International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

A Study Of Attack And Defense Strategies

Japneet Kaur[1]
Manpreet kaur [2]
Abhishek Sinha[3]
Ms. Rimanpal Kaur[4]

Abstract

The initial design of internet and web protocols [1] assumed an environment where servers, clients, and routers cooperate and follow standard protocols except for unintentional errors. However, as the amount sensitivity of usage increased, concerns about security, fraud and attacks became important. In particular, since currently internet access is widely available, it is very easy for attackers to obtain many client (and even host) connections and addresses, and use them to launch different attacks, both on the networking itself and on other hosts and clients. Today's attackers are more likely to host their malicious files on the web. They may even update those files constantly using automated tools. When you are surfing the Internet, it is easy to visit sites you think are safe but are not. These sites can introduce malware when you click the site itself, when you download a file from the site manually and install it, or worse, when you are conned into believing the site you are visiting is a real site, but in fact is nothing more than a fake used to garner your personal information. From a network security [2] perspective, a browser is essentially a somewhat controlled hole in your organization’s firewall that leads to the heart of what it is you are trying to protect. While browser designers do try to limit what attackers can do from within a browser, much of the security relies far too heavily on the browser user, who often has other interests besides security. There are limits to what a browser developer can compensate for, and browser users will not always accept the constraints of security that a browser establishes.

1 M.Tech-CSE, CGC Technical Campus, Jhanjeri, Mohal
2 M.Tech-CSE, CGC Technical Campus, Jhanjeri, Mohali
3 M.Tech-CSE, CGC Technical Campus, Jhanjeri, Mohali
4 Asistant Professor, CGC Technical Campus, Jhanjeri, Mohali

CHANDIGARH GROUP OF COLLEGES
Campus : Jhanjeri, Mohali

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Introduction

Open Browser Engineering Issues

Other than the general design of HTTP, HTML, and related mechanisms discussed previously, a handful of browser engineering decisions tend to contribute to a disproportional of day-to-day security woes [1]. Understanding these properties is sometimes important for properly assessing the likelihood and maximum impact of security breaches, and hence determining the safety of user data. Some of the

pivotal, open-ended issues include: Relatively unsafe core programming languages:

• C++ is used for a majority of code in Internet Explorer, Firefox, Safari, Opera, and Chrome; C is used in certain high performance or low-level areas, such as image manipulation libraries. The choice of C and C++ means that browsers are regularly plagued by memory management [2] and integer overflow problems, despite considerable ongoing audit efforts. No security compartmentalization:

• Once control of the process is seized due to common implementation flaws, most browsers provide essentially unconstrained access to the user context they are running in. This means that browser bugs - historically, very common - easily lead to total system integrity loss.

Inconsistent and haphazard data storage practices:

• Browsers use a mix of random storage methods to keep temporary files, downloads, configuration data, and sensitive records such as passwords, browsing history, saved cookies, or cache entries. These methods include system registry, database container files, drop-off directories, text-based configs [5] (CSV, INI, tab-delimited, XML), and proprietary binary files. The data may be stored in user home directories, system-wide temporary directories, or global program installation folders. Controlling the permissions on all these resources and manipulating them securely is relatively difficult, contributing to many problems, particularly in multi-user systems, or when multiple browsers are used by the same user.

Web technologies are used in browser chrome:

• JavaScript, HTML, and XML are all used to a varying degree to implement some browser internals and various diagnostic and error pages in most browsers [4]. This choice contributes to an elevated risk of HTML injection flaws that permit web

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

content to gain elevated chrome privileges, which - depending on the browser - may carry the permission to read or write files, access arbitrary sites on the Internet, or alter browser settings. The problem is particularly pronounced for Firefox, which implements much of its user interface in this manner.

Inconsistent and overly complex security UIs:

- a vast majority of browsers employ highly inconsistent UI elements and security messaging, including several styles of modal prompts, interstitials, icons, color codes, and messages that pop up either on the bottom or on the top of the document window. Usability studies consistently show that at least some of these features are easily misidentified, misunderstood, or trivial to spoof (this is particularly the case for interstitials and notification bars that are not anchored in browser UI). Although a gradual improvement may be observed in certain aspects, further coordinated work in this area seems to be necessary.

Phishing Techniques

Link manipulation:

- Most methods of phishing use some form of technical deception designed to make a link in an e-mail (and the spoofed website it leads to) appear to belong to the spoofed organization. Misspelled URLs or the use of sub domains are common tricks used by phishers [1]. In the following example URL,http://www.yourbank.example.com/, it appears as though the URL will take you to the example section of the your bank website; actually this URL points to the "your bank" (i.e. phishing) section of the example website. Another common trick is to make the anchor text for a link appear to be valid, when the link actually goes to the phishes' site. The following example link,http://en.wikipedia.org/wiki/Genuine, appears to take you to an article entitled "Genuine"; clicking on it will in fact take you to the article entitled "Deception". In the lower left hand corner of most browsers you can preview and verify where the link is going to take you.

Filter evasion:

- Phone phishing: Not all phishing attacks require a fake website. Messages that claimed to be from a bank told users to dial a phone number regarding problems with their bank accounts. Once the phone number (owned by the phisher, and provided by a Voice over IP service) was dialed, prompts told users to enter their account numbers and PIN. Vishing (voice phishing) sometimes uses fake caller-ID data to give the

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

appearance that calls come from a trusted organization.•Phishers have used images instead of text to make it harder for antiphishing filters to detect text commonly used in phishing e-mails.

Website forgery:

• Once a victim visits the phishing website the deception is not over. Some phishing scams use JavaScript commands in order to alter the address bar. This is done either by placing a picture of a legitimate URL over the address bar, or by closing the original address bar and opening a new one with the legitimate URL.

Cracking the Information:Phishes, pretending to be legitimate companies, may use email to request personal information and direct recipients to respond through malicious web sites

•Phishers tend to use emotional language using scare tactics or urgent requests to entice recipients to respond

• The phish sites can look remarkably like legitimate sites because they tend to use the copyrighted images from legitimate sites

• Requests for confidential information via email or Instant Message tend to not be legitimate

• Fraudulent messages are often not personalized and may share similar properties like details in the header and footer.

Designing the browser

Proactive and reactive developers can generate an endless series of software updates. As a responsible defender, your dilemma is that allowing these updates in to your users without testing may break applications or even introduce security holes, but not allowing them may leave your enterprise open to even more serious attacks. Distributed management provides some help in this area, but all major browsers are weaker than many defenders would like them to be. Microsoft provides the free Internet Explorer Administration Kit[5], which sets the bar for enterprise browser deployment and management tools, but that bar is lower than many would care for Firefox ADM, an open source project for managing collections of Firefox browsers, is far more limited but a step in the right direction. Front Motion provides a Web based tool [3] that allows a defender to create packages with approved software, configuration, and plug-ins for Firefox. All are available for Windows platforms only.

International Journal Of Business Management
Available at www.ijbm.co.in
ISSN NO. 2349-3402
VOL. 2(1),2015

Firefox and Google's Chrome browser have implemented sandboxes, in which the browser runs code (such as JavaScript or Flash) in a compartmentalized area of the program that provides only limited resources for the program and whose design is heavily scrutinized for security flaws. Internet Explorer uses a zone-based security model, in which security features are enabled or disabled depending on the site being accessed.

Under Vista, Internet Explorer runs in what is known as Protected Mode, which limits the operating-system privileges the browser program [3]can exercise. Open source developers especially must be very careful about designing and implementing sandbox systems, because their sandbox source code is available to the attacker for study and testing. This is, of course, no surprise to the sandbox developers and one reason why open source sandboxes tend to improve quickly.

Conclusion

Browsers are at the heart of the Internet experience, and as such they are also at the heart of many of the security problems that plague users and developers alike. As the sensitivity of internet usage increased concerns about security, fraud and attacks became important. There are limits to what a browser developer can compensate for, and browser users will not always accept the constraints of security that a browser establishes. Attack and defense strategies are coevolving, as are the use and threat models. As always, anybody can break into anything if they have sufficient skill, motivation and opportunity. The job of browser developers, network administrators, and browser users is to modulate those three quantities to minimize the number of successful attacks.

References

[1]. Thomas Wadlow and Vlad Gorelik.Security in the Browser. Communications of the ACM, pages 40-45, Volume 52, Issue 5 (May 2009)

[2].http://code.google.com/p/browsersec/wiki/Main.

[3].http://www.isecpartners.com/files/iSECattacking_AJAX_Applications.BH2006.pdf.

[4].http://en.wikipedia.org/wiki/HTP_cookie.

[5].http://en.wikipedia.org/wiki/Phishing