SCARECROW: Scalable Malware Reporting, Detection and Analysis

[1]Osamah L. Barakat, [2]S. J. Hashim, [2]R.S.A., [2]Abdul Rahman Ramli, [2]Fazirulhisyam Hashim,
[2]Khairulmizam Samsudin, [3]Ibrahim Ahmed Al-baltah, [2]Mohammed Mustafa Al-Habshi
[*1] *Faculty of Computer and Information Technology, Sana'a University,*
o.barakat@su.edu.ye
[2] *Faculty of Engineering, Universiti Putra Malaysia,*
sjh@upm.edu.my
[3] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia,*
abou_amel@yahoo.com

Abstract

Malware is the main computer security threat that can cause damage to user's devices and company's infrastructure. End users who want to download executable files from the Internet are currently presented by a binary choice (OK or Cancel) but there is no viable third alternative for uncertainty (Not Sure). Reporting to any security agency or company for status inquiry regarding executable files normally lack of efficiency in terms of reporting back to the users in a timely manner. As a consequence, developing a more efficient approach that provide a prompt response to the users on reported suspicious files is important in order to encourage more end users engagement in malware reporting thus ultimately reducing the number of unknown malware in the wild. This study proposes a new automatic and scalable malware analyzer that is able to quickly scrutinize and help generate report for each malware detected. The implementation of the approach includes both the client (user's system) and the backend processing (security agency). The client side provides a user friendly and integrated reporting mechanism. The backend is based on both static and dynamic analysis for comprehensive malware detection and profiling. The backend utilizes cloud computing infrastructure to scale, speed up and automate the overall analysis and feedback processes. The system provides a win-win situation for both end user and security agency by providing sustainable and successful symbiotic anti-malware eco-system.

Keywords*: Malware, Malware analysis, Virtual machines, Cloud computing, Scalability*

1. Introduction

Nowadays, cloud computing has emerged as one of the leading Information Technology (IT) paradigms [1], [2]. As with any expanding IT technology, this comes with many advantages such as scalability, power efficiency, and effective resource management [3]. Furthermore, one of the essential advantages of cloud computing is outsourcing, meaning that intensive operations are moved from desktops to more powerful machines which provide major improvements in performance.

The last two decades have witnessed accelerating yet paced Internet-based applications from peer-to-peer to social networking. At the same time, the availability of the Internet has seen the global proliferation of malicious software such as trojans, viruses and worms. The Internet has becomes a suitable environment for proliferation of application software which include both the useful and the dangerous. Thus, computer and mobile device maintenance requires protection from malwares. For example, according to a report [4], there were 5.5 billion attacks blocked in 2011 compared to only 3 billion attacks in 2010. New malwares are now introduced on a daily basis. Unfortunately, the popularity of peer-to-peer networking and social networking accelerate their distribution because users are sharing programs or executable files that can be malicious if installed inside their computer and mobile devices [5].

Alerting users to malware before downloading the executable file is more highly desirable than ever to prevent security incident. This is not an easy task since introducing knowledge of submitted

malware requires substantial computing resources [6]. Performing malware detection and analysis manually and off-line also requires enormous man power. A close look at the literature shows that the response time in this area of computing is very slow. We have therefore developed a new scalable cloud based system called SCARECROW that aims at automating the process of malware detection and analysis. In addition we have adopted crowdsourcing technique to engage end users for reporting suspicious executable files before downloading.

SCARECROW accelerates the malware reporting, detection and analysis cycle. The detection process is based on static and dynamic analysis. For static analysis, a complete existing malware signature database is used and result can be known in immediately. For dynamic analysis, suspicious unknown programs or executable files there is a further analysis for any malicious attribute and prompt response will be generated for the end users. The response generated by SCARECROW is to inform the end users that either the submitted file is malware free or that malware has been detected. If malware has been detected, the feedback includes summary of information about the malware such as name, behavior and severity so as to increase awareness of end users.

SCARECROW can benefits end users and cyber security experts at the same time. End users are then able to get experts' opinions about any program they wish to download. SCARECROW provides them with three choices when they wish to download files from the Internet without limited to binary choice 'OK' or 'Cancel'. They can confirm downloading, cancel downloading, or submit the file cyber security agency or company. By choosing the third choice, the file is not first downloaded to the end users' machine. Instead, the download link is sent directly to the cyber security lab for inspection. As an alternative, any user who has downloaded the suspicious executable files can upload through the website. Cyber security experts will be happy for the increasing number of collaborative end users who keep reporting suspicious files to SCARECROW. This can improves their ability to detect and any new malware in a faster time frame than current situation that be up to few years. Thus, their database on new malwares signature will improve rapidly. Furthermore, it can provide them more opportunities to study the malware possibly giving insight into zero-day malware that exploits unknown computer vulnerabilities.

The cloud based platform is scalable due to a virtual machine (VM) instance is launched for every submitted program or executable file. The system automatically determines the type of operating system required for running the program. For example, an EXE program is run under the Windows operating system VM while APK program is run under Android operating system VM.

As shown in Figure 1, the general workflow in SCARECROW starts when an end user reports by submitting a link to a file or a file to SCARECROW. When request arrived, it gets the file and analyzes the file statically by checking the file's signature against malware white and black. If a match is found, notification email will be sent immediately to the corresponding end user. If no match is found, then it starts a new dynamic analysis task and sends it to cloud enabled malware analyzer. A new VM will be started for each submitted malware which analyzes it and produces a behavioral report. Based on that report, detection system will try to guess if it is safe program or not. Security expert will further verify the file status from the automated report and possibly perform further analysis. Eventually, if the file is deemed to dangerous it will be added to existing black list.

The rest of this article is presented as follows: Section 2 briefly presents and reviews related work. Section 3 introduces the proposed system. Section 4 reviews the proposed system and provides a comparison to normal anti-virus programs. Section 5 concludes the presentation and outlines work for the future.

2. Related Work

Malware analyzers have adapted and used VM as a core element in their systems [7], [8]. The main purpose for using VM is to isolate and prevent any actions conducted by suspected software to propagate in real machines, in addition to providing an identical environment to real systems. Most of those malware analyzers are now available to the public as a service which may be available for free (e.g. "Anubis Malware Analyzer" [9] and "Cuckoo Sandbox" [10]) or as commercial offerings (e.g. "Norman SandBox" [11]). Despite the success stories surrounding VM usage in malware analyzers, it is no longer considered to be a perfect tool to detect and expose malicious software. The reason involves malware developers having means to detect virtual environments [12], [13], and developing

malware which behaves normally with no suspicious instructions once a virtual environment is detected.

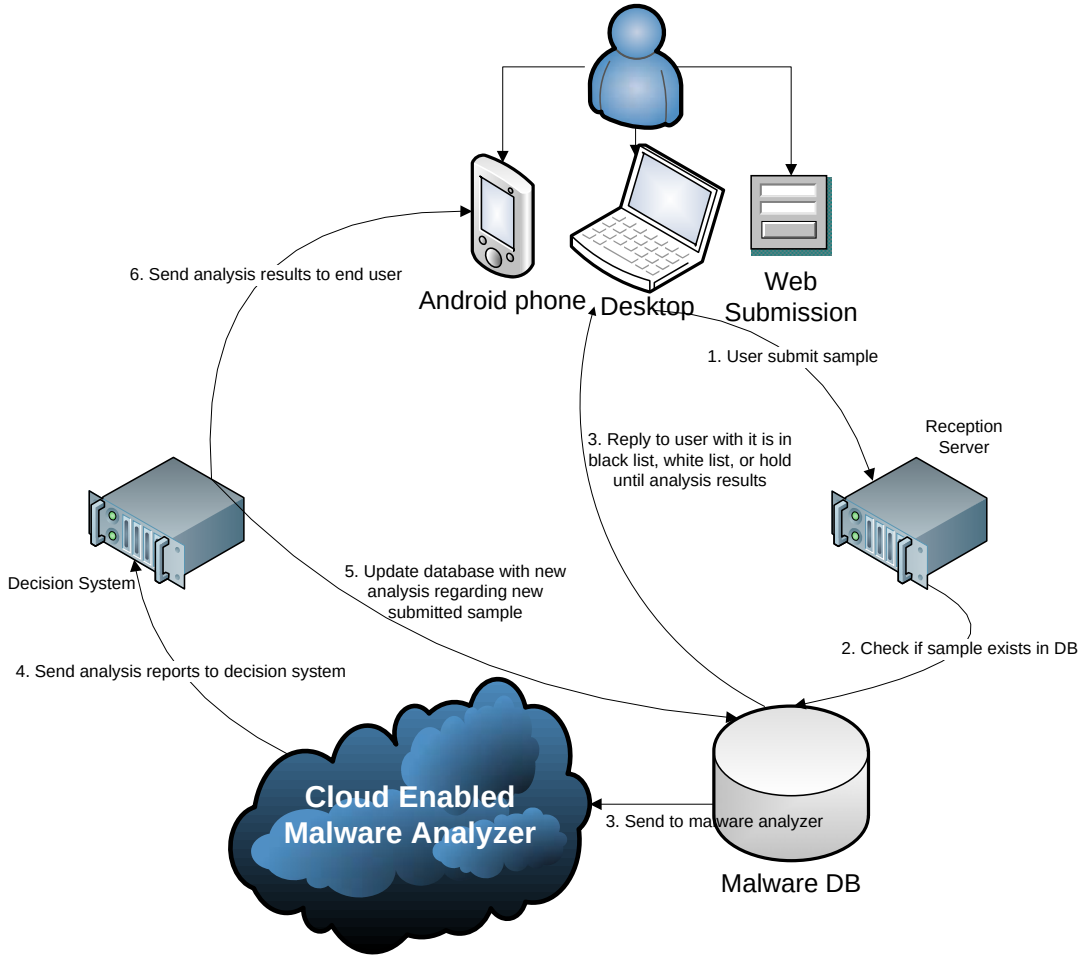

Figure 1. SCARECROW Work Flow

There have been many approaches to evading virtual environment detection [14], [15], [16]. The main variable here is the virtual machine. It still acts as a main player in malware analysis systems because attackers do not always anticipate the normal user accessing virtual systems as they are not certified experts who analyzes programs. As a result, cloud computing, where virtualization is a main component [17], can be used to advantage in the malware analysis industry.

Cloud computing is presently utilized in the malware analysis industry in three different ways. The first, as proposed in [18], is where most of the analysis work is conducted in the cloud environment and only system related function calls will be executed on the client's personal computer (PC) in order to trick malware programs. The second utilizes the cloud system to move all scanning procedures to cloud systems which yield fewer false negatives and facilitate many small engines in customers' machines. Examples of this, uCLAVS and CloudAV have been proposed in [19], [20] respectively. The third utilizes the cloud system in order to detect malware attacks using the Hadoop environment [21] as proposed in [22]. However, most of the current approaches lack an ease-of-use interface and require users to have some technical knowledge to employ them. Current approaches also involve slow response times. The advantages of the developing SCARECROW system compared with others are: it provides easy submission mechanisms, it has a fast response time to the end user, and it supports many file formats including Android mobile applications.

3. SCARECROW System

3.1 SCARECROW System Architecture

Figure 2 illustrates the layered architecture of the SCARECROW system. It is composed of three different layers: the submission layer, the application layer, and the database layer. More details about these layers are presented in the following subsections.

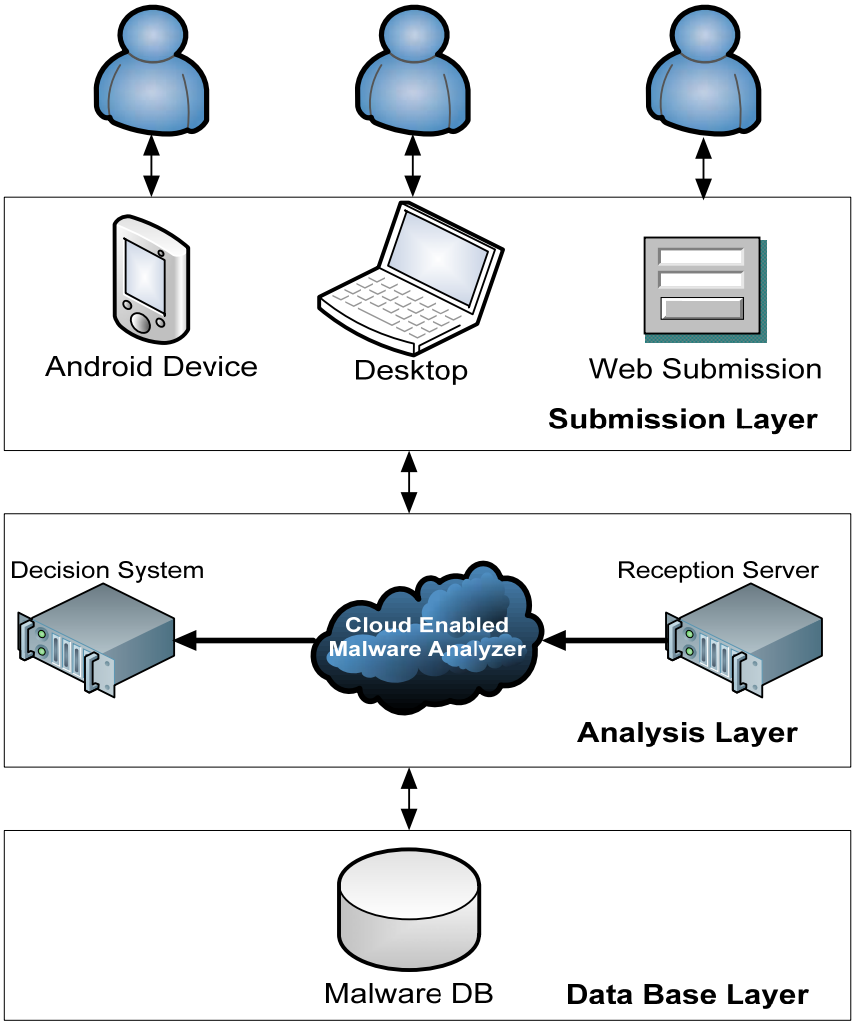

Figure 2. SCARECROW System Architecture

3.1.1 Submission Layer

As we have mentioned above, the main purpose of SCARECROW is to help end users make the correct decision about downloading and running any executable file from the Internet. SCARECROW therefore provides easy submission mechanisms for the end user to submit the target file. Furthermore, SCARECROW is not only targeted to help the computers' end users, it also targets Android devices end users such as those with smartphones and tablets. This is due what has become the continuing growth in Android device usage. On 12th September 2012, Stephen Shankland [23] stated that 500

million devices have been activated with 1.3 million users every day. As can be seen in Figure 1, the file can be submitted from Android devices, computers, and from websites.

3.1.2 Application Layer

The application layer exists between the submission and database layers and is the core layer for SCARECROW functions. This layer consists of three main components: the reception server, the cloud enabled malware analyzer, and the decision system. The reception server is responsible for receiving the submitted files from users. The submitted file is then analyzed in the cloud enabled malware analyzer. The decision system is responsible for making the appropriate decision about the submitted file with the help and verification from security expert in order to reduce false positives.

3.1.3 Database Layer

This database stores most of the necessary information about previously detected malware, and categorizes them into white and black lists. The former list contains programs with proven 100% free of any malicious activity (i.e., operating system updates and core files), while the latter list contains malicious programs known to result in harmful actions to the end users' programs and devices.

3.2 SCARECROW Malware Reporting

The reporting processes involve three phases: submission/reporting, analysis, and feedback/ notification.

3.2.1 Submission/Reporting Phase

SCARECROW provides two simple submission methods to allow clients to send files for analysis. In the first method, users can submit the files by uploading them through the SCARECROW website. In the second method, users install the client software provided by SCARECROW to submit the URL of the suspicious file before download it into their computers or Android devices. SCARECROW provides two types of client software. The first is compatible with a PC platform and supports computer users. The second is compatible with the Android platform (e.g., mobile phones) to support Android device users.

If using the SCARECROW website method, users fill in some basic information such as user name and email address in order to allow SCARECROW to send the related feedback for their submitted files. Subsequently, users are permitted to send either a single file or a group of files to SCARECROW.

Users should also download SCARECROW client software from the SCARECROW website and install it in their machines with the aim of using the second submission method. This program works to monitor Transmission Control Protocol/Internet Protocol (TCP/IP) stacks. Once the user tries to download an executable file, the SCARECROW program interrupts this action asking the user to select from among three choices: 1) confirm downloading, 2) cancel downloading, or 3) submit the file to a cyber-security agency. When the user chooses the third option, the URL link of this file and this user's email is sent to a SCARECROW reception server. This method allows the analysis process to take place without downloading the suspicious file, preventing any risks of infection to the user's machine.

Generally, the two mentioned methods will send all of the reports to the reception server. Based on the arrival request and before sending files for analysis, the reception server downloads sample files if the request came through the second method.

In all cases, any submitted file should be statically checked against black and white list. Those two lists will save analysis time and allow SCARECROW to respond immediately to end users. The two lists have been already fed with world known safe malicious free programs information (white list) and confirmed malicious files (black list).

Both lists uses cloud computing storage and computation thus user's computer and mobile devices will not be burdened by huge list storage or the processing overhead. All analysis requests against both lists will be on cloud which will be faster and has comprehensive databases. Cloud computing in its nature, depending on available resources, can analyze the file quickly and store big databases. This

feature yields higher performance and lower false positive compared to analysis being performed on normal user's computer or mobile device.

3.2.2 Analysis Phase

SCARECROW uses a malware analyzer with cloud computing features enabled to enhance the analysis phase. This malware analyzer is being developed to expand and shrink based on number of submitted files. Thus, end users will get a much faster response with SCARECROW compared to normal and standalone analyzer. All submitted analysis requests until this stage should have all needed information and sample files that have been downloaded by the reception server if they are submitted through the second method. Subsequently, a quick search against the malware database is performed to check whether the submitted file is a well-known malware or not. The result of this initial examination is either a malware free decision when it is found in the white list, or known malware finding when it is found in the black list. After that, the end user will be immediately notified. Otherwise, it is considered an unknown file which goes to the cloud enabled malware analyzer for further investigation and the user is notified to wait until a full analysis report is ready. The cloud enabled malware analyzer is then processes the requested analysis tasks. After verification from the security expert, and possibly further analysis, the analysis result summary is sent to the end user for every reported file.

3.2.3 Notification Phase

All reports which have been generated from the analysis phase are sent to the decision system. This system processes those reports to come up with a final decision regarding each submitted sample's behavior. The results of these decisions are then sent to the database to update the black list, and to the end user's nominated email address. Summary of the analysis result sent to the user shall not only include whether the submitted file was a malware or not, but will also include information about how dangerous the file is in those cases in which it is malware. Information about how to deal with such files is also included to raise users' knowledge and awareness towards safely dealing with such dangerous files.

4. Implementation

In this section more implementation details will be presented and explained. It contains three main parts; the malware analyzer which utilizes cloud computing, the desktop submission tool, and the web submission interface.

4.1 Cloud Enabled Malware Analyzer

The main player in SCARECROW is the cloud enabled malware analyzer [24]. This component consists of two main parts of the malware analyzer and the cloud environment as shown in Figure 3. These components are combined together to deliver automated and scalable solution for malware reporting, detection and analysis.

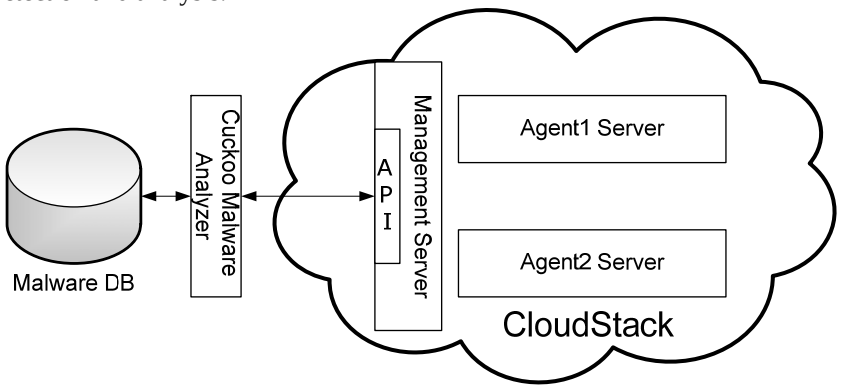

Figure 3. Cloud Enabled Malware Analyzer

4.1.1 Malware Analyzer

SCARECROW adapts Cuckoo SandBox as a malware analyzer. Cuckoo SandBox is an open-source malware analyzer that started as Google summer of code 2010 within the Honeynet project. In fact, Cuckoo SandBox was chosen from among a considerable number of malware analyzers to be adapted in SCARECROW, because it has some features that are required by SCARECROW.

These features are: availability of source code, customizability, and architecture appropriateness. The analyzer's source code should be available and organized to facilitate possible and successful integration with the other components. After that, there is the analyzer software customizability, which will ease adding new features and integrate easily with cloud environment. The final feature that is required by SCARECROW is the architecture appropriateness to cloud environment, which is highly needed by SCARECROW. All of these features gave Cuckoo the preference to be chosen compared with other malware analyzers.

Building and operating a malware analysis laboratory using Cuckoo is simple and straight forward. Cuckoo SandBox contains three main modules: core module, analysis module and finally submission module. SCARECROW gets most of Cuckoo features including analysis reports which can be exported to many formats.

The first module, which is the core module, is responsible for performing many functions. It begins by preparing and creating a pool of available virtual machines. After that, the module fetches the next malware from the waiting queue, which implemented as a database, and once there is an available VM in the pool, a new analysis task will be launched. Then, SCARECROW revokes a new VM and sends the malware sample to it and the analyzer module starts its job immediately in this VM. The analyzer module after finishing its work will send all log information back, which triggers the core module threat to continue working. Core module will call reporting functions to process log files received from analyzer module. At the end, copy of all analysis results are saved to a folder named with the task number.

The second module is the analyzer module, which starts when the VM already in running state and get the proper command from core module. It starts by checking the type of malware file, and then it runs the malware in a suspended state to allow injection of the monitor DLL. After the DLL get injected, the analyzer module resumes the malware file and waits until there is no more activity or waiting timer times out. Finally, this module will send back all files and logs to main server and terminates.

The third module called submission module, which is responsible mainly to submit malware files to the database. Submitting malware could be done either by programming method using submit class that designed and programmed to be called by any Python program, or by using the command line to invoke the Python interpreter with submission module. In SCARECROW submission process heavily depends on the desktop tool and the web site.

With standalone Cuckoo software, it is impossible to be run under cloud environment. Therefore, many modifications had been conducted, and new Python's modules had been written to make it possible for Cuckoo to run under CloudStack private cloud environment.

The core module is working based on main three files cuckoo.py, cloudstackapi.py, and db.py. First file contains the main source code of the core module, and it was altered to suit the new environment. The main changes to that file were done through adding some variables to track the current total number of deployed instances against the maximum possible instances that could be run in cloud environment, which defined in the configuration file. Other changes were done in analysis class, where new functions had been added while existed one had been changed. The added functions give the class ability to send malware to deployed instance and to run analyzer module on it, whereas modified functions reflect the new workflow besides dealing with cloud environment. The second file contains only two functions; one to deploy new instances whiles the second one to destroy an existed instance. The third file contains functions to write and read from the database. Some functions were altered to reflect the new scheme besides adding new function to set actual start time of the analysis task. All those changes had been tested many times to confirm its correctness.

Furthermore, the analyzer module, had only slight changes to reflect the new scenario as malware and configurations will be sent throw SSH despite a file share feature; which used by the standalone Cuckoo. The same alteration was done with submission module and the third module, as only changes were done to reflect the new database scheme.

4.1.2 Cloud Computing Environment

CloudStack is an open-source software written in Java and owned by Citrix systems and used to establish IAAS cloud service model. There are three editions of CloudStack: the Enterprise Edition, the Service Provider Edition, and the open-source Community Edition which is used to build the required cloud environment for this study.

Based on SCARECROW needs, the cloud environment should have two important requirements private and isolated environment. The private environment feature is needed to prevent exposing private information by suspicious software under analysis to public users. While the isolated cloud environment is important to prevent any damage that may cause by malicious software under analysis process. As a result, whatever the severity of analyzed malware is, there will be no harm to the real networks even if the malware try to escape from VM. Achieving full isolation can be reached by implementing local DNS server instead of using the global Internet.

The minimum components to build a CloudStack cloud are management server, agent server (host), primary storage, and secondary storage. These components could be duplicated for purpose of scaling. In addition, CloudStack cloud composes of multiple levels of infrastructure, which are: zone, pod, cluster, and host. Due to proof of concept implementation, only minimum components had been considered.

4.2 Desktop Submission Tool

As a part of SCARECROW system, there is a desktop submission tool. End users are expected to use this tool to submit any suspicious file before they download it from Internet. The purpose of implementing this tool is to enables users to enquire SCARECROW database without downloading the suspicious file and upload it again.

4.2.1 Tool Specification Requirements

To develop this tool, it is required to define the needed requirements and conditions that should be considered during the development and implementation, in order to control the behaviors and tasks of this tool. This tool is able to monitor the behavior of the internet browser during downloading executable files.

The system should behave seamless without annoying end-use application interaction. This tool provides the end user with three options during downloading executable files which are “Ok, Sure”, “No, Cancel”, and “Not sure, send to verify with feedback”. The reporting should include information about the file name, its download URL link, and the end user email for feedback of the file’s status.

4.2.2 Tool Implementation

Based on the requirements, to monitor the internet browser during downloading windows-based executable file, the implementation was done from the network interface level. The reason is to avoid using web browsers application programming interface (API) in order to make the software web browser agnostic. WinPcap was used to provide access to the network interface for the Windows operating system, which is an open source packet capturing library that is equivalent to LibPcap. If required, this design choice will enable easy desktop software porting to UNIX/Linux user’s computer.

In order to make the system to behave seamlessly, two main components were implemented, namely, background resident and HTTP analyzer. The former component was designed to run as a concurrent background resident application that is only seen in system tray, same as anti-virus software. Since monitoring software would consume CPU during operations, the system was designed to be suspended all time and active with low latency when web browsers has interactive session with end user. Therefore, CPU usage is limited. The system monitors HTTP protocol as the main stream communication between the web browser and web server. Other protocols such as FTP, SMTP were not considered at the moment for the system prototype, as they are least known to be used to transfer executable files by advertisers and vendors.

The latter component is HTTP analyzer that responsible for analyzing HTTP request and response. When user interacts with a web browser (e.g. Firefox, Google Chrome and Microsoft Internet Explorer), the system starts monitoring the TCP packets. These packets are analyzed to extract the embedded HTTP header of the request and response.

If the HTTP header has defined content-disposition type of attachment with filename that have extension of executable file, the system will launch a window interface that offers user with the three aforementioned options, which halts the screen for input/output until the end-users select an option. If user selects the third option (Not sure, send to verify with feedback), the window interface will send a request to expert's database server with information of such as filename, URL download, and user email.

The system is designed to provide a simple method of crowdsourcing data collection of downloading unknown Windows based executable files for the less technically inclined users.

4.3 Web Submission Interface

SCARECROW's web site implements the second submission mechanism that allows user to submit their files by uploading them directly through the web site to SCARECROW system. This submission mechanism had been implemented as an accessible web site, which running under Linux operating system. This Web site was implemented as a tool for transforming into practice the entire process of the web submission mechanism. Figure 4 shows the snapshot of the SCARECROW Web site.

The web site had been implemented using ASP.NET (Active Server Pages) technology, and the behind code was written using C# programming language based on Mono framework version 3.1. Mono is an open source development environment for implementing Microsoft's.NET framework to allow developers to develop cross-platform applications. For designing the graphical user interface, MonoDevelop version 2.8 was used as an IDE (Integrated Development Environment) which integrated with Mono to develop our Web site project as a cross-platform project. This Web site had been successfully integrated with the back-end application to provide harmonious interaction between users and SCARECROW system.

The main target function that is provided by SCARECROW Web site is to easily enable users to submit their suspicious file directly to SCARECROW system. Upon successful submission, a message will instantly appear to notify users regarding to their submission. Three types of messages that provided by the system, namely, 'the reported file has been recognized as malware free', 'the file is malware' or 'the file has been sent for further analysis please wait until it finish'. The first message will be displayed and sent to user if the submitted file was found in the white list. The second message will be displayed and sent to user if the submitted file was found in the black list. The third message will be displayed and sent to inform user that the submitted file had not been found in both white and black lists, thus it is sent to the malware analyzer for further analysis.

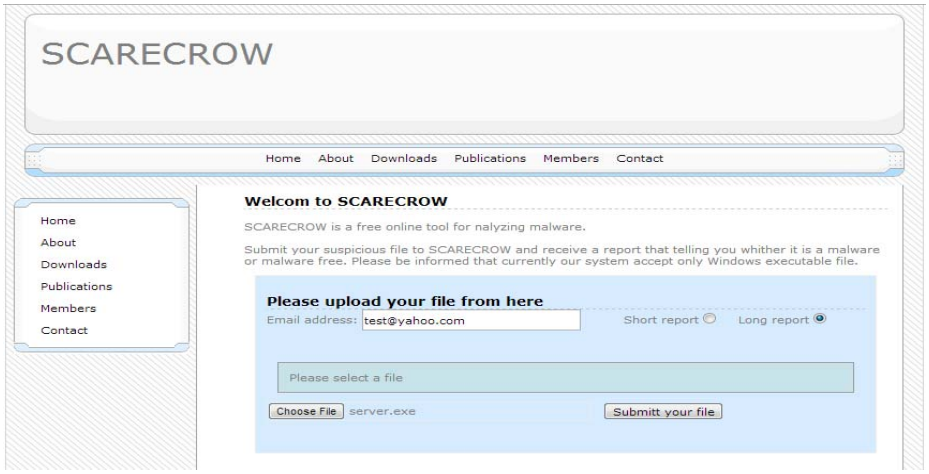

Figure 4. A Snapshot of SCARECROW Web Site

5. Discussion and Comparison

The SCARECROW system presents some noticeable advantages compared to normal anti-virus software. To reveal some of these advantages, Table 1 summarizes a comparison between SCARECROW and normal PC anti-virus based on certain relevant criteria.

Table 1: Comparison between normal anti-virus software and SCARECROW

	PC anti-virus	***SCARECROW***
Signature	Out dated malware signature database needs to be updated frequently with potential of disturbing PC's operation.	Signatures are kept up-to-date in the cloud by the CERT experts without the potential of disturbing PC.
False negative	High false negative since detection is based on outdated and limited malware signature database.	Low false negative since detection is based on comprehensive databases (blacklist and white list) and the potential of new malware detection.
Polymorphic malware	Polymorphic malware that can change its signature after downloaded to PC and can evade static analysis detection.	Polymorphic malware that changes its nature when running can be detected in the VM sandbox during dynamic analysis.
Overhead	Scanning and detection can slow down the host PC.	Does not slow down the PC since detection is performed in the cloud.
Malware fingerprinting	More difficult for expert to hunt the new malware in a timely manner.	Easy to identify new malware from suspecting normal users with the potential for "zero-day" malware detection.
User Decision Downloading EXE	Limited to "OK" or "Cancel" when downloading executable application limits users' options.	Users are given more informed decision with not only by "OK" or "Cancel", but also with "Not Sure, submit the file to cyber security laboratory" before the malware touches the users' computer.
CERT Reporting	Limited numbers of users are reporting malware.	More malware reporting by users.
Malware based Security Incident	Isolated and micro view of malware based security incident.	Centralized and macro/micro view of malware based security incident.

The malware signatures database is frequently updated by the SCARECROW system to keep the database up-to-date, and will be stored in a cloud computing environment. The reason behind storing the database in a cloud environment is to avoid encumbering the PC with a huge quantity of discovered malware signatures. In addition, searching and matching against such an immense database will provide more accurate results by limiting the number of false negatives, although it will take a longer time to report back to the user on the final decision when the initial categorization is "Unknown" rather than "Black" or "White".

From other perspectives, carrying out the analysis process in the cloud environment comes with many positive points. First, the analysis load will move from the user's PC to faster machines in the cloud environment thus no slowing down of user's computer and no power wastage of user's mobile device. Secondly, the static analysis databases are always up-to-date as opposed to out-of-date once downloaded malware database. Thirdly, the dynamic analysis process will be under the control of security experts thus all malware actions and activities will be recorded and analyzed. As a result, it will be easier for experts to find new malware and expose its signature to the world thus help reducing the number of malware in the wild. Finally, experts will gain more benefit when other users start to report the same newly discovered malware, allowing further investigation based on activity, infection and propagation of the new malware.

The user's third choice, cloud analysis beyond "OK" and "Cancel", takes some time to get back to the user with an appropriate decision, but guides unsophisticated users to download only safe

applications from the Internet. Therefore, SCARECROW system can empowers normal computer and mobile device users to report suspicious malware while at the same time increase the number of reported malware for security agency analysis. This symbiotic relationship between end users and security expert will eventually increase the probability of new malware detection and decrease the time spend by unknown malware in the wild.

6. Conclusion

Research in malware reporting, detection and analysis needs further efforts to reach the maturity level. Towards that end, this paper has introduced a new, ongoing approach called SCARECROW to advance malware analysis based on utilization of crowdsourcing. The aim of the proposed approach is to provide an environment where end users can submit and get a feedback about any suspected file they find on the Internet. The feedback sent to the end user includes not only detected or undetected malware, but includes summary of information regarding the detected malware in order to increase the end user's awareness. The SCARECROW reporting process revolves around three phases: submission/reporting, analysis, and feedback. In the submission/reporting phase, SCARECROW provides two simple options that allow users to choose whether to submit the suspicious files directly through the website or indirectly by submitting link through the installed client software. After that, the submitted files move on to the analysis phase, and a feedback based on the analysis is generated in the final phase.

Acknowledgments

The work presented in this paper was supported by MOSTI's (Ministry of Science, Technology and Innovation, Malaysia) fund.

6. References

[1] T. Velte, A. Velte, and R. Elsenpeter, *Cloud Computing, A Practical Approach*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2010.

[2] X. Wu, "MWfSCC: A Migrating Workflow System Based on Cloud Computing Paradigm," *Journal of Convergence Information Technology*, vol. 7, no. 23, pp. 537–546, Dec. 2012.

[3] J. Baliga and R. Ayre, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149 – 167, 2011.

[4] G. Egan, K. Haley, D. Mckinney, T. Millington, J. Mulcahy, T. Parsons, A. Watson, M. Nisbet, N. Johnston, and S. Hittel, "Internet Security Threat Report," 2012.

[5] S. Abraham and I. Chengalur-Smith, "An overview of social engineering malware: Trends, tactics, and implications," *Technology in Society*, vol. 32, no. 3, pp. 183–196, Aug. 2010.

[6] S. J. Hashim, K. Jumari, and M. Ismail, "Computer network intrusion detection software development," in *2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No.00CH37119)*, 2000, vol. 2, pp. 117–123.

[7] C. Lin, L. Bo, H. Huaping, and Z. Jing, "RkFinder: A Practical Anti-malware system Based on VMM," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 19, pp. 530–537, Oct. 2012.

[8] U. Bayer, C. Kruegel, and E. Kirda, "TTAnalyze: A Tool for Analyzing Malware," in *15th Annual Conference of the European Institute for Computer Antivirus Research (EICAR)*, 2006.

[9] "Anubis Malware Analyzer." [Online]. Available: http://anubis.iseclab.org/. [Accessed: 08-Jul-2013].

[10] "Cuckoo sandbox." [Online]. Available: http://cuckoobox.org/. [Accessed: 08-Jul-2013].

[11] "Norman sandBox." [Online]. Available: http://www.norman.com/security_center/security_tools/. [Accessed: 08-Jul-2013].

[12] T. Raffetseder, C. Kruegel, and E. Kirda, "Detecting System Emulators," in in *Lecture Notes in Computer Science*, vol. 4779, J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–18.

[13] X. Chen, J. Andersen, Z. M. Mao, M. Bailey, and J. Nazario, "Towards an Understanding of Anti-virtualization and Anti-Debugging Behavior in Modern Malware," in *2008 IEEE*

International Conference on Dependable Systems and Networks With FTCS and DCC (DSN), 2008, pp. 177–186.

[14] J. Van Randwyk, K. Chiang, L. Lloyd, and K. Vanderveen, "Farm: An automated malware analysis environment," in *2008 42nd Annual IEEE International Carnahan Conference on Security Technology*, 2008, pp. 321–325.

[15] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether," in *Proceedings of the 15th ACM conference on Computer and communications security - CCS '08*, 2008, pp. 51–62.

[16] A. M. Nguyen, N. Schear, H. Jung, A. Godiyal, S. T. King, and H. D. Nguyen, "MAVMM: Lightweight and Purpose Built VMM for Malware Analysis," in *2009 Annual Computer Security Applications Conference*, 2009, pp. 441–450.

[17] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[18] L. Martignoni, R. Paleari, and D. Bruschi, "A Framework for Behavior-Based Malware Analysis in The Cloud," in *5th International Conference, ICISS 2009*, 2009, vol. 5905, pp. 178–192.

[19] C. Adrian Martinez, G. Isaza Echeverri, and A. G. Castillo Sanz, "Malware Detection Based on Cloud Computing Integrating Intrusion Ontology Representation," in *2010 IEEE Latin-American Conference on Communications*, 2010, pp. 1–6.

[20] J. Oberheide, E. Cooke, and F. Jahanian, "CloudAV: N-version Antivirus in The Network Cloud," in *Proceedings of the 17th conference on Security symposium*, 2008, pp. 91–106.

[21] "Hadoop." [Online]. Available: http://hadoop.apache.org/. [Accessed: 27-Jul-2012].

[22] S.-T. Liu and Y.-M. Chen, "Retrospective Detection of Malware Attacks by Cloud Computing," in *2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2010, pp. 510–517.

[23] S. Shankland, "Google: 500 million Android devices activated," 2012. [Online]. Available: http://news.cnet.com/8301-1035_3-57510994-94/google-500-million-android-devices-activated/.

[24] O. L. Barakat, S. J. Hashim, R. S. A. Raja Abdullah, A. R. Ramli, F. Hashim, K. Samsudin and M. A. Rahman, "Malware analysis performance enhancement using cloud computing," *Computer Virology and Hacking Techniques*, June 2013.