Middle-East Journal of Scientific Research 17 (4): 455-464, 2013
ISSN 1990-9233
© IDOSI Publications, 2013
DOI: 10.5829/idosi.mejsr.2013.17.04.12159

Protecting from Zero-Day Malware Attacks

Abid Shahzad, Mureed Hussain and Muhammad Naeem Ahmed Khan

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST),
H-8/4, Islamabad 44000, Pakistan

Abstract: The menace of malware is becoming more harmful and ominous for the enterprises as well as the home users. The malware attacks usually cause users to render their critical data in the hands of nefarious persons. Shielding against the malware attacks seems a challenging job for IT administrators. The common solutions that provide protection against malware are known as signature based anti-malware solution. These solutions works on the blacklisting technique which seems unsuccessful when it comes to sophisticated zero-day malware attacks. However, a newly emerging technique, the whitelisting provides best protection against zero-day malware attacks by only allowing legitimate services, processes, applications and websites to run on the machine. The whitelisting technique maintains list of the trusted applications and allows only these listed executables to execute only while preventing all other threats to be executed. In this paper we proposed a light weight zero-day anti-malware solution. The solution uses whitelisting and also accounts for specific advantages of blacklisting technique. The validation of the proposed solution proves effectiveness and efficiency. It requires low CPU and memory usage and does not require bandwidth or Internet. In short, it is simple and light weight as compared to signature based anti-malware solutions.

Key words: Malware Analysis · Signature-based Technique · Blacklisting · Whitelisting · Zero-day Malware

INTRODUCTION

Malware threats are growing day by day at a rapid pace. Most of the malware exploit the vulnerable entry points of any computer network and abuse these network weaknesses to achieve their goals by stealing the critical information. Over the past many years, different measures are in use to provide protections against these malware attacks. Therefore, different levels of security mechanisms and technologies are in use at network, application and host levels. It seems a serious challenge for the organizations is to keep data confidentially, availability and integrity intact by averting the modern malware attacks.

The recent sophisticated malware attacks resulted in data theft and information loss for many organizations and home users resulting in serious financial loss. One of the most recent malware attacks are the blended attacks [1]. The blended attacks are launched by professional hackers which always have some sort of malicious motive behind the attack. These attackers target the users and send them an attractive email or link. As soon as the user clicks the link or accesses the email, a PDF or Word document opens on the user machine. Actually these types of documents contain malware payload which executes on the user machines immediately when user tries to open them. Afterwards, the malware resides on the user's machine and steals the required information from that machine or uses that machine as bot machine to capture the information from the network. Stuxnet is one such example of sophisticated malware, which was launched to attack different high profile organizations to steal the valuable information and cause data damages.

Malware are also an enormous huge threat to web based applications and services. Computing future is linked with cloud computing driven by web 3.0, but impending malware threats is very alarming which can jeopardize the cloud computing future services. In addition, mobile phones such as smart phone users are also victim of recent malware attacks. The attackers always find mobile phone and tablet users an easy target to launch malware attack.

Corresponding Author: Abid Shahzad, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST),
H-8/4, Islamabad 44000, Pakistan.

For the last two decades or so the software industry is producing anti-malware solutions which are mainly signature based which use the blacklisting technique. These solutions maintain a repository of the known threats. This repository gets the regular updates from the solution provider. Unfortunately, these existing signature based anti-malware solutions do not seem very useful against the sophisticated zero-day malware. So, the limitations of signature based anti-malware solutions motivated the anti-malware software industry to shift its focus towards a new effective approach called whitelisting. The initial research has shown that whitelisting seems comparatively more effective for protection against zero-day malware attacks.

The whitelisting technique only allows trusted services, processes and applications to run on the machines. Whitelisting automatically removes the chances of execution of other suspicious applications which contain malware. We can easily control the execution of malware by maintaining a whitelist which contains details of those applications which are needed by the users. The whitelisting improves protection against malware but its management is very difficult. This creates rigidity in the network environment because most of the time users require executing new applications or their updates. If a user runs an application which is not in the whitelist, then the user has to contact the administrator to get that application included in the whitelist in order to execute it.

Rest of this paper is organized as follows: Section II gives an overview of techniques, methods and frameworks based on whitelisting to provide solutions against zero-day malware attacks. Section III provides the limitations of the existing anti-malware solutions and techniques. Proposed light weight zero-day anti-malware solution is presented in section IV and validation results of proposed solution are discussed in section V. The last section concludes with summary and possible directions for future research.

Literature Review: The task of protecting networks from recent malware is becoming more and more challenging. The existing signature based anti-malware solutions are not good enough to provide complete security against all types of malware attacks. In other words, the signature based solutions seems to be useless against zero-day sophisticated malware attacks. The signature based anti-malware solutions works on blacklisting, it checks the repository and blocks any applications which seems suspicious i.e., if its entry is found in the database. In whitelisting, only those applications are allowed to execute which are listed in the whitelist maintained by the administrator. Organizations use different technologies to protect their network. Some common technologies are anti-malware, intrusion detection and prevention systems, firewalls, encryption/decryption devices etc. When we talk about positive security model, the known good is whitelisted. In this model, all the trusted applications are added in the list. This model works similar to the access lists in routers or firewalls. Another model is negative security model in which the known malicious applications are blacklisted. The blacklist is maintained just like the antivirus signatures scanners. All the current anti-virus solutions work on the basis of negative security model. A shift from negative security model towards positive security model has been tried in [1]. However, in mission critical environments, where we cannot compromise on the security of information, we should only use application behavior whitelisting.

The modern smart phones have many new features and functionalities which provides both computer and mobile services. The excessive use of smart phones makes them vulnerable to malware attacks. For attackers, the smart phone users are always an easy target to launch an attack and get the user information and private/personal data. The latest phones provide three computation, communications and sensing functionalities. These functionalities though facilitate users, but raise the security concerns as well. Every smart phone has sensors like microphone, camera and GPS receiver. Cai *et al*. [2] claim that the attackers can launch sniffing attacks using the mobile phone sensors. Though enterprises use different technologies and solutions, but such technologies seems useless when it comes to zero-day malware attacks [3]. The existing anti-malware solutions which are signature based blacklisting solutions have been proved to fail against such attacks. The main problem in signature based solution is their high false positive and false negative rates. Keeping these problems in mind, the world is shifting towards the whitelisting technology which provides best protection against sophisticated zero-day malware attacks. General whitelisting architecture which is basically a client server architecture is proposed in [3]. Whenever a client wants to execute an application, the activity log is sent to the server, which maintains the whitelist, for granting execution permission. The checks if the requested application is present in the whitelist database. If it is found the permission is granted otherwise the application execution will be denied by the server. The main problems while implementing whitelisting solutions are maintaining the legitimate application database, integration of patch management servers with application whitelisting servers, verification of digital certificates which are legitimate but are stolen from other users. The application whitelisting performance decreases considerably when it is extended to link with DLL whitelisting.

The organizations and corporate rely on the web based solutions to expand their businesses across the world. As the use of web services is increasing, the numbers of phishing attacks are also increasing with rapid speed. The phishing attacks are of serious concern for the organizations like banking and financial institutions. A solution to detect phishing attacks on the web services is presented in [4] which is based on the personalized whitelisiting along with a support vector classifier SVM. The whitelisting approach can also be used to block the suspicious web pages. The traditional solution which is used to detect phishing attacks is blacklisting technique, but it suffers from a caveat that it cannot detect zero-day phishing attacks. The alternative to overcome this limitation is to use whitelisting, but it is always impossible to maintain such a long whitelist which contains all the legitimate websites. These limitations can be overcome by changing the working procedures, for example, some organizations only allow those websites to run on their networks which relate to their official business. To this end, Gates *et al*. [5] proposed the idea of personalized whitelisting technique to protect the hosts from sophisticated malware attacks. In such approaches, a whitelist is maintained on the user machine.

Phishing websites is a serious concern for users because they can lose their financial information such as credit card information and other bank account details, username and pin code to some hacker or attacker. The current blacklisting software used to protect the users from phishing websites attacks seems partially effective. Kang *et al*. [6] proposed a Phishing Guard framework to protect the users to be victim of phishing websites as well as DNS pharming attacks. The framework works on whitelisting technique which uses similarity check of URL to warn the user against any phishing site. Under phishing attack, the attacker sends a spoofed email to the Internet users for enticing them to visit the phishing site. DNS farming is also a serious attack which misleads the user to use phishing sites or servers. In pharming attack, the attacker changes local host files of the user's machine to link the legitimate financial site to the phishing site. When the user opens perceivably a legitimate website, the local host entry shifts the page to the phishing site instead of the original site. The pharming can be checked against local, network and Internet DNS.

Distributed Denial of Service (DDoS) attacks results in non-availability of critical business services. When an enterprise network is under attack, its website is not available to its intended customers/users. Initially attacker hacks a machine of the victim and makes it a bot machine - often called as botnet. These botnet machines generate a large amount of traffic for the web server at which the critical business site is hosted. These botnets keeps the web server busy with dummy traffic. During this time, the server is crippled to handle legitimate users' requests which eventually results in unavailability of the website. The attackers use different IP spoofing techniques to avoid detections and filtering of the source IP of botnets. Therefore, current anti-malware solutions can be easily fooled through DDoS attacks. However, mitigation against DDoS attacks can be done through maintaining a whitelist that contains entire source IPs which have been previously used to access the critical website [7]. Under the situation of DDoS attack, the IPs present in the whitelist will be given priority and every request which contains the listed IPs in their source will be entertained first.

Devising effective mechanisms to enhance security in the distributed grid environment has been an area of active research during the last decade. The identity reporting is one of the techniques to provide security in a distributed environment. The identity reporting ascertains the applications running on the machines of the grid to establish trust in the environment. Application whitelisting is also used in the trusting computing environment to achieve protection against attacks. However, whitelisting has severe limitations in distributed virtual environment mainly due to its management across different administrative domains. The whitelist of one domain can conflict with other domain of the same environment which results in untrustworthiness of different services to different users. For example, one application may be whitelisted in one domain, but it may not be legitimate for other domains. Such a problem causes rigidity in the grid environment and has thus corroborated whitelisting as unsuccessful in distributed system environment. To this end, a mechanism to update and mange whitelist from a centralized location using configuration manager has been proposed in [8].

Eggendorfer [9] claims that tar pit SMTP simulator is much effective for email servers to handle spam, as it reduces the number of spam and makes job of email server much easier. The tar pit SMTP simulator works by identifying all the legitimate email senders and keeps these senders away from being getting marked as spam by the spam filters. The tar pit simulator can be used in combination with whitelisting. The whitelist will be used to keep record of all the legitimate systems or senders which tar pit simulator identifies as legitimate email senders. Whitelist can help minimize load of tar pit simulator which results in low memory usage at original email server.

Phishing attacks are more serious problem than viruses and malware because of the financial loss that they can cause. For this very particular reason, the solutions to detect phishing attacks are mostly used by the banks and financial institutions. The browsers security toolbars provided by different antimalware solution companies are not much effective as they do not provide adequate protections against the phishing attacks. Again, an anti-phishing solution based on the whitelisting could be a possible answer [10]. However, management of whitelisting would be an extra job for the home user. In short, the idea is to maintain an anti-phishing whitelist that contains list of all the trusted websites of financial institutions which a user has accessed in the past.

Problem Statement

Signature Based Blacklisting Anti-Malware Technique: Right from the beginning of revolution in the computer technology, the malware have always been a threat for the home users and organizations. Over the last two decades, the software industry is producing signature based anti-malware solutions to provide protection against the malware. The most common technique that anti-malware solutions use is blacklisting technique. However, these signature based blacklisting solutions seem vulnerable to zero-day malware attacks and have some prominent drawbacks as described in the subsequent paragraphs.

Drawbacks of Blacklisting Solutions:

- While downloading new signatures from solution provider to update local blacklisting database, the user actually gives control of his/her machine to the anti-malware vendor. These solutions download updates regularly which results in high bandwidth requirement and high CPU and memory utilization.
- These solutions needs remediation against all known malware attacks and update blacklisting database accordingly. However, these solutions have no protection against zero-day malware attacks because they do not have any verification mechanism about the trustworthiness of the software other than checking their signatures in the blacklisting database [11].
- Due to continuous scanning of user machine and IP traffic, the machine's performance degrades and its response becomes slow.
- The solution providers require users to download the updates on regular basis for keeping blacklisting database up-to-date, which disturb normal official working at the user side.

However, with the passage of time, advancements have been made in the blacklisting solutions. Heuristics blacklisting solutions have been introduced which seem effective to some extent as they partially intercept zero-day malware attacks as different variants of malware use generic signatures. These solutions do need to download the updates because they do not completely depend on the definition update files. Due to this, sometimes such solutions detect malware whose signature is not even present in the blacklisting database.

Drawbacks of Heuristic Based Anti-Malware:

- Solutions for some malware are based on assumptions which can lead to misleading results.
- False positive rate is very high. For example, while dealing with large number of emails, some legitimate emails can be identified as spam because of some matching pattern.
- This technique is still in infancy stage and needs further improvement to achieve desired results.

Whitelisting: a Solution for Protection Against Sophisticated Zero-Day Malware Attacks: The sophisticated nature of malware pushed anti-malware software industry to move towards whitelisting technique. The whitelisting technique totally works opposite to blacklisting technique. The whitelisting technique maintains a list of those executable applications, email addresses, website URLs and IP addresses which are allowed to run or open on the user machine. The most common form of whitelisting is the application whitelisting technique. For example, the legitimate applications present in the whitelist will only be allowed to execute on the system while other application will be denied execution. Another common whitelisting technique is email based whitelisting. In email whitelisting, the administrator allows only the legitimate email addresses from whom the users can receive emails. Given below are some significant advantages of whitelisting.

Benefits of Whitelisting over Blacklisting Solutions:

- Whitelisting solutions do not need signature updates.
- They provide protection against sophisticated zero-day malware attacks.
- The machine and traffic scanning is not required by these solutions which results in high CPU and memory availability for other application and processes.
- Only legitimate executables, processes and applications will be allowed to run while all other applications or software will be denied to install or run.
- Due to whitelisting, no unlicensed application or software will run on the machine which will eliminate any license or copyright claim by any vendor.

Light Weight Zero-Day Anti-Malware Solution: Countering malware threats have always been a serious challenge for the network and information security professionals. To defend against malware, a number of signature based blacklisting anti-malware solutions are in use. This study found that these blacklisting anti-malware solutions are helpless against zero-day malware threats as they only provide protection against the known malware threats. In spite of this, there are some positive points about blacklisting technique; however, the only technique which can provide better solution against zero-day malware threats is the whitelisting technique. So, keeping in view the limitations and positive points of whitelisting technique, we propose a zero-day anti-malware solution which uses both the whitelisting and some advantageous aspects of blacklisting technique. Our solution is likely to provide better protection against-zero-day malware threats. The proposed solution monitors and controls the execution of any legitimate or auto-run malware process, application or installer. The solution accounts for both automated and user-triggered execution of processes. Our solution provides protection against the execution of malware, unauthorized applications and installation of illegal software. Our proposed solution is equally useful for different domain users, specifically home and mission critical domain users. The ordinary home users can feel uneasy by intermittently allowing or disallowing new executable on their systems. However, this rigidity introduced into the system would ensure absolute security. The mission critical domain users can benefit from the proposed solution by allowing/blocking execution of processes, applications or installers on their machines. In some domains, our solution would need proper management and could create rigidity in the environment because there are different kind of users who needs different kind of processes and applications to run on a daily basis. We recommend making our solution autorun on the system startup by editing the Windows registry.

Current Uses of Whitelisting and Blacklisting Techniques: Almost all of the signature-based anti-malware solutions use blacklisting technique. These solutions maintain a database of malicious software or websites in the form of signatures/URLs. They block viruses, Trojan horses and malware to some extent by matching them with their signature database. On the other hand, whitelisting is mainly used in email anti-spam filters or email gateways. The administrators maintain a whitelist of all the legitimate email domains, addresses and the IPs. In this way, the particular organizations which adopt whitelisting approach only receive emails from the listed domains, addresses and IPs. The blacklisting technique is also in use in email spam filters. Much like whitelisting, the administrator maintains a blacklist of all the fraud, spammer email domains, addresses and the IPs. This way the organizations remains protected from spam and harmful emails.

Proposed Solution: Our proposed solution consists of a software utility and the two databases – the whitelisting and blacklisting databases. Figures 1 and 2 show both phases of the proposed solution respectively.

Phase I: In the first phase, as shown in Figure 1, the user is required to maintain whitelisting and blacklisting database of legitimate processes, auto-run malware processes, applications and installers. The list of processes, applications and installers in the whitelisting and blacklisting needs to be mutually exclusive. If any process, application or installers executes on the system,

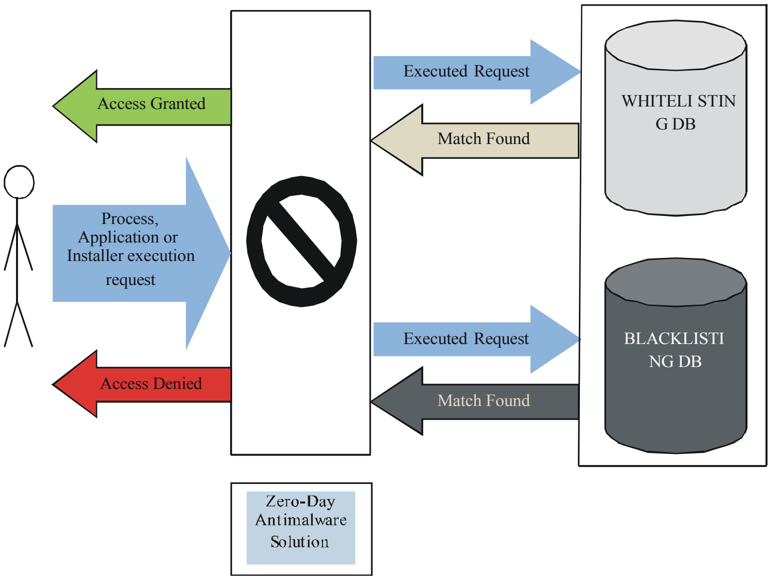


Fig. 1: Phase I of zero-day antimalware solution

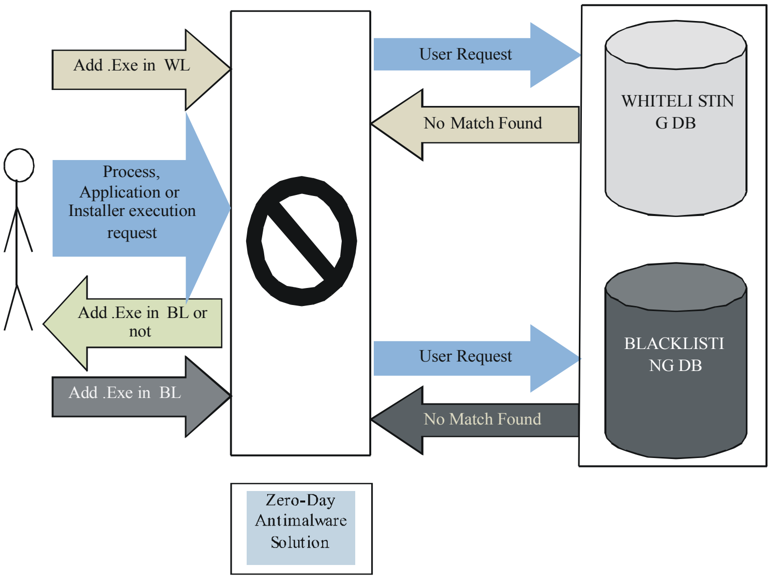


Fig. 2: Phase II of zero-day antimalware solution

the proposed solution checks the executed request and matches it with the whitelisting and blacklisting databases. If signature of the requested executable is found in the whitelisting database, then access for that particular request will be granted. On the other hand, if a match is found in the blacklisting database then request will be denied immediately.

Phase II: In the second phase, as shown in Figure 2, our proposed solution checks if any process, application or installer executes on the system and solution does not find a match of the requested executable in any of the database (i.e., whitelisting or blacklisting), then our proposed solution will prompt user to perform an action either to mark the requested executable as blacklist or whitelist. If the requested executable is a legitimate process or application and user is aware of that, then user needs to mark it whitelist. Once the user will mark it as whitelist, the entry against that particular executable will be stored in the whitelisting database. If the user thinks that the executable is a malware or not a legitimate one, then user can opt to mark it as blacklist. The entry against that executable will be stored in the blacklisting database accordingly.

A word of caution is that in case the user accidently kills/quits our solution while he/she continues working on the system. Then, in the meanwhile until our solution is loaded again, the user will be able to execute any application, process or installer even if it was part of the blacklisting database. However, an important feature of our solution is that when our solution is loaded again, it will scan OS process manager. If any process is found running on the system which has its entry in the blacklisting database, our solution will instantly terminate it.

Advantages of the Proposed Solution

Protection Against Zero-Day Malware Threats: The solution uses characteristics of both blacklisting and the whitelisting techniques. The whitelisting part of the solution will help protect from any new malware threat if the blacklisting database does not have information about it. The whitelisting will not allow that malware to execute because it is not present in the whitelisting database as well. The solution will only allow the executables which is present in the whitelisting database. The blacklisting database will also be available to protect against any malware which have record in the blacklisting database.

No Updates Required: The solution does not require automatic updating the whitelisting and blacklisting databases through Internet. Rather, the users have to maintain or update both the databases as and when required.

Processes and Application Management: As the solution monitors all the executables either it is a process, a malware or any application on the system, it helps users to allow or deny its execution. The trusted processes and applications can be whitelisted and unwanted processes and applications can be blacklisted accordingly. So, in this way, we can restrict network users to run unauthorized applications on the system.

Software Control: Our solution will restrict the user to install any prohibited or unauthorized software. If the user tries to install software and its binary is new for the solution then it will prompt the user for action. The system administrator can control and manage the installations of different software using our solution.

Low CPU and Memory Requirements: The proposed solution does not need scanning of the system to detect and remove malware. It will make detections as soon as the new process or application will try to execute. Therefore, our solution does not need high CPU and memory requirements as no scanning of the system is involved.

Validation of the Proposed Solution: We have implemented our solution in C# using Visual Studio 2010. The solution is a Windows Application built using. Net Framework 3.5. We implemented our proposed solution on a network machine. Our system precisely detected all the binaries whenever an executable/ application attempted to run on the system. For this purpose, we validated the functionality of the solution against various types of executables, for example, legitimate processes, malware processes and user applications.

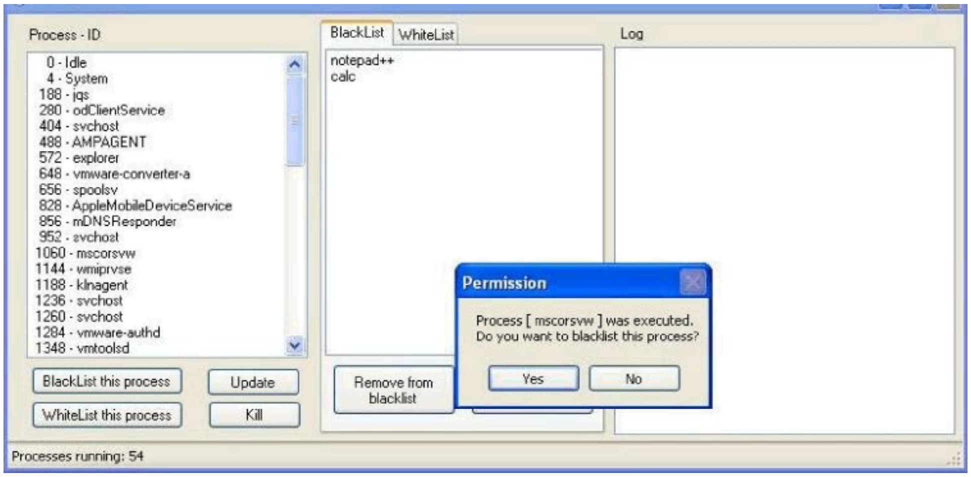
Figure 3 shows a legitimate windows process “mscorsvw” trying to run on the system. However, our system does not have any information about this process in both the databases (whitelisting and blacklisting); therefore, it prompts the user to either mark it legitimate or illegitimate. Since an ordinary user is generally unaware of such type of Windows processes, therefore, we blacklisted that process to stop it getting run on our machine. This scenario in our experimentation is in fact meant to exhibit how to block the undesired processes/applications.

Figure 4 illustrates that the process which have blacklisted in the previous step (Figure 3) is now included into blacklisting database to prevent its future execution on our system.

In Figure 5, the fallout of the executing a new application “Firefox” is shown as both the databases have no prior information about this application. As soon as the Firefox application is launched, the system prompted user for supplying an appropriate action. Since, Firefox is commonly known web browser and is generally considered as a legitimate application, we want let it to execute.

As a result of the user action explained above, an entry for Firefox application is created into whitelisting database. Figure 6 shows Firefox application has its entry in whitelisting database. Now our solution will always allow the execution of Firefox on our machine.

In the next step, we executed a real malware “New Folder” in our experimentation to validate our solution against averting malware threats. Our solution immediately responded for the necessary user action as shown in Figure 7. We blacklisted this malware to stop it current as well as future execution on our machine.



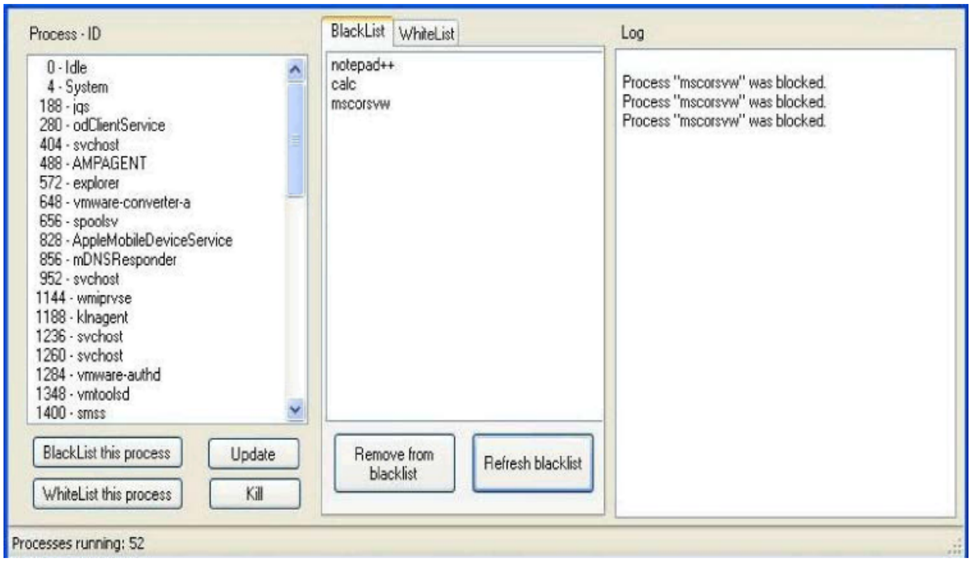
```
0 - Idle
4 - System
188 - jqs
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1060 - mscorsvw
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1348 - vmtoolsd
```

```
notepad++
calc
```

Process [mscorsvw] was executed.
Do you want to blacklist this process?

Processes running: 54

Fig. 3: Blocking a backend windows process



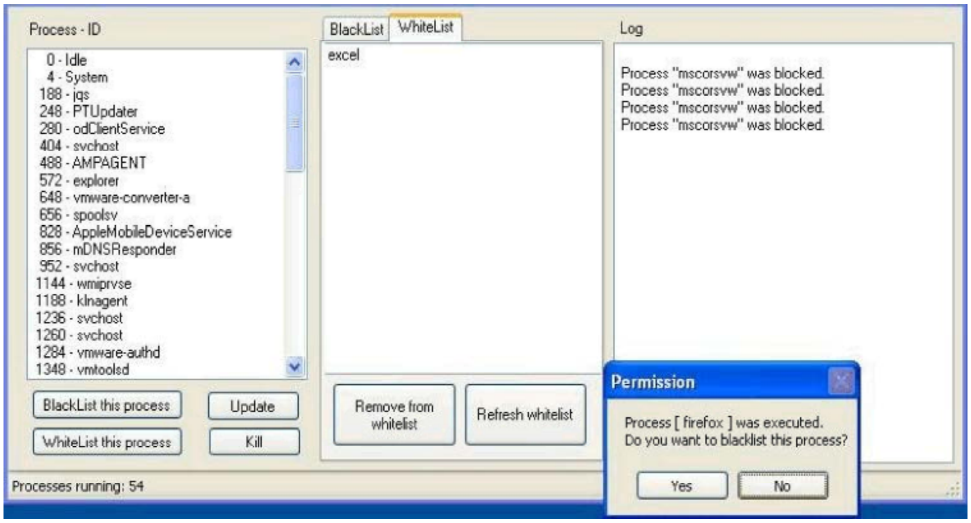
```
0 - Idle
4 - System
188 - jqs
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1348 - vmtoolsd
1400 - smss
```

```
notepad++
calc
mscorsvw
```

```
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
```

Processes running: 52

Fig. 4: Snapshot of blacklisting database



```
0 - Idle
4 - System
188 - jqs
248 - PTUpdater
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1348 - vmtoolsd
```

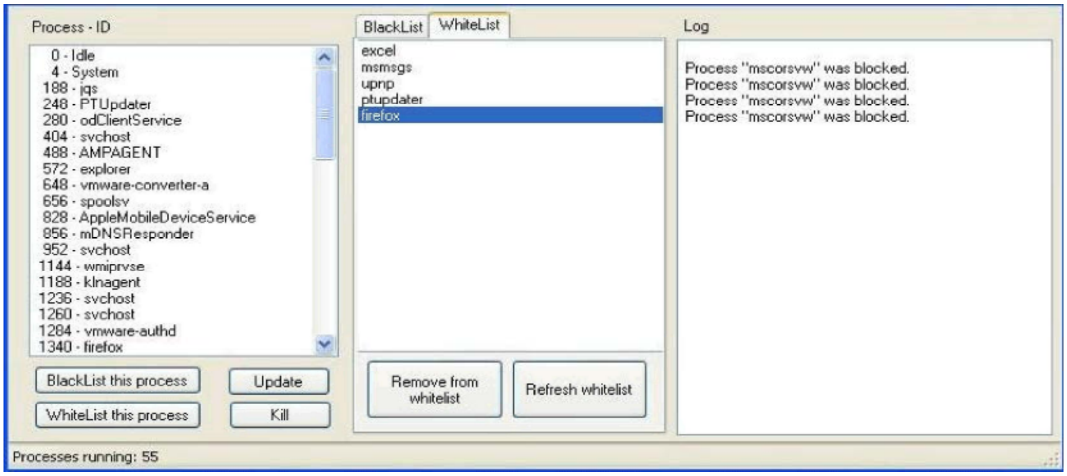
```
excel
```

```
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
```

Process [firefox] was executed.
Do you want to blacklist this process?

Processes running: 54

Fig. 5: Execution of firefox application



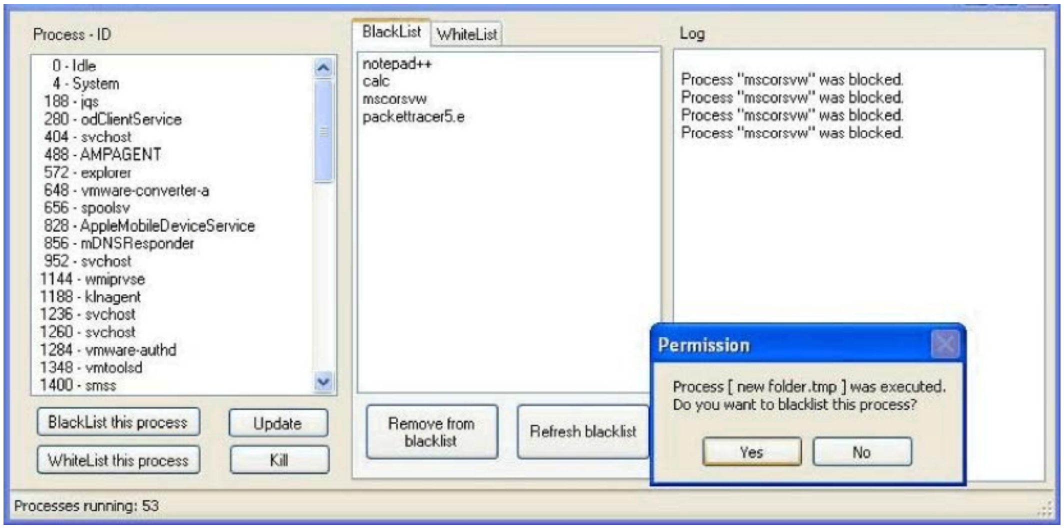
```
0 - Idle
4 - System
188 - jqs
248 - PTUpdater
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1340 - firefox
```

```
excel
msmsgs
upnp
ptupdater
firefox
```

```
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
```

```
Processes running: 55
```

Fig. 6: Snapshot of whitelisting database



```
0 - Idle
4 - System
188 - jqs
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1348 - vmtoolsd
1400 - smss
```

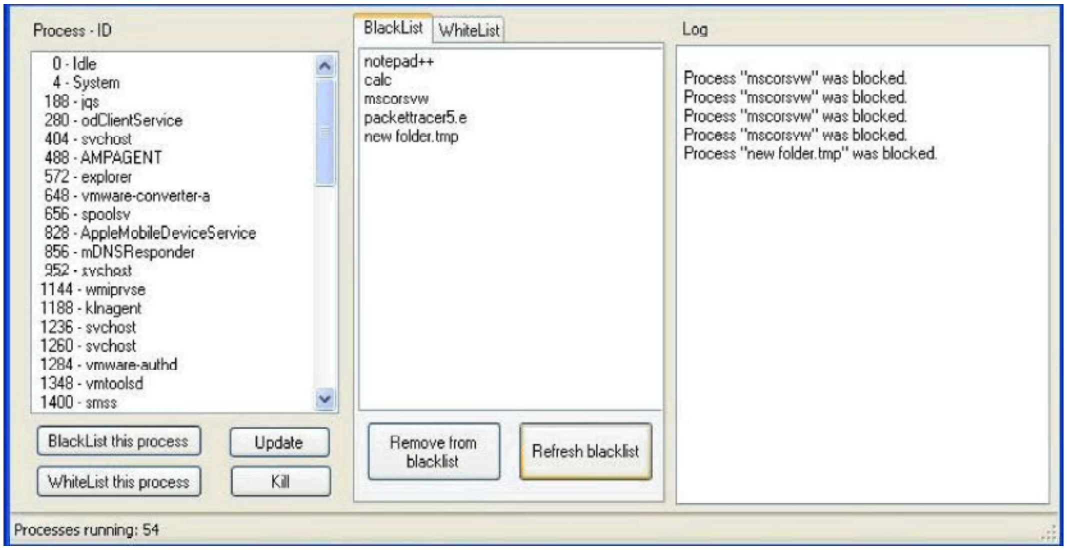
```
notepad++
calc
mscorsvw
packettracer5.e
```

```
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
```

```
Process [ new folder.tmp ] was executed.
Do you want to blacklist this process?
```

```
Processes running: 53
```

Fig. 7: Execution of a malware



```
0 - Idle
4 - System
188 - jqs
280 - odClientService
404 - svchost
488 - AMPAGENT
572 - explorer
648 - vmware-converter-a
656 - spoolsv
828 - AppleMobileDeviceService
856 - mDNSResponder
952 - svchost
1144 - wmiprvse
1188 - klnagent
1236 - svchost
1260 - svchost
1284 - vmware-authd
1348 - vmtoolsd
1400 - smss
```

```
notepad++
calc
mscorsvw
packettracer5.e
new folder.tmp
```

```
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "mscorsvw" was blocked.
Process "new folder.tmp" was blocked.
```

```
Processes running: 54
```

Fig. 8: Snapshot showing that malware has been blacklisted

The Figure 8 confirms that the malware which we have blacklisted in the above step is now blacklisted.

CONCLUSION

In this paper, we have discussed the harmful impact of the malware attacks which usually result in significant losses in the form of data and information. We have also discussed the problem and limitations of the existing signature based anti-malware solutions which prove useless against zero-day malware attacks. For years, whitelisting technique has been considered to be the only solution to provide protection against zero-day malware attacks. In view of this, we have proposed a zero-day anti-malware solution to provide protection against these malware attacks. Our solution can help users to control the execution of different processes, applications and installers on the system. The main idea behind this research is to provide a light weight zero-day anti-malware solution for the home as well as mission critical domain users. On the basis of the validation results shown in this paper, we can conclude that our solution provides better protection against zero-day malware attacks as compared to the traditional signature based anti-malware solutions. In future, we intend to extend our work to implement a centralized solution for large organizations and enterprises. Implementing a new solution with modifications which can analyze behavior of legitimate whitelisted processes and applications to avoid intrusion in the domain and provide better security is also envisaged to be another possible future dimension to this research.

REFERENCES

1. Eswari, P.R.L. and N.S.C. Babu, 2010. “A practical business security framework to combat malware threat”, IEEE 2012 World Congress on Internet Security World CIS, pp: 77-80.
2. Liang, C.A.I., S. Machiraju and H. Chen, 2009. “Defending against sensor-sniffing attacks on mobile phones”, SIGCOMM Workshop on Networking Systems and Applications on Mobile Handhelds, pp: 31-36.
3. Pareek, H., S. Romana and P.R.L. Eswari, 2012. “Application whitelisting approaches and challenges”, International Journal of Computer Science Engineering and Information Technology IJCSEIT, 2: 13-18.
4. Belabed, A., E. Aimeur and A. Chikh, 2012. “A personalized whitelist approach for phishing webpage detection”, 7th International Conference on Availability, Reliability and Security ARES, pp: 249-254.
5. Gates, C., N. Li and J. Chen, 2012. “Codesheild towards personalized application whitelisting”, 28th Annual Computer Security Applications, pp: 279-288.
6. Kang, J.M. and D.H. Lee, 2007. “Advanced whitelist approach for preventing access to phishing sites”, International Conference on Convergence Information Technology, pp: 491-496.
7. Yoon, M., 2010. “Using whitelisting to mitigate ddos attacks on critical internet sites”, IEEE Communications Magazine, pp: 110-115.
8. Huh, J.H., J. Lyle, C. Namiluko and A. Martin, 2009. “Application whitelists in virtual organizations”, Oxford University Computing Laboratory, pp: 1-1.
9. Eggendofer, T., 2008. “Combining the smtp tar pit simulator with whitelisting”, Security and Management Conference, pp: 333-339.
10. Wang, Y., R. Agrawal and B.Y. Choi, 2008. “Light weight anti-phishing with user whitelisting in a web browser”, IEEE Region Five Conference, pp: 1-4.
11. Faronics, 2011. “Blacklisting vs whitelisting software solutions”, Intelligent Solutions for Absolute Control Whitepaper, pp: 1-6.