RESEARCH ARTICLE OPEN ACCESS

Behavior Based Spyware Identification

Pooja Singhal [1], Dharna [2]
Department Of Information Security and Management
Indira Gandhi Delhi Technical University for Women
Delhi - India

ABSTRACT
Spyware is a potentially unwanted program that resides in the user's machine to transmit the information, private and confidential to the user, to the third party without the user's consent, control, knowledge and permission. Spyware affects the user's privacy in a way that some of the spyware programs may display some advertisements on user's screen and log information about the user's activity including email addresses, web browsing history, online buying activities, etc. All the anti-spywares developed yet are based on the established signatures or are stateless in nature. This research is based on developing a patch management technique that will be stateful in nature and will revert back the changes occurred in the behaviour of the system because of the presence of spyware programs in the computer system. Technique will be developed using the pattern matching techniques.

Keywords :— Spyware, malware, privacy, information, stateful

I. INTRODUCTION

With the increase in usage of internet and its underlying technologies, unprecedented opportunities to gain unauthorized access to data, change data, destroy data, make unauthorized use of computer resources, interfere with the intended use of computer resources have been exploited time by time via many types of malware including, but is not limited to computer virus, worms, Trojan horses, etc. [1]. Spyware is a potentially unwanted program that resides in the user's machine to transmit the information, private and confidential to the user, to the third party without the user's consent, control, knowledge and permission. Spyware affects the user's privacy in a way that some of the spyware programs may display some advertisements on user's screen and log information about the user's activity including email addresses, web browsing history, online buying activities, etc. If your computer starts to behave strangely, you might be experiencing spyware symptoms or have other unwanted software installed on your computer [2].

The installed spyware may be capable of capturing keystrokes, taking screenshots, saving authentication credentials, storing personal email addresses and web form data, and thus may obtain behavioural and personal information about users. It may also communicate system configuration including hardware and software, system accounts, location information, and information about other aspects of the system to a third party [3].

- **Pop-up advertisements all the time.** Some unwanted software will bombard you with pop-up ads that aren't related to a particular website you're visiting. These ads are often for adult or other websites you may find objectionable. If you see pop-up ads as soon as you turn on your computer or when you're not even browsing the web, you might have spyware or other unwanted software on your computer.
- **Settings change and these can't be changed back to the way they were.** Some unwanted software can change home page or search page settings. Even if settings are adjusted, they revert back every time you restart your computer.
- **Web browser contains additional components that it downloads itself.** Spyware and other unwanted software can add toolbars to your web browser that you don't want or need. Even if you remove these toolbars, they might return each time you restart your computer.
- **Computer seems sluggish.** Spyware and other unwanted software are not designed to be efficient. The resources these programs use to track your activities and deliver advertisements can slow down your computer and errors in the software can make computer crash. If you notice a sudden increase in the number of times a certain program crashes, or if your computer is slower than normal at performing routine tasks, you may have spyware or other unwanted software on your machine.

A. *Spyware Classification*

According to terminology used in SpyBot S&D, Spywares are classified as follows [4]:

- **Cookies and Web bugs**: On the behalf of web servers, cookies store state information on individual's client web browser. However, many sites use the same advertisement providers; they track the behaviour of the users across web sites. Cookies ad web bugs both rely on the existing web browser function and do not contain any code of their own.
- **Browser hijackers**: Hijackers what generally do is; they change the user's web browser settings by either installing a browser extension, modifying Windows registry entries or directly modifying or replacing browser preference files.
- **Keylogger**: Keylogger are the kind of software or hardware that records all keystrokes made by the users in order to find the sensitive information such as passwords, credit card numbers and more. The log is accessed by the attacker either offline or online.
- **Tracks**: Information recorded by operating system or application activities the user has performed such as visited websites, recently opened files and programs maintained by operating system.
- **Malware**: Malicious software such as viruses, worms, and Trojan horses.
- **Adware**: Software that displays the advertisement according to the user's current activity or browsing activity to the third party.

B. *Spyware Signatures*

Spyware signatures are being identified using the following grounds:

- Type of headers (creator of the spyware)
- Pattern or metadata which to be acting like a malicious activity
- Language used to develop the spyware
- Timestamp of the file used as spyware

Spyware programs are being analysed either on the basis of their signatures or their behaviours. Classification determines the association between the signature and behaviour of the spyware programs:

- One-to-one: Spyware having single signature and results in the same behaviour each time it is invoked.
- One-to-many: Spyware programs created by same author but to perform different function that results in the polymorphic behaviour of the system
- Many-to-one: Same Spyware programs written by different authors so that the spywares behave in the same way but each has its own signature. For instance, keyloggers. Keyloggers are developed by many companies in the market thus each keylogger has its own developing companies signature but all keyloggers are performing and behaving in the same manner.
- Many-to-many: Different spyware signatures with different behaviour. They are difficult to analyse.

Spyware programs utilize the critical areas of the system to survive the reboots and mini-installers help them to re-install after they have been detected and removed. These critical areas where self-healing spywares strive for their survival for a longer period may include [1]:

Arbitrary location:

It would be very easy for user to discover the spyware program if they reside in very obvious places such as C:\Program Files. Therefore they are usually scattered in arbitrary locations such as temporary directories (e.g. Temporary Internet Files) and privileged system directories (e.g. %windowsdirectory%\system32) to bypass the straight forward inspection.

Randomized Filename:

Filenames of the spyware programs can be randomized (either partially of fully) for different users on different machines. For example, Look2Me spyware programs would generate randomized filename.

Manipulated time property or system calls:

Spyware programs may alter the time properties (creation, modification, access time) of the system when they reside deep inside the system. When anti-spywares try to look for those spyware, they sort the results by time to look for new suspicious files.

Legitimate DLL as disguised:

Windows interface system generally automatically load the DLL files, the spyware programs force other DLL files and processes to load them. Spyware programs can also replace the existing DLL files. For example, they can replace system DLL files with spyware infected spyware. By this, user cannot make difference between a bad DLL and a good DLL.

Existing solutions for fighting spyware either require users to manually examine the system or use signature-based antispyware tools (a few freely available are Lavasoft AdAware, Spybot Search & Destroy, and Microsoft Windows Antispyware) to identify and remove known spyware[2]. In practice, it is essential to install multiple anti-spyware tools in order to minimize false negatives of spyware detection. Some of these tools have provided real-time monitoring features (e.g. Spybot's TeaTimer and Microsoft Windows AntiSpyware Real-Time Protection) that warn users when a program is attempting to make changes to critical areas of Windows system registry. Most of these anti-spyware tools developed yet use signatures to detect the spyware programs. Over time, spyware programs have grown more resilient to this technique; once detected and removed, they re-install themselves over the system. Since current anti-spyware tools are stateless they

fail to permanently remove these self-healing spyware programs.

The spyware creators have developed this feature that provides a few self-defence workarounds to increase their survivability by recovering after being removed by any anti-spyware tool.

This paper is organized as follows Section II describes literature survey used to work on this research. Section III explains how the identification process has been applied. Section IV discusses the results and Section V concludes the paper with future possible work.

II. LITERATURE SURVEY

Within research areas [5], several studies on spyware have been presented. Installing the threats on a computer and testing it with different Anti-Spyware scanners is one of the approaches used by newspapers and magazines to rate and review the Anti-Spyware products. Sometimes the system is tested only with a newly operating system installed with security patches such as Windows and then the system is bombarded with Spyware. These types of tests are also used to track the performance of the system before and after the Spyware are installed on the computer. Tzu-yen Wang [6], A surveillance spyware detection based on data mining methods was considered in which three kinds of information about file are collected as potential behaviour, impact on system files and network traffic. The first one is a static analysis and next two are dynamic analyses. Behaviour of an executable program was predicted by analysing DLLs. Ming-Wei Wu, Sy-Yen Kuo, Yi-Min Wang [2] developed Stateful threat aware removal system (STARS) to keep the track of activities performed by running processes and follow up the effectiveness of a spyware removal task over time. However, STARS are not able to detect hidden registries and DLL injection. As registries are more complicated to maintain than files and it requires remote thread monitoring to identify DLL injection. Amar Al-Anwar, Yousra Alkabani, M. Watheq El-Kharashi, Hassan Bedour [3] presented a methodology that aims to protect from hardware Spywares embedded in third party IPs without trying to detect the Spywares. The method operates at run time instead of the traditional test-time techniques and also protects from Trojans. While this method introduces a significantly larger overhead, it provides higher levels of Spyware protection. However, it can only protect from spying Trojans by decreasing the probability of being able to send information. Additionally, it will not really detect a Trojan or protect from circuit failure. Jonathan L. Edwards [1] gave a system, method and Computer Program Product for scanning the plurality of names in a registry for complete Search history of a computer. In particular, a change in a registry of a computer is first identified then a scan is performed based on whether the change in the registry is identifies. Steven Gribble, Seattle, Henry Levy, Seattle, Alexander Moshchuk, Seattle, Tanya Bragin, Seattle [7] developed a tool that uses a virtual machine (VM) to sandbox and analyze potentially malicious content. By installing and running executables within a clean VM environment, commercial anti-spyware tools can be employed to determine whether a specific executable contains piggy-backed spyware. Suchita Yadav, Ravi Randale [8] detected and prevented the keylogger spyware attack in which the detection is performed by the help of honeypot and keystroke agent. The prevention is performed by the help of encryption algorithm. This original logfile encrypted and send to the honeypot system for detection. After inspecting this logfile the honeypot system delete keylogger if required and finally keylogger program which sent to hacker is not original logfile but scrambled logfile. Mohammad Wazid, Avita Katal, R.H. Goudar, D.P. Singh and Asit Tyagi [9] proposed a framework for detection and prevention of keylogger spyware. For detection and prevention purpose, a detection prevention server has been installed that will automatically remove the keylogger spyware program from the system when detected. Easwar A. Nyshadham and Eric Acjerman [10] argued that aversion to spyware risk is contributed by the people's inability to judge likelihood of risk. They used decision theory to conduct an experiment to a) assess the separate contributions of standard risk aversion and aversion to ambiguity to overall risk and b) examine whether peoples traits (optimism/pessimism, tolerance for ambiguity) and perception of information explain the patterns in the parameters corresponding to risk and ambiguity functions. Parmjit Kaur, Sumit Sharma [11] proposed a hybrid approach for detection of malicious applications in android application with the help of antiviruses. Hao Wang, Somesh Jha and Vinod ganapathy [12] proposed a tool, NetSpy, for automatically generating network-level signatures for spyware. It determines whether an untrusted program is spyware by correlating user input with network traffic generated by untrusted party. Abhay Mittal [13] proposed a technique which utilises the fundamentals of application layer and network layer to eliminate the spyware programs. It scans the HTTP requests at the browser and suggested a new add-on at the both DNS and network layer in order to detect and remove the unwanted program. This technique focuses only on network-based detection and uses NetSpy concepts for detection mechanism.

III. IDENTIFICATION PROCESS

A. *Operating system based survey*

1. *Using Performance Analyser*

A freshly installed computer system (installed with anti-virus) including Registries, DLLs, Applications, Drives, Files, Folders has been monitored and scanned along with CPU utilization and network utilization to determine how the computer system behaves and works in the normal working condition.

Though, there is not any specific product for this work, Performance analysers have been used as Task Manager for monitoring CPU utilization and open source product named as Blueproject software Systracer v2.10 which analysis the following registries as:

- *HKEY_CLASSES_ROOT*
- *HKEY_CURRENT_CONFIG*
- *HKEY_CURRENT_USER*
- HKEY_LOCAL_MACHINE
- HKEY_USERS

Applications as:

- Startup
- Services
- Drivers
- Running processes
- Loaded DLLs
- Programs
- Opened Handles
- Opened Ports

Several Spyware programs classified as Internet Spyware, Desktop spyware and Keyloggers has been installed in the computer system to determine the behaviour and working of the system in the presence of spyware programs.

Installed spywares include:

PowerSpy as internet Spyware
SSPro as desktop spyware
Spytech SpyAgent keylogger

Behaviour and working of the computer system after the installation of above mentioned spyware program results in following changes:
Scanning and monitoring process has been performed by using BlueProject Software Systracer v2.10 and following changes have been observed:

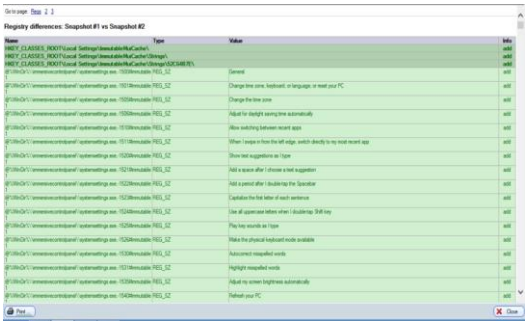


Fig1: Example of Registries added

Fig2: Example of Registries deleted

Difference in addition and deletion of the applications (shows the difference in loaded DLLs):

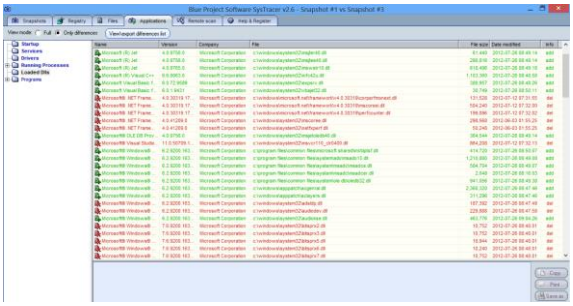


Fig3: Example of Addition and Deletion of Applications

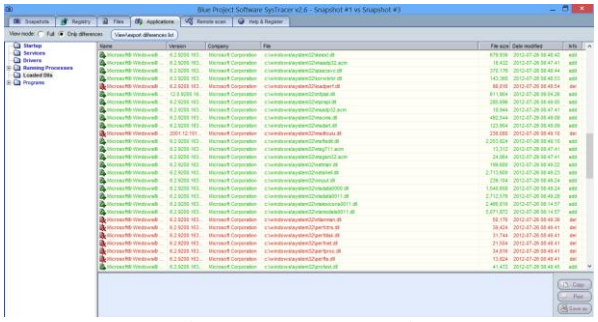


Fig4: Example of Addition and Deletion of Applications

Existence of the spyware affects the Services in the following manner:

Fig5: Example of Services Affected

CPU utilization:

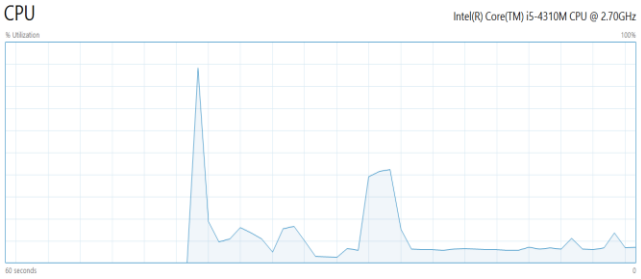


Fig6:Affect on CPU Utilization

Fig7:Affect on CPU Utilization

2. Using Command-line utilities

However, instead of using the product analyser, this task could have been performed manually either by making changes in the value of registries. This could have been done by exporting the registries of the system and then changing the value of registries or by using the command

- dir/a/s > "filename" ("Result"): It results in a result.txt file containing all the information about the computer system including files, folders, applications, drives, processes and hidden folders with their timestamp of creation, date of creation, path of file, each file size and folder size.

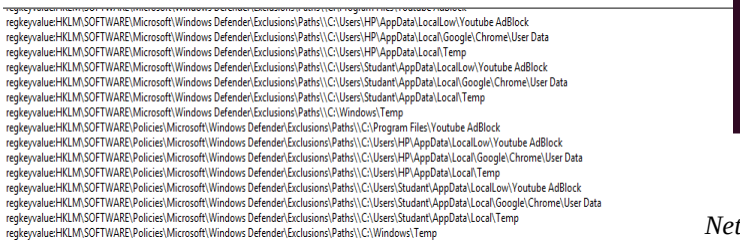


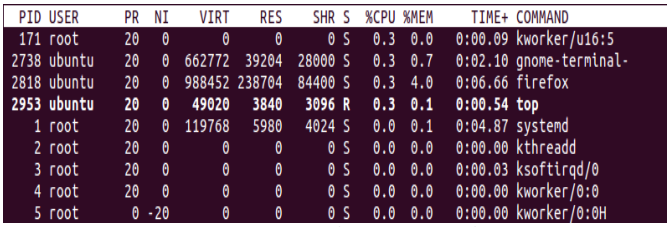
Fig8: Result of Using Utilities

3. Using Scanning Process

```
Void DirSearch (String* sDir)
{
try
{
//Find the subfolders in the folder
String* d[]=Directory::GetDirectories(sDir);
int numDirs= d->get_Length();
for(int i=0;i<numDirs;i++)
{//do something with file
}
//recurse into the next directories
DirSearch(d);
}
}
Catch(System::Exception* e)
{
MessgaeBox::Show(e->Message);
}
}
```

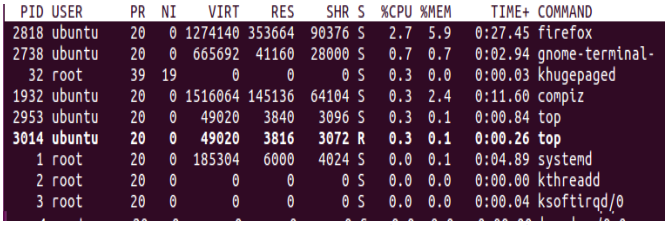
B. APPLICATION BASED SURVEY

Mozilla Firefox has been analysed with its CPU and network utilization in the presence of Spyware program that affects the browser:



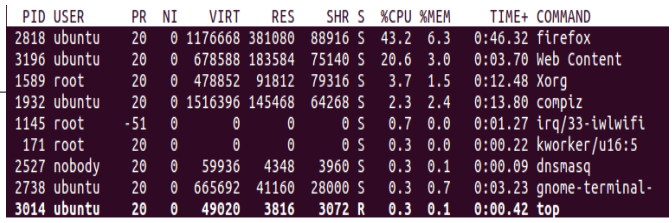
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
171	root	20	0	0	0	0	S	0.3	0.0	0:00.09	kworker/u16:5
2738	ubuntu	20	0	662772	39204	28000	S	0.3	0.7	0:02.10	gnome-terminal-
2818	ubuntu	20	0	988452	238704	84400	S	0.3	4.0	0:06.66	firefox
2953	**ubuntu**	**20**	**0**	**49020**	**3840**	**3096**	**R**	**0.3**	**0.1**	**0:00.54**	**top**
1	root	20	0	119768	5980	4024	S	0.0	0.1	0:04.87	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.03	ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H

Fig9:CPU utilization for Mozilla Firefox



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2818	ubuntu	20	0	1274140	353664	90376	S	2.7	5.9	0:27.45	firefox
2738	ubuntu	20	0	665692	41160	28000	S	0.7	0.7	0:02.94	gnome-terminal-
32	root	39	19	0	0	0	S	0.3	0.0	0:00.03	khugepaged
1932	ubuntu	20	0	1516064	145136	64104	S	0.3	2.4	0:11.60	compiz
2953	ubuntu	20	0	49020	3840	3096	S	0.3	0.1	0:00.84	top
3014	**ubuntu**	**20**	**0**	**49020**	**3816**	**3072**	**R**	**0.3**	**0.1**	**0:00.26**	**top**
1	root	20	0	185304	6000	4024	S	0.0	0.1	0:04.89	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.04	ksoftirqd/0

Fig10:Change in CPU utilization for Mozilla Firefox



PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2818	ubuntu	20	0	1176668	381080	88916	S	43.2	6.3	0:46.32	firefox
3196	ubuntu	20	0	678588	183584	75140	S	20.6	3.0	0:03.70	Web Content
1589	root	20	0	478852	91812	79316	S	3.7	1.5	0:12.48	Xorg
1932	ubuntu	20	0	1516396	145468	64268	S	2.3	2.4	0:13.80	compiz
1145	root	-51	0	0	0	0	S	0.7	0.0	0:01.27	irq/33-iwlwifi
171	root	20	0	0	0	0	S	0.3	0.0	0:00.22	kworker/u16:5
2527	nobody	20	0	59936	4348	3960	S	0.3	0.1	0:00.09	dnsmasq
2738	ubuntu	20	0	665692	41160	28000	S	0.3	0.7	0:03.23	gnome-terminal-
3014	**ubuntu**	**20**	**0**	**49020**	**3816**	**3072**	**R**	**0.3**	**0.1**	**0:00.42**	**top**

Fig11:Change in CPU utilization for Mozilla Firefox

Network Utilization

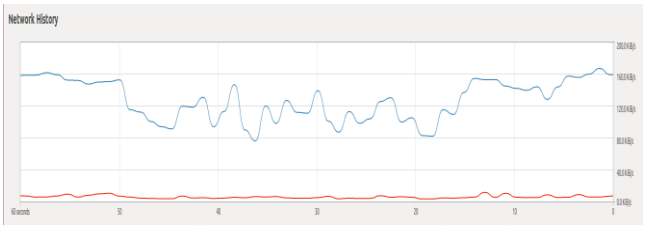


Fig12:Affect on Network Utilization for Firefox

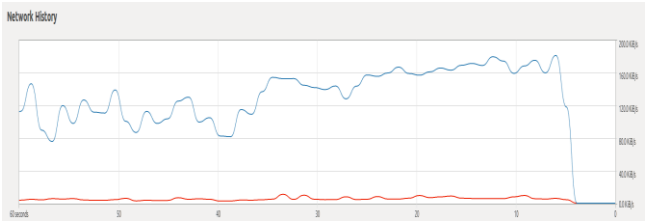


Fig13:Affect on Network Utilization for Firefox

IV. RESULTS

The tables below show the difference between number of registries, number of services, number of files and folders, number of applications, CPU utilization before and after the installation of spyware programs in the freshly installed Windows operating system.

Number of registries	Before the installation of spyware programs	After the installation of spyware programs
Registry keys	204955	181500
Registry values	322628	294879

TABLE 1: DIFFERENCE IN NUMBER OF REGISTRIES

Number of Files	Before the installation of spyware programs	After the installation of spyware programs
Files	62991	63268
Folders	13727	13768

TABLE 2: DIFFERENCE IN NUMBER OF FILES

Number of services	Before the installation of spyware programs	After the installation of spyware programs
Manual	110	113
Auto	43	46
Disabled	5	4

TABLE 3: DIFFERENCE IN NUMBER OF SERVICES

Number of Applications	Before the installation of spyware programs	After the installation of spyware programs
Running processes	37	43
DLLs	540	563
Programs installed	3	4
Programs found	509	517

TABLE 4: DIFFERENCE IN NUMBER OF APPLICATIONS

CPU utilization	Before the installation of spyware programs	After the installation of spyware programs
Range(in percentage)	0-20	40-100

TABLE 5: DIFFERENCE IN CPU UTILIZATION

Also, application based survey involves monitoring of network utilization and CPU utilization because of the presence of browser spyware for Mozilla Firefox.

CPU utilization	Before the installation of spyware programs	After the installation of spyware programs
Range(in percentage)	0.3-2.7	22.9-68.4

TABLE 6: DIFFERENCE IN CPU UTILIZATION FOR FIREFOX

Network utilization	Before the installation of spyware programs	After the installation of spyware programs
Range(in percentage)	0-60	40-100

TABLE 7: DIFFERENCE IN NETWORK UTILIZATION FOR FIREFOX

V. CONCLUSION AND FUTURE WORK

This survey shows how severely the presence of spyware programs affects functioning of the freshly installed operating system along with the presence of anti-viruses and firewalls. Inspired by the different scanning methods, three different methods have been used to scan the computer system to monitor the functioning of the system that involves the use of performance analyser, command line utilities and program code.

Based on the changes identified and signatures developed earlier, a patch management model/technique will be developed to mitigate the spyware program which will detect the spyware and will revert them back from the system without affecting the systems' working. This patch management technique will be stateful in nature and will identify spyware based on the behaviour and working of the system by identifying the malicious activities being performed in the system.

REFERENCES

[1] Jonathan L. Edwards," System, Method and Computer Program product for accelerating Malwares/Spywares scanning", United States Patent Edwards, July 19 2011

[2] M.W.Wu, Y. Huang, Y.M.Wang, and S.Y Kuo, "A Stateful Approach to Spyware Detection and Removal", 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06), IEEE, 2006

[3] Amr Al-Anwar, Yousra Alkabani, M. Watheq El-Kharashi, Hassan Bedour," Defeating Hardware Spyware in Third Party IPs", IEEE, 2013

[4] Stefan Saroiu, Steven D. Gribble, and Henry M. Levy," Measurement and Analysis of Spyware in a University Environment", IEEE

[5] Rehan Shams, Muhammad Farhan, Sajid Ahmed khan Fahad Hashmi," Comparing Anti-Spyware Products – A different Approach", IEEE, 2013

[6] Tzu-Yen Wang, Shi-Jinn Horng, Ming-Yang Su, Chin-Hsiung Wu, Peng-Chu Wang, Wei-Zen Su," A Surveillance Spyware Detection System Based on Data Mining Methods" 2006 IEEE Congress on Evolutionary Computation

[7] Steven Gribble, Seattle, Henry Levy, Seattle, Alexander Moshchuk, Seattle, Tanya Bragin, Seattle," DETECTION OF SPYWARE THREATS WITHIN VIRTUAL MACHINE", United States Patent Gribble et al., Jun. 5, 2012

[8] Randale," Detection and Prevention of Keylogger Spyware Attack", International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE) Volume 1, Special Issue, Vivruti 2015.

[9] Mohammad Wazid, Avita Katal, R.H. Goudar, D.P. Singh and Asit Tyagi," A Framework for Detection and Prevention of Novel Keylogger Spyware Attacks", IEEE, 2012

[10] EaswarA. Nyshadham, Eric Acjerman, "Spyware-Risk and ambiguity attitudes", Graduate school of computer and information sciences,2012

[11] Parmjit Kaur, Sumit Sharma," Spyware Detection in Android using Hybridization of Description analysis, Permission mapping and interface analysis", International Conference on Information and Communication Technologies, 2014

[12] Hao Wang, Somesh Jha and Vinod ganapathy," NetSpy: automatic generation of spyware signatures for NIDS", 22nd Annual computer security application conference, IEEE, 2006.

[13] Abhay Mittal, "Resolving the Menace of Spyware through Implementations in Application Layer and Network Layer", 2012, IEEE