On the Ethics of Exploits and Exploitation

Max Goodman

03/10/2009

1 Introduction

Exploit, n.
1. To make use of selfishly or unethically [1].
2. In computer security, an unethical or illegal attack that takes advantage of some vulnerability [2].

To *exploit* someone is to manipulate them selfishly or unethically, often against their own interests. In the computing world, the word takes on a double meaning: to utilize a flaw or hidden opening to take unauthorized control of a computing system. Exploitation of both forms can occur via computers in the digital age, for both beneficial and detrimental causes: from stealing credit card numbers, to circumventing copy control mechanisms, to both infecting and protecting computers from viruses.

In today's world, technology plays vital roles in business, entertainment, and government. As the ubiquity of computing increases, people are putting increasing faith in the technology they depend upon. In 2009, people bank, invest, view medical information, and communicate, all through the same wires and web browser software. People rely upon computers and software to be safe, secure, and free of critical flaws.

The ubiquity of software has given great power to people who are skilled at controlling it. Devastating personal attacks and billions of dollars of damage are within the reach of a single person typing at a terminal. As a computer programmer, I know there will almost always be cases of bugs in software I use and write. There may also be cases where I am tempted to exploit these bugs in my line of work. As an ethical computer user, I ask, how do I define the ethics of exploitation?

I believe that while it is wrong to exploit a human being, exploiting computer flaws can be done for right and wrong reasons. In this paper, I will explore two contemporary examples of technological exploitation. To conclude, I will discuss perspectives for evaluating these actions.

2 The Storm Worm

On January 19, 2007, a new and unfamiliar worm entered the computer security scene. The Storm worm had a unique mix of characteristics specifically designed to make it difficult to track, difficult to discover, and dangerous to both infected computers and Internet servers. As security expert Bruce Schneier put it, the Storm worm was "the most successful example we have of a new breed of worm"[3].

What was it about the Storm worm that made it stand out as a threat? In short, because infected computers form a peer-to-peer network known as a "botnet". Botnets are made up of thousands to millions of infected computers whose resources are pooled by hackers to send email spam and overwhelm web servers. In addition, infected computers can be manipulated individually to install spyware and other viruses.

In the past, large botnets have been based on centralized control. Once law enforcement tracked down virus creators, they would simply shut down the botnet control servers. Without commands, the botnet would become defunct, and activity would vanish rapidly. The fact that Storm worm computers exist in a decentralized peer-to-peer "cloud" nullified this approach. The Storm worm botnet uses a web of relationships between infected nodes ("zombies"), making it difficult to determine the origin of commands, or even to determine the size of the network.

How then, does one address such a threat? In late December 2009, four German researchers presented an answer. In their presentation to the 25th annual Chaos Communication Congress (an international hacking conference), Georg Wicherski, Tillmann Werner, Felix Leder, and Mark Schlösser presented a powerful approach to dismantling the Storm worm botnet[4].

The researchers discovered vulnerabilities in the network protocols of the Storm worm and designed techniques for exploiting them to take control of infected computers. Their efforts resulted in a piece of software able to infiltrate the Storm network, take control of infected computers, and clean the Storm worm from them. Thanks to their discoveries, the researchers could launch an attack on the Storm worm botnet that would bring this 2 year case to a close.

The catch? They didn't. "If we released the cleanup code and the distributed cleanup stuff, maybe someone else'll push the button, but we wont – because it's illegal in Germany, and even releasing the code with all pieces is illegal because of the infamous 202c, because it can take over computers with this, so we'll only be releasing bits and pieces and someone else has to assemble this"[4]. For legal and ethical reasons, the programmers chose not to put their software into action.

The "infamous 202c"[5] is a German law, adopted in 2007, which criminalizes using or distributing passwords or computer programs to gain unauthorized access to data. In essence, 202c makes using and publishing technological exploits illegal. Because of this law, the researchers who cracked the Storm worm could talk about their findings, but couldn't share their full source code or their complete methods. As ethical hackers, the researchers followed their laws to the letter. Thus, the Storm worm lives on.

3 Circumventing DVD CSS

Many a beginning Linux user has discovered, to their chagrin, that they are unable to play DVD movies in Linux. The problem isn't a technical one; they may have all the necessary hardware and media player software. What they are missing is one tiny detail: a tiny 40-bit key required to decrypt the contents of the disk[6]. This fact can be very frustrating to consumers, since they purchased their DVDs with the expectation of playing them.

Nearly all DVD movies are encrypted using CSS (Content Scramble System), an anti-piracy measure. The idea was to parcel out decryption keys to trustworthy vendors, so that copyrighted DVD movies would only be usable with licensed DVD players[6]. Unfortunately, this scheme excludes open source software players, because it is based on the idea of keeping the decryption keys secret. Even if an open source developer licensed a CSS key, they would be unable to distribute the key with their source.

However, there is one way to get DVD playback working in Linux: by installing a program that cracks the CSS encryption codes. It was found that the encryption used by CSS is very weak, and can be cracked within seconds by a modern computer. This vulnerability in the CSS encryption can be exploited to remove the need for a licensed decryption key.

In mid 1999, the first CSS cracking programs were developed, including the famous “DeCSS”[6]. Unlike the Storm worm researchers, these hackers released their code, and were swiftly met with legal charges from organizations such as the Motion Picture Association of America (MPAA). The problem is that cracking CSS is illegal in the U.S. because of the Digital Millennium Copyright Act (DMCA), a law that “imposes civil and possible criminal liability for the circumvention of access control measures and for the distribution of technology to circumvent access or copy controls”[6].

By breaking the encryption of CSS, users are actively circumventing the copy control system, even if they do not intend to make a copy. Thus, Linux users are faced with an ethical conundrum: to play a DVD movie that was legally purchased on the Linux operating system, users are forced to break the law.

In response to the legal threats, DeCSS users spread the software over thousands of copies on the Internet[6], assuring that the software could not be taken down. Variants of the original CSS crackers are still available to this day. It is debated whether these programs were first developed in the interests of piracy or open source playback[6], but the continued availability of such software benefits both.

4 Discussion

Both the Storm worm research and the DVD CSS circumvention share a common thread: technological vulnerabilities were exploited to benefit a group of people. Vulnerabilities in the Storm worm were exploited to clean PCs, and the vulnerable CSS encryption scheme was exploited to allow unlimited access to movies. In each case, the desired action wasn't possible without exploiting an unintended vulnerability. Was it right to do so?

To justify using the Storm worm software, one has to address both taking unauthorized control of computers, and (in Germany) the willful act of breaking the law. In the case of the DVD CSS circumvention, bypassing movie copy protection could be viewed as an act of piracy, exploiting the people who made and invested in the film. These are not simple, one-sided issues – as few questions of ethics are – and require careful consideration of the circumstances.

When assessing the ethics of technical exploits, it is important to consider whether people were exploited in the process. In both cases, while it is possible to argue that both technical exploits lead to human exploitation, the exploitation is indirect. This follows from the definition of exploitation used in the introduction: "To make use of selfishly or unethically"[1]. While these exploits do take unauthorized control of technology selfishly, they do not directly use human beings selfishly. In other words, the exploitation is of human property. Considering this, we can restrict the question of ethics to matters of property.

From a deontological (rule-based) perspective, we could consider exploitation in any form to be wrong. From this point of view, one could argue that the technical exploitation in both the Storm worm and DVD cases is wrong. The problem with this point of view is that it is a personal thing: different people may have different views on what exploitation is, and whether it is right.

Alternatively, from a utilitarian standpoint (considering the net good), we might measure each form of exploitation by their end results. Exploiting computers infected with the Storm worm would hopefully result in removed infections, reducing damage to both the computer owner and other Internet users. However, there is the risk of the process going wrong, resulting in unauthorized damage to computers. The Storm worm cleanup should be judged in terms of the benefit to the commons (the removal of a harmful botnet, as well as personal exploitation) versus the risk of damage to the commons.

Similarly, the creation of software that cracks the DVD copy protection scheme can be viewed in two lights. The net benefit of the software's existence is that consumers are enabled to freely view their legally purchased DVDs. Unfortunately, the same technology can also be used for the purpose of piracy. Thus, one must weigh the freedom this software has given to consumers against the power it gives to pirates.

5 My Opinion

With knowledge comes great power, and knowledge of technology can be used to exploit machines and their users. I believe that exploitation of technology is not inherently right or wrong; rather, one must consider exploits by their net effect (a form of utilitarianism), and by taking the complex human effects into account. In this paper, I have presented two cases where I feel people benefit from technological exploitation.

Human and technological exploits are a difficult topic to discuss neutrally, because they are most commonly associated with their abuse. It is far more common to hear about unethical technological exploitation (e.g. Storm worms and DVD piracy) than ethical uses of hacking and exploits. How, specifically, can exploits be used ethically? I believe that in some cases, the rules put in place by legal and technical limitations are fundamentally wrong, and that in these cases it is right for individuals to use exploits to bypass the limitations.

In both of the cases I have described, exploiting technology enables individuals to act beyond the limits dictated to them. The Storm worm exploit takes control of computers remotely through vulnerabilities in the worm design. The DVD CSS exploit decrypts movie data without a key through vulnerabilities in the cryptographic design. In my opinion, the results of these exploits are more beneficial than they are harmful.

In the case of the Storm worm, I think that the benefits of cleaning up the botnet far outweigh the risks. The removal of the Storm infections benefit everybody. However, as an action for the good of the commons, I think that any sweeping action should be considered by a body responsible for representing and protecting the commons. I believe that it would be ethical to take control of people's computers for the strict purpose of cleaning them, and that a government or law enforcement body should have the authority to do so.

The predicament of technological exploits is reminiscent of the common phrase: "if guns are outlawed, only outlaws will have guns." I think that for ethical computer users to be competent with the reality of exploitable technology, they will need to be able to study and practice the art of exploitation. For this reason, I cannot fault the act of discovering exploits alone. Similarly, I think that the distribution of such exploits (such as prohibited by 202c) is acceptable for the same reasons. Exploits are not created; they are discovered – and once an exploit is discovered, it is not beneficial to try to hide it.

Considering the DVD CSS exploit, as a Linux user myself, I have a personal stake in the ethics of using and distributing this software. However, even when I attempt to view the situation from the point of view of the DVD producers and copyright holders, I find the use software to be acceptable. In my opinion, the DVD copy protection scheme was intended for a world of proprietary hardware and software vendors, where it was feasible to manage the decryption of disk content without the consumer ever having to know about it. The reality is that the surge of open source media player software broke this model.

If I shipped a product that was completely broken for a segment of my customers, I would find it acceptable for them to bypass protection mechanisms to use the goods that I sold them. Thus, I believe that the use of DVD CSS cracking software to watch movies is ethical. By extension, I believe the distribution of such software is ethical as well.

References

[1] Houghton Mifflin Company. Exploit, 2004. Available from World Wide Web: `http://dictionary.reference.com/browse/exploit`.

[2] Computer Language Company Inc. Exploit, 2008. Available from World Wide Web: `http://dictionary.reference.com/browse/exploit`.

[3] Bruce Schneier. Gathering 'Storm' superworm poses grave threat to PC nets, October 2007. Available from World Wide Web: `http://www.wired.com/politics/security/commentary/securitymatters/2007/10/securitymatters_1004`.

[4] Georg Wicherski, Tillmann Werner, Felix Leder, and Mark Schlosser. Stormfucker: Owning the storm botnet, February 2009. Available from World Wide Web: `http://media.ccc.de/browse/congress/2008/25c3-3000-en-stormfucker_owning_the_storm_botnet.html`.

[5] Strafgesetzbuch. Vorbereiten des ausspahens und abfangens von daten, November 2007. Available from World Wide Web: `http://dejure.org/gesetze/StGB/202c.html`.

[6] Rob Warren, Paul Fenimore, Sean Standish, Eric Seppanen, and Wendy Seltzer. The openlaw DVD/DeCSS forum frequently asked questions (FAQ) list, May 2000. Available from World Wide Web: `http://cyber.law.harvard.edu/openlaw/DVD/dvd-discuss-faq.html`.