AN ISP BASED NOTIFICATION AND DETECTION SYSTEM TO MAXIMIZE EFFICIENCY OF CLIENT HONEYPOTS IN PROTECTION OF END USERS

Masood Mansoori[1] and Ray Hunt[2]

[1, 2] Faculty of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand

[1] masood.mansoori@gmail.com

[2] ray.hunt@canterbury.ac.nz

ABSTRACT

End users are increasingly vulnerable to attacks directed at web browsers which make the most of popularity of today's web services. While organizations deploy several layers of security to protect their systems and data against unauthorised access, surveys reveal that a large fraction of end users do not utilize and/or are not familiar with any security tools. End users' hesitation and unfamiliarity with security products contribute vastly to the number of online DDoS attacks, malware and Spam distribution. This work on progress paper proposes a design focused on the notion of increased participation of internet service providers in protecting end users. The proposed design takes advantage of three different detection tools to identify the maliciousness of a website content and alerts users through utilising Internet Content Adaptation Protocol (ICAP) by an In-Browser cross-platform messaging system. The system also incorporates the users' online behaviour analysis to minimize the scanning intervals of malicious websites database by client honeypots. Findings from our proof of concept design and other research indicate that such a design can provide a reliable hybrid detection mechanism while introducing low delay time into user browsing experience.

KEYWORDS

Browser Vulnerability, Client Honeypot, ICAP

1. INTRODUCTION

The first implementations of bait system concepts were introduced in information security in the late 1980s [1]. The idea has always been to use deception against attackers by mimicking valuable and vulnerable resources to lure them into exploiting those resources. The purpose of such a strategy is twofold; to gather information on the nature of the attack, attacker, tools and techniques used in the process, and to protect operational hosts and servers.. The first implementations of honeypots focused on imitating server side services and were mainly designed to protect servers and production systems, and gather information on attackers and attacks directed at those services. In recent years, a change in attack behaviour has been detected across the internet. Instead of putting all the efforts to target and gain access to production servers, attackers are targeting end-user systems.

Online attacks targeting users' operating system through browser and browser vulnerabilities are common threats. Attackers use social engineering techniques to lure users into downloading and installing malicious software or malware without disclosing their actual intend. Prompting users to install plugins to watch online videos and mimicking themselves as free antivirus software are two examples of common techniques used by attackers to induce users into installing them. Malicious website are also deployed which contain code that exploit vulnerabilities of popular web browsers, their extensions and plug-ins. Once such a website is visited by a vulnerable browser, the malicious code is rendered and executed by the browser's

DOI : 10.5121/ijnsa.2011.3505

engine, resulting in exploitation of a particular vulnerability associated with the browser or the operating system and its applications. The infected client system may then fetch and install malware from a malware server or allow an attacker to gain a full control over the system. Client honeypots have been designed to identify such malicious websites, using signature, state, anomaly and machine learning based detection techniques. The trend change, from targeting server side services to client side applications can be the result of the following factors in two different parts of network design:

2. ATTACKS ON SERVER AND PRODUCTION ENVIRONMENTS

Servers form the backbone of information and contain the data vital for the organization to function properly and invaluable in terms of capital. Such information directly affects the everyday functions of the company and may cost a company millions if not available for short periods of time. An example is the webserver of a bank that allows customers to perform online transactions. These servers are ideal targets for hackers to attack and exploit. Gaining access to such high profile targets can have numerous advantages in terms of financial benefits and fame within underground community. These attacks can be self or even politically motivated. Taking down such systems can cause the largest damage, creating a single point of failure which affects the entire organization directly. Many companies therefore spend a significant amount in protecting their servers from attacks and unauthorized access. Some of the techniques currently used to protect servers and production hosts and network resources of organizations are:

2.1. Strict Security Policies

Designing a security policy and enforcing it is without a doubt, one of the most critical aspects of a good information security system. Authentication systems based on level of access and user privileges ensure that only users with right permissions are authorized to access the resources on a system or network. Systems operating on principle of granting access based on users' roles in organizations are rightfully called, Role Based Access Control (RBAC). If implemented correctly, RBAC can effectively protect resources against unauthorized access. The effectiveness and efficiency of this system can be seen in the popularity of RBAC model implemented in almost all organizations [2, 3]. Although a user with low privileges may gain a higher access rights through exploiting vulnerabilities in the host system, detection and mitigation of such attacks are beyond the scopes of such systems. Moreover there are security mechanisms in place which will detect any attempts to bypass policies and gain a higher privilege. One of such mechanisms is an Intrusion Detection System (IDS).

2.2. Intrusion Detection/Prevention Systems

Once thought as one solution for all against security attacks, intrusion detection systems have been around for many years and have evolved from simple signature based detection to detection based on anomaly and Artificial Intelligence (AI). Newer IDSs perform Deep Packet Inspection in hardware which allows for faster inspection of packets and can cope with high bandwidth networks. While intrusion detection systems are designed to perform a passive detection and employ a warning approach, Intrusion Prevention Systems (IPS) take a more active approach by responding to an attack through predefined actions (i.e. resetting the connection). Although host and network based implementations of intrusion detection and prevention exist to suit different needs and security requirements of a organization, their role in information security structure has been demoted because of design issues with reliable detection, costs, management overhead and risk associated with implementing such systems.

Intrusion detection systems tend to generate high amounts data and false positives which require extensive analysis efforts. Shortcomings in detection capabilities also effect the implementation of proper prevention systems since companies would not take the risk of denying legitimate access to resources to customers based on unreliable detection of such systems.

2.3. Firewalls

Firewalls are justly the most effective barrier against attacks on production networks. They isolate the network from outside environment and based on the predefined rules, decide what packets get to enter or exit the network. Efficiency of firewalls resides in their simplicity in concept and design, their extensibility and reliability. Predefined actions can be performed on a packet based on different characteristics such as source or destination IP address, port in use and protocol. They can also be configured to isolate the network by running as a proxy, interacting with outside networks on behalf of the internal hosts or implementing NAT to deny direct access from outside networks to internal hosts. Firewalls however fail to detect attacks which are originated from inside or in case if someone is attacking the internal resources from within the network.

2.4. Honeypots

Honeypots are system resources placed on production and research networks of companies or educational institutes to detect and monitor current state of attacks on hosts and servers. Honeypots also play an important role in protecting servers and hosts against attacks targeted at resources available on a production network by directing attacks to decoy systems. These decoy systems are placed within production network and mimic hosts and servers, diverting attacker's attention away from the real ones. Once implemented properly with other technologies such as intrusion detection systems and firewalls, honeypots become highly effective tools against external and internal attacks. A honeypot placed in front of the firewall can help administrator monitor and obtain a detailed picture of the frequency, nature and types of attack. On the other hand, a honeypot placed behind a firewall and within a production environment allows administrators to monitor and detect attacks that have bypassed primary security tools (i.e. Firewall, IDS). Honeypots are effective tools to detect internal attacks and propagation of worms within an internal network which other tools such as firewalls fail to achieve.

2.5. Trained IT Personnel

Companies generally have dedicated network administrators who are highly trained about computers and are responsible for updating users' operating system with the latest security updates, running network forensics tools, patching vulnerabilities and installing and maintaining antivirus software on each host. They are also responsible for monitoring information security policies violations, keeping an eye on employees to operate based on predefined policies in terms of computer usage, either within the internal or outside networks. Network administrators act as monitoring, enforcement and management force behind the information security of an organization. Figure 1 illustrates the results of a survey on the use of security products deployed in companies.

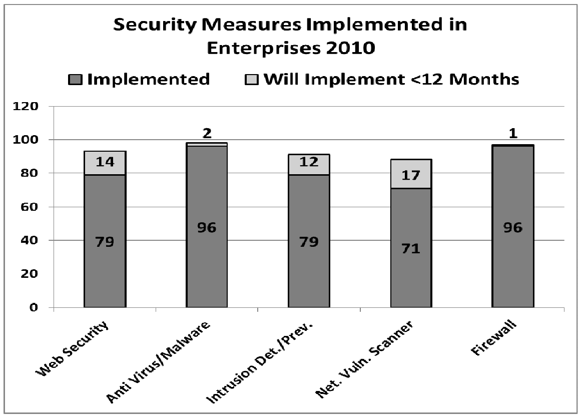

Figure 1. Ratio of security product use in enterprises IT [4]

3. END USER, AN EASIER TARGET

End users, in contrast to hosts and servers in production environments, do not possess the luxury of host or network based intrusion detection or prevention systems, nor do they have someone else responsible for updating their systems with latest updates. They have to rely on self knowledge of security tools, built-in operating system security features and third party software to provide security functions. Security of home users mainly revolves around the following security mechanisms:

3.1. Secure Operating System

Operating system is the platform on which all other user applications run and resembles the foundation of a building. Microsoft is the dominant player in operating system market and has a total share of 94 percent of all operating systems in use. While its latest operating system, Windows 7 is the most stable and most secure operating system, its slow adaptation in home users and corporations has led to wide spread use of older operating systems. According to statistics of the last three month (Nov-Dec-Jan 2010), Windows XP, a popular but vulnerable operating system has a 50% market share while windows Vista owns 15% followed by Widows 7, 28%. Microsoft provides constant updates for these operating systems; it is up to the user to install these updates. A quick look at the operating system security however, confirms that even with constant update and release of service packs, the number of discovered vulnerabilities have not decreased dramatically. Based on the report be Secunia, 75 vulnerabilities were discovered in Windows 7 alone in 2010. Windows Vista follows by 89 followed by Windows XP 97 [5].

3.2. Client Firewall

Most recent operating systems come with built in and “enabled by default” firewall package. Starting with Windows XP service pack 2 and since, firewall has been enabled by default on all Microsoft operating systems. This provides basic protection for an average home user. Based on a latest study in European Union countries in 2010 [Internet usage in 2010 – Households and Individuals], less than 50 percent of the users had their firewalled enabled. Users decide to disable windows firewall because of compatibility issues with other programs. This results in significant threat to the security of the host system.

3.3. Antivirus

Antivirus software is the basic security tool installed in end user computer. They mostly rely on signature based detection where executable files are matched against a signature database of known viruses. New versions have run-time scanning feature that scans the file in real time and avoids execution, if a threat is detected. Signature based detection however results in the antivirus engine failing to detect variants of known viruses, therefore a constant update of antivirus signature database is essential to provide basic protection. Although an antivirus is fairly effective in detection of known attacks if updated regularly, they are unable to protect users from remote port attacks or attacks directed at user applications from internet. Latest survey shows that, 25 % of users disabled their antivirus software because they believe this software have negative impact on their PCs' performance [6].While another study by a research group had similar results showing around 23% had absolutely no active security software installed. Rightfully assuming that every computer without any active protection would be infected with one type of virus or malware, 23% makes a huge impact not only on the infected computers but overall security of the internet as these infected hosts will be used to attack other hosts across the internet, be a part of DDoS attacks or exploited to deliver Spam [7].

3.4. Applications

Operating system vulnerabilities are most discussed in security community but vulnerabilities in user applications cover most of the vulnerabilities found in an end user system. A study by Secunia points to 729 (Windows XP), 722(Windows Vista) and 709 (Windows 7) available vulnerabilities, in top 50 applications on a typical user system [8]. All these applications are offered by 14 vendors. Based on these numbers, more secured operating system does not necessarily provide a more robust platform unless applications are developed in a secured manner or patched properly [8]. With increasing number of users online, application vulnerabilities are the easiest and the main target of such attacks. Based on [8], 84% of all attacks were classified to be "from remote" while local network or local system each had 7 % of total attacks. These results show the importance of implementing a robust internet security while browsing the internet, especially in applications which directly interact with web contents.

Internet browsers are the main tools used to browse and retrieve contents from the internet. With increasing popularity of dynamic internet contents and online services (i.e. online banking), browsers' roles have been more than just to view static contents but rather a part of a new wave of user interaction and experience with the web. With popularity of cloud computing, browser's role will even be more immanent in the usability of novel operating systems. Since browsers are the main tools to interact with online contents, they are a gateway into the host operating systems and local networks. If a browser is exploited, attackers would be able to gain access to system resources and install malicious code on the local system. The infected host then fetches more malicious content from remote locations and targets local hosts. All this is done without firewall blocking any connections. Any NAT implementations are bypassed since once a host is infected all connections are initiated from inside, which is permitted by firewall.

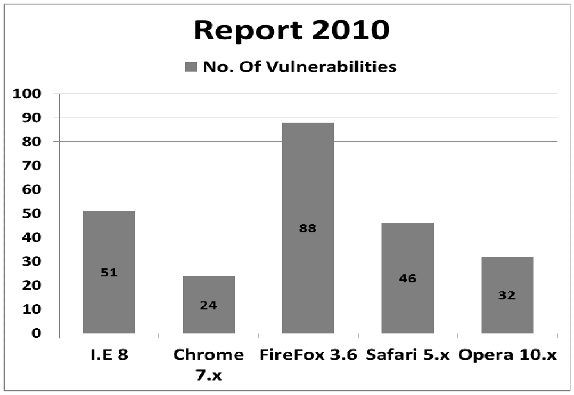

Figure 2. Number of vulnerabilities in popular web browsers [9]

As shown in the figure 2, a vast majority of application vulnerabilities consist of vulnerabilities in user browsers. The vulnerabilities are easy to exploit by malicious webservers which host exploits on websites. Studies illustrate that most of the remote attacks are directed at exploiting such vulnerabilities[8]. Certainly updating to the latest version of the browser and patching operating system reduces the risk of infection dramatically, however as statistics show, even latest versions of the browsers have high number of vulnerabilities.

A method to complement the security and safeguard a user's experience online is to deploy client honeypots and identify such malicious websites. A database of malicious websites allows a user to check his desired URL against the database of known malicious websites and domains in real time and be notified of the security risks. Such database already exists in form of browser plug-ins and APIs [10, 11]. A fundamental drawbacks of such implementations of user notification is the fact that internet consists of roughly 255 million websites [12]. Even if a website takes 1 second to be scanned by a honeypot system, it would take 2951 days for a single honeypot to scan the entire internet. Even with extensive hardware resources, the idea of scanning a massive portion of the internet on regular bases does not sound feasible [13]. Other elements to remember in the nature of malicious websites are:

- Malicious websites change their malicious content frequently, a website which is hosting exploit a day might not host it the next day [14].
- New malicious websites are constantly added. This is either within the certain domains operated by hackers or by targeting legitimate websites and using them as a base to host and deliver exploits [15].
- Malicious content are delivered to users in form of advertisements placed on legitimate websites[15].
- In order to avoid detection and maximize their lifetime, malicious websites do not necessarily deliver exploits on every visit. Exploits might be delivered based on user's browser version, user input, geographical location or visit patterns[16]. Tools such as MPack allows attackers to specify exploit delivery based on different criteria [17].

Scanning the entire internet on regular basis is not feasible nor is able to detect malicious websites effectively and with high accuracy. On the other hand, real time scanning with current honeypot designs, affects user experience negatively as the fastest implementations takes between 3 to 5 seconds for each URL. Moreover, crawling suspicious domains only or crawling internet based on search queries submitted to search engines provides a narrow perspective on the total number of online malicious contents. Any legitimate website is susceptible to SQL injection attacks and hosting malicious content without webmaster's discern.

4. PROPOSED SOLUTION

To effectively combat online threats and infection of end user operating systems, the fundamental design of host security must change and not rely solely on end user to provide and maintain his security. End user's unfamiliarity with security tools and products combined with their unwillingness to deploy and update their security products, contributes vastly to the infection rate on the internet. The proposed idea is based on the notion of increased participation of ISPs in providing near real time security to their customers. A Similar initiative has been implemented by Comcast. Although Comcast's design is proprietary and relies on Deep Packet Inspection (DPI) techniques, the same concept, utilizing free and open source tools can be developed. The benefit of such a system relies on its real time notification system which can be achieved by an implementation of web notification system and ICAP open source protocol.

4.1. Design Challenges

The challenges that the new design should overcome include:

- **Scalability:** One of major attributes in determining the effectiveness of any security tool is its capacity to cover a large number of users. A security mechanism is as good as the number of hosts/services it protects. Proxy based applications are designed to meet scalability setbacks as a proxy architecture is intended to service a large number of hosts. Given that home users do not belong to an enterprise where proxy architecture could be applied, for maximum efficiency, the proposed system should be deployed at ISP level to cover as many hosts as possible.

- **Simplicity and Compatibility:** Today's networks are diverse in terms of operating systems running on individual hosts. One of the challenges in any adaptation of a security tool is its ability to run effectively on any operating system and should not require installation of a particular operating system, protocol or application. It should require minimal resources and be backward compatible, not only with different operating systems from different vendors but obsolete systems running outdated OS without a need to upgrade to a newer version. The system should also be simple to install and manage and as transparent as possible, as users tend to avoid complicated applications.

- **Central Management:** A centralized management provides an easy mean to control all aspects of the system from a single location. Instead of relying on end users who might not be familiar with the system or reluctant to make changes and update the tool the system should be managed and controlled from a central location avoiding the need of end user involvement in installation, control and management of the system.

- **Regular Scanning:** Scanning the entire internet for malicious websites is neither efficient nor feasible in terms of resources (i.e. hardware, bandwidth, operating) and cost associated with those resources. The cost of finding a single malicious website varies depending on the algorithm used to gather and inspect websites, the speed and the accuracy of the detection. According to [18], a sequential scanning may cost as much as "0.293 US dollars for a base rate of 0.4% and 0.021 US dollars for a base rate of 5.4%". While other algorithms such as Bulk or DAC, used by Capture-HPC lower the cost of detecting a Malicious URL, yet these algorithms do not focus on real life URLs gathered from end users. Targeted scanning of suspicious domains on the other hand excludes potentially a large number of legitimate websites which might have been hacked to host malicious content. On the other hand, real time scanning with current honeypot tools and hardware resources affects user experience negatively. The proposed system should be able to provide real time notification alerts based on recent scanning of URLs with detection based on current available honeypot systems and integration of other security mechanisms (i.e. Antivirus, Intrusion Detection Systems and Malicious website lists).

4.2. System Design

ISPs, depending on the size have more than a few hundred users and access to qualified personnel to manage and control such a system on behalf of customers, therefore implementation of such a design at ISP level solves scalability and central management design issues. In order to meet simplicity and compatibility issue, the system must be integrated into an application or a service available almost on every operating system regardless of the vendor. Such application should also have direct access to internet to provide constant and up to date notification while browsing a website. A web browser meets all the requirements of simplicity and compatibility.

Browsers are available in all operating system and used primarily to browse and fetch website content over HTTP protocol, which avoids the need for a new protocol to deliver the notification. Notification exchanges between clients and the central system located at ISP can be achieved through deployment of ICAP service in conjunction with web notification system. ICAP service is a light weight protocol based on request/response design, allowing ICAP clients to pass HTTP contents to an ICAP service for transformation or modification. The modified contents are then passed to the client for further processing and demonstration. Web notification system allows the utilization of such client server based content modification to pass security notifications to client computers within client browsers. Since web based notification system relies on web browsers to deliver notifications to end users, this approach can easily be implemented in almost all home user computers. Web based notification system is also fully compatible with all operating systems and browser versions, since notifications are appended to URL displayed on a user computer in plain HTML or java script.

Detection mechanism of the proposed design is based on client honeypots. The type of client honeypot deployed, is not the design concern and focus of this paper but rather how it manages to keep up with the changing dynamics of malicious websites. As we discussed earlier, considering the size of the Internet, scanning the entire internet is not practical to achieve. In order to overcome issues with scanning and updating the malicious database, a different alternative should be considered based on user browsing habits should be analyzed and considered.

Studies show that users in general exhibit specific patterns online in terms of performing search queries, visiting websites and using browser specific features. Users exhibit behaviours such as “Backtracking” where a user clicks the “Back” command to exit a server through the path he used to enter that particular site. A behaviour related to this paper is how users interact within a particular website, how often they visit, how many visits they perform and how many internal links are clicked in a common website. Studies suggest that users exhibit patterns of rarely traversing more than two layers in a hyperlink before returning to the entry point regardless of hyperlink per page ratios [19, 20]. As stated by the author “a typical user only requests one page per site and then leaves” [21]. As we all know, some websites are more popular than others and receive higher number of clicks. A study conducted with data gathered from ten proxies indicates that a low number of servers (25%) contribute to a large amount of all online accesses (80-90%) while servers that are accessed once have a low share of 5%. [20, 22].

Regardless of the type of honeypot used in the proposed design, user browsing habits can be used to overcome the difficulties in updating the malicious website database. The main idea is to prioritise websites, not based on their suspicious likeliness but rather on their popularity with end users. There are few rules that could help with prioritising URLs:

- **Highly trusted domains:** any website belonging to this category could be safely ignored without any scanning. This category may include websites in known domains (i.e. Google) or websites with digital signatures (i.e. banks)
- **Dynamically generated and highly visited websites:** Websites that contain dynamic contents and are visited regularly but do not belong to a trusted domain. These websites are the main focus of the proposed design. These websites are accessed by many users and are likely to be hacked to host malicious web content without webmaster’s alert. Since many users access these websites, their infection ratio is high causing large number of infections in a very short period of time. They have the highest priority in the proposed design in terms of scanning interval. Different ISPs might have a different policy to rate a website to belong to such a group but as a rule of thumb, minimum of 5 visits in a 24 hour can be considered as the base option.
- **Websites with low access ratio:** In order to maximize the effectiveness and focus the detection on the second category, Websites that are rarely accessed are not scanned by the honeypot but rather by an antivirus engine and matched against public malicious website databases (i.e. Google Safe Browsing). Scanning with antivirus engines and matching

against APIs are not resource intensive and introduce a minimal delay in a user's browsing experience while providing a fairy adequate level of detection.

The proposed system implements a three way detection and verification mechanism to ensure a high effectiveness in detection of malicious websites. The first step in the deployment of the system is to gather URLs browsed by users and rank them accordingly based on their number of visits and domains. Each ISP might have a different policy on how to rank a website, what domains are considered safe and what error messages should be displayed to the end user.

An initial data gathering period ranks and prioritizes the websites and categorizes them according to their content, domain or number of visits. The results are then saved in a database accesses by the honeypots. The websites belonging to the highly visited and dynamic websites category are scanned using a preferably high interaction honeypot and results of the analysis are saved. To maximize effectiveness of this system based on the user browsing behaviour research, all links from this websites are also extracted and scanned, one or two levels down the initial page (Figure 3).

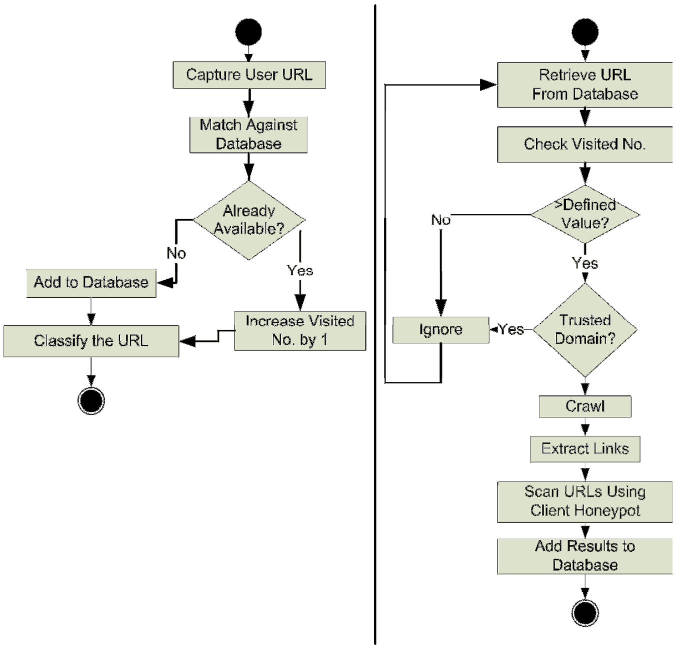

Figure 3. Illustration of the processes of adding URLs to database and Honeyclient scanning

After the initial state, each time a user visits a website, the URL is matched against the internal database of websites, if a match exists; the website is delivered to the user based on the state of the website in the databases:

- If website is marked to be malicious, the content is passed to ICAP service where a user notification is inserted into the website content and passed back to the client.
- If the URL is not marked as malicious, the URL is matched against the database of external malicious websites (APIs), if a match is found, the same process of web content modification of ICAP server takes place.
- The web content is then retrieved and scanned using an antivirus engine (i.e. ClamAV), if a match is found, the same process of web content modification of ICAP server takes place.
- If no match is found in any of the databases, the content is delivered to the user without any modifications or warning messages.

Incremental update of a ranking value assigned to each website will cause the URLs to be ranked and assigned to either of the two categories. Removing the websites with low ranking value, which are not visited by many users frequently, assures that the internal database does not expand exponentially.

Regular scanning of the internal database is performed frequently by a high or low interaction honeypot to minimize the time lap between each scan and provide the most updated analysis. Since the size of the database will be much smaller than the entire internet, comparatively very little in terms of resources are needed to achieve scanning process.

The design could be made more secure by integrating virus scanning engines into the design. Proxy based virus engines are common and freely available to download. In order to maximize effectiveness of such a system, all websites, regardless of being detected as malicious or not, or belonging to a low ranked category can be scanned using the antivirus engine. Since these systems introduce very little in terms of processing delay, the user experience will not be affected drastically. Moreover these systems do not require expensive hardware to implement and operate. Figure 4 and 5 illustrate the overall design and processes of the system.

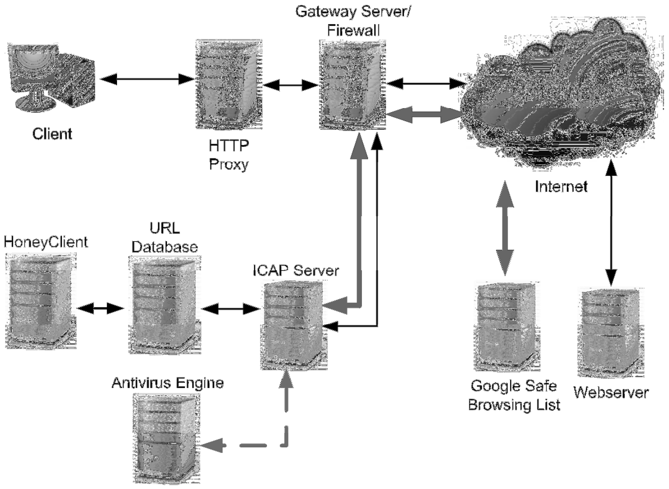

Figure 4. Overall architecture of the system

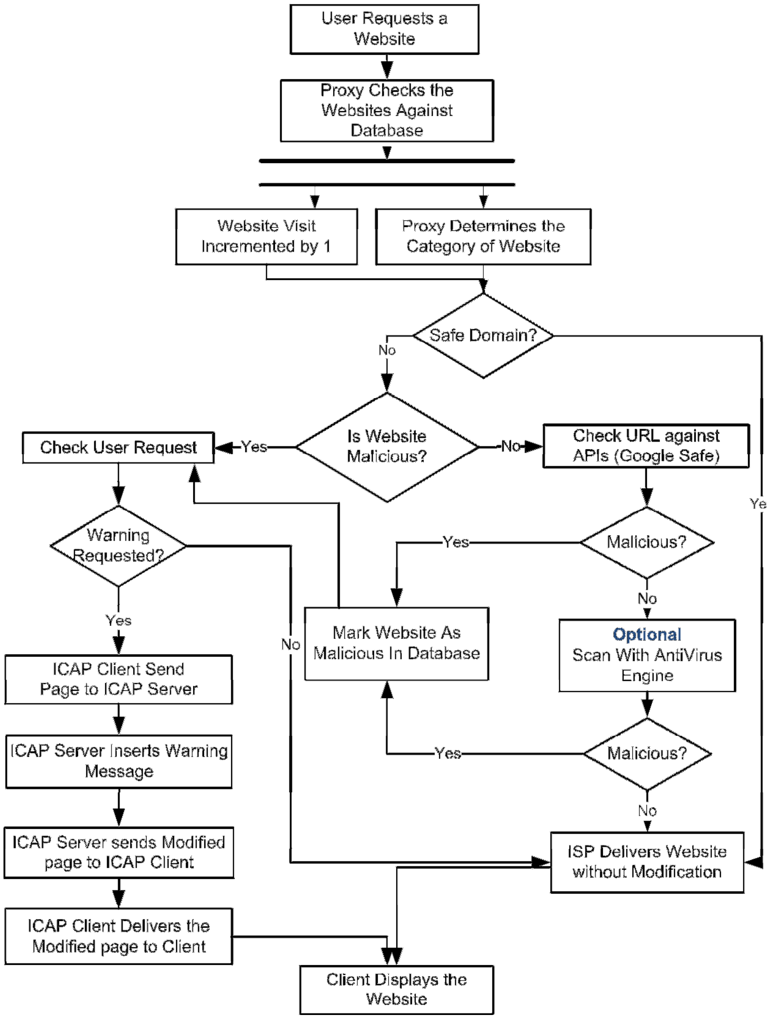
User Requests a Website
Proxy Checks the Websites Against Database
Website Visit Incremented by 1
Proxy Determines the Category of Website
Safe Domain?
No
Yes
Is Website Malicious?
Yes
No
Check User Request
Check URL against APIs (Google Safe)
Warning Requested?
Yes
No
Malicious?
Yes
No
Mark Website As Malicious In Database
Optional
Scan With AntiVirus Engine
Malicious?
Yes
No
ICAP Client Send Page to ICAP Server
ICAP Server Inserts Warning Message
ICAP Server sends Modified page to ICAP Client
ISP Delivers Website without Modification
ICAP Client Delivers the Modified page to Client
Client Displays the Website

Figure 5. Overall flow of the design functions

Once an attack is detected, The ICAP service modifies the website, adds a warning message to the HTML file received from the ICAP client and sends the content back to the ICAP client to be displayed on the client's computer. An option should be available to allow users not to be notified in case they do not want to receive any notifications. Multiple users might be located behind a NAT therefore keeping a track of which users are interested in receiving the warning messages is difficult and complicated. A full state handling should be implemented to link each request to its response. In scenarios where a single user is surfing the web with a public IP address or where users of an organization are located behind a static NAT, request and response messages can be achieved by measures such as page URL or client IP address.

5. TESTBED AND PROOF OF CONCEPT SYSTEM

Architecture of the proposed system is constructed based on the previous work to design an effective notification system based on ICAP protocol. The system integrates a combination of detection tools and ICAP protocol to provide a reliable detection and notification system to end users [23]. In order to analyse the effectiveness of such a design, the system can be evaluated based on the performance of each individual component. The effectiveness of the system can be measured with two different aspects:

5.1. Detection Modules

5.1.1. Detection Rate

The effectiveness of detection tools such as antivirus and client honeypots are typically measured by the detection rate and number of false alarms they generate. Different client honeypots are used to detect various types of attacks. While high interaction client honeypots generally have a higher detection rate, they are less effective in detection of malicious websites that require an input to present the user with malicious content or operate on a time trigger. This system was deployed using a high interaction honeypot namely Capture-HPC. Capture-HPC is the most widely used high interaction honeypot and has proven effective in detection of malicious website attacks in previous research. [24-28]. High interaction client honeypots have a relatively higher detection rate, lower false positive rate but a lower processing speed than a low interaction honeypot [18, 28].

Antivirus software's efficiency has always been questioned in detection of zero-day viruses, malicious software and attacks, given the fact that they rely on a signature to detect the malicious behaviour. Creating reliable signatures which effectively detect malicious behaviour while minimizing false positives and false negatives requires constant monitoring of threats and this process is performed by security vendors and individuals. Once a signature is created for an attack, the antivirus software is highly reliable in detection of that specific threat.

A study conducted in 2010 by [29] focused on drive-by downloads reveals that prior to update of the popular Norton antivirus database, it managed to detect 66% of the malware while an update after 30 days enabled the antivirus to achieve 91% detection rate. Our tests also showed a high rate of false positive and false negative using ClamAV antivirus. A commercial antivirus engine is highly recommended because of regular and up to date signatures provided by the vendor. Nevertheless, compared to an unprotected system, a 66% detection rate is nonetheless comparatively highly effective in protecting end users against popular threats. Detection rate however varies based on the product and the vendor. Kaspersky antivirus was able to detect a significant 91% of all the threats in the initial test and 98% following the update. Honeyware, a low interaction client honeypot using multiple antivirus engines, achieved a 98% detection rate in which Avira antivirus had the highest detection rate. This shows a great promise in the deployment of signature based detection tools against web based browser attacks in future systems[30].

5.1.2. Performance

Performance of security tools are also measured by the time required to complete the scanning and detection task. This system aims at providing real time detection and notification to users therefore any delay introduced to user browsing experience should be measured and minimized. The client honeypot should be able to scan a large number of websites on the background with high detection and speed rate to minimize the time between each scanning of the entire gathered URLs. Antivirus module on the other hand scans websites in real time before being passed to an end user; therefore it should be capable of providing an adequate level of detection while introducing the least amount of delay.

In the designed proof of concept system [Four systems of Core2 duo 2.1GHz with 2GB RAM], capture-HPC was able to scan 10000 crawled websites from social networking website Twitter, per day per system and between 15000 to 20000 links, gathered from known malicious website lists. The higher number of visited websites by high interaction honeypot in the known malicious website category was due to DNS errors affirming the changing dynamics of malicious websites.

Based on our experiments, the antivirus introduced an average of 100ms to scan files up to 100KB in size and roughly less than a second for files of 1MB. The time varies depending on the size and the content of the webpage. In order to minimize the delay, any content larger than 1 MB can delivered without scanning.

The Google Safe Browsing API introduces so little delay into user browsing experience that could be easily ignored. It is due to the fact that most browsers (i.e. Firefox) download and keep a local database of the hashed malicious website list. Every URL is hashed and matched against the local database upon every visit. The local database is sequentially and periodically updated.

5.2. Notification Modules

Assessing the performance of the notification system relies on two factors:

- The number of browser and operating system it supports
- The lag it causes before delivering the content to the user

Traffic from ICAP client to ICAP server and back to user browser are delivered using TCP protocol therefore the ICAP protocol messaging system is supported by different browsers and operating systems. Warning messages inserted into the website code by the ICAP server on the other hand, can be in form of any scripting languages (i.e. Java script). Java script messages are easily rendered by all browsers and supported by all operating systems. In our implementation, the warning message consisted of HTML DIV tag, CSS to format the layer, and JavaScript code to minimize the bottom of the screen. Chrome and Firefox were tested and successfully displayed the warning message.

Table 1. Results of Performance Testing

ICAP Response Content Insertion - Static	Little performance impact, except in high concurrency
ICAP Response Content Antivirus Scanning	Moderate performance impact, except in high concurrency
ICAP Request & Response Multiple Operations	Moderate performance impact

6. CONCLUSION

Malicious websites are an increasing threat against security of end user systems. Attackers take advantage of the vulnerabilities in client web browsers to take control over a system by either luring users into visiting infected websites that host vulnerability exploits or hijacking legitimate websites and inserting malicious content into them. Studies show that a high number of end users are not familiar with security products or reluctant to use and update their security products. In order to combat web based browser attacks, internet service providers and large organizations need to step up their efforts to directly protect their users against these attacks. Our proof of concept system with minimal hardware configurations, deployed at proxy level shows that a combination of client honeypots, real time antivirus scanning and matching user entered URLS against malicious website lists, provide an adequate level of protection while introducing low delay into a user online browsing experience. The design also builds upon studies on user browsing habits to minimize the scanning interval of URL database hence overcoming the difficulty of detecting websites which change their malicious content frequently. The system also employs a notification system alerting user of a threat by sending warning messages through ICAP protocol if a website is detected to be malicious.

ACKNOWLEDGEMENTS

We would like to sincerely thank Michael Pearce for his contribution and help in writing this paper.

REFERENCES

[1]. Cheswick, B. (1990), An Evening with Berferd in which a cracker is Lured, Endured, and Studied: *Citeseer*.

[2]. Ramaswamy, C. and R. Sandhu. (1998), Role-based access control features in commercial database management systems: *Citeseer*.

[3]. WA, J. (1998), "A Revised Model for Role-Based Access Control": Citeseer.

[4]. Aberdeen Group, S. (December, 2010), "Managing Vulnerabilities and Threats (No, Anti-Virus is Not Enough)". December; Available from: http://secunia.com/gfx/pdf/Aberdeen_Group_Research_Brief_december_2010.pdf.

[5]. Secunia. (2010), "Factsheets By Windows Operating System - 2010". 20 - March - 2011]; Available from: http://secunia.com/resources/factsheets/2010_win_os/.

[6]. Avira. (December 2010), "Anti-Virus Users Are Restless, Avira Survey Finds". 24 - April - 2011]; Available from: http://www.avira.com/en/press-details/nid/482/news/avira+survey+restless+antivirus+users.

[7]. PcPitstop. (2010), "The State of PC Security". 20 - December 2010]; Available from: http://techtalk.pcpitstop.com/2010/05/13/the-state-of-pc-security/.

[8]. Secunia. (2010), "Secunia Yearly Report - 2010". Available from: secunia.com/gfx/pdf/Secunia_Yearly_Report_2010.pdf.

[9]. Secunia. (2010), "Research Reports, Factsheet by Browser - 2010". [cited 2011 5 - January]; Available from: http://secunia.com/resources/factsheets/2010_browsers/.

[10]. Google. ("Google Safe Browsing API". [cited 2010 20-January]; Available from: code.google.com/apis/safebrowsing.

[11]. McAfee. ("SiteAdvisor: Website Safety Ratings and Secure Search". [cited 2011 5 - January]; Available from: www.siteadvisor.com/.

[12]. Pingdom. (2010), "Internet 2010 in numbers". Available from: http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers/.

[13]. Dewald, A. and F. Freiling. (2010), ADSandbox: Sandboxing JavaScript to fight Malicious Websites. in *In Proc. of ACM Symposium on Applied Computing*. Sierre, Switzerland: *ACM*.

[14]. Provos, N., et al. (2007), The ghost in the browser analysis of web-based malware: *USENIX Association*.

[15]. Provos, N., et al. (2008), "All Your iFrames Point to Us", in 17th Usenix Security Symposium: San Jose, United States.

[16]. Finjan Security. (2008), "Web security trends report - Q2/2008, 2008". [cited 2011 4 September]; Available from: http://www.finjan.com/GetObject.aspx?ObjId=620&Openform=50.

[17]. Symantec. (2009), "Mpack, packed full of badness". Available from: http://www.symantec.com/connect/blogs/mpack-packed-full-badness.

[18]. Seifert, C., P. Komisarczuk, and I. Welch. (2009), True Positive Cost Curve: A Cost-Based Evaluation Method for High-Interaction Client Honeypots: *IEEE*.

[19]. Catledge, L.D. and J.E. Pitkow. (1995), "Characterizing browsing strategies in the World-Wide Web". Computer Networks and ISDN systems. 27(6): p. 1065-1073.

[20]. Tauscher, L. and S. Greenberg. (1997), "How people revisit web pages: empirical findings and implications for the design of history systems". International Journal of Human Computer Studies. 47: p. 97-138.

[21]. Huberman, B.A., et al. (1998), "Strong regularities in world wide web surfing". Science. 280(5360): p. 95.

[22]. Abdulla, G., E.A. Fox, and M. Abrams. (1997), Shared user behavior on the World Wide Web: *Citeseer*.

[23]. Pearce, M. and R. Hunt. (2010), "Development and Evaluation of a Secure Web Gateway Using Existing ICAP Open Source Tools".

[24]. Büscher, A., M. Meier, and R. Benzmüller. (2010), Throwing a MonkeyWrench into Web Attackers Plans: *Springer*.

[25]. Seifert, C., et al. (2008), Identification of Malicious Web Pages Through Analysis of Underlying DNS and Web Server Relationships: *Citeseer*.

[26]. Seifert, C., I. Welch, and P. Komisarczuk. (2008), Identification of malicious web pages with static heuristics: *IEEE*.

[27]. Wondracek, G., et al. (2010), Is the Internet for porn? An insight into the online adult industry: *Citeseer*.

[28]. Chen, K.Z., et al. (2011), "WebPatrol: Automated Collection and Replay of Web-based Malware Scenarios".

[29]. Narvaez, J., et al. (2010), Drive-by-downloads. in *Proceedings of the 43rd Hawaii International Conference on System Sciences*. Hawaii, USA: *IEEE Computer Society*.

[30]. Alosefer, Y. and O. Rana. (2010), "Honeyware: A Web-Based Low Interaction Client Honeypot", in Proceedings of the 2010 Third International Conference on Software Testing, Verification, and Validation Workshops, IEEE Computer Society. p. 410-417.

Masood Mansoori received his Bachelor’s Degree in Computer Science and Information Technology from Eastern Mediterranean University, Cyprus and his Master Degree in Data Communications and Networking from University Malaya, Malaysia in 2010. He is currently doing his PhD studies in University of Canterbury, New Zealand. His research interests include computer networks, network and system security and primarily client and server honeypots.

Ray Hunt is an Associate Professor specializing in Networks and Security in the College of Engineering at the University of Canterbury, Christchurch, New Zealand. His areas of teaching and research are networks, security and forensics He heads the Computer Security and Forensics Post Graduate program at University of Canterbury.