Proposition de recherche doctorale

Automated Monitoring and Analysis of Malicious Code on the Internet

Mots clés :

- **Directeur de thèse** : davide BALZAROTTI
- **Co-encadrant(s)** :
- **Unité de recherche** : Laboratoire de recherche d'EURECOM
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

introduction : The world wide web has become an integral part in the lives of hundreds of millions of people, who routinely use online services to store and manage sensitive information. Unfortunately, the popularity of the web has also attracted miscreants who attempt to abuse the Internet and its users to make illegal profits. A common scheme to make money involves the installation of malicious software on a large number of hosts. The installed malware programs typically connect to a command and control (C&C) infrastructure. In this fashion, the infected hosts form a botnet, which is a network of machines under the direct control of cyber criminals. As a recent study has shown [17], a botnet can contain hundreds of thousands of compromised hosts, and it can generate significant income for the botmaster who controls it. Malicious web content has become one of the most effective mechanisms for cyber criminals to distribute malicious code. In particular, attackers frequently use drive-by-download exploits to compromise a large number of users. Other common infection mechanisms include the distribution of malware programs over Peer-to-Peer networks or malicious and misleading websites. Other than malware, several other threats are posed by the Internet today, such as the distribution of rogue antivirus programs and of phishing pages that lure web users into providing sensitive information to malicious organizations. Besides this, also several kinds of fraud are performed daily and on large scales, such as click-fraud and scams. Given the rising threat posed by malicious web pages, it is not surprising that researchers have started to investigate techniques to protect web users. As such, several approaches have been proposed to identify, analyze and block malware spreading on the Internet. The aim of this thesis is to study the approaches that have been recently proposed to target these problems, and to finally be able to develop a new system, based on a combination of static and dynamic analysis techniques, for the automatic detection of malware on the Internet. Our new approach will use several characterizing sets of information do detect whether a machine, or a web site, in general, is infected with malware. To do so, we expect to derive information from the content of web pages, the format of URLs, the patterns appearing in DNS requests, as well as from the use of pre-existing tools made available to us. Our final aim is to be able to effectively detect and fight the security threats that have been emerging on the Internet in the last years, such as drive-by-download malware, phishing and rogue antivirus software. Drive-by-download malware A drive-by-download attack installs malware on a victim machine exploiting vulnerabilities in a web browser or in one of the browser's plugins. In order for it to work, the attacker usually injects malicious scripting code into compromised web sites or hosts it in a server under his own control. When a victim visits a malicious web page, the malicious code is executed, and, if the victim's browser is vulnerable, the browser is compromised, infecting the victim's computer with malware. This kind of attacks have become pervasive over the last few years [5, 6]. Phishing Phishing web sites are used to steal users' sensitive information, such as login credentials and credit card numbers, in order for the criminals to be able to sell the stolen data in the underground market. The way a phishing web site works is simple: the visitor is presented a web page that resembles in every detail a login page of a legitimate bank, or online service. The user does not usually realize he is visiting a fake web page; he's invited to enter his login credentials, which, once submitted, are sent to the criminals. Rogue antivirus software Rogue antivirus programs lure the visitor of a web page into believing that his machine is infected by some kind of virus, presenting themselves as the only way to remove the malware. Once the user accepts to install the rogue antivirus (with the purpose of getting rid of the virus), this piece of software asks for a payment in order to be activated and remove the "virus" that it claims to have detected. The main reason why Rogue AVs are so successful nowadays is their persistence and ability to scare visitors, making them believe they are infected by a virus. A study of this phenomenon has been presented in [13].

Résumé du projet de recherche (Langue 2)

ExistingWork For the identification of web-related threats, such as drive-by-downloads, phishing pages and rogue antivirus programs, one of the most successful approaches is the use of blacklists, useful to prevent users to be infected by a malicious web page once it has been discovered and reported. Some common blacklist services are the ones made available by Google Safe Browsing [15], SpamHaus [8], SpamCop [2] and Blacklist Monitor [10]. These blacklists store URLs that were found to be malicious. The lists are queried by a browser before visiting a web page. When the URL is found on the blacklist, the connection is terminated or a warning is displayed. Of course, to be able to build and maintain such a blacklist, automated detection mechanisms are required that can find on the Internet web pages containing malicious content. The automated identification of web pages containing malware is usually carried out using dynamic analysis tools called honeyclients, such as the MITRE HoneyClient [18], Microsoft's HoneyMonkey [19], Capture-HPC [16], or Google Honeyclient [12]. These systems use traditional browsers running in a monitored environment to detect signs of successful attacks. Other more recent approaches for the detection of malicious web pages include tools such as Wepawet [3], PhoneyC [11], and JSUnpack [7], which rely on instrumented JavaScript run-time environments to detect the execution of malicious scripts. Besides malware detection systems studied for the identification of malicious web pages, also several tools for the in-depth analysis of malware itself have been published. Such tools are able to provide a great level of detail on how the malware behaves. Examples of this kind of tools are Anubis [1] and CWSandbox [14]. The detection of online threats can be carried out also using static approaches. These approaches rely on the analysis of the static aspects of a web page, such as its textual content, and characteristics of its HTML code and JavaScript code, as well as characteristics associated to the URL of the resource (DNS information, IP address, etc.). The advantage of static approaches is their speed, but this usually goes together with a lower precision of analysis. Examples of systems based on static analysis are the one presented in [9] for the detection of scam and phishing pages, and CaffeineMonkey [4] for the detection of malicious JavaScript code. 2 References [1] U. Bayer, A. Moser, C. Kruegel, and E. Kirda. Dynamic Analysis of Malicious Code. In Journal in Computer Virology, 2006. [2] Cisco Systems, Inc. SpamCop. http://www.spamcop.net/. [3] M. Cova, C. Kruegel, and G. Vigna. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code. In Proceedings of the International World Wide Web Conference (WWW), 2010. [4] B. Feinstein and D. Peck. Caffeine Monkey: Automated Collection, Detection and Analysis of Malicious JavaScript. In Proceedings of the Black Hat Security Conference, 2007. [5] D. Goodin. SQL injection taints BusinessWeek.com. http://www.theregister. co.uk/2008/09/16/businessweek_hacked/, September 2008. [6] D. Goodin. Potent malware link infects almost 300,000 webpages. http://www. theregister.co.uk/2009/12/10/mass_web_attack/, December 2010. [7] JSUnpack. http://jsunpack.jeek.org, 2010. [8] T. S. P. Ltd. SpamHaus. http://www.spamhaus.org/. [9] J. Ma, L. Saul, S. Savage, and G. Voelker. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2009. [10] Mailworkz. Blacklist Monitor. http://www.blacklistmonitor.com/. [11] J. Nazario. PhoneyC: A Virtual Client Honeypot. In Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2009. [12] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All Your iFrames Point to Us. In Proceedings of the USENIX Security Symposium, 2008. [13] M. A. Rajab, L. Ballard, P. Mavrommatis, N. Provos, and X. Zhao. The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution. In Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2010. [14] K. Rieck, P. Trinius, C. Willems, and T. Holz. Automatic Analysis of Malware Behavior using Machine Learning. Technical report, Berlin Institute of Technology, University of Mannheim, Vienna University of Technology, 2009. [15] Google Safe Browsing API. http://code.google.com/apis/ safebrowsing/, 2008. [16] C. Seifert and R. Steenson. Capture-HPC. https://projects.honeynet.org/ capture-hpc, 2008. [17] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2009. 3 [18] H. P. Team. HoneyClient. http://www.honeyclient.org/, 2010. [19] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites that Exploit Browser Vulnerabilities. In Proceedings of the Symposium on Network and Distributed System Security (NDSS), 2006. 4