Computer Security and Rootkits

by

Jameel Alsalam,
Somnath Banerjee,
Grant Musick
and
Rares Saftoiu

Table of Contents

1. Introduction

Computer security has been a problem since the inception of computing itself. In the beginning, computer security was enforced by restricting physical access to the equipment itself. However, as networked computing became more and more prevalent, the restriction of physical access was rendered insufficient. An attacker hundreds of miles away was now able to compromise a computer system remotely. With the dawn of the Internet age and the interconnection of millions of computers, remote attacks have become commonplace.

In response, a new branch of computing came into existence. As attacks became more and more frequent, began to offer their services to secure computer systems. Thus came the anti-virus and firewall vendors who help protect computer systems by keeping them clean and shielding them from remote attacks.

The establishment of the computer security industry gave rise to an arms race between two groups – the criminals or black hats trying to get their malicious software onto other computer systems and the security vendors or white hats trying to detect it and remove it before it causes damage.

In the middle of this cyber arms race is an attack technique that has made life extraordinarily difficult for the white hats. It is a technique that hides what an attacker is doing to a computer system in such a way that the computer's owner may never figure out that someone else is running his or her machine. The technique is called "rooting" a system and at the heart of it is a software package called a rootkit.

2. What is a rootkit?

A rootkit is a suite of one or more programs that allows a third party to hide files and activities from the administrator of a computer system. An intruder takes advantage of one or more known vulnerabilities on a particular computing platform to deliver and install the rootkit. Once the rootkit is in place, the intruder can use the infected system while remaining undetected.

The original intent of rootkits (circa 1996) appears to have centered simply on hiding programs that would allow an attacker to "sniff" or spy on traffic going to and from a computer system. They earned the name "rootkits" because they were mainly used on Unix derived computer systems where the top-level administrative account is called "root". Thus to "root" a system is to obtain top-level administrative privileges and hence obtain full control of the system.[1]

In the last few years however, attacks have grown in sophistication and the targets have changed. Home users (especially users of Microsoft Windows family of operating systems) are now the primary targets. Not only are attackers trying to harvest information from the targeted computer systems, they also take over the systems and use them as building blocks for "bot-nets". These bot-nets can then be used in a variety of ways, including spam forwarding and extortion. While performing the latter, an attacker demands money from a corporation, threatening a DDOS attack against their corporate website(s) as a consequence of non-payment. Even well-known and respected companies have been known to use rootkits as a way to enforce their own DRM policies or keep end-users from eliminating adware.

Rootkits of varying levels of sophistication are known to exist for all major operating system platforms, although all types can be dangerous if a system administrator is not wary[2]:

The earliest type of rootkit consists of replacement programs for commonly used administrative programs on the computer system. The replacement programs hide the activity of the actual payload from the system administrator. These kits are almost exclusive the Unix/Linux family of operating systems. These will be referred to as first generation rootkits.

The second generation rootkits use "hooks" to redirect application calls to operating system (kernel) functions. The hook is used to intercept such legitimate function calls and reroute them to the rootkit's own version of the function. They are more difficult to detect because they have moved a level closer to the kernel. A cousin to this type of rootkit is a library kit that subverts non-system calls from standard applications, such as a word processor, thereby controlling the application's interaction with the underlying operating system.

Attackers then began to employ even more sophisticated techniques. Third generation rootkits use techniques that go deeper into the system and modify dynamic data structures in the kernel itself. This type of modification is very difficult to detect and, for the most part, can only be spotted by using heuristics that analyze statistical patterns of memory usage.

The latest and greatest of rootkits are now aiming to subvert the memory of other applications running on the system, presenting the applications with a false view of the underlying system. This would effectively hide the rootkit from any antivirus scanning

software. This type of rootkit was the subject of a paper at the Blackhat conference in the summer of 2005 in Las Vegas.

3. Current vulnerability assessment

3.1. Windows vulnerabilities (2k/2k3/XP/64-bit)

Windows is by all accounts the most popular operating system on the planet. Although exact market share numbers are not known, it is estimated that Windows has at least 90% of the home user market share, therefore constituting a prime target for attacks.

Unfortunately, Windows has some of the biggest security holes as well. Almost everyone running Windows at home runs it as the top-level or root account "Administrator" or as an account in the "Administrators" group. Indeed, many programs for the operating system will not install or run correctly if they are not accessed from an account with top-level permissions.

Windows rootkits primarily consist of the second and third generation types that subvert the operating system kernel through device drivers. They will either install new versions of the device drivers on the system to "hook" and divert regular operating system calls to their own code or they will "patch" their own code into existing device drivers. With the elevated permissions that most Windows users run under, it is trivial to install this type of rootkit once you trick the user into running malicious code on the system.

There are multiple methods by which rootkits get installed on Windows. One common approach is to use web-page links to trick the user into running an ActiveX control or some other program containing a malicious payload. When the user clicks on a

link, the rootkit and payload are installed on the computer just like any other installable piece of software.

Windows also has tight integration of many of its subsystems, providing many venues for potential exploitation. For example, the image processing subsystem can be exploited to run arbitrary code with top-level permissions. Such was the case with a vulnerability found last year (2004), where a user could exploit a buffer overrun in the Windows image handling software to execute arbitrary code on a user's system. This vulnerability made it trivial to compromise systems. All a malicious website would have to do is redirect the user to a web page with a bad image on it. Once the user reached the malicious web page, the system was automatically compromised.[3]

Another attack vector is through system level applications such as SQL Server 2000. The so-called SQL Slammer worm caused a lot of mayhem in 2003 by exploiting a buffer-overflow vulnerability to turn each system into a virus generating machine. Although this particular exploit was easily detectable, a more delicate use of this vulnerability would have been to surreptitiously install a payload to monitor information on the system.

As recent as several months ago, in October 2005, the first rootkit infections through instant messaging (IM) software were reported. The rootkits are installed when the attacker convinces the IM target to click on a link in their IM client to launch the install program[4].

As if all the illegal vectors were not enough, rootkits can also come from organizations you would expect to be upstanding citizens. Earlier year Sony used a rootkit to enforce their own DRM scheme on home users' computer systems. The

software was bundled with certain music CDs. When a user listened to a CD on his or her computer, the software was installed and hid several pieces of code intended to stop people from abusively copying the music they purchased.

However, it only took a couple of days before people started realizing they could use this "benign" rootkit for other activities, such as hiding their malware, spyware, or as in a more unusual example, hiding cheat programs for the popular online role-playing game World of Warcraft[5].

3.2 Linux/Unix/OS X vulnerabilities

The other major family of operating systems consists of Linux, Unix and OSX. Together they form an extended family which share the same basic heritage and kernel design. These systems will collectively be referred to as *Nix systems hereafter.

*Nix systems tend to be less vulnerable than Windows systems for several reasons. They are not as widely deployed, which means less exposure to attacks. Where they are deployed, they are often (with the exception of OS X) deployed by people with superior knowledge of computers. Their users are encouraged by the system to not logon as "root", making it more difficult for software to be installed by a tricked user. Moreover, these systems may not support dynamically loaded device driver modules for the kernel, which removes an entire class of vulnerabilities.[6]

On the other hand, *Nix systems are often used for mission-critical roles and can be much more valuable targets for an attacker to subvert. Another potential problem lies in the fact that the underlying system source code for some of these systems (e.g. Linux, Mac OS) is freely available. Attackers can examine such code for vulnerabilities and devise new attacks.

Attacks on *Nix machines tend to be different than those on Windows. Users are encouraged not to run as "root", therefore limiting the amount of control they have on the system. Even if a user were to click on a malicious website that attempted to execute code on their system, chances are the user would not have sufficient permissions to install anything particularly harmful.

Instead, attacks on the *Nix systems tend to revolve around remote attacks through services with known vulnerabilities and through gaining access to the system via an account with a weak password. Once an attacker gains access to a system, he or she then uses known vulnerabilities within the system to elevate his or her privilege to that of the root account. Often, attackers will use these exploits to open up an SSH terminal session and start installing their payloads.

An example of one such attack used a Solaris (Unix) vulnerability reported by CERT March 30, 2001 where a buffer overflow attack in snmpXdmid allowed an attacker to post a rootkit on the system. This rootkit added new versions of the ps, netstat, ls and find commands so that it could hide its activities. To control the system and communicate, the rootkit added root privilege telnet and ssh sessions on the system to facilitate remote control, as well as an IRC proxy. Last, but not least, it installed a packet sniffer to monitor network traffic. The system logs were altered to hide the infection.[7]

One type of vulnerability that plagues the *Nix systems in much the same way Windows is plagued by device driver rootkits involves technology called Loadable Kernel Modules (LKM). Although LKM can be used for many purposes, they often are allowed so that new device drivers don't need to be compiled into the *Nix operating

system kernel. However, this vulnerability can be mostly mitigated by compiling the *Nix kernel with LKM turned off.[8]

One last note about vulnerabilities in the *Nix systems: there appear to be no known rootkits for OSX systems, at least not in the wild. A rootkit was recently made for OSX as a proof-of-concept, but it could not practically be spread from system to system. It should be noted, though, that OSX is based on a Unix variant called FreeBSD and any rootkit written for that system could potentially affect OSX.[9]

4. Estimated Attack Damage

As can be seen in the previous section, all modern computer systems contain multiple vulnerabilities which can be exploited by a rootkit. But what are the consequences of having your computer system "rooted"?

The effects range from the traumatizing, but relatively harmless identity theft for individual home users, to a coordinated cyber assault on critical governmental or enterprise computer systems. In this section we will discuss the losses for home and enterprise users separately since they each have different uses for computer systems and different stakes to lose when they get compromised.

4.1 Home users

4.1.1 Identity and private information theft

This includes stealing of credit card information, bank account information, social security numbers, software license activation keys, etc., and using the stolen identity for profit or clandestine activities.

A rootkit can be used to hide several common surveillance techniques, such as a key-logger that captures input from the keyboard or a packet sniffer that monitors traffic to and from a server. According to some estimates, 80 to 90 percent of computers are infected with spyware. Assuming that fifteen percent of these users transfer identity related information, there are over twenty six million people that are likely victims of identity theft[10]. Another estimate by UW professor Steven Gribble states that between 75 and 80 percent of computers were infected with spyware one time or another[11].

With no way to detect the spyware, an attacker can easily monitor computers and harvest credit card information. Although major credit cards provide zero liability in the event credit cards are compromised, recovering from identity theft can still be a hassle for the consumer. Consumers could spend a lot of time trying to pursue the issue with the credit card company, have to wait for a week or two for a new card, reset their automatic payments, and have their credit history wiped out[12].

Less commonly, harvesting information could be used for identity theft to enable illegal immigration, terrorism, espionage, or changing identity permanently. It may also be a means of blackmail, especially if medical privacy or political privacy has been breached and revealing the activities undertaken by the thief under the name of the victim might have serious consequences such as loss of job or marriage.

Stolen social security numbers can be used by illegal immigrants to find jobs and obtain other services, which may cause serious problems for the victim[13]. Identity theft can be a source of funding for terrorist groups. Terrorists can use stolen information to pay for their operations as well as create false identities[14].

4.1.2 Turning home user's computers into zombies for unethical/criminal activity

When a user's computer has been infected, an attacker can use this machine as a zombie. These machines under the control of the attacker can also be used to launch distributed denial of service (DDOS) attacks or simply to send out spam.

According to Newsweek, (November 7, 2005) an incredible 60% of all spam is estimated to come from so-called zombie computers. A consumer education site states "the widespread use of zombie computers to commit crimes over the Internet presents a very real danger to law-abiding computer users"[15]. The cost to the users is inconvenience due to sluggish Internet connection and slowing down of overall computer performance, as well as the cost of time spent in researching a solution[16].

4.1.3 Loss of time/money/confidence

If a user knows that her system cannot be trusted or is not convenient to use anymore then she has to spend the time in taking remedial measures. This includes buying new hardware/software or seeking professional help. Some estimates put 445 million pounds in the UK in lost time, productivity and in computer repairs[17]. Another outcome is feeling of vulnerability and loss of confidence, these have grave implications on e-Commerce and will be discussed with respect to costs for the enterprise.

4.2 Enterprise and Governmental Systems

All of the above also impact the corporate world, as most of the employees and customers of any corporation are also home users. The possible losses for corporate users span a wide range depending on the size, location, revenue, and nature of the

organization's type of business. Information corruption or theft from rootkits could threaten life, property, goodwill, and revenue, amongst a host of things.

For safety critical information systems, such as those that control nuclear power plants and air traffic control systems, failure to respond at an optimum level could endanger lives, a particularly desirable outcome for terrorist organizations. The results of failure in non-safety related systems include the possibility that data will be corrupted or stolen, or services will be unavailable. These are serious outcomes are not perceived to be as serious as those associated with safety critical systems. Financial consequences can also be grave. Some possibilities are quantified below and their impact on the bottom line calculated. Losses can be divided into direct costs, indirect costs, and failure costs.

4.2.1 Direct costs

Direct costs include purchasing, installing, and administering security measures that can be unambiguously associated with loss prevention. These include purchase of products such as firewalls and anti virus software, rootkit detection kits etc. Another direct cost is the maintenance of hot sites. Direct costs vary widely depending on the requirements of the corporation or governmental entity. Specialized groups, such as the U.S. military, usually have the most demanding requirements.

However, other institutions such as banking, nuclear facilities, and air traffic control systems also have stringent security requirements. Some sources put the federal information security spending at $6.1 billion this year and projected to grow by 20 percent over the next five years[18].

Studies also indicate that information security in the Air Traffic Control Systems is weak and vulnerable[19]. About 8 percent of all crashes have resulted from Other Human

Error, which includes Air Traffic Controller error[20]. An active series of attacks could push that number up significantly and/or paralyze the U.S. transportation infrastructure.

There is rising trend of cyber attacks on corporations.[21] Financial institutions understandably are the number one targets of these attacks. Profits from cyber crimes have surpassed those from drug trafficking, with a turnover of $105 billion in 2004.[22] Corporations are also increasing their security budget to counter an increased threat from cyber attacks[23]. A Deloitte survey in 2003 shows financial services companies are spending approximately 6% of their corporate IT budget on security[24].

Of those surveyed 47% percent hired extra security staff compared to 2001. According to another report companies spend from 5.5% (for large companies) to nearly 20% (for small companies) of their IT budget on security. [25] If we assume that the IT budget for a small company is a hundred thousand dollars a year, then twenty thousand dollars are spend on security. Similarly, if a very large company spent $100 million on IT a year, $5.5 million is being spent on security. A substantial portion of the security budget is being spent on web intrusion protection services, the market for which is expected to be $700 million in 2006[26].

As mentioned above, spam zombies are also an ever-present problem. One estimate placed the per-worker cost of spam at $1,934 in 2004[27]. This cost more than doubled from 2003 where the estimated cost per-worker was $874.

4.2.2 Indirect costs

These are less obvious costs associated with loss prevention. Additional security measures can affect system performance, employee morale, or retraining requirements.

One example is the implementation of software changes to improve security in order to mitigate the risk of attack. An indirect cost could be the time spent installing a new technology and working out any of the glitches. The introduction of security improvements increases system complexity. Changes to complex systems lead to the introduction of bugs and thereby increase the cost of system maintenance and troubleshooting.

Another element that adds to the indirect cost is the hassle factor. Employees often see security policies as a barrier to productivity[28]. Increased security controls will force corporate end users to take additional steps to log in and access information as well as follow elaborate security policies and procedures. Forcing users to employ multiple different monthly passwords for logging onto different systems reduces the productivity of most users. In the company one of the authors works for, Cell Therapeutics Inc., users complain a great deal about having to change passwords often, having to make them non-dictionary words and having to use weird combinations of upper and lower and other special characters. As this is a company policy, users follow it, albeit reluctantly.

Finally, another important indirect cost is loss of reputation or customer trust. A Federal Trade Commission consumer survey placed the number of Americans victimized by identity thieves at 10 million in 2003, with consumers losing $5 billion and businesses $48 billion[29]. The news about widespread security breaches has hurt consumer confidence[30]. There is a growing fear among consumers of victimization through identity theft. A survey of 1500 users suggests a quarter of the users have stopped buying online[31]. A Gartner research report also suggests fewer people are buying online because

of security concerns[32]. Cyber crimes in which confidentiality is violated cause a measurable negative impact on stock market value to the tune of 5%[33].

4.2.3 Failure Costs

Failure costs arise when an attack occurs with a consequent loss or impairment of service. These include loss of confidential information, theft of intellectual property, increase in spam, degraded performance, network congestion and instability, drains on productivity, increased help desk costs, and inability to ensure regulatory compliance. Complaints from employees about slow computers, program crashes, or slow network connections may be the first signs of infection. These problems can signal that programs are working in the background, connecting to remote servers to upload or download information. The results are a productivity drain for employees and a computing resource/bandwidth drain for the company. Some estimates put the boot times of a computer with spyware to be 4 to 8 times higher than a computer without spyware[34]. According to Microsoft's Jeffrey Friedberg, spyware causes half of the computer crashes reported by customers, at a cost of millions of dollars[35]. The same article states that spyware accounts for more than 12 percent of Dell's technical support calls.

Failure is of particular challenge to organizations trying to demonstrate compliance with government regulations for information security. These regulations include the HIPAA, established to ensure privacy of patient information, the Sarbanes-Oxley Act, established to ensure that financial statements are resistant to fraud, the Gramm-Leach-Bliley Act, established to safeguard customer information, and the California Data Privacy Law (California SB 1386), established to protect the confidential information of state residents. Simple non-compliance under HIPAA can bring fines up

to $50,000 [36]. Similarly compliance with Sarbanes-Oxley Act is very important to an organization because failure can bring senior management up to twenty years of jail time and $5 million in fines [37].

5. Preventative and Responsive measures

Although rootkits can be used for a number of malicious purposes, it is important to note that rootkits themselves pose no inherent threat. They are merely tools used to compromise a computer system, allowing hackers use of the system for further attacks. Rootkits are used to disguise programs running on a computer system, allowing the attacker to escape detection while using the system.

5.1 Technical Responses

Unfortunately, there is no easy way to combat rootkits. Rootkits are designed by nature to be stealthy and undetectable. Unlike viruses or previous breeds of spyware, rootkits alter the structure of the underlying operating system itself, rendering most current techniques for virus and spyware detection ineffective.

In a computer system, the operating system is the interface between the hardware and the software. An application running on a particular operating system is a client of that system, and, in order to operate correctly, relies on information provided by the underlying system. After the installation of a rootkit, the underlying operating system can no longer be trusted. Unfortunately, any anti-virus or spyware detection software running on a particular system is also a client of the operating system and is inherently dependent on it. Current virus scanning software examines each file on a computer

system, checking it for a match in a database of known virus signatures. If any part of a particular file matches a given signature, a virus has been found.

The antivirus software relies on the operating system to provide the list of files to be examined, assuming the operating system will provide a complete list of files. If a rootkit is present on the system, the list of files reported by the operating system is controlled by the rootkit. The rootkit can simply choose not to report the spyware/virus infected files and the scanning software will never know it has been deceived. The rootkit uses similar techniques to alter other points of information collection within the operating system, thereby rendering itself, as well as any programs under its protection, invisible to the system's users and administrators.

Current research into rootkit removal examines a number of different approaches. Common rootkit signatures are being added to common virus scanners. Anti-virus software companies continually update the list of virus and malware signatures that their products recognize. The McAfee website lists both the FU rootkit and Hacker Defender rootkit in their database of threats which their product can handle. Norton Antivirus, produced by Symantec, does not list either of these rootkits, and acknowledge the difficulty in detecting these products once they are installed. Microsoft's Malicious Software Removal Tool advertises that it can remove the Hacker Defender family.[38] This approach however, is vulnerable the rootkit's ability to hide desired files and running programs.

One of the most effective approaches to rootkit detection is the creation of new software designed to specifically detect the behavior of rootkit infected systems. This generalized solution relies on the ability to use different operating system facilities to

query files and programs. If two different methods for obtaining a list of files yield different results, the system is likely to be infected.

One way to obtain the two different file lists is by using the system's standard way to report information, and in addition, using a method that is less likely to be compromised. The non-compromised list can be constructed in a variety of ways – through known good backups, lower level and less commonly used API's which are less likely to be infected, or through the examination of resources from a known good operating system, which could be deployed from a different machine or a clean boot CD. F-Secure Blacklight and RootKitRevealer from SysInternals are programs which use this technique to detect rootkits. Although this approach is able to detect a wide variety of rootkits, even the low-level scan could be intercepted by a well-designed rootkit, necessitating an external viewpoint to accurately detect file and process-hiding.

VICE is a program designed by Jamie Butler, designer of the FU rootkit, which catalogues the API hooks which may be potentially used by a rootkit. It could not detect rootkits that use other techniques, such those that directly modify kernel data structures. Although effective, this method cannot guarantee 100% coverage. A rootkit may hook into API's that are not being monitored. Moreover, it is difficult to automatically distinguish legitimate from malicious API hooks.

Another way to keep track of changes potentially made by a rootkit is by using a "cross-time diff" comparison to catalog changes as they are being made to the system. To use this method effectively, the rootkit detection software must be installed in a known clean state. From this point forward, the detection software can compare the present state to the known clean state, and allow the user to merge desired system

changes into the known-clean state. If any suspect changes appear, the user can revert to the last known clean state. Tripwire and Strider Ghostbuster are two software implementations of this approach. The advantage of this method is that it can detect a wide variety of malware. The major disadvantage is that a system incurs many benign changes during its lifetime, changes that must be recognized by the user as not being malicious.[39]

Responding to a rootkit threat involves both the detection of the rootkit and its removal. Since rootkits alter the operating system itself, they are often difficult to remove without destabilizing a system. Moreover, it is very hard, if not impossible to know when and if a system is clean. The methods surveyed above are all detection only or preventative measures. In most cases, users are advised to perform a clean install once a system has been deemed compromised.

5.2 Policy Responses

5.2.1 Existing Policy

Although rootkits have been around since the mid 1990's, the average computer user has not been aware of their existence until just recently, and only due to the media attention generated by Sony's use of a rootkit to enforce copy protection of its music CDs. Although consumer awareness of spyware in general has been on the rise, rootkits still remain relatively unknown.

It is then not surprising that legislation pertaining to rootkits specifically does not exist. A survey of existing computer security legislation shows attempts to tackle spyware and the means by which spyware propagates. The lack of rootkit specific legislation can also be attributed to the use of rootkits as tools to propagate spyware or

other malware. Since the rootkit is used behind the scenes, the visible consequences are not the actions of the rootkits themselves but those of the programs they are protecting.

Although rootkits are not equivalent to spyware, it seems that they will be put to use mainly as tools to aid in the counter-detection and counter-removal of spyware, rendering spyware more resilient and stealthy. Although other uses for rootkits, such as deliberate attacks on high-value systems, can be envisioned, it seems that the bulk of the potential damage to society, both monetarily and socially, will be in the form of magnifying the threat of spyware. Several surveys place the number of computers infested with spyware above 80%. If the spyware in these systems would be hardened against detection and removal, significantly more effort and resources will have to allocate to dealing with the spyware threat.

While the legal distinction between adware, malware, and rootkits is not easily made, policy makers can agree on a set of principles governing computing best practices. At the core of these best practices is the idea that a user should have control over his or her computer. The user should be informed as to what software is on his or her system, what said software is doing, and be able to uninstall said software easily. Moreover, software running on a user's computer must obtain consent from the user before transmitting any personally identifiable data, such as browsing history and buying habits.[40]

Current legislation that addresses the issue of spyware is generally very new. The first state to pass anti-spyware legislation is Utah, in 2004. There is no current federal legislation addressing the issue of spyware, although a discussion on the subject and several proposals are being considered. Current spyware legislation recognizes that one

of the major hurdles in attempting to legislate spyware is the issue of definition and attempts to take the approach described above. If a piece of software is out of the control of the user, it is considered spyware. Although Utah's Spyware Control Act contains a two page definition of spyware, three main points emerge. A program is classified as spyware if:

1) monitors the computer's usage and sends information about the computer's usage to a third party,
2) without obtaining user consent for doing so, and
3) without providing the user with an easy and straightforward way to uninstall the program.[41]

Similar provisions exist in most spyware laws that have been passed. By the above definition, rootkits fall into the category of spyware. However, this definition only applies in the state of Utah. There are currently 29 states which have passed some form of spyware legislation, each with its own definition.[42] Policy makers should be worried about the growing patchwork of spyware legislation. As the problem of spyware becomes more and more pressing, more states are bound to pass similar laws. A vast number of different laws addressing the same issue may make it difficult for legitimate businesses to conform and easier for illegitimate businesses to escape prosecution.

In view off this, the Federal Trade Commission (FTC) held a workshop on spyware in 2004. However the conclusions reached by this committee were that no federal spyware legislation should be created. Instead, law enforcing agencies were recommended to bolster enforcement of the problem and legitimate companies were

encouraged to self regulate by steering clear of entering into agreements with disreputable partners.[43] However, a survey by the Center of Democracy and Technology (CDT) shows that illegitimate companies hide themselves through many layers of intermediaries, such that the provider of the malware can be far removed from the company that ultimately delivers it, most of the time unknowingly.[44]

5.2.2 Policy Recommendations

Although the FTC advocates industry self-regulation and the enforcement of current anti-spyware statues, in the past year only two cases have been brought to court on a federal level.[45] Moreover, spyware over the last year has become more aggressive, using increasingly sophisticated techniques to thwart detection and removal. Businesses are increasingly plagued by technical support calls involving spyware, both from internally and external customers. As mentioned above, in the past year, the number one issue Dell's technical support department has had to deal with was spyware.

The FTC's decision to call for more industry self-regulation and enforcement of existing statues has failed.[46] The FTC itself has recognized this and there are now several proposals in the congress which aim to create federal anti-spyware legislation.[47] Moreover, legislating against spyware based on programmatic behavioral criteria is an effective way to go after spyware. If a program fails certain basic requirements, such as informing its user of its intent and allowing the user to uninstall it, it is classified as spyware and action can be taken against the producer of the software.[48] Moreover, a series of federal statues will more effectively undermine the vast networks spyware creators use to avoid prosecution.

Although the new legislation will provide a more effective way to combat spyware, a major problem remaining is the vulnerability of the Windows operating system to the installation and propagation of software. Windows runs on over 90% of systems and over 80% of Windows systems are or have been infected with spyware at one time or another. Microsoft has been aware of this problem for several years. Indeed, the first anti-spyware solutions date from more than 4 years ago. However, Microsoft, who is in the best position to harden the Windows operating system, has done little. Only in the past year has Microsoft released Service Pack 2, which fixes a host of vulnerabilities. In addition, in the past few months Microsoft has released its own anti-spyware application.[49]

Moreover, anti-virus vendors have also been slow to provide anti-spyware solutions, only getting on the bandwagon due to the recent media attention brought to the issue. Although companies are finally taking steps to combat the situation, policy makers should consider a policy that would provide incentive for Microsoft and other vendors to come up with solutions for the spyware threat. A monetary incentive, perhaps in the form of tax breaks, could be offered to companies that actively battle the threat. Such a policy would decrease the response time of industry and provide us with a faster recovery from the current problem.

However, this would still not deal with the situation presented in the Sony case where a trusted corporate entity actively engaged in distributing what most people would describe as a rootkit. Almost all security vendors refused to classify the Sony sponsored program as a rootkit or malware until there was a massive public backlash concerning the situation. Therefore, if anti-virus vendors can be trusted to be technically proficient in

fending off rootkits, there remains the question about whether they can be trusted as members of a corporate fraternity that are either unwilling or afraid to turn against their brothers. This breach of trust on the part of the anti-virus/anti-spyware branch of industry once again proves that the FTC's call for industry self-regulation was not sufficient. Although Sony has significant resources and legal clout at its disposal, the FTC should not hesitate to enforce anti-spyware statutes, even when powerful players, such as Sony, commit violations.

The FTC should perhaps create or hire an independent organization as a consultant. This independent organization could monitor the industry and take action against any observed violations. It was the security experts form sysinternals.com, a web page dedicated to providing users with security related tools and utilities, who first discovered the Sony rootkit. Luckily, the problem was so widespread, that the mainstream media brought it out of the computer expert world. Smaller or stealthier violations could go unnoticed, and the lack of media pressure could result in a lack of retaliatory action. If these experts had the ear of the FTC, one could be more certain that violators would be prosecuted.

6. Cost analysis of proposed measures

The cost effectiveness of different solutions is highly dependent upon the perspective in question.

6.1 Technical Perspective

For computer security companies, defensive technical solutions are the only solutions that are feasible. In the context of this research, rootkits represent a jump in the

escalating competition between hackers and computer security companies. Rootkits may necessitate a shift in how computer security is implemented. Because rootkits could potentially make changes at a very low level of the operating system, it may be necessary for operating systems themselves to incorporate more checks that changes being made in their functionality are legitimate. Microsoft releases new operating systems approximately every three years, which means that this is the timescale on which to implement changes that might fundamentally change vulnerabilities to rootkit exploitation.

Current research and development funding in industry is restricted to just a few companies. Microsoft spends $7 billion on research and development, but gives no information on how much of this is on product development vs. long-term research, and within long-term research how much is spent on computer security or rootkit specific topics. As Bill Gates noted in a speech, because past research in industry has sometimes benefited competitors, companies are hesitant to invest in long-range research.[50]

Most basic research is conducted in Universities and funded by federal agencies like DARPA and NSF. In recent years funding for these basic research agencies has risen more slowly than in the past, and has declined in some areas. For fiscal year 2006, NSF is requesting $620 million for the Computer and Information Science and Engineering Directorate (CISED), a 1.1% increase over 2005.[51] One priority of the Information Technology Research, a subdivision of the CISED, for 2006 is "Trustworthy Computing". This subdivision is requesting $145 million, a 16% decline from the previous year.[52] Other federal agencies, such as the Department of Homeland Security also fund fundamental research in computer security.

The optimum amount of money necessary to spend in a research program or funding agency to develop concrete benefits to society will always be difficult to determine. Defending against rootkits requires a shift in how computer security programs work, but could probably be implemented in a fairly short amount of time on a fairly low budget, given the fact that several of the most effective programs currently in distribution are made by smaller computer security companies. This indicates that at least some of the technical advances necessary to implement rootkit security are highly accessible. F-secure, a computer security company based in Helsinki which distributes Blacklight, the most commercialized of the rootkit defense tools, had $12.5 million in revenues related to anti-virus and intrusion prevention in the one-year period before April 2005[53]. While comprehensive solutions will require much more work than the tools currently available, this may indicate that some progress can be made in this area quickly and inexpensively.

6.2 Policy Perspective

It is difficult to analyze the cost effectiveness of spyware legislation. The oldest legislation is less than two years old, and although many lawsuits have been brought under several of the statutes, most of these lawsuits are still in pending litigation

A series of cases involving several known adware/spyware creators have been brought to court. Claria, one of the companies sued, has been settling, winning, and losing cases based on different statues and in different states since 2001. Since some of the settlements are private and the costs of litigation are not clear, it is hard to determine how much money was spent/gained. As of 2004, all cases involving Claria had been settled.

WhenU, a company which creates software, that when installed, places its own ads on an underlying web site's ads, has been sued by several companies, including Weight Watchers and Wells Fargo. The Virginia court dismissed the charges, issuing a summary judgment in the favor of WhenU. Moreover, in 2004, WhenU sued the state of Utah, claiming that Utah's Spyware Control Act was invalid. A Utah state court ruled in favor of WhenU, stating that WhenU need not comply with the provisions of the act. WhenU's argument was based on the idea that WhenU could not differentiate between users in Utah and users outside of the state. [54]

No surveys have yet been done to determine how cost effective this legislation has been because there simply isn't enough data available. Unfortunately, since no legislative precedents exist, policy makers must simply make the best informed guess as to what policy will be potentially successful.

Another problem with spyware legislation is that it may take several years to pass, especially in the case of federal legislation. In the computer industry, several years is a long time. By the time the legislation has passed, the playing field could be completely different, and the legislation itself may be rendered ineffective.

Despite all this uncertainty, one thing is clear: the spyware problem must be addressed. The WhenU vs. Utah case shows once again that state legislation is not enough to address the problem. Fortunately, federal anti-spyware legislation will be introduced over the next few years. The new legislation will definitely lead to an increase in costs. Any legislation bears the inherent costs of writing, passing, and enforcing said legislation. The only way to determine if such legislation will be cost effective is to wait until it is passed and sufficient cases have been brought to trial.

Moreover, any industry incentives, such as providing tax breaks for operating systems vendors who combat spyware, will also have significant costs. It is also hard to say how cost effective these measures will be, although one advantage of such measures is that the money will be spent after certain conditions are satisfied, and thus the taxpayers see what they pay for before they actually commit the necessary funds.

7. Conclusion

Rootkits will be with us for a while. Since there is no simple way to legislate against rootkits, an effective legal solution to the problem cannot be expected anytime soon. Moreover, rootkits are a very effective a tool for hiding malicious software, and blackhat hackers will continue to use them as long as there is profit to be made. The only way they will be defeated is by the technology vendors making some fundamental shifts in the way they approach networked computing. Reduced permission levels, general bug fixes and safe zones within a computer's memory space to install suspicious software are just a few of the techniques that can be used.

But each new security technique has accompanying complexity, which means that there will be flaws in the security software itself that can be exploited. So while technology may be the only way out of the problem of rootkits, it will be a long difficult slog before a permanent solution is found.

8. Glossary

ActiveX – Microsoft technology that allows execution of programs within a web browser

Adware – program that actively delivers advertising content to a computer user

Bot-net – a collection of computers that have been compromised by one or more attackers and can be collectively controlled by remote; often used to carry out DDOS attacks

Distributed Denial of Service (DDOS) attack – an attack which employs multiple computers to generate superfluous network traffic to and from a target host, thereby preventing legitimate traffic from getting through and effectively taking the host offline

Device driver – a program that controls a device on a computer system. The disk drive, CD driver, mouse and keyboard are all examples of devices controlled by a device driver.

Digital Rights Management (DRM) – a system that attempts to protect copyrighted digital information traveling across a network

Hot site – a physical location where an organization may continue computer operations in the case of a major disruption.

Kernel – the part of an operating system that is typically responsible for memory, task, process and disk management; essentially the "brains" of a computer system.

Loadable Kernel Modules (LKM) – a technology for the Linux family of operating systems which allows device drivers to be loaded into the kernel on the fly, as opposed to being specified before the kernel is built

Malware – a piece of software that an end user would consider harmful to his or her system. Negative behaviors include damaging a computer or simply reducing its efficiency. Also commonly referred to as **adware** or **spyware**

Spyware – a piece of software which monitors the activity of a computer system and periodically reports it to a third party using a network connection; often used for illicit purposes.

Sniffer – a piece of software on a user's system that monitors network traffic and/or user input/output from the computer system.

Zombie - a computer attached to the Internet that has been compromised by an attacker and can be remotely controlled; used as a primary building block for bot-nets.

9. Endnotes

[1] http://www.cs.wright.edu/people/faculty/pmateti/Courses/499/Fortification/obrien.html

[2] http://www.phrack.org/phrack/63/p63-0x08_Raising_The_Bar_For_Windows_Rootkit_Detection.txt

[3] http://www.microsoft.com/technet/security/Bulletin/MS04-028.mspx

[4] http://www.eweek.com/article2/0,1895,1879157,00.asp

[5] http://www.theregister.co.uk/2005/11/04/secfocus_wow_bot/

[6] http://la-samhna.de/library/rootkits/index.html

[7] http://www.cert.org/advisories/CA-2001-05.html

[8] http://www.rootsecure.net/content/downloads/pdf/unix_rootkits_overview.pdf

[9] http://www.theregister.co.uk/2004/10/25/mac_rootkit_opener/

[10] http://www.prnewswire.com/cgi-bin/stories.pl?ACCT=104&STORY=/www/story/10-11-2005/0004165297&EDATE

[11] http://seattlepi.nwsource.com/local/216184_spyware16.html

[12] http://moneycentral.msn.com/content/Banking/creditcardsmarts/P87328.asp

[13] http://www.msnbc.msn.com/id/6814673/

[14] http://kyl.senate.gov/legis_center/subdocs/2yrsafter.pdf

[15] http://www.consumer-action.org/English/PressReleases/2005_10_27_PR.php

[16] http://www.usatoday.com/tech/news/computersecurity/2004-09-08-zombieuser_x.htm

[17] http://www.theregister.co.uk/2005/10/20/webroot_uk_spyware_guesstimates/

[18] http://www.esecurityplanet.com/trends/article.php/3507741

[19] http://www.gao.gov/archive/1998/ai98155.pdf

[20] http://www.airlinesafety.com/editorials/HumanErrorVsTerrorism

[21] http://msnbc.msn.com/id/7257289/

[22] http://msnbc.msn.com/id/10241467/

[23] http://www.computerworld.com/securitytopics/security/story/0,10801,81261,00.html

[24] http://www.deloitte.com/dtt/cda/doc/content/Global%20Security%20Survey%202003.pdf

[25] http://infosecuritymag.techtarget.com/2003/mar/cisosurvey.shtml

[26] http://plaza.ufl.edu/joserb/fourthpage.html

[27] http://www.eweek.com/article2/0,1895,1608661,00.asp

[28] http://www.csoonline.com/analyst/report3303.html

[29] http://www.usatoday.com/tech/news/computersecurity/2005-11-02-cybercrime-online-accounts_x.htm

[30] http://www.infoworld.com/article/05/07/01/27OPsecadvise_1.html

[31] http://msnbc.msn.com/id/9826022/from/RL.2/

[32] http://news.zdnet.co.uk/internet/security/0,39020375,39205460,00.htm

[33] http://www.optimizemag.com/article/showArticle.jhtml?articleId=18700435&pgno=4

[34] http://www.keytech.co.uk/Information%20Centre/Document%20Library/spyware%20whitepaper.pdf

[35] http://www.wired.com/news/technology/0,1282,63345,00.html

[36] http://dce.ucf.edu/hipaa/about.htm

[37] http://www.eweek.com/article2/0,1895,1864756,00.asp

[38] http://www.microsoft.com/security/malwareremove/families.mspx

[39] ftp://ftp.research.microsoft.com/pub/tr/TR-2005-25.pdf

[40] http://commerce.senate.gov/hearings/testimony.cfm?id=1496&wit_id=2092

[41] http://www.benedelman.org/spyware/utah-mar04/

[42] http://www.benedelman.org/spyware/legislation/

[43] http://www.ftc.gov/bcp/workshops/spyware/index.htm

[44] http://commerce.senate.gov/hearings/testimony.cfm?id=1496&wit_id=2218

[45] https://netfiles.uiuc.edu/ehowes/www/one-year-on.htm

[46] https://netfiles.uiuc.edu/ehowes/www/one-year-on.htm

[47] http://www.thexlab.com/faqs/usspywarelaw.html

[48] http://writ.news.findlaw.com/ramasastry/20040601.html

[49] http://www.microsoft.com/athome/security/spyware/default.mspx

[50] http://www.microsoft.com/billgates/speeches/2005/07-18FacultySummit.asp

[51] http://www.nsf.gov/about/budget/fy2006/pdf/4-ResearchandRelatedActivities/2-ComputerandInformationScienceandEngineering/17-FY2006.pdf

[52] http://www.ibid.com

[53] http://www.f-secure.com/investor-relations/news/items/news_2005042600.shtml

[54] http://www.benedelman.org/spyware/#suits