

White Paper:
Understanding Malware and Internet Browser
Security



May 2006
By Mike Belton, Security Engineer, CISSP

Contents

| | |
|--|----|
| Introduction..... | 3 |
| What is Malware?..... | 4 |
| Defining Malware..... | 5 |
| Authorized or Unauthorized..... | 6 |
| Consequences of Malware Infections..... | 6 |
| Does Your Network Have a Malware Problem?..... | 7 |
| How Does Malware Get Into Your Network?..... | 7 |
| Web Browser Software and Script Interpretation..... | 7 |
| Living with the Threat..... | 9 |
| What can be done to Defend Against Malware?..... | 10 |
| Education..... | 10 |
| Anti-Virus Solutions..... | 10 |
| Host Intrusion Prevention Systems..... | 11 |
| Information Security Policy: A Foundation for Defense..... | 12 |
| Conclusion..... | 13 |
| For More Information..... | 14 |

Introduction

The issue of unauthorized software in the enterprise environment is becoming a critical resource and productivity issue. The problem has risen to the legislative level. In 2004 the U.S. House of Representatives passed HR2929, the so-called Spyware Act, which is currently being reviewed by the Senate. Recently, the Senate Commerce, Science and Transportation Committee approved S.2145, a bill that would outlaw the practice of installing software that collects personally identifying information without the end-user's consent. Additionally, various state governments have enacted their own legislation to better define and regulate this unauthorized software, which is typically referred to as malware. Software companies have responded by pointing out that their product comes with an End User License Agreement (EULA) that explains what their software does and what the end user is agreeing to. These companies claim that by entering into this binding legal agreement with the software company, the end user has consented to any actions the malware might perform.

The traditional method for detecting malware as it enters a computing device is through file scanning. Scanning is performed on any file that the computing system interacts with, including files downloaded from the Internet, files sent via email, and files on removable media. However, many companies that produce scanning software have been sued by the software producers for mislabeling some third-party software as a particular type of threat. The software producers argue that their software has legitimate uses and is delivered with a license. Therefore, if anti-virus companies classify these products as 'spyware' or 'adware' or 'trojan,' they are destroying product image and limiting sales.

All of this sets the stage for a very dynamic security issue that involves multiple layers of defense and end user education. To truly attack and mitigate the issue of malware, one needs to understand the policy issues, business drivers, and computer literacy of any particular business unit. You must also create and enforce policies regarding the types of access and the types of permissions that are appropriate for each part of your company. As with most corporations, your network has a very diverse user-base with unique computing requirements, business functions, and knowledge levels. This makes the task of fighting malware much more difficult.

What is Malware?

Classic malware includes familiar software such as:

- Viruses
- Trojan horses
- Worms
- Dialers
- Rootkits

With the advent, and subsequent evolution of the web browser, new forms of malware have been created. This expanded definition of malware includes software such as:

- Adware
- Spyware
- Hijackers
- Toolbars

Table 1 provides definitions and more details on each category of malware.

TABLE 1 – MALWARE DEFINITIONS

| MALWARE CATEGORY | DEFINITIONS |
|------------------|---|
| VIRUS | A <i>virus</i> is a piece of software that self-replicates by injecting copies of itself into other executables or documents. |
| TROJAN HORSE | A <i>trojan horse</i> is a piece of software that appears to be a legitimate application but performs malicious actions while running. |
| WORM | A <i>worm</i> is software that is able to replicate itself across a network and infect other systems. Worms come in many shapes and sizes and can propagate very quickly across an internal network. |
| DIALER | A <i>dialer</i> is software that uses a modem connection to place unauthorized phone calls. Dialers have been around a very long time and have historically been used to generate revenue by placing phone calls that are billed back to the victim. |
| ROOTKIT | A <i>rootkit</i> is a piece of software that modifies a host operating system to conceal files, processes, memory usage and more. Rootkits are difficult to discover and remove. |
| KEYLOGGER | A <i>keylogger</i> is software that records any keystrokes entered by the end-user and stores that data locally, or transmits it across a network to a third-party. While it is true that keyloggers can perform useful functions in an enterprise environment, they are generally considered malicious. Additionally, keyloggers have been implemented in software and hardware. |
| ADWARE | <i>Adware</i> is software that presents advertisements to the end-user. Typically, this software is installed without the end-user's knowledge and is often difficult to remove. |
| SPYWARE | <i>Spyware</i> is software that gathers information about an end-user's computer use and either stores that data locally or transmits it across a public network to a third-party. Essentially, spyware creates a profile of end-user computing habits. |
| HIJACKER | A <i>hijacker</i> is software that takes control of the end-user's web browser. Hijackers are used for a variety of purposes. Historically, hijackers have been used to generate 'clicks' that translate to advertising revenue. |
| TOOLBAR | A <i>toolbar</i> is a piece of software that embeds itself in an end-user's web browser. Typically, toolbars provide enhanced functionality but can also be used for malicious purposes. Likewise, a toolbar can function as adware, spyware or a hijacker. |

Defining Malware

The full definition of malware is a bit of an anomaly; one person's malware is another person's mission-critical application, is another person's productivity tool. A classic example of this would be a piece of software named WeatherBug®.

The WeatherBug® software is typically installed by an end-user to deliver localized weather information. While this is a seemingly harmless task, the WeatherBug® software generates revenue for its creator by providing highly targeted advertising based on actions taken by the user. This software monitors the end-user, generates information related to what the user is doing at any given time and connects to public networks to make use of that information. This is risky behavior, since most

organizations and their employees do not want to transmit personal information over the Internet, nor do they want to be added to questionable advertising databases.

Authorized or Unauthorized

Given the landscape of malware, it is useful to consider that software is either authorized or unauthorized. Likewise, specific actions that software might perform can be classified as authorized or unauthorized. Finally, if this type of software has been knowingly installed by an end-user, the actions of the end-user can be classified as authorized or unauthorized. In all of these cases, a simple definition of what is authorized and what is not will enhance an organization's ability to quickly decide on the applicability and desirability of any particular software. This task is best accomplished via a set of comprehensive policies that define authorized activities and other expectations an organization has concerning proper use of a computer.

Consequences of Malware Infections

Malware presents a complex and costly issue for corporate networks. While precise numbers are difficult to acquire, Berbee's Security team has witnessed several different scenarios. In some of the more damaging instances of mass malware infections, organizations have had to power down entire networks in the middle of the workday to handle infections one machine at a time. This process involves at least one employee physically working with every infected computer. In other scenarios it has taken two or three dedicated personnel over one week (roughly 120 person hours) to completely eliminate the presence of a network worm. These scenarios typically depend on current staffing and network defenses, and can be translated into a rough calculation of what an outbreak and recovery might cost your organization based on your staffing level, wage information and financial implications.

Does Your Network Have a Malware Problem?

The answer to this question is, it depends on your perspective. Without some way of classifying what exactly your organization considers 'malware' to be, we can't possibly answer this question. Currently, many organizations do not have the necessary policy statements in place to answer this question. While it is not always immediately obvious, the power of a well-planned information security policy is invaluable. To better quantify the malware issue and ensure solid decision-making, some type of test, standardized across the organization, should be considered. Given that no two networks are exactly the same, the specifics of this test will be unique to your environment. That said, some immediate questions should be:

- How do we define malware?
- Based on our definition, are there active instances of malware currently infecting our network?
- What level of service disruption do we consider acceptable (minutes, hours, days, weeks?)
- Do we have operating procedures in place to effectively mitigate damages from an infection?
- Do we have appropriate defenses in place to defend against our definition of malware?

The answers to these questions will probably vary between the various divisions or business units that form your organization. If your organization has suffered a malware incident in the last 36 months it is prudent to thoroughly analyze that incident and define the costs associated with mitigation and recovery. This analysis will typically highlight deficiencies in defense technologies, response capabilities, and overall preparedness.

How Does Malware Get Into Your Network?

Malware is introduced into the corporate network through a variety of mechanisms. The primary delivery mechanism is via the Internet, specifically the World Wide Web, as infections can and do happen without the user's consent or knowledge simply by viewing a web page. The standard web browser deployed in most corporate networks is Microsoft's Internet Explorer. The Internet Explorer product has a long history of security problems. As of this writing, there have been more than 90 vulnerabilities affecting the Internet Explorer 6.x product since 2003. Currently, there are 19 vulnerabilities in the latest fully patched version of the Internet Explorer product that are remotely exploitable. The potential for damage ranges from unauthorized information disclosure to complete system compromise.

Given this knowledge, it is difficult to completely secure Internet Explorer, and certain types of malware activity cannot be stopped or managed. Completely negating this issue means switching your organization to another browser platform, which may be very hard to accomplish logistically. Even if your organization were to switch to another browser, there will still be particular business units within the organization that will require the use of Internet Explorer for certain tasks and, in the case of web software developers, for usability studies and testing. At this time, it is likely that there are a number of Internet Explorer versions running at different patch levels in your corporate network. In this case, patch management is the real issue. To negate this issue, it is important for any organization to standardize deployed versions of Internet Explorer to the latest fully patched release and implement a comprehensive patching policy and management system to ensure timely application of new patches.

Web Browser Software and Script Interpretation

Every major web browser is approaching a level of maturity found in most Operating Systems. Web browsers now contain mature language interpreters, also called runtime environments, for executing web-based software and scripts. These virtual environments allow for an amazing level of control over

the endpoint and, depending on the level of privilege granted to the end user, can be exploited to perform any number of tasks. The immediate danger here is two-fold. First, many organizations, for a variety of reasons, grant the end-user a high level of privilege – including administrative access of the operating system. Second, the web browser generally communicates over channels that are viewed as legitimate and the content of those transmissions can be encrypted with SSL, further reducing the ability to react and respond at the network level. Given these factors, attackers have been focusing on exploiting browser technologies to bypass many of the typical defense mechanisms. These new types of attacks create a complex situation that can be extremely difficult to defend against.

There are three primary languages in most web browsers and these languages approach security in a different way. This level of complexity is positive in that it creates a rich development environment for building web-based applications and services. The drawback however, is that malicious software, targeted at these runtime environments can be used to perform tasks such as disabling anti-virus software, or installing new pieces of malware such as keyloggers or backdoors. The companies that develop browser-based runtime environments have attempted to address the potential for mis-use by implementing security models in the language and interpreters. These languages include:

- Microsoft ActiveX®
- Java
- JavaScript

The first runtime environment we will examine is Microsoft's ActiveX technology. ActiveX programs are referred to as 'controls,' and the ActiveX interpreter does not limit the level of access granted to any ActiveX control. When an ActiveX control attempts to execute the user is presented with a dialog box asking if this is acceptable. At this point in the process security is under the control of the individual end user. This person might not be qualified to make decisions about software security. ActiveX controls allow for complete access to the underlying Operating System and are executed with the User Rights of the current user. While it is considered to be poor practice, many users in an organization have 'Administrator' privileges – this level of permission allows for full control and manipulation of the underlying operating system.

The second runtime environment is Java from Sun Microsystems. Programs written in Java are called 'applets.' Java employs a completely different security model that allows for any program to be downloaded and executed. The Java runtime environment limits the type of actions an applet can perform and in this way limits the destructiveness of any rogue application. The effectiveness of this security depends on the secure implementation of the runtime environment.

The third method of programming in web browsers is a scripting language called JavaScript. JavaScript was created by Netscape Communications to allow non-programmers access to enhanced web programming features. JavaScript is certainly not benign, however, its functionality and access to applications outside the web browser is rather limited. JavaScript is traditionally used to mount attacks against the browser itself.

While these three browser technologies are generally included with every enterprise-grade web browser, the potential threats do not end here. A large number of applications are being developed to integrate with network-based services. These applications are also vulnerable to attack and manipulation and include applications like:

- Macromedia Shockwave™
- Macromedia Flash™
- AOL Instant Messenger®
- Microsoft Instant Messenger
- RealPlayer™
- Windows Media Player
- Others

Living with the Threat

Each technology listed above has had at least one published vulnerability in the past year, and some of them have had more than one. Given that most of this software is generally permitted in corporate environments, the issue of defense becomes extremely complex.

While it is possible to create and maintain a list of trusted software that is allowed to execute inside runtime environments, it is a time-consuming task that requires vigilance and near daily maintenance to ensure a proper defense. Likewise, it is possible to limit the end-users' ability to install these types of applications. However, most of this software does produce positive business benefits. In many cases, other businesses (and possibly your business partners) are relying on these technologies to increase workflow, reduce costs, and enhance communications between business units, partners and customers. Therefore, these are becoming essential tools for conducting business in the 21st century.

What can be done to Defend Against Malware?

There is no silver bullet.

Defending against malware is a convoluted issue due to the fact that most organizations have not definitively defined it. Malware is 'bad software' - the meaning of 'bad' exists in a user's mind and in an organization's policy statements. Understanding the issue fully involves determining what types of malware the company is exposed to, analyzing the associated threats, and making risk management decisions. This decision will require a fair understanding of the unique issues at each physical site and then balance the goals of security and usability.

Education

Security is a process that requires vigilance from everyone involved. This doesn't mean everyone needs to understand the complexity of any particular security issue. It does mean that every associate should be able to recognize behavior that doesn't promote a secure environment. This is an incredible issue. If an individual asks a member of your organization for their password, the presumption is that the associate would recognize the threat and not disclose the information. On the other hand asking the same associate, 'what kinds of computers do you use at work?' might not be considered a security issue. Clearly user education is a primary issue in maintaining a secure computing environment.

Anti-Virus Solutions

Moving away from the user, most organizations utilize anti-virus solutions as the second line of defense. This software performs well at stopping known malware threats. There are, however, some issues associated with this defense. The first issue concerns a rogue ActiveX control and its ability to bypass or disable the anti-virus product. Legitimate software is never this aggressive due to the potential illegality of this action. It is highly unlikely that an associate or your organization would have a legitimate business-related reason to be viewing sites that break criminal computer laws to install a piece of software. The issue of disabling a scanner via ActiveX applies equally to anti-spyware scanners.

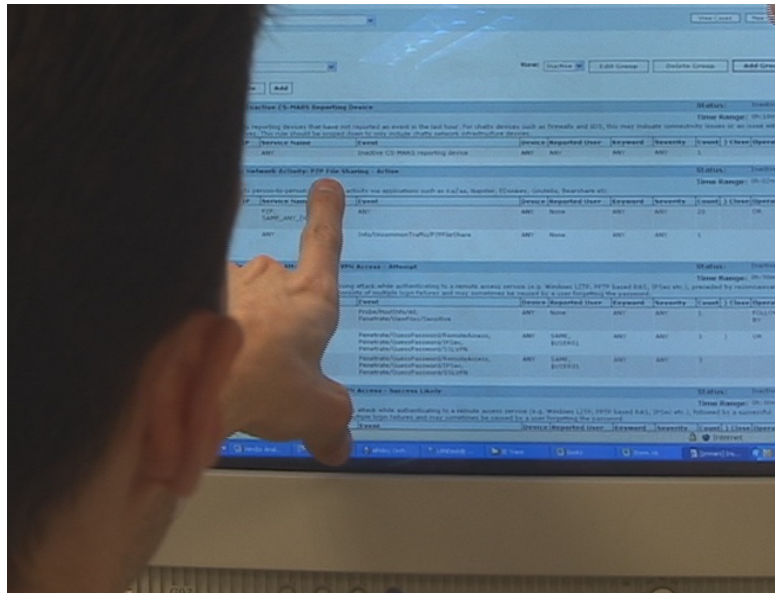


Figure 1 - Defending Against Malware

Currently most organizations have deployed an anti-virus solution that scans for malicious software as they are being passed to a computing device. This anti-virus software is typically updated regularly and provides a good baseline protection against undesired software installations. As mentioned previously, most malware exists somewhere between 'good' and 'bad' and is therefore not scanned for as aggressively. While an anti-virus solution is a proper software guardian, the unique bugs found in the most web browsers allow for a malicious web site operator to bypass, even disable, the virus scanner for a period. This allows them to install the malware without detection by the virus scanner. Any

scanning solution, from traditional AV products to so-called anti-spyware scanners, suffers from this issue.

Host Intrusion Prevention Systems

Along with signature-based anti-virus solutions, many organizations are deploying host intrusion prevention systems (HIPS). These systems can be combined with anti-virus solutions to create a strong defense against attacks that have not been identified and therefore are not recognized by signature-based solutions. By combining anti-virus solutions with host intrusion prevention and possibly a host-based firewall such as the one included with Windows XP, your organization will become highly resistant to a wide variety of attacks beyond malware. In practice this type of defense posture discourages all but the most determined adversary, and any attack against this type of defense will require a large amount of planning and consideration, most likely involving physical access to the internal network.

Along with these solutions, there are a variety of options available for hardening operating systems to negate the damage a rogue application can perform. A full analysis of operating system and subsystems security is outside the scope of this paper.

Information Security Policy: A Foundation for Defense

Attacking the issue of malware from a policy perspective involves understanding what features define malware to your organization. These features range from licensing issues to application features.

To continue with the previous example, a quick review of the WeatherBug® website shows that the developers do not in any way consider this product to be 'spyware' or 'adware' because it does not fit a strict definition of those terms. Since this software does not fit a particular definition of 'malware' popular anti-spyware and anti-virus programs no longer search for or classify it. This software does, however, monitor end-user behaviors, open windows and dialog boxes, connect to the public Internet through ports that are difficult to filter at the firewall level, and adversely impact resources such as network bandwidth. Does your organization consider this to be malware even though the developers and the person who installed it do not? Beyond that, does your information security policy govern these types of activities on your network? This is software that performs unattended tasks and utilizes network connectivity to contact unknown third parties. This type of connectivity simply increases risk while providing very little benefit or, arguably, no business-related benefit at all.

In the previous scenario, fighting malware is extremely difficult because there has been no policy statement defining this as positive or negative behavior. The company is, in effect, asking for a 'service' when the associate installs the software.

This type of software is delivered with some type of license that the end user consents to prior to installation. Unfortunately, it would be too difficult to analyze the various software licenses in existence and create any type of guidance across them. This moves the policy issue into the arena of application features. In this area the company can find more concrete answers. Does software that connects out through the company firewall to communicate with a third party qualify as malware? A yes answer begins to define the policy statement. Does software that automatically displays dialog boxes or 'pop-ups' qualify as spyware regardless of the developer's view? Again, a yes answer points towards the policy solution.

These types of considerations should be examined to determine how invasive the issue of malware is to your organization. If the issue is spot infections at certain areas and is being caused by software installed by an employee, a policy statement defining what is acceptable will solve the problem faster than any particular technical solution.

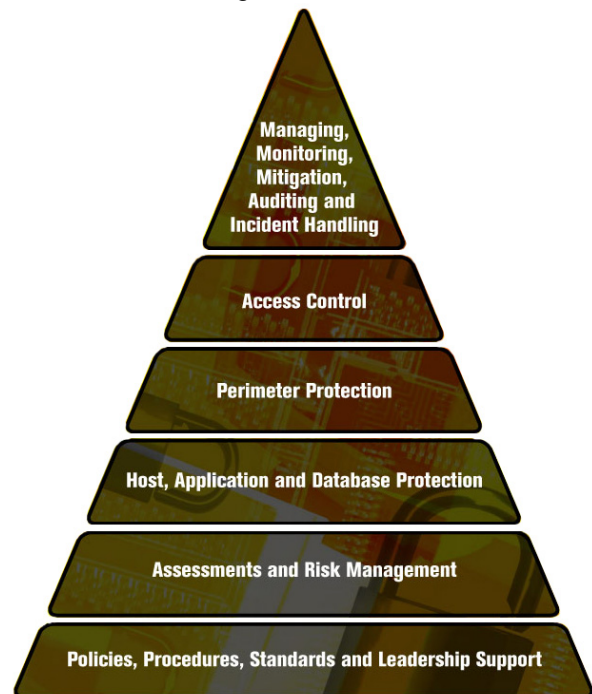


Figure 2 - Security Policy is the Foundation for Defense

Conclusion

New forms of malware are under continual development and the issues shows no signs of slowing down. The ultimate outlook of this is not dire. Properly defending against this rather complex issue involves commitment from your organizations' executives, comprehensive definition of the issues involved and education. There are a vast number of policy and technical security tools available. Defining the toolset that is most appropriate for your organization will provide the best possible defense, balanced against usability and fiscal responsibility. For many organizations, this type of issue will expose other potential lapses in defense, possibly due to a misallocation of existing resources or an implementation that has not been optimized for performance and security. Ultimately, most organizations will be confronted with this issue in the lifetime of their network. Properly preparing for that time will greatly ease the overall level of discomfort experienced during that period.

For More Information

For more information on Berbee's approach to security, visit
<http://www.berbee.com/public/solutions/security.aspx>