# ATE: Anti-malware Technique Evaluator

Manuel Corregedor
Academy of Computer Science and Software Engineering
Corner Kingsway and University Road
Auckland Park, Johannesburg, South Africa, 2006
Email: mrcorregedor@uj.ac.za

Sebastiaan Von Solms
Academy of Computer Science and Software Engineering
Corner Kingsway and University Road
Auckland Park, Johannesburg, South Africa, 2006
Email: basievs@uj.ac.za

***Abstract*—Current commercial anti-malware products fail to guarantee a 100% detection and prevention of malware. This paper proposes an evaluation framework called ATE (Anti-malware Technique Evaluator) that can be used to evaluate commercial anti-malware products. ATE identifies the vulnerabilities in anti-malware products by providing a set of requirements that must be fulfilled by the anti-malware product being evaluated. The ATE requirements used for evaluating anti-malware products go beyond the usual false positives, false negative, performance etc requirements employed by current anti-malware product evaluations.**

**Keywords: malware, social engineering, rootkits, anti-malware evaluation**

## I. Introduction

The commercial products available to detect and prevent malware make use of one or both of the following approaches: Signature based detection and heuristic based detection [1]. Looking at the statistics in [2] we can see that although malware detection techniques are detecting and preventing malware, they do not guarantee a 100% detection and or prevention of malware. In order to better understand and / or determine why anti-malware products are not working we have created a framework called ATE (Anti-malware Technique Evaluator). We use ATE to evaluate current commercial anti-malware products used by the home user.

The remainder of this paper will be structured as follows: sections II-VI will discuss our ATE framework, section VII will provide the justification for ATE's layer weights, section VIII will briefly discuss the results of an ATE evaluation and lastly section IX will provide the conclusion and future work.

## II. Overview of ATE (Anti-malware Technique Evaluator)

Our ATE framework consists of four layers namely the User Layer, Dependability Layer, Accuracy Layer and the Easily Go Undetected (EGU) Layer. To evaluate a commercial anti-malware product X, the product X is evaluated against each of the four layers of ATE. Each layer of ATE consists of a number of components and each component has a number of requirements that an anti-malware product being evaluated is measured against. The requirements of the components each have a different score associated with them. The score for a requirement is awarded to an anti-malware product when it fulfills that specific requirement. The requirement scores of a respective component collectively add up to a total possible score of 100. The requirement scores are used to calculate that component's respective Component Score (CS).

The CS is calculated by adding up the scores attained by the anti-malware product for the requirements in that requirement's respective component and multiplying the result by the component's weight. The weights of each component in a layer add up to 1. Each layer also has a weight associated with it and all the layer weights add up to 1. The Layer Scores (LSs) are calculated by adding up all the component scores of the respective layer and multiplying the result with the respective layer's weight. The four layer scores are then summed up to give us a final score out of 100 for the anti-malware product being evaluated. A commercial anti-malware product is evaluated against each of the four layers of ATE by checking which requirements the anti-malware product fulfills in the respective ATE layer. We will discuss the User Layer of our ATE framework next.

## III. User Layer

The User Layer deals specifically with aspects regarding the user of the anti-malware product. As illustrated in Fig.1, we look at two components in this layer namely the Susceptibility to Social Engineering and the Susceptibility to Human Error components. Each of these components contain 2 requirements. It should be noted that the social engineering and human error components of the User Layer are not usually considered as being a part of anti-malware product evaluations. This can be seen in the evaluations performed by organisations such as AV-TEST [3] and AV-Comparatives [4]. Furthermore from our literature study and research over the past 2 years we found no other evaluation framework and / or method that considers human error and social engineering as part of the anti-malware product evaluation.

This layer consists of two component scores namely CS1 and CS2. The weights for both components are 0.5. We will now discuss each component in turn starting with the Susceptibility to Social Engineering component.

### A. Susceptibility to Social Engineering

This is a very important component to consider with regards to how effective an anti-malware product is. The reason for this is that the user in most cases is the one who decides whether or not they want to execute a file be

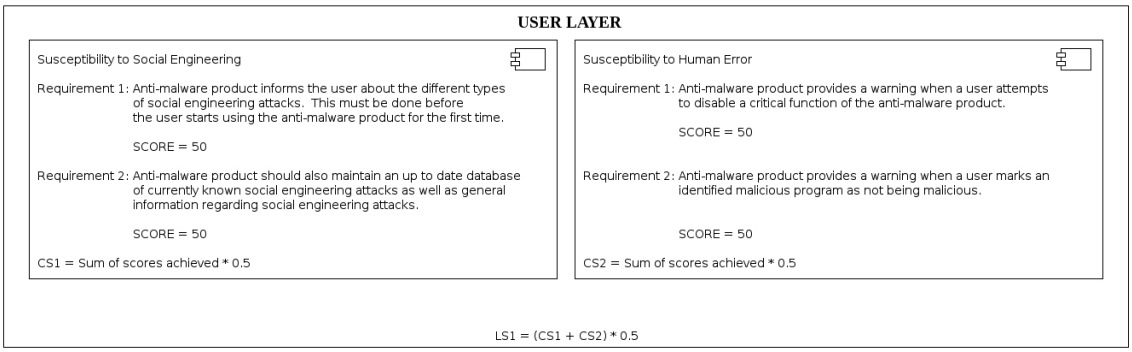


Fig. 1. User Layer of ATE framework [By author]

it malicious or not. This may be due to the user believing that the program is useful or that it comes from a trusted source. The report in [2] identifies social engineering as the preferred method to spread malware. Although many users are aware of social engineering attacks by means of e-mail [2], the majority of users are unaware of the dangers of executing files from unknown sources. The two requirements of this component are listed in Fig.1 and focus specifically on reducing the chances of a social engineering attack being successful. The two requirements of this component will be discussed next.

*1) Susceptibility to Social Engineering Requirements:* An anti-malware product will fulfill the first requirement if, when it runs for the first time after installation, it informs the user about the different types of social engineering attacks. An anti-malware product will fulfill the second requirement if it maintains an up to date database with information regarding social engineering attacks. The database must be easily accessible to the user of the anti-malware product. The next section will discuss the second component of the User Layer namely the Susceptibility to Human Error component.

### B. Susceptibility to Human Error

Human error and lack of understanding play a role in the effectiveness of the anti-malware product. This is because the user may choose to disable a feature provided by the anti-malware product due to failing to understand the feature or because the feature affects the performance of the computer. A user may also decide to ignore a program once it has been detected as being malicious because of fear or lack of knowledge of which action should be taken [5]. This could result in the PC being infected with malware. The two requirements of this component are listed in Fig.1 and focus specifically on reducing the chances of a human error occuring. Additional information regarding the two requirements will be given next.

*1) Susceptibility to Human Error Requirements:* The anti-malware product will fulfill the first requirement if it provides a warning message when a user attempts to disable one of its critical functions. The warning message must contain an explanation of what the function provides and the consequences of disabling the function in order to encourage the user to leave the function on. The explanation should be easily understood by all users.

In order to fulfill the second requirement, the anti-malware product must strongly advise the user not to ignore a program that has been identified as being malicious unless they are a 100% sure. The anti-malware product must display a percentage regarding the certainty that the file is malicious. This will help the user in making the correct decision. It should be noted that the anti-malware product will be considered as not meeting requirement 2 should it not allow the user to mark a program as not being malicious. The prior is important because anti-malware products can have false positives. A false positive could result in more problems should the anti-malware product not allow the user to mark a program as not being malicious. We have now discussed the two components of our User Layer that is part of our ATE framework. The next section will provide the justification for the chosen requirement scores as well as the chosen weights of the components i.e. how these components will be scored.

### C. Justification for chosen requirement scores and component weights

The User Layer of ATE deals specifically with human error as well as social engineering. Both of these components are closely related as social engineering in some aspects relies on there being a human error in judgment whereas human error can also be brought about due to social engineering. Therefore each component score, namely CS1 and CS2, are given an equal weight of 0.5. The requirements in each component are also of equal importance and have hence been given a score of 50 each. This is due to the fact that should one requirement not be fulfilled a human error or social engineering attack may occur and / or be exploited. This concludes the discussion about the User Layer of our ATE framework. The next section will discuss the Dependability Layer of our ATE framework.

## IV. Dependability Layer

This layer looks at any dependency that the anti-malware product might have on some or other resource in order to work

correctly. As illustrated in Fig.2, we look at one component in this layer namely the Dependency on the Operating System (OS) to provide information component. The component only contains 1 requirement. This layer consists of one component score denoted as CS1. The weight for the component is therefore 1.

Lastly it should be noted that the Dependability Layer checks for a requirement which is not usually considered or taken into account when organisations such as AV-TEST [3] and AV-Comparatives [4] perform anti-malware product evaluations. Furthermore according to our knowledge and from our extensive literature study no other framework or method exists that evaluates anti-malware products by checking for such requirements. The next section will discuss the only component of this layer, namely the Dependency on Operating System to Provide Information component.

### A. Dependency on Operating System to Provide Information

In order for an anti-malware product to perform a scan and / or to monitor for malware it requires certain data such as the list of processes currently running, the list of files in directories, access to memory etc. This data is provided by the OS through application programming interfaces (APIs). However as can be seen in [6] this data cannot always be trusted due to the existence of malware but more specifically rootkits that can change the data being requested in order for the malware to hide itself or other malware. Furthermore this component was included as a result of our work done in [7], where we found that should a rootkit compromise a system you cannot trust the OS. The requirement of this component is listed in Fig 2. Additional information for the requirement will be given next.

*1) Dependency on Operating System to Provide Information Requirement:* Our Evader Rootkit prototype from our previous work in [7] will be used to check this requirement by hiding files on disk from the anti-malware product while it performs a scan. The next section will provide the justification for the chosen requirement scores and component weights.

### B. Justification for chosen scores and weights

The Dependability Layer contains only one component which has only one requirement. Therefore the chosen score for the requirement was a 100 and the chosen weight for the component was 1.

This concludes the discussion about the Dependability Layer of our ATE framework. The next section will discuss the Accuracy Layer of our ATE framework.

## V. Accuracy Layer

The Accuracy Layer is used to evaluate the anti-malware product's accuracy. As illustrated in Fig.3 there is only one component in this layer namely the Accuracy component which has three requirements. This layer consists of one component score denoted as CS1. The weight for the component is therefore 1. We will now discuss the Accuracy component in more detail.

### A. Accuracy

An anti-malware product should not produce any false positives or false negatives. This is becoming more challenging to do because malware authors started using techniques to generate metamorphic or polymorphic malware [8]. In order to deal with this problem, heuristic based techniques allow a user to control the sensitivity of the technique by adjusting a detection threshold. However raising the sensitivity could cause benign programs to be detected as malicious, which is an example of a false positive. Lowering the sensitivity could result in malicious programs not being detected which is an example of a false negative. The requirements of this component are listed in Fig.3. Additional information (not shown in Fig.3) for the requirements will be given next.

*1) Accuracy Requirements:* For the first requirement, false negatives can be checked by making use of malware collections such as those provided by VX Heavens [9] and Offensive Computing [10]. Should the anti-malware product fail to detect any of the samples used then a false negative would have occurred.

Checking the second requirement is important because a high amount of false positives will result in a user loosing faith in the product and / or stop using it [11]. False positives are checked by scanning benign samples. If any file is detected as being malicious then a false positive would have occurred. This may seem to be an overly strict requirement, however the reader needs to keep in mind that the goal of ATE is to identify why commercial anti-malware products are not working. The prior is achieved by identifying the vulnerabilities that exist in the products. One such vulnerability is the possibility of a false positive occuring i.e. a single false positive could result in significant damage occuring, therefore an anti-malware product should stive to achieve such a requirement.

The third requirement differs from the requirements of the Susceptibility to Human Error component because in this case the anti-malware product is not allowing the user to disable a critical function but it is instead allowing the user to reduce its accuracy and / or effectiveness. To fulfill this requirement the anti-malware product must warn the user of the dangers of lowering or increasing the accuracy of the anti-malware product. Furthermore help should be available to assist the user in choosing the best setting for him/her.

We have now discussed the Accuracy component of our Accuracy Layer that is part of our ATE framework. The next section will provide the justification for the chosen requirement scores and component weights.

### B. Justification for chosen scores and weight

The requirements regarding false negatives and false positives, requirements 1 and 2 respectively can be initially considered as contributing a score of 25 each and a combined score of 50 for the Accuracy Layer. The reason for this is that a user will loose faith in a product should a false positive or false negative occur. However looking at the detection rates and false positive results of anti-malware products evaluated

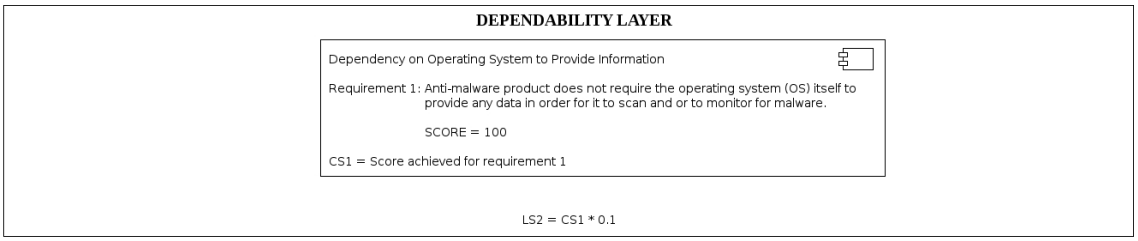


Fig. 2. Dependability Layer of ATE framework [By author]

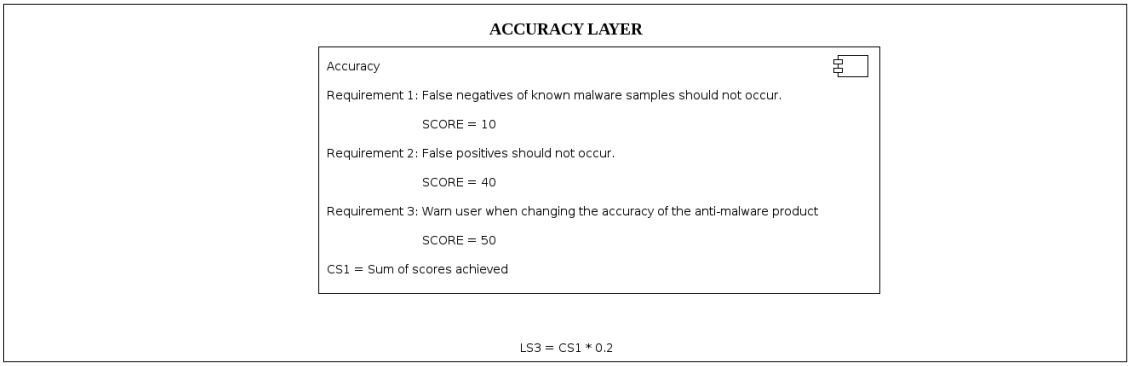


Fig. 3. Accuracy Layer of ATE framework [By author]

in [12], we can see that the average detection rate of the anti-malware products evaluated in [12] is 96%. This indicates that on average 4% of the scanned malware resulted in false negatives. Also the average percentage of false positives was 19%. Therefore the difference in percentage between 4% of false negatives and 19% of false positives is 15%. This indicates that false positives are 15% higher than false negatives which makes false positives more of a greater concern than false negatives. Therefore the initial scores assigned were adjusted as follows: the false negatives requirement from 25 to 10 and the false positives requirement score from 25 to 40.

The remaining score of 50 is assigned to the third requirement i.e. warning the user when the accuracy of the anti-malware product is changed. This is important because should a user lower the accuracy then false negatives would occur whereas raising the accuracy could result in more false positives. Lastly the chosen component weight was 1 because there is only one component in this layer. The next section will discuss the last layer of our ATE framework namely the EGU (Easily Go Undetected) Layer.

## VI. EGU (Easily Go Undetected) Layer

We have introduced this layer to address the toughest problems faced by anti-malware products namely zero day malware, privilege level escalation, rootkits, metamorphic and polymorphic malware [1], [13]. It should be noted, as in the case of the User Layer and Dependability Layer, that the EGU Layer checks for requirements which are not usually considered or taken into account when organisations such as AV-TEST [3] and AV-Comparatives [4] perform anti-malware product evaluations. Furthermore according to our knowledge and from our extensive literature study no other framework or method exists that evaluates anti-malware products by checking for such requirements.

The EGU Layer makes use of two rootkits that we developed in our previous work [7] to check for the requirements of this layer. In our previous work we named the two rootkits The Sabotager and The Evader respectively. Recall that our Evader rootkit is also used to check a requirement of the Dependability Layer. We will provide a brief summary of our rootkits in the next section.

### A. The Sabotager and The Evader Rootkits

*1) The Sabotager Rootkit:* The Sabotager Rootkit can successfully sabotage 32 bit Windows XP or 32 bit Windows 7 OS by preventing it from booting. It also deletes previous restore points created by Microsoft's System Restore utility and creates a restore point after its installation is complete. This ensures that the user will not be able to easily recover from the bug check (blue screen) causing the computer not to boot. We have packaged the rootkit as an installer for a game. During the installation a message is displayed notifying the user that the installation failed. The message also displays a link to some site where the user is lead to believe they can get help on the error but which could instead be a malicious site.

*2) The Evader Rootkit:* The Evader Rootkit, as with The Sabotager Rootkit, makes use of social engineering in order to be installed i.e. it is packaged as an installer for a game. The main goal of The Evader rootkit is to disable any anti-malware products running on the computer. The Evader Rootkit also

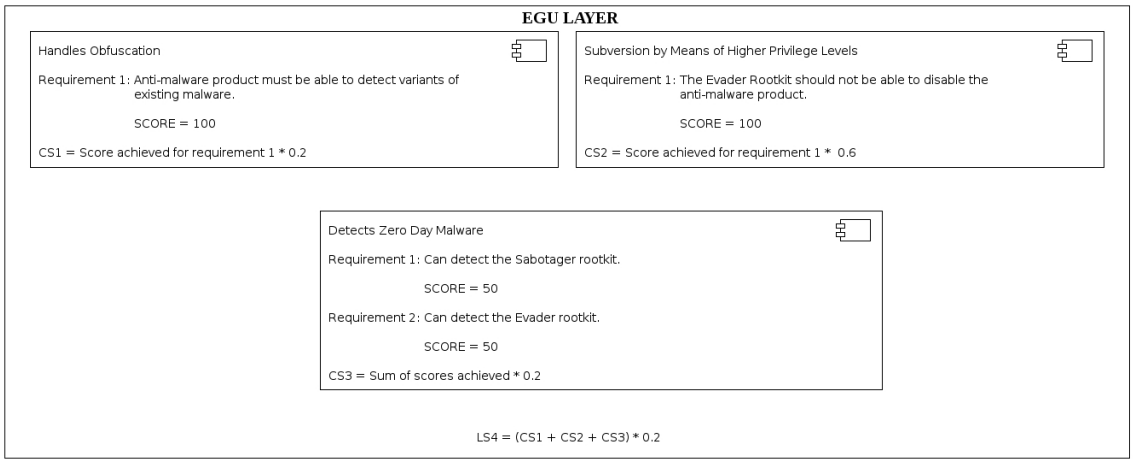


Fig. 4. EGU Layer of ATE framework [By author]

logs any key pressed and stores them in a text file. The Evader rootkit is also able to hide any file on disk. This concludes the discussion of our two rootkits used in the EGU Layer. We will now discuss each component of the EGU Layer (see Fig.4) in turn starting with the Detects Zero Day Malware component.

### *B. Detects Zero Day Malware*

The previous section briefly discussed the rootkits we developed in our previous work in [7]. The rootkits were developed in order to assist us in the evaluation of the requirements for the EGU Layer. This section will discuss the first component of our EGU Layer that addresses zero day malware. Zero day malware refers to an unknown malware instance that has been released into the wild. Signature based techniques are significantly impacted by zero day malware because they require the signature of the malware instance in order to be able to detect it [13]. A signature however is only extracted after a large amount of computers have been infected by the malware. Other techniques such as the heuristic based techniques are also unable to deal with all zero day malware. The two requirements of this component are listed in Fig.4 and discussed next.

*1) Detects Zero Day Malware Requirements.:* The Sabotager and The Evader rootkits are currently not wide spread and no specific signature exists for them. Therefore according to the definition of zero-day malware the Sabotager and Evader rootkits are zero-day malware. Therefore in order for an anti-malware product to fulfill the first and second requirement it must be able to detect the Sabotager rootkit and Evader rootkit respectively. The next section will discuss the Subversion by Means of Higher Privilege Levels component of the EGU layer.

### *C. Subversion by Means of Higher Privilege Levels*

User mode applications cannot perform any instruction directly that affects the control of the machine or does Input/Output, for example reading hard drives. All such instructions must go through the OS [14]. The only other software that is allowed to run in kernel mode are device drivers, also known as kernel-mode drivers. Kernel-mode drivers have the same privileges as the OS itself i.e. they can do anything the OS can do [6], [14]. This means that malware that makes use of a kernel-mode driver has the same privileges as the OS allowing it essentially to do whatever the OS can do. This includes but is not limited to: hiding itself by manipulating the list of running processes, concealing files on a hard drive etc [6]. Our Evader Rootkit runs as a kernel-mode driver and therefore has such capabilities. Particularly it can disable anti-malware products and hide files on disk. Therefore the Evader rootkit is used to test for the requirement of this component. It should also be noted that anti-malware products also make use of kernel-mode drivers to scan for malware [6]. The requirement of this component is listed in Fig.4 and additional information regarding the requirement is provided next.

*1) Subversion by Means of Higher Privilege Levels Requirement:* Our Evader Rootkit will attempt to disable the anti-malware product. If the anti-malware product is not disabled by our Evader rootkit then it will have fulfilled the requirement. The next section will discuss the Handles Obfuscation component of the EGU layer.

### *D. Handles Obfuscation*

Malware authors have started to resort to sophisticated hiding techniques based on code obfuscation [13]. Code obfuscation is the process of applying semantic preserving transformations to a program such that it becomes unintelligible but still maintains similar behaviour [15]. Therefore malware authors can change the malware such that it results in a different signature being produced but leaves the semantics or rather the malicious behaviour of the malware the same. There are many different ways to obfuscate code as highlighted in [13].

Metamorphic malware makes use of code obfuscation in

order to change its code structure in such a way that very little bytes remain that can be used as a signature [8]. Polymorphic malware encrypts its payload using different keys each time to make it undetectable. Polymorphic malware however can be identified by the signature of its decryptor [8]. The polymorphic malware therefore makes use of metamorphic decryptors to avoid such detection. The requirement of this component is listed in Fig.4 and additional information regarding the requirement is provided next.

*1) Handles Obfuscation Requirement:* In order to check for the requirement of this component we obfuscated a malware instance using PEScrambler, which is a tool that is used to obfuscate Win32 binaries automatically [16]. The anti-malware product fulfills the requirement if it successfully detects the obfuscated malware instance. We have now discussed the EGU Layer. The next section will provide the justification for the chosen requirement scores and component weights.

### *E. Justification for chosen scores and weights*

The EGU Layer consists of three components. The Handles Obfuscation component only has one requirement and as such it is given a score of 100. The Detects Zero Day Malware component has two requirements of which one addresses the detection of the Evader rootkit and the other the detection of the Sabotager rootkit. It was decided to give each requirement a score of 50 each because both rootkits are equally representative of zero day malware. The Subversion by Means of Higher Privilege levels component only has one requirement and as such it is given a score of 100.

The Subversion by Means of Higher Privilege levels component was given a component weight of 0.6 and the remaining two components, namely the Detects Zero Day Malware component and the Handles Obfuscation component, were given a component weight of 0.2 each. The rationale for this is that anti-malware products are already extensively tested for being able to detect zero day malware and handling obfuscation. Lastly it is more important to ensure that the anti-malware product cannot be disabled because if it can then being able to handle obfuscation and detect zero day malware is a mute point i.e. irrelevant. We have now looked at all the layers of our ATE framework. The next section will discuss the allocation of the layer weights.

## VII. Justification of the layer weights

The following weights were assigned to each layer:

1) User Layer a weight of 0.5
2) Dependability Layer a weight of 0.1
3) Accuracy Layer a weight of 0.2
4) EGU Layer a weight of 0.2

The Dependability Layer, Accuracy Layer and EGU Layer all deal with evaluating the anti-malware's technical aspects whereas the User Layer is not directly related to the anti-malware product per se. The statistics in reports such as [2], [17] show that there has been an increasing trend in attackers making use of social engineering in order to spread malware, to execute malware or both. This is especially the case with rogue security software also known as scare ware. Therefore in order for an anti-malware product to be successful it must not only address the technical aspects i.e. malware related but also the user aspects that affect the usefulness of the product. The prior is the reason why the User Layer has been given a weight of 0.5 and the remaining more technical layers have a collective weight of 0.5. Also as pointed out in [18] the total weight for malware related categories should not exceed 50% because there are too many other things that make anti-malware products good or bad.

The Accuracy and EGU Layers are collectively more important than the Dependability Layer due to the fact that they are both related to how effectively an anti-malware product will be able to detect and / or prevent malware. Furthermore as pointed out if such requirements are not met the user will loose faith in the anti-malware product. Therefore the Accuracy Layer and EGU Layer are each allocated a weight of 0.20. The remaining weight of 0.1 is assigned to the Dependability Layer.

This concludes the discussion of our proposed evaluation framework called ATE (Anti-malware Technique Evaluator) that can be used to evaluate current anti-malware products but more importantly help to better understand and / or determine why anti-malware products are not working. The next setion will discuss the results of an ATE evaluation.

## VIII. Results of an ATE Evaluation

ATE has been implemented and 9 commercial anti-malware products were evaluated. The products evaluated were: Avira Free Antivirus, avast! Internet Security 6, ESET Smart Security 5, AVG Anti-Virus Free Edition 2012, McAfee AntiVirus Plus, Microsoft Security Essentials, Ad-Aware Free Internet Security, Kaspersky Anti-Virus 2012 and Norton AntiVirus 2012. All of the anti-malware products were disabled during the evaluation except for Norton. Furthermore the evaluation showed that the majority of the products evaluated were dependent on the OS and susceptible to social engineering and human error. Due to the limited space in this paper we can not discuss all of the results in depth nor the procedure used to do the evaluation. It should however be noted that the ATE evaluation procedure followed adhered to the aspects that the Anti-Malware Testing Standards Organization (AMTSO) recommends when performing whole-product testing. AMTSO is a non-profit organisation that was established to ensure that anti-malware programs are tested objectively [19]. Lastly in order to better understand and to visually see the ATE evaluation process, we encourage the reader to go to http://adam.uj.ac.za/ATE. This website contains the videos of all of the anti-malware product ATE evaluations that we have performed. The evaluation videos are stored under each anti-malware product's respective directory and are named according to the respective layer evaluated. Alternatively if the videos are not available at the aforementioned address, you may contact us using the contact details on this paper to request the original videos. The next section will discuss the results of our User Layer.

### A. User Layer Results

The User Layer results showed that four out of the nine anti-malware products evaluated, namely Avira, AVG, Microsoft Security Essentials and Ad-Aware all got a final layer score of zero. The remaining anti-malware products namely: avast!, ESET, McAfee, Kaspersky and Norton met the first requirement of the Susceptibility to human error i.e. to provide a warning when a user attempts to disable a critical function of the technique. As such they all got a final score of 12.5 out of a possible 50. The results showed that anti-malware vendors are not putting enough effort into reducing human error and social engineering which lead to the spread of malicious software. This concludes our discussion regarding the User Layer results, the next section will discuss the Accuracy Layer Results.

### B. Accuracy Layer Results

The Accuracy layer evaluation was performed with all updates installed for the anti-malware products and the OS up until the 27th of November 2011. The results showed that all of the anti-malware products were able to fulfill the false positives requirement except for AVG. The reason AVG failed was because it detected a Windows XP notepad.exe file, that we obfuscated, as being malicious. None of anti-malware products fulfilled requirement 3 as they failed to generate a warning when we reduced their respective settings. This once again, as in the case of the User Layer, leaves an opportunity for human error to occur which would result in malware spreading. However the most concerning result is that none of the anti-malware products fulfilled the false negatives requirement. We only went as far as finding one false negative per anti-malware product i.e. once the false negative was found the evaluation was stopped. The results have shown once again that the anti-malware products are susceptible to human error. However more importantly the results in this section have proven that anti-malware products are not able to detect all samples that are considered to be malicious. The next section will discuss the EGU layer results.

### C. EGU Layer Results

The evaluation of the EGU layer's Detects Zero Day Malware component and Handles Obfuscation component were performed with all updates installed for the anti-malware products and the OS up until the 28th of November 2011. The Subversion by Means of Higher Privilege Levels component was evaluated with all updates installed for the anti-malware products and the OS up until the 29th of November 2011. The results showed that all the anti-malware products were able to detect our obfuscated malware. However it should be noted that this does not guarantee that the anti-malware products will be able to detect all obfuscated malware. The results also showed that Avira was the only anti-malware product that was able to detect our Evader rootkit by making use of a generic detection routine. Whereas no anti-malware product was able to detect our Sabotager rootkit. All of the anti-malware products were disabled during the evaluation by our Evader rootkit except for Norton. Also we found that Kaspersky blocked the installation of our Sabotager and Evader rootkits by using a heuristic pattern based on the fact, from what we can deduce, that our rootkits were hidden files and were drivers being installed. Kaspersky however allowed us to give our installer permission to install the rootkits which once again leaves an opportunity for human error to occur. Lastly recall that Avira was able to detect our Evader rootkit. Therefore in order to install our rootkits and disable Avira we had to add several exceptions into Avira for our rootkit files. Alternatively we could have changed the extension of our Evader and Sabotager rootkits from .SYS to something else which would have resulted in Avira not automatically detecting our rootkit. The next section will discuss the Dependability Layer results.

### D. Dependability Layer Results

The results of this layer are of a very high concern as only two out of the nine anti-malware products evaluated passed the dependability test. The two anti-malware products that passed the test were ESET and Norton. This is of a high concern because it means that malware can be hidden in the file system in such a way that the other anti-malware products will not know it's there. This vulnerability is due to the fact the anti-malware products are using the OS to get the list of files to scan. The next section will look at the final scores for the anti-malware programs.

### E. Final Results

The final results showed that Norton performed significantly better than all the other anti-malware products. This can be attributed to its EGU layer results. On the other hand AVG was the anti-malware product that performed the worst, which can be attributed to AVG failing the false positive requirement of the Accuracy layer. However it should be noted that there is no winner or better product. The results instead show that all anti-malware products have failed because none of them got a score of a 100 which means they all have some vulnerability that can be exploited by malware. Therefore we can see it as all anti-malware products failing but some failing less than the others. Recall that the purpose of the ATE evaluation is not to find the best anti-malware product but instead to evaluate each product and discover its vulnerabilities. This however does not mean that the final results cannot be used or have no value. The final results of several ATE evaluations can be used to compare and / or to check if anti-malware products are indeed improving over time. The next section will look at the additional results which were gathered during the ATE evaluation.

### F. VirusTotal Results

In addition to our evaluation results we also scanned our rootkits using www.virustotal.com for the first time on the 30th of November 2011. Virustotal is a service that analyses suspicious files and URLs using several anti-malware engines. It should be noted that VirusTotal is a free independent service.

For the first test we submitted our entire installation archive i.e. installer and rootkits. The detection rate for the first scan was 14.6%. For the second scan we submitted only the Evader rootkit which resulted in a detection rate of only 2.3%. The Avira anti-virus engine was the one that detected the Evader rootkit. This coincides with our results from the ATE evaluations. For the third scan we submitted our Sabotager rootkit which was not detected as malicious by any anti-virus engine. For the fourth scan we submitted our rootkit installer executable which resulted in a detection rate of 14%. This was an interesting result which lead us to create a batch file that performs the installation of the rootkits. We then submitted the installation batch file to VirusTotal which resulted in a 0% detection rate. Therefore our batch file is performing the exact same malicious activity as the executable installer but is not being detected as malicious. The next section will provide the conclusion and future work.

## IX. Conclusion and Future Work

This paper has discussed our proposed evaluation framework called ATE (Anti-malware Technique Evaluator) that can be used to evaluate current anti-malware products but more importantly help to better understand and / or determine why anti-malware products are not working. The framework looks at different requirements that anti-malware products can be measured against. An anti-malware product is evaluated against each layer of the ATE framework. The layers consist of components which in turn consist of requirements that an anti-malware product should have. Should the anti-malware product fulfill i.e. have the requirement it will be given that requirement's score otherwise it gets zero. The scores are summed up and adjusted according to the component and layer weights. A final score is then calculated for the anti-malware product being evaluated.

The results of an ATE evaluation were also briefly discussed. All of the anti-malware programs evaluated using ATE all had a final score of less than 50 out of 100 indicating that all of the anti-malware programs have several vulnerabilities that need to be addressed. The User Layer of our ATE framework is one of the layers where most of the anti-malware products got their lowest scores. We therefore encourage anti-malware developers to take more responsibility in educating the home user about malware and how to make well informed decisions when confronted with possible malware. This can be achieved because anti-malware companies are already exchanging malware samples with each other weekly, daily or more than once a day essentially using live feeds [20] and such could also exchange social engineering attack data. The other layer where low scores were attained by the anti-malware products was the EGU Layer. This layer demonstrated how vulnerable an anti-malware product was against a direct attack and zero-day malware. All of the anti-malware products evaluated were disabled except for Norton.

The results of the Dependability layer showed that only two out of the nine anti-malware products evaluated passed the dependability test. This is of a high concern because it means that malware can be hidden in the file system in such a way that the anti-malware product will not be able to scan it. Lastly the results of the Accuracy layer were very worrying as they showed that none of the anti-malware products evaluated were able to detect all of the malware samples used. This indicates that there is a lack of collaboration between anti-malware vendors which is most certainly due to competition between them. If collaboration had occurred between the anti-malware vendors then it would have been possible that all the anti-malware products would have passed the false negative test. Future work will be to find a better way of generating malware variants ,so as not to rely on existing tools, and performing more ATE evaluations in order to have a comparative analysis of results for the same products over time. Lastly a tool will be created to perform the evaluations automatically.

## References


[1] Yin, H., Song, D.: Panorama: capturing system-wide information flow for malware detection and analysis. In: Proceedings of the 14th ACM conference on Computer and communications security,pp.116–127. ACM, New York (October 2007)

[2] Volume 9 of the Microsoft Security Intelligence Report (SIR). http://www.microsoft.com/security/sir/default.aspx, (Referenced on 12 November 2010)

[3] AV-Test. http://www.av-test.org/, (Referenced on 31 October 2011)

[4] AV-Comparatives. http://www.av-comparatives.org/, (Referenced on 31 October 2011)

[5] Morales, J., Sandhu, R., Shouhuai, X.: Evaluating Detection and Treatment Effectiveness of Commercial Anti-malware Programs. In: Proceedings of the 5th International Conference on Malicious and Unwanted Software (MALWARE),pp. 31–38. IEEE, (2010)

[6] Blunden, B.: The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System. In: Wordware Publishing, Inc., Texas (2009)

[7] Corregedor, M., Von Solms, S.: Implementing Rootkits to Address Operating System Vulnerabilities. In: Information Security South Africa 2011 (ISSA 2011),pp. 1 – 8. IEEE, Johannesburg (August 2011)

[8] Leder, F., Steinbock, B., Martini, P.: Classification and detection of metamorphic malware using value set analysis. In: 2009 4th International Conference on Malicious and Unwanted Software (MALWARE),pp. 39–46. IEEE, New York (October 2009)

[9] VX Heavens. http://vx.netlux.org/, (Referenced on 15 February 2011)

[10] Offensive Computing. http://www.offensivecomputing.net/, (Referenced on 1 March 2011)

[11] Christodorescu, M., Jha, S.: Testing malware detectors. In: Proceeedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis,pp. 34–44. ACM, New York (2004)

[12] AV-Comparatives: On-demand Detection of Malicious Software August 2011. http://www.av-comparatives.org/images/stories/test/ondret/avc_od_aug2011.pdf, (Referenced on 27 September 2011)

[13] Dalla Preda, M., Christodorescu, M., Jha, S., Debray, S.: A semantics-based approach to malware detection. In: ACM Transactions on Programming Languages and Systems,Vol.30 ,. ACM, New York (August 2008)

[14] TanenBaum, A., Woodhull, A.: Operating Systems Design and Implementation. In: ,Chapter 1. , New Jersey (2006)

[15] Lee, J., Jeong, K., Lee, H.: Detecting Metamorphic Malwares using Code Graphs. In: Proceedings of the 2010 ACM Symposium on Applied Computing,pp. 1970–1977. ACM, New York (March 2010)

[16] pescrambler. http://code.google.com/p/pescrambler/, (Referenced on 18 November 2011)

[17] Volume 11 of the Microsoft Security Intelligence Report (SIR). http://www.microsoft.com/security/sir, (Referenced on 31 October 2011)

[18] Marx, A.: Guideline to anti-malware software testing. In: In European Institute for Computer Anti-Virus Re-search (EICAR) 2000 Best Paper Proceedings,pp. 218–253. , (2000)

[19] AMTSO - Anti-Malware Testing Standards Organization. http://www.amtso.org/index.html, (Referenced on 13 October 2011)

[20] Muttik, I., Vignoles, J.: "Rebuilding anti-malware testing for the future". In Virus Bulletin Conference, (2008)