



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## TAKD EVASION ON KEYLOGGER

Shubham Kumar, Anjish Kumar, Dr. Brijesh Kr. Singh

Student, Student, Associate Professor  
Computer Science and Technology,  
Galgotias University, Greater Noida, India

**Abstract:** The development of technology is absolutely fast and in the field of cybersecurity its increasing exponentially. Keylogger is a tool which is rapidly developing because it is rarely recognized as a malicious program by antivirus. We are trying to develop a software which will focus on behaviour based approach of this problem rather focusing on signature based approach. We are going to work on immune based inspired Dendritic Cell Algorithm (DCA). It will perform multi data fusion. We will use malware detecting techniques like Anti Hook technique and HoneyID through this we can overcome these kind of attacks.

**Index Terms** – Dendritic Cell, fusion, malware.

### I. INTRODUCTION

Keylogger is the action of recording the key stroke on a keyboard, typically in a covert manner. Keylogger is a tool that is rapidly developing because this application is very rarely recognized as a malicious program by antivirus, this will quite easily record all activities related to keystrokes. In keystroke a particular insidious type of spyware can record and steal data from consecutive keystroke that the user enters on device Keyloggers steal the confidential information and they completely run in stealth mode. When Keyloggers is installed in a computer, it is not shown either in start-up icons or anywhere else on the computer that is being monitored. The keylogging detection technique discussed here uses active and passive analysis mechanisms when deployed at the host and active analysis at check-points (which means routing components such as a router, firewall, gateway, IDS. etc). We used a keylogger that is written with C# language because many hacker use it and have many function that suitable for this work such as connecting to mail services. For encoding the source codes and string we will use smart assembly and Multimedia Builder. Smart assembly is popular software in encoding domain. For testing the level of encoding we will test keylogger with BinText Tools that it extract all text from any file and we can see unencoded text and for final testing the keylogger again popular antivirus we use online labs such as Jotti and Virus-Total.

#### 1.1 Formulation of Problem

With increasing growth of communication networks, social interactions and financial transactions have been migrate to virtual environments. Internet is one of the most substantial platform for most people's social interactions and transactions. However, the notable challenge in online transactions is security in cyber environments and to understand the hazards accompanied with this communication platform. Because of the increased use of Internet and virtual environments in daily affairs such as financial transactions, this platform has become the focus of attackers and swindlers, for stealing user's information by keyloggers that they are one kind of malwares. The first malware was a virus which was written in early 1980s with the purpose of disrupting stored information in computer systems. Then first network worms were born in 1988 for contaminating SunOS and VAX BSD systems. It attacked these systems through network vulnerability and after inserting, ran a disruptive program on the system. Current situation is also similar to this as every day new advancement in malwares are coming which makes it hard for the softwares to detect the malicious code installed in the systems. Still software companies are working over Anti-Hook techniques and HoneyID techniques for the solution. The keyloggers that most Up-to- date antivirus and anti-adware tool cannot detect them are named as undetectable keyloggers. At first we make a keylogger with C# language, then use smart assembly tools for string encoding the keylogger and Multimedia builder software for embedding the keylogger in another file and at last we test popular antiviruses against keylogger.

#### 1.2 Tools and Technology Used

For fighting against such a thing which is advancing day by day we need to be one step ahead of it. We goanna make keystroke detector which can stand enough against the advancement of malwares. But you have to do these things always :

- Always keep your desktop update.

- Always Use 2 -step verification.
- Install anti malware program.
- Avoid downloading crack software.

Some approaches based on API calls focus on searching only those APIs that can be used to intercept keystrokes, either statically or dynamically. Unfortunately, these APIs are also used by legitimate applications, which makes these approaches heavily prone to false positives. We will use an immune inspired algorithm - dendritic cell algorithm (DCA) to correlate multiple types of API.

The input signals are derived from the frequency of invocations of keystroke tracking functions (PAMPs and safe signal-2), the time difference between two consecutive WriteFile calls (danger signal-1), the relation between different categories of function calls (danger signal-2) and the time difference between two outgoing consecutive communication functions (safe signal-1). The process (identified by Process ID) which causes the calls is defined as antigens. The DCA correlates these antigens with input signals, resulting in a pairing between signal evidences and antigen suspects, and the identification of the keylogger process in the end. However, as the DCA distinguishes between normal and potentially malicious antigens on the basis of neighbouring antigens, the crafty attackers can exploit this correlating feature to evade detection by reducing the 'concentration' of antigens in dendritic cells. We will design a keystroke agent. By invoking system kernel, the agent simulates keyboard event completely. Because a keylogger tracks keystrokes from all applications (including keystroke agent application) in order to log sensitive data entered in them, it could see the simulated keystrokes since they are the same with the real keystrokes. But the keylogger doesn't understand what it sees and it can't tell the keystrokes generated by real users via keyboard from the ones generated by phantom users via our keystroke agent.

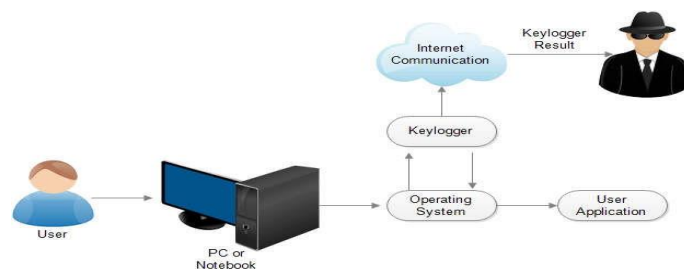


Fig.1 Basic Working of Keylogger

## II. LITERATURE SURVEY

As we know for the prevention of keylogger virtual keyboards plays the most vital role but, apart from that we need to find out some techniques or algorithm which can play important role in evading the malicious program. Currently many research papers have been published related to this regarding the prevention of keystroke attacks. Divyadev Pillai & Irfan Siddavatam in their research paper mentioned that the different keyloggers which are available or being installed are detected using Support Vector Machine learning algorithm. After various analysis the result has been generated and it is counter verified with some of the already available anti-keylogger tools. Also, some Password-based confirmation convention to show how representation can upgrade ease of use and security. Also, some of the recent papers claims that mouse - keystroking could be the future of keystroking. Also, one paper was published by R Sreeram Sreenivas who is currently pursuing his PhD from Anna University, Coimbatore under the supervision of Dr R Anitha, Dr R Anitha, PSG College of Technology, Coimbatore, India describing detection of keyloggers based on traffic analysis with periodic behavior. They described client level logging detection and they performed the testing of keyloggers by attacking systems in which newly and famous antivirus and malware detection softwares were installed like Kaspersky Internet Security 14.0.0.4651, Kingsoft Internet Security 2013, Lavasoft Ad-Aware Free Antivirus, McAfee Internet Security 16.8.708 and more. All of them concluded that the stealthy keylogger cannot be detected by many Antiviruses software as running on the victim's machine. The user has no way to determine the presence of keylogger on his machine, therefore, he turn into a victim of the identity theft. Currently many researches are going on for the proper evasion of keylogger.

## III. WORKING FUNCTIONALITY

### 3.1 Functionality

Keyboard is primary target of most common keyloggers; it consists of matrix of circuit with keys also known as key matrix, there are many different types of key matrix depending on keyboard manufactures. However, the circuit closes key matrix when the user presses key, then keyboard processor and ROM detect this event. The processor translates the circuit location to a character or control code and sends to keyboard buffer. Although they have different implications and different information capturing process, these keylogger share one thing in common; they save captured sensitive data and information in a log file.

**Hardware keylogger :** Hardware keylogger is physical device located between the keyboard and the computer. There are two connection methods; keyloggers can be connected between the keyboard and computer directly. Examples of this method are PS/2 and the USP keylogger.

**Acoustic keylogger :** Unlike hardware keylogger. Acoustic keylogger on analysis and captures the sound of individual keystrokes. Special equipment is required to listen to the sound of the user's typing. Parabolic microphones are utilized to record a long distance, so this microphone is used to pick up the keyboard sound from hundred feet away of targeted area or work.

Wireless keylogger : Wireless keylogger exploits Bluetooth interfaces to transfer captured data to a log file up to the distance of 100M. The primary target of this wireless key logger is to intercept transmitted packet from wireless keyboard that uses 27 MHz RF connection of encrypted RF transported keystroke character.

Software keylogger :Software keylogger intercept data travelling along the keyboard and the operating system. It collects keystroke events, stores them in a remote location, and then transmits to the attacker who installed the keylogger. Research about removal of spyware parasite reported a total of 540 keyloggers and they were mostly software-based. Window operating system has many event mechanisms, for examples, when a character is pressed on the keyboard or mouse clicked; the keyboard driver on the operating system translates this event into window message called WM\_KEYDOWN.

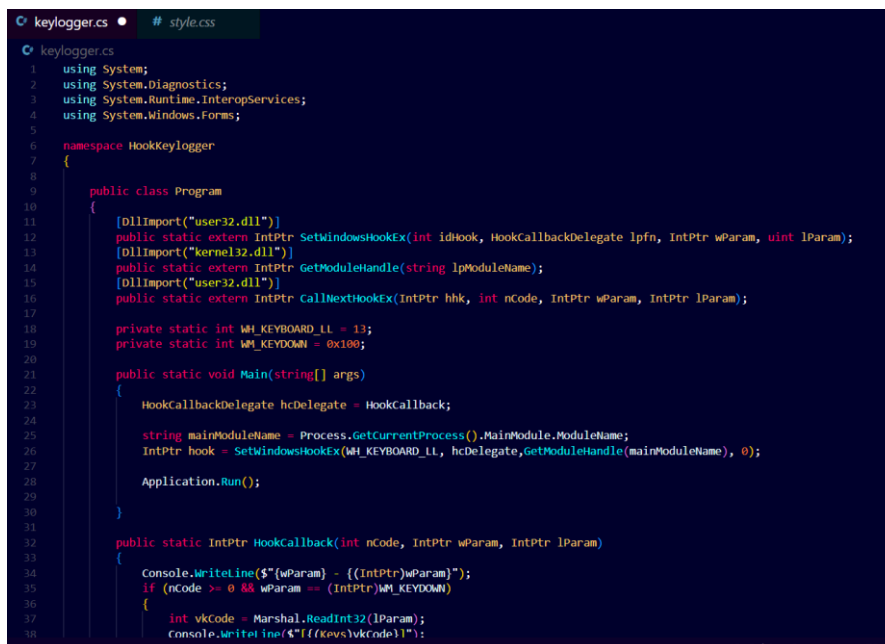
The main idea behind keyloggers is to get in between any two links in the chain of events between when a key is pressed and when information about that keystroke is displayed on the monitor. This can be achieved using video surveillance, a hardware bug in the keyboard, wiring or the computer itself, intercepting input/ output, substituting the keyboard driver, the filter driver in the keyboard stack, intercepting kernel functions by any means possible (substituting addresses in system tables, splicing function code, etc.), intercepting DLL functions in user mode, and, finally, requesting information from the keyboard using standard documented methods.

### 3.2 Making of Keylogger

The most common methods used to construct keylogging software are as follows:

- A system hook which intercepts notification that a key has been pressed (installed using WinAPI SetWindowsHook for messages sent by the window procedure. It is most often written in C.
- A cyclical information keyboard request from the keyboard (using WinAPI Get(Async)KeyState or GetKeyboardState - most often written in Visual Basic, sometimes in Borland Delphi.
- Using a filter driver (requires specialized knowledge and is written in C.

We used a keylogger that is written with C# language. This keylogger can send captured information to predefined email address and use a Gmail account for sending email. Also it save the captured data in txt file in the place that keylogger located. All source code have been append.



```

keylogger.cs # style.css
keylogger.cs
1 using System;
2 using System.Diagnostics;
3 using System.Runtime.InteropServices;
4 using System.Windows.Forms;
5
6 namespace HookKeyLogger
7 {
8
9     public class Program
10     {
11         [DllImport("user32.dll")]
12         public static extern IntPtr SetWindowsHookEx(int idHook, HookCallbackDelegate lpfn, IntPtr wParam, uint lParam);
13         [DllImport("kernel32.dll")]
14         public static extern IntPtr GetModuleHandle(string lpModuleName);
15         [DllImport("user32.dll")]
16         public static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode, IntPtr wParam, IntPtr lParam);
17
18         private static int WM_KEYBOARD_LL = 13;
19         private static int WM_KEYDOWN = 0x100;
20
21         public static void Main(string[] args)
22         {
23             HookCallbackDelegate hcDelegate = HookCallback;
24
25             string mainModuleName = Process.GetCurrentProcess().MainModule.ModuleName;
26             IntPtr hook = SetWindowsHookEx(WM_KEYBOARD_LL, hcDelegate, GetModuleHandle(mainModuleName), 0);
27
28             Application.Run();
29         }
30
31         public static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)
32         {
33             Console.WriteLine($"{wParam} - {(IntPtr)wParam}");
34             if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
35             {
36                 int vkCode = Marshal.ReadInt32(lParam);
37                 Console.WriteLine($"{((Keys)vkCode)}");
38             }
39         }
40     }
41 }

```

Fig. 2 Source Code of Keylogger

### 3.3 Encoding the String of Keylogger

The encoding stage involve some task that describe below. This stage is critical because if this stage done good many security software cannot detect the keylogger. We use smart assembly (a tools that programmer use to secure their code against cracker) for encoding.

Assigning a strong name key : This task protect keylogger from assembly edition and text extraction.

Pruning code : Pruning task removes code that will never be executed at runtime. Pruning also removes metadata such as design attributes and names of events and properties, it is hard for people to use reverse engineering and analyse the keylogger structure. Also this task improve the performance and speed of keylogger and reduce the size of keylogger up to 30%.

Obfuscating with name mangling : Obfuscating task change class name and methods to unreadable name and increase security of code thus it hard for security analyzer to understand the structure of keylogger.

Control flow Obfuscating : Convert source code to spaghetti code. It convert code to complex and unstructured code. It is difficult to read or follow by a people and security software because it cannot be organized and in many time and vice-versa.

References dynamic proxy : This task create a proxy for external call and hide all call to external of codes. This dynamic proxy secure keylogger from security software and increase undetectable rate, so security software cannot track keylogger and after detect it.

Encoding strings :This task protect form passwords (such as passwords, query and information) by encoding them.



Fig. 3 Encoding the strings of keylogger

A sample of social engineering attack. On this task, the key logger will embedded in another file then will set for embedded keylogger like a deceptive icon. Also, on this task the extension of keylogger file will change to another extension (such as jpg, mp4) by Extension spoofing that use security hole in the Windows operating system and allow attacker to change extension of keylogger file to any extension. To do this stage we can use Multimedia Builder software. Now for testing security software against undetectable keyloggers, we used best antivirus currently in the market :

- AVIRA Internet Security 14.0.3.350
- Bitdefender Internet Security 17.26.0.1106
- F-Secure Internet Security 14.99.103
- Fortinet FortiClient 5.0.8.344
- Kaspersky Internet Security 14.0.0.4651
- McAfee Internet Security 16.8.708
- Microsoft Security Essentials 4.4.304.0
- Panda Cloud Free Antivirus 2.3.0

RANK	ANTIVIRUS NAME
1	Kaspersky Lab
2	F - Secure
3	E - Scan
4	Fortinet
5	Emsisoft
6	Bitdefender
7	Lavasoft
8	BullGuard
9	Lavasoft
10	Qiboo (en)
11	McAfee
12	Panda
13	AVIRA
14	Tencent
15	ESET

Table 1 : Report based on detection rates and false alarms

Here, we tried to show that keyloggers can be undetectable from Up-to-date antivirus and anti-malware tool and existing technique can be fail against advanced keyloggers. We explained stage and task of creation of undetectable keyloggers. We described a new challenge that must attract by Antivirus Company’s.

Antivirus Software	Making the Keylogger	Encoding the String	Embedding the Keylogger
Kaspersky Lab	Found Nothing	Found Nothing	Found Nothing
F - Secure	Detected	Found Nothing	Found Nothing
E - Scan	Detected	Found Nothing	Found Nothing
Fortinet	Found Nothing	Found Nothing	Found Nothing
Emsisoft	Detected	Found Nothing	Found Nothing
Bitdefender	Detected	Found Nothing	Found Nothing
360 Internet	Detected	Detected	Found Nothing
McAfee	Found Nothing	Found Nothing	Found Nothing
Panda	Found Nothing	Found Nothing	Found Nothing
AVIRA	Detected	Detected	Found Nothing
Tencent	Found Nothing	Found Nothing	Found Nothing
Trend Micro	Found Nothing	Found Nothing	Found Nothing
ESET	Detected	Found Nothing	Found Nothing
Threat Track Vipre	Found Nothing	Found Nothing	Found Nothing

Table 2 : Scan Results In Every Stage of Making Keylogger

### 3.4 Detection Technique

There are only a few existing techniques for software keylogger detection. Most of these techniques use signature based detection. The biggest disadvantage of this technique is that it has nothing to do against novel keyloggers. Another category of detection method is behaviour based detection. In this technique instead of looking for the specific file signature, the behaviour of the application is scrutinized. As keyloggers always use Windows hooks, Aslam disassembles all running processes searching for SetWindowsHookEx used by some keyloggers to find key logger processes. This method could discover keyloggers never seen before. However, it is easy for keylogger developers to evade this detection technique by using different methods to log the user activities other than using SetWindowsHookEx. Meanwhile, since legitimate applications also use this function to set hooks, this method inevitably has a high false positive rate. In addition, disassembling all processes searching for SetWindowsHookEx is a tedious task.

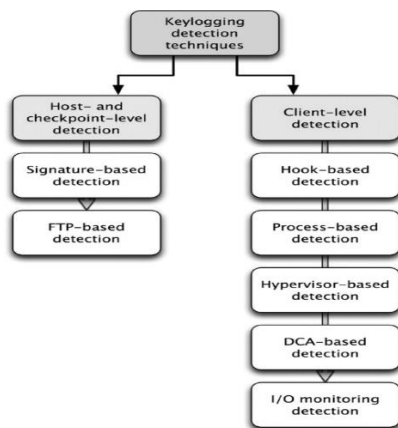


Fig. 4 Classification of existing keylogging detection techniques

Most of the existing methods use client level detection techniques, employing methods such as the hook-based mechanism. This scans all the running processes and the DLLs related to them. It checks the hooks used by these processes and triggers a threat when any suspicious behaviour is noted. The technique requires a lot of computation and the false positive rate is very high. Also one of the famous detection technique is the Dendritic Cell Algorithm (DCA) is used for detect Keyloggers installed on a user computer. The detection is based on correlation between different behaviour such as Keylogging, file access and network communication. Existing keylogging detection techniques work predominately at the client level, or at the host and checkpoint levels using signatures. This motivated us to come up with an anomaly-based detection mechanism that can also be integrated in signature- and log-based detection techniques.

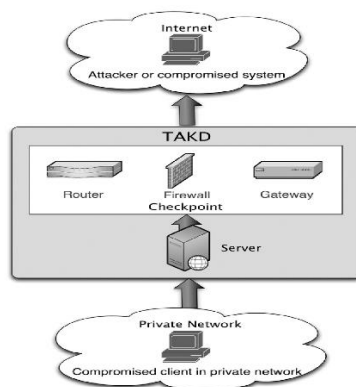


Fig. 5 Overview of traffic analysis keylogger detection

Since a remote keylogger expects to upload the log file to a server, it tries to communicate with the remote host and establishes a connection at regular intervals. This results in intermittent but regular traffic over the Internet. Moreover, the file size is almost identical for every transfer because the interval for log accumulation is preset in the keylogger. This makes the delay between the transfers (latency) constant. Our algorithm is based on traffic analysis based on such periodic behaviour, so keyloggers that exhibit non-periodic behaviour will be ignored. The important observations here are that the source and destination do not change. However, genuine traffic uploads occur from the same source to several destinations with random time intervals.

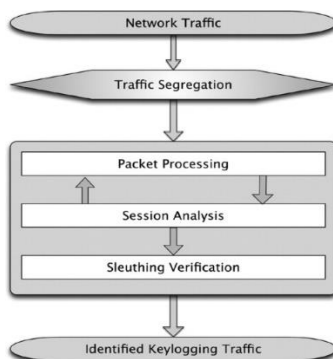


Fig. 6 Block diagram of keylogging detection (TAKD mechanism)

#### IV. RESULTS AND DISCUSSIONS

According to our detection algorithm Keylogging detection using traffic analysis consists of four modules. They are: traffic segregation; packet processing; session analysis; sleuthing verification.

The full TAKD algorithm consists of six steps:

Step 1 : Segregate the TCP traffic.

Step 2 : Identify the type of log sending mechanism i.e. FTP. SMTP. SMB. web hosting, etc.

Step 3 : Accumulate the source and destination details – i.e. source IP and port, destination IP and port, etc.

Step 4 : Calculate consecutive delay latencies (DLi).

Step 5 : Verify if these delay latencies are equal. If so, drop the traffic and alert the victims.

Step 6 : Else, release the traffic segregated from the network.

While examining log-sending mechanisms, we found that every keylogger had features such as variable time intervals between 1 min and 60 hours. The time interval is fixed by the attacker at the time of keylogger installation. Log size varies between 1 MB and 9.999MB. Moreover, there are options like filters that enable the keylogger to collect only specific data. This study raised some questions, such as what features or options an attacker might use to maintain stealth. Some of the properties of the keyloggers tested are shown in Table 1. The keyloggers were studied and tested in real time to verify features - for example, that the latency of the keylogging traffic is constant because the log sending interval is fixed by the attacker.

Investigating the consecutive flows generated from a source to the same destination with the constant latency helped in identifying the keylogging flow with periodicity among other network flows. Figure 7 shows the normal network traffic of our network. Figure 8 shows the FTP traffic segregated from the network traffic. Figure 9 is the SMTP traffic segregated from the network traffic. Figure 10 shows the network share traffic segregated from the network. Figure 11 shows the keylogging flow generated by one of the tested keyloggers. And Figure 12 shows keylogging traffic identified by the TAKD algorithm.

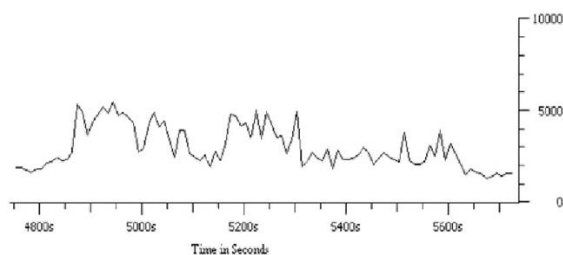


Fig. 7 Normal Network Traffic

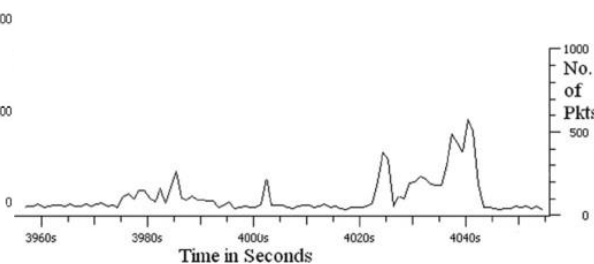


Fig. 8 FTP traffic segregated from network

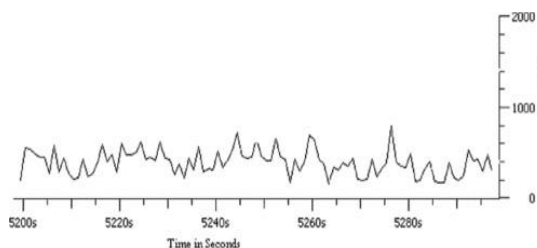


Fig. 9 SMTP traffic segregated from network

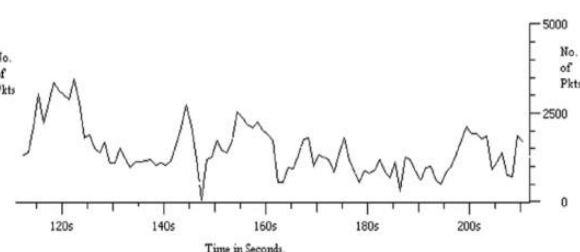


Fig. 10 Network share traffic segregated

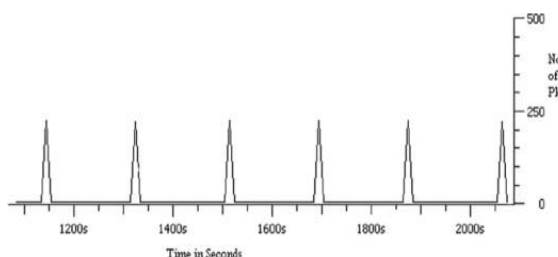


Fig. 11 Traffic flow generated by keylogger

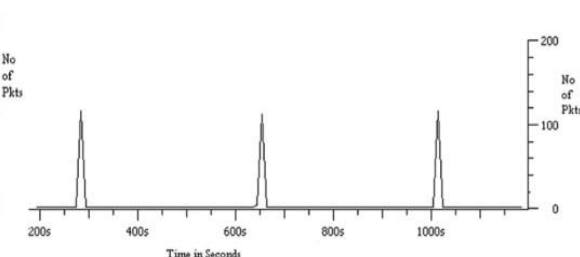


Fig. 12 Keylogging traffic identified by TAKD

#### V. CONCLUSION

This paper mainly focuses on various software Keyloggers and its effects on computer system. Keyloggers records all the keystrokes and user activities on the computer, steals confidential information like password, credit card number and send to the attacker. The TAKD algorithm provides an accurate means of detecting keyloggers by traffic analysis. The algorithm can be easily incorporated into routing devices such as a router, gateway, firewall, IDS and so on to increase its keylogging detection efficiency. Future work might include extending the detection algorithm to perform quantitative analysis for irregular time intervals - for example, where botnets or P2P servers are employed.

**VI. REFERENCES**

- [1] "Keyloggers in Cybersecurity Education," Christopher A. Wood, Rajendra Kk. Raj, Proceeding of the 2010 International Conference 2010.
- [2] "Detecting Bots Based on Keylogging Activities," Y. Allammadi, U. Aickelin, In the Proceeding of 3rd International Conference on Availability, Reliability and Security, pp.896-902, 2008
- [3] "Detecting Software Keyloggers with Dendritic Cell Algorithm," Jun Fu, Yiwen Liang, Chengyu Tan, Xiaofei Xiong, In Proceeding of the 2010 International Conference on Communication and Mobile Computing, pp.111-115, April 12-14, 2010.
- [4] Don't Fall Victim to Key loggers: <http://www.makeuseof.com/tag/dont-fall-victim-to-keyloggers-use-these-importantanti-keylogger-tools/> Last accessed: Jan 2014.
- [5] "How to Login From an Internet Café Without Worrying About Keyloggers," Cormac Herley, Dinei Florencio, In Proceeding of the Association for Computing Machinery, Inc, pp.2, July 2006.
- [6] "HoneyID: Unveiling Hidden Spywares by Generating Bogus Events," J. Han, J.K won, H. Lee, In the proceeding of IFIP 23<sup>rd</sup> International Information Security Conference, pp.669-673, 2008.
- [7] "Overview of Detecting Keyloggers:" <http://www.sandboxie.com/> Last accessed: Feb 2014.
- [8] "Kaspersky Lab Keyloggers: How they work and how to detect them." <http://www.viruslist.com/en/analysis?pubid=204791931>. Last accessed: Jan, 2014.
- [9] "Security Technology Ltd. Testing and reviews of keyloggers, monitoring products and spy software." <http://www.keylogger.org>. Last accessed: Dec 2013.