, if an attacker acquires a small set of computers with enough bandwidth from which he could launch his attack, he could potentially attack millions of computers in hours without the need of propagation. If the attack exploits a vulnerability in a UDP application (e.g., in a remote procedure call (RPC) program), a single packet to each computer may be all that is necessary. If the attack succeeds just by downloading the email (e.g., without needing the user to click on an attachment), then the attacker can simply buy millions of email addresses from bulk email providers. Also, if the attack can be launched by users simply reading a web page (e.g., a bug in the HTML renderer or any of the supporting libraries or applications the web browser might use to display a page), then an attacker could reach millions of computers by penetrating a small number of high-profile sites such as Yahoo! and eBay.

The user could voluntarily place the malicious code on their computer. If the attacker can create a "must have" application, he can insert Trojan code inside the application and let millions of users install the application on their systems. As mentioned previously, KaZaA is an example of this: an application that millions have voluntarily downloaded but that also contains additional functionality via AltNet. According to Brilliant Digital Entertainment's annual report (form 10KSB) filed with the SEC, AltNet provides technology that can "push" content such as music and news directly to the user's computer[17]. A malicious attacker could exploit a capability such as this to destroy millions of computers.

A similar approach would be for an attacker to gain employment for a developer of widely used applications. Microsoft would be an obvious target. Malicious code embedded into the operating system, the web browser, or the Office applications could reach 90% or more of the world's computers. The only issue left is how the attacker would synchronize the attack so that it occurs nearly simultaneously.

[14] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0655
[15] http://online.securityfocus.com/bid/1503
[16] http://www.kb.cert.org/vuls/id/443699
[17] http://news.com.com/2009-1023-873905.html

4 Summary

This paper focused on what can be considered a strategic malicious code attack, what type of damage might be caused to the targeted systems, and how the attack might be delivered. Unfortunately, while we have only scratched the surface of these issues, we can already see that the problem is vast. Sadly, virtually all of the techniques described in this paper are already known and many have actually been implemented in small-scale attacks. The only thing missing is a malicious attacker to put all the pieces together.

We believe a deep understanding of strategic malicious code attacks is necessary to craft policy, develop concepts of operations, appropriate funding, and deploy assets. Difficult questions will have to be asked. For example, should large ISPs or backbone providers be legally required to deploy mechanisms necessary to ameliorate the effects of a malcode attack? The price of not understanding the problem and asking the hard questions could be the destruction of our global economy and our way of life along with it.