Nathan Olla

Survey of Vulnerability of GNOME/Mozilla Desktop Environment to Parasites

GSEC Practical Assignment

Version 1.4c

Abstract:

A high level evaluation of the current resilience to introduction of software parasites in the GNOME/Mozilla desktop environment is presented in comparison to Windows XP/Internet Explorer.   An overview of how various software parasites makes it onto Windows XP systems is given and the GNOME environment is evaluated against the same threats. A classification system for infection vectors is mentioned. The purpose is to gain some idea of veracity behind the ideas of a non-Windows desktop being less vulnerable to this type of threat.

INTRO

Definition

Software parasites are a recent addition to the list of software hazards that internet users must be wary of.   Software parasites are known by several names, according to a taxonomy that takes into account either what the software's purpose is (e.g. spyware, adware) or what the software actually does (e.g. dialer, downloader, keylogger).   Currently, "spyware" is used often colloquially to refer to all of these types of software, but some sources narrow the definition somewhat, such as the definition of spyware given by Wikipedia

> ...computer software that gathers information about a computer user and then transmits this information to an external entity without the knowledge or informed consent of the user.

Throughout this research, it was discovered that a large body of software exists that is lumped together as "spyware", but is different in functionality. Substantially, it is not concerned with gathering data that is useful for advertisements. Therefore, for this work the terminology preferred is the more accurate nomenclature given by Andrew Clover of doxdesk.com, and will therefore refer to the class of software as 'parasites', following his lead. Parasite software in its many forms has become a serious issue, and one not well or rigorously studied until this point. As this paper is primarily concerned with the study of spyware as it presents a security threat, only forms of spyware that install executable code will be considered. The basis for this distinction is to limit the scope of the discussion to the most threatening forms of spyware – those that pose the most grave security threat.

Why Adware, Spyware and other Parasites Exist

The reason software parasites exist is an interesting topic in and of itself. Parasites exists because the information gathered is valuable enough to someone to expend the time, energy and money to craft the code, to host the web pages that exploit the security holes and in doing so, accept some level of risk for their actions. Spyware, software that gathers information and sends it to a third party without the explicit knowledgeable consent of the user has clear commercial value to the company that collects the data. In the case of the more commercial forms of spyware or adware, such as Claris/Gator/GAIN, the connection between the value of the information and the software is direct and clear. In a recent segment of the public radio program Marketplace, discussing the economic value of "social topology studies" futurist Andrew Zolli noted the value of being able to track interactions in social groups in real time – "Social Network Cartography" [marketplace citation]. While the mechanism that would be used to track and gather this information was never discussed, the idea of tracking activities on blog viewership, peer-to-peer file transfers and the like in real time accurately describes the functions of some spyware applications. Other types of spyware have been noted to gather different types of information for purposes other than commercial advertising. In "Spiders, Spam, and Spyware: New Media and the Market for Political Information", authors Howard and Milstein describe the market for political information and two companies use of spyware, spam and related techniques to gather this information.

A Security Threat

There is mounting evidence to support the notion that even simple spyware and adware represent a gathering storm of security threats. In the study "Measurement and Analysis of Spyware in a University Environment", Saroiu et al. concluded after finding security issues in two wide spread spyware programs, on of which they never received a resolution from, that "the potential for spyware to cause substantial security problems is real." (pg. 11). In my own experiments, and on internet forums [moz] for example, it has been noted that parasitic software often piggybacks on each other, such as in the case of the malicious XPI "Content Access Plugin 1.01", which installed an XPI that installed Internet Explorer targeted malware from various websites. The secretive nature of this software presents a security issue in and of itself, but the fact that it installs unknown code from untrusted sources with functionality is catastrophic from a security standpoint.

Importance of Evaluating Alternative Platforms

To date, it is true nearly all parasites primarily target users of the Microsoft Windows family of operating systems, using the Internet Explorer web browser that is included with that system. It is safe to assume that this is due to the overwhelming market dominance enjoyed by the Windows platform. Other platforms and other browsers have not been entirely unscathed by adware or spyware in the past, but the browser based spyware seems limited to non-existant for those platforms. For example, Apple Macintosh platform has issues with adware included in the 3rd party applications, such as the Limewire P2P application[http://www.sfgate.com/cgi-bin/article.cgi?file=/news/archive/2002/01/04/financial1758EST0370.DTL] in the past, but there seems to be little evidence of browser-infecting spyware for the Mac at this time. Alternate browsers have been targeted as well, such as the previously noted malicious XPI for Mozilla based browsers.

Emergence of Linux and Mozilla Based Browsers

In the past few years, there has be ever growing support for Linux and as of 2004, every major PC manufacturer now has some level of support for the Linux operating system. A few vendors such as HP, IBM and Dell offer Linux on laptops, desktops or workstations. Sun Microsystems announced an entire desktop to server infrastructure aimed at business and government customers called the “Java Desktop System” which is based on the SuSE Linux distribution, and have announced a few deployments. Governments in Europe and Asia have announced wholesale Windows to Linux migrations, and the trend seems unlikely to diminish in the near future. The reasons for this migration and the emergence of Linux as a commercially viable, widely deployed Operating System are far beyond the scope of this paper to discuss. However, it can be stated, based on popular impressions, that there exists the perception that the Linux environment is more resilient to these sorts of security issues, as an open source operating system. Sun Microsystems even includes this as a marketing bullet in their literature on the Sun Java Desktop System, stating that it “..is less susceptible to virus and worm attacks due in part to a superior security architecture”[Sun]. In this light, as more and more adoptions along the lines of the City of Munich’s recent program occur, it stands to reason that Linux will become a target of more software parasites, and that resilience will likely be tested.

Although Linux based operating systems offer an abundance of choices in user interface environments – the choice of KDE vs. GNOME vs. other – this does not present much of an issue. At the current moment, the large commercial backers of Linux have converged on the GNOME system as the default desktop, and the Mozilla based browser appears poised to be the default under GNOME, and likely to be used under KDE as well. Sun and Novell have announced that this will be the supported desktop environment, under the Java Desktop and SuSE, respectively. In this light, focus on GNOME and Mozilla-based browsers and any architectural resilience or susceptibility to the techniques used by parasites is most appropriate.

Spyware Classifications

Spyware has remained elusive to define partially because it has come to mean a large class of software that is annoying or invasive from the users perspective. This has lead to efforts to classify spyware formally and an whole informal nomenclature. For purposes of evaluating how susceptible a particular platform, it is useful to break parasites into two very broad categories, based on whether or not the software contains only data or code and data.

- **Active Parasites**
  - **Install executable or interpreted code**
    - **Ex. Anything that installs a BHO, dialers**

- **Passive Parasites**
  - **Do not contain install executable or interpreted code**
    - **Ex. Tracking cookies, some registry entries**

In the course of the research, the focus on which specific types of spyware were interesting narrowed considerably. Parasites have been ably categorized by other research efforts, but in general these consider first and foremost the functions or purpose of the parasite, such as adware, browser hijackers, downloader Trojans, tracking cookies and so on. For the purposes of this paper, ware be usefully put into the categories based not on the functionality, but on the infection vector.

Another Classification Scheme

There are many developments occurring in software parasites evolution. On one hand you have infamous companies like Claria (ex-Gator) who are attempting the Herculean task of moving their business from something that consumers endure to something consumers see at best as an annoyance. On the other hand, there is the “drive-by-download” phenomenon, in which methods ranging from the merely unethical to the likely illegal are used to insert the assorted parasites and Trojans are installed. There exists a middle ground of sorts, in which the user is cajoled or deceived into authorizing the installation of the parasitic software through misleading window titles, graphics and occasionally, through a sheer panicked response to dozens of pop up windows, error dialogs and the like appearing rushing at the user much faster than they can respond to them.

As an example of the former category, spy- and ad-ware that is striving for consumer acceptance are the more successful P2P applications. For example., the Kazaa Corporation and several other companies have come under fire for bundling spyware with little to no acknowledgement of that to the user – at best the privacy agreement was obtuse. Newer versions however, along similar lines to the Claria Corporation, are explicit in the announcement of the additional adware, and they offer a non-encumbered version of the software for a fee. This mainstreaming of adware will probably continue into the future. As these businesses seek legitimacy in the marketplace, they accept potential liability for security and privacy concerns where failure to uphold legal requirements could cost them even more bad publicity, lawsuits and other troubles. Therefore, it is reasonable that these companies, the ones that expect to survive, will begin to behave more like traditional software vendors, while simultaneously trying to find an acceptable balance between effective targeted advertising and end user acceptance.

This sort of software, if the businesses achieve the legitimacy that they desire, and given their business performance there is little doubt that they will, will certinaly find it's way onto other platforms should those platforms gain strength In the markets. Therefore, Linux and it's host of browsers will experience this sort of spyware should it achieve a sufficient consumer install base as to provide monetary motivation for the targeted advertising firms to produce linux/mozilla versions of the software. Although unpleasant and irritating to many users, there is an undeniable transactional element to the software that falls into this category, in that it is bundled with commercial software that the user specifically installed for no initial purchase cost.

The second class of spyware that is found on the internet is of an undeniably darker nature – the lines a considerably less blurry whether this software constitutes malware or not, and it lacks the transactional element that to some degree legitimizes some forms of spyware and adware. These programs, while they attempt to posture themselves as offering a quid pro quo, in fact offer very little to no value and mistreat the users system in assorted ways. These are the sorts of software parasites that the user agrees to install from a website, such as "adult content dialers" or gambling applications which claim to give the user access to desired content at extremely inflated prices.

This sort of software is impossible to defend against for any platform in consideration, whether it is Windows or a UNIX-like operating system, because the ability to take away the users ability to install software that they willing wish to install has a large negative impact on usability. The default Linux security position of running as an ordinary user has some small effect on the probability of widespread creation of this software, in part due to difficulties in install system level software for an ordinary user. It is entirely possible for authors to target specific linux system, such as Fedora or Redhat however, which come with specific software to make installation of 3rd party programs easier for the end user.

The final class of parasite based on invection method is without a doubt purely malware – it does not pretend to offer any useful function what so ever.
The stub downloader and other trojans, combined with websites that actively employ exploits to insert themselves into a victim's system are clearly in same league as viruses, worms and traditional trojans.   By using vulnerabilities in the operating system or browser application, these trojans actively exploit the system in a fashion little different than a human attacker would use.

This sort of parasite's impact on linux and mozilla is little different than Windows. First and foremost, with web integrated desktop environments the norm, it seems extremely likely that security concerns in the browser or it's libraries or in plugins could lead to similar exploits being used against non-windows systems. One tiny advantage that linux systems have is that the tendency to run as "administrator" as in windows is not present, and therefore likely to limit in scope some of the software damage. However, the fact that the there are sites and the people who run them willing to engage in exploits initiated installation of arbitrary software suggests that they would be willing to use any local exploit to elevate privileges however necessary.

A Fundamental Issue

No browser/platform can be completely free of this unless there exist zero exploitable bugs in any part of any linked library or any external protocol handler, and that all parts of the system use good input validation to assure that no protocol handler can be exploited to grant otherwise denied privileges to external data.

Again, no browser architecture that allows third party libraries to be installed can be free of this completely, and depending on the options to the operating system, no library that uses shared libraries can necessarily be free of this. By using techniques such as LD_PRELOAD (GNU Libc) or DLL injection techniques, if the code can be inserted onto the system and the environment even temporarily modified, arbitrary code embedding seems exteremely diffeiclut to diffuse in any DAC system. Indeed, Trojans are the fundamental architectural issue with DAC systems according to [NCSB Paper], and indeed the behaviour of, for example, Troj/Psyme-AS and Download.Ject caused processes to run in the context of the user running internet explorer, causing download and installment of arbitrary code, which could then be called in a variety of ways.

It's all about improper input validation in the end. One method modern software in a DAC system uses to mitigate the Trojan horse effect is to attempt to limit the ways in which software or data can come into the systems such as by filtering proxies, inline antivirus scanners, restricting the logic flow in a some what controlled fashion, such as the Java VM's environment, but ultimately there will be bugs in those systems as well, and have been in the past. There is perhaps no reasonable, perfect solution to this issue in the immediate future, short of migrating to a MAC system such as Red Hat intends to take their linux distribution to. Even then, a bug in the security policy could allow the intended access control to be completely bypassed.

The ongoing insecurity of the web browser as a class of applications seems unlikely to abate in the near future. The susceptibility of one platform, Microsoft Internet Explorer on Windows, compared to the theoretical security advantages of running a different platform seem to be temporary advantages -- advantages brought by market realities rather than overall architectures. Thus, it is true that running an alternate browser on an alternate operating system currently grants an advantage in security, it is extremely difficult to characterize this as the result of a superior, more secure architecture. Much of the Internet Explorer/Windows security issue can be addressed by following some of the several security blueprints for Windows, applying smart GPO's and by running as an unprivileged user, which can mitigate that advantage to some extent. I conclude that as a class of Applications in general, the web browser has an extremely difficult task of getting untrustable data from an unknown external source and properly balancing it's behavior in terms of both usability and security.

Attempts to Mitigate

There have been attempts and will continue to be to mitigate the vulnerability of the browser and associated desktop environment to the three classes of parasite. These include access control systems and attempts to provide a form of limited 'jail' for plugins. Anecdotally, people seem unaware of the Zone-like option available in Mozilla-based browsers, called CAPS. CAPS refers to capabilities, and is mechanism by which access to specific methods can be controlled for certain URLs, hosts, networks and so on. CAPS provide a superset of what control IE zones grant, giving a more fine grained permission system, down to the method level. In general, there is no interface currently for arbitrarily setting policies however. Some discreet security setting interfaces exist for images, scripts and software installations, and more have been added in various Firefox and Mozilla releases. For example, in the preferences, an option to control which non-Mozilla sites can install software as added in Firefox 1.0RC)[firefox release notes/press release]. CAPS and Zones can be one element used to attempt to minimize the infection vectors by preventing certain websites from using client side scripts, loading poisoned images and the like. In order to limit the damage caused by malicioius scripts or malformed content some other mechanisms are used. IE, for example, now includes the "Kill Bit", a registry setting for each ActiveX object, which can be used to cause IE to completely ignore the object when called by external sources – this is basically a no-execute bit in *nix parlance. Other mechanims combating plugin-wrought violence, recent Firefox builds have introduced the concept of the whitelisted distribution site. Thus, in the default installation, sites other than update.mozilla.com will cause alarms, and force the user to manually download the software rather than launch an installation.

For Windows XP Service Pack 2, Microsoft has release a large number of security features for Internet Explorer that should dramatically reduce the impact of many of these vulnerabilities. Discussion of each of the features is beyond the scope of this study, but a detailed discussion is available at http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.mspx Some of the features include an Addin-manager (to control BHO's for example), a lock down of the Local Machine Zone, and improvements and bugfixes for issues such as cross domain access to cache items and more thorough enforcement of ActiveX security settings for remote content.

A Few Words on Windows vs. Linux

From experiments on a VMWare installed image, one of the most fundamental issues between the two operating system is the system level access that a user is granted. On a default Windows XP Professional or Home installation, the primary user has administrative access. When an exploit occurs, such as in the third class of parasite, the damage is somewhat limited by the capabilities of the user who has been exploited. In both environments however, once the ability to run arbitrary code has been established, any local privilege escalation attack is fair game and can be used to take control of the box.

The Future

The approach taken by Microsoft for XP SP2 of restricting the Local Machine Zone with policy meant to limit the damage does not currently have anything analogous in the Mozilla/Linux environment. In addition, many of the fixes that are being applied via SP2 in terms of the user interface also are without analogous counterpart. In some part, alternate platforms and alternate browsers have an opportunity to exploit the bad press and unfortunate security issue that have plagued Internet Explorer and Windows in the past year. As it stands now, many of the issues seem inadequately addressed by current Mozilla browsers and so it seems likely that history will repeat itself unless security is made into a higher priority for Mozilla browsers.

References and Sources

Source #1 (internet)
SANS Handler's Diary, July 23rd 2004
Handler on Duty - Tom Liston of http://www.labreatechnologies.com
http://isc.sans.org/diary.php
http://isc.sans.org/diary.php?date=2004-07-23&isc=00ee9070d060393ec1a20ebfef2b48b7
Retrieved Friday, September 17, 2004

Source #2 (internet)
Anatomy of a Drive-by Download
https://netfiles.uiuc.edu/ehowes/www/dbd-anatomy.pdf

Source #3 (internet)
Measurement and Analysis of Spyware in a University Environment Stefan Saroiu, Steven D. Gribble, and Henry M. Levy Department of Computer Science & Engineering University of Washington
{tzoompy,gribble,levy}@cs.washington.edu
http://www.cs.washington.edu/homes/gribble/papers/spyware.pdf
Retrieved Friday, September 10, 2004

Source #4
Marketplace, September 13, 2004. "Internet trends" Interview with Andrew Zolli and David Brown (host). Entire program available at
http://marketplace.publicradio.org/shows/2004/09/13_mpp.html
Audio Segment at
http://www.marketplace.org/play/audio.php?media=/2004/09/13_mpp&start=00:00:21:57.0&end=00:00:26:25.0
Retrieved Friday, September 17, 2004

Source #5
Wikipedia
http://en.wikipedia.org/wiki/Spyware
Retrieved September 17, 2004
http://en.wikipedia.org/w/wiki.phtml?title=Spyware&oldid=5944539

Source #6
Sun Microsystems, Sun Java Desktop System
http://wwws.sun.com/software/javadesktopsystem/

Source #7
Spiders, Spam, and Spyware: New Media and the Market for Political Information 1 [in Mia Consalvo, Ed., Internet Studies 1.0 (Peter Lang: 2003)]
Philip N. Howard Tema J. Milstein Department of Communication University of Washington

Release Notes - Firefox Preview Release ("Greenlane")
http://www.mozilla.org/products/firefox/
retrieved Sep 20, 2004

Microsoft Security Bulletin MS04-024
Vulnerability in Windows Shell Could Allow Remote Code Execution (839645)
http://www.microsoft.com/technet/security/bulletin/ms04-024.mspx
retrieved Sept. 20, 2004

Internet Explorer Update to Disable ADODB.Stream ActiveX Control
Original release date: July 2, 2004
Last revised: --
Source: US-CERT
http://www.us-cert.gov/cas/techalerts/TA04-184A.html
http://www.us-cert.gov/cas/techalerts/TA04-184A.html

How to Stop an ActiveX Control from Running in Internet Explorer
http://support.microsoft.com/default.aspx?kbid=240797

Multiple vulnerabilities in Mozilla products
Original release date: September 17, 2004
http://www.us-cert.gov/cas/techalerts/TA04-261A.html

Source #8
The Anatomy of a "Drive-by-Download" by Eric L. Howes

MozillaZine Forums
“Site(s) try t install malicious XPI when I visit them” thread,
http://forums.mozillazine.org/viewtopic.php?t=64341&postdays=0&postorder=asc&postsperpage=15&start=0
retrieved Sept 19, 2004

Source #9
Computer Genomics: Towards Self- Change and Configuration Management Yi-Min Wang Microsoft Research, Redmond, Washington, USA

Source #10

Spyware Analysis, presentation by Jan Monsch
Compass Secuirty Network Computing AG
http://www.csnc.ch

CAPS
Mozilla Capabilities Systems
[http://www.mozilla.org/projects/security/components/ConfigPolicy.html]

Source #11
“Build a Managed BHO and Plug into the Browser”
by Michele Leroux Bustamante
http://www.15seconds.com/issue/040331.htm
retrieved September 17,2004

“A GUIDE TO UNDERSTANDING DISCRENTIONARY ACCESS CONTROL IN TRUSTED SYSTEMS”
30 September 1987
http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-003.html