Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FOUR-PHASE PROTOCOL FOR DETECTION, DELETION, PROTECTION AND RECOVERY FROM AUTORUN VIRUS

[1]KHALID M. HOSNY, [2]AMEER E. GOUDA, [3]EHAB R. MOHAMED

[1]Vice Dean, Faculty Of Computers And Informatics, Department Of Information Technology, Zagazig University, Zagazig 44519, Egypt.

[2]Teaching Assistant, Faculty Of Computers And Informatics, Department Of Information Technology, Zagazig University, Zagazig 44519, Egypt.

[3]Lecturer, Faculty Of Computers And Informatics, Department Of Information Technology, Zagazig University, Zagazig 44519, Egypt..

ABSTRACT

Keeping any computing system stable is a very sentenced job, since virus programs can easily infect any computing system via external devices such as pen drive, floppy disk, Memory sticks, DVDs, CD ROMs, and USB devices. The auto run virus is one of the most dangerous attacks that affect different computing system. In this paper, a detection, deletion, protection and recovery (DDPR) protocol is proposed. This protocol consists of four phases. The first phase of the protocol is working toward detecting the system's infection with the virus. If the virus is detected, the second phase of the protocol is deleting the virus and all of its associated accessories from the system. Then the third phase is a full protection of the computing system from possible future infection by this virus. Finally, the fourth phase of the protocol is the process of recovering the destroyed or tampered data/information. The proposed DDPR protocol introduces a lightweight contribution that is represented in detecting and preventing both autorun virus and also the DDoS attack at it's early stages. DDPR has been tested with different computer devices and platforms. The obtained results clearly show that the proposed protocol is superior to all existing algorithms and solutions.
Keywords: *Virus, Autorun Virus, Protection, System Maintenance, DDPR Protocol.*

1. INTRODUCTION

An autorun virus is a virus that spreads the autorun.inf file in the operating system of windows which in turn results in launching auto play malicious programs and files that are stored in external devices [4]. The autorun virus exploits that the auto play feature is enabled by default in the operating system of windows to infiltrate to the computing system causing tremendous prejudice [5]. It moreover duplicates itself onto the computing system by creating a set of copies of autorun.inf and executable files on all drives of the system [6]. The virus surreptitiously may force the user to open malicious websites [7]. It might also install a key logger in the computing system that can capture web site activity, login credentials, usernames, passwords, account numbers, credit card information and other personal and sensitive information [8]. The autorun virus has many harmful effects on the network, as in some cases the virus enforces the user to be directed to open suspicious sites or route traffic to a particular server which may exhausts network resources or cause the service to break down from that server, the so-called denial-of-service attack[9]. Huge amount of work has been done in removing the autorun virus. Among these works the (USB FlashDrive Autorun Antivirus) application from Abhi-soft Technologies Company. (USB Flash Drive Autorun Antivirus) is an antivirus for only USB drive. It tries to provide protection against autorun viruses. It automatically detects and deletes the virus from drive but it has insufficiency speed in scanning and consuming the system resources and work only on flash drives [10].One other solution in dealing with the autorun virus was (Autorun Virus Remover) . It tries to protect computers from malicious threats of the virus infected portable disks. It scans the USB device when it is inserted into the computer and the application will detect the threat and try to delete it. One of the most important disadvantages of this application is that it does not make any future protection from the virus. It failed with experiment in recovering the files may be damaged by the virus and also it is not free [11].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This paper aims to propose an integrated and efficient protection system against the autorun virus to protect the main components of the network namely the client and the server. The motivation behind this research was fixing the insufficiency of the previous methods and remedying their shortage.

This research proposed a new hybrid four phase detection, deletion, protection, and recovery (DDPR) protocol that detects, deletes autorun virus from the infected devices. It also protects the computing system from infecting again and protects the network from exhausting its resources with a fake traffic launched by the virus and recovering the infected data due to the autorun or any other virus.

The rest of this paper is organized as follows. In section 2, the proposed DDPR protocol is presented. The simulation using three different platforms is presented in section 3. Finally, Section 4 concludes this paper and introduces the limitation of the solution then the future work will be discussed as an entry to the protection of hosts in SDN.

2. THE PROPOSED PROTOCOL

An autorun virus publishes an automatic autorun.inf files in a windows operating system that, in turn, starts automatic malicious software and files stored in external devices[12].In general, to erase any file that contains any of these four attributes (system, hidden, archive and read-only) those attributes must be removed from the file first [13]. As is known the autorun.inf file may contain (system, hidden, archive and read-only) attributes thus it is needed to strip these attributes from autorun.inf file before deleting it. As a result, the proposed DDPR Protocol depends on Four-Phase solution
(Detection-Removal-Protection-Recovery) that first work on removing the four attributes (system, hidden, archive and read-only) then detects, deletes autorun virus from the infected devices. It protects the computing system from infecting again by inserting a fake file named "AUTORUN.INF" in the root of all drivers of the computing system and its peripherals. Finally, it recovers the infected data due to the autorun or any other virus as seen in the flowchart in figure (1).

The DDPR works on the root of the drivers deleting the origin of the virus. The rest of virus copies become a dead copy with no effect as shown in the following steps:

1. Check if malware detected in the computing system.
2. Check if drive contains autorun.inf file.
3. Check if autorun.inf file has malware entry.
4. Strip the system, hidden, archive and read-only attributes from all hidden files.
5. Scan to detect the autorun.inf virus file.
6. Detect autorun.inf virus files.
7. Remove detected autorun virus file.
8. Spread a fake file named AUTORUN.INF with all letters in upper case to be different from the virus file and give the AUTORUN.INF the four attributes (system, hidden, archive and read-only) so that it can't be deleted by any future Attack.
9. Recovering the files that the virus has been damaged.

And the pseudo code of the proposed DDPR protocol is shown below in table 1.

3. SIMULATION

The DDPR had been evaluated through a variety of criteria, namely detection accuracy, protection accuracy, scan time, resource consumption (utilization), average recovery time, single drive recovery option, application freeware, application size and application setup/portability. The next subsections will discuss the experimental results and evaluation of applying the DDPR on each of the desktop computer, the laptop, and the server in different environments.

3.1 Specifications Of Platforms Used In Testing DDPR Protocol

DDPR has been tested on many different devices and platforms. It has been tested on a desktop computer, laptop and a real server. For the desktop computer, it was of type Dell with a processor of a Liquid-Cooled Intel i7-6700K 4.2GHz Boost Quad-Core and ram of 16GB DDR4. While the laptop was of type Dell Latitude E-6420 with a processor of an Intel –core i5-2520M CPU @ 2.50 GHz (4 CPUs), ~ 2.5 GHz and ram of 8GB DDR3. Finally, the server was of type Dell with a processor of Dual Xeon E5372 Quad Core (Clover town) 2000MHz and ram of 64096MB DDR Registered ECC. These extensive experiments were done on different operating systems as listed in table (2). The DDPR also has been tested on a

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

real local network by using a set of devices connected to a switch of a star topology shape [14]to demonstrate the effect of the Protocol's application on the network. Network components, specifications and configurations are listed in table(3).

3.2 Detection, Deletion, Protection Phases Of DDPR Protocol

3.2.1 Implementation of DDPR protocol on the PC

DDPR has been tested on a Dell desktop computer with specifications shown in Table 1 and it also recorded a rate of 100% accuracy and efficiency in detecting –removing and protecting the computer as shown in figure (2).

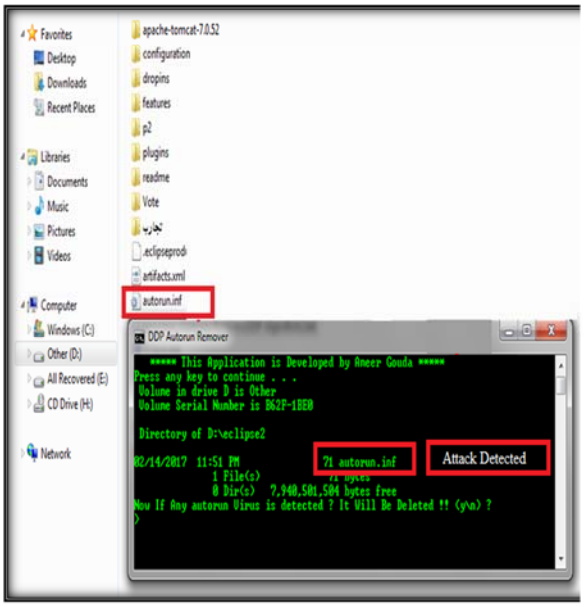


Figure 2: Detection Phase Of The Proposed DDPR Protocol.

3.2.2 Implementation of DDPR Protocol on Laptop

DDPR has been tested on a Dell Latitude E-6420 laptop with specifications shown in Table 1viewed above and it also recorded a rate of 100% accuracy and efficiency in detecting –deleting and protecting from the autorun virus as shown in figure (3).

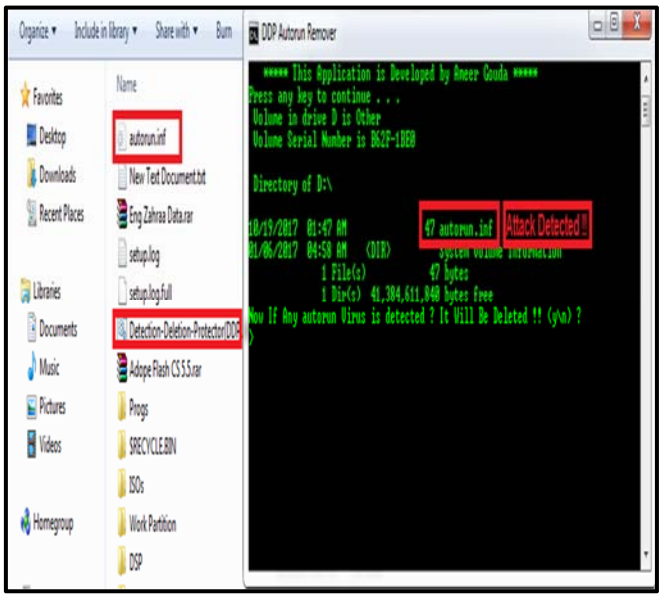


Figure 3: Detection Phase Of The Proposed DDPR Protocol.

After testing DDPR on both a desktop computer and Laptop, it was discovered that it does not consume from either CPU or memory compared with the normal state were the DDPR is not running as shown in figure(4) and figure (5)respectively.

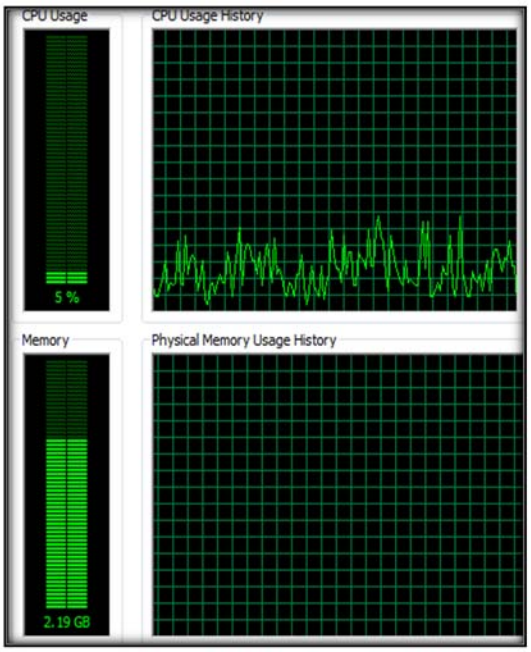


Figure 4: Memory/Cpu Consumption While The DDPR Protocol Is Running.

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

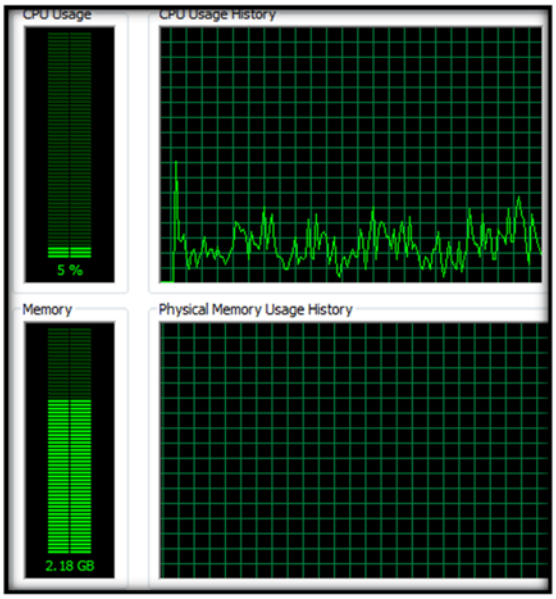


Figure 5: Memory/Cpu Consumption Without The DDPR Protocol Is Running.

3.2.3 Implementation of DDPR protocol on a real server in a real network

DDPR has been tested on a real server and it has recorded a rate of 100% accuracy and efficiency in detecting –removing and protecting the server as shown in Figure (6).The autorun virus has many harmful effects on the server and the entire network connected to it [15]. As in some cases the virus compels the user to go to open suspicious sites or route traffic to a particular site [16]. That may cause the service to break down from that site, the so-called denial-of-service attack [17]. It may also cause the consumption of all device resources, resulting in the user losing control of his personal computer [18]. This is contrary to the case that the server is part of a network of servers that can be attacked all of which could lead to a verified disaster [19].

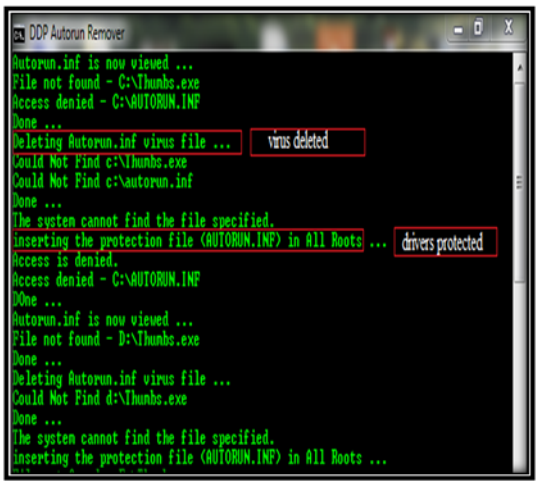


Figure 6: The Two Phases Deletion And Protection Of The Proposed DDPR Protocol.

3.2.4 Effects of applying DDPR on the whole devices on a real network

Autorun virus file (autorun.inf) can launch a DDOS Attack by launching executable files that run an attack on a host or a server[20] as seen in Figure (7) and Figure (8) causing:

1. The host to stop working as shown in Figure 7 as the attack exhausts all the host resources.
2. Exhaust the server resources.
3. Prevent legitimate users from accessing the server.
4. Increase the network traffic as shown in Figure (13) captured by Wireshark during the attack.

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 7: DDOS Attack Launched By Autorun Virus And The Resource Monitor During The Attack.

By using the proposed protocol, the host, the server and the network are protected. The developed solution removes the autorun virus file that is responsible for launching a DDOS attack on determined host, server or a subnet of hosts or servers. Network Specifications were DDPR has tested are mentioned in Table (3) below.

Case study 1: Single host attacking a subnet

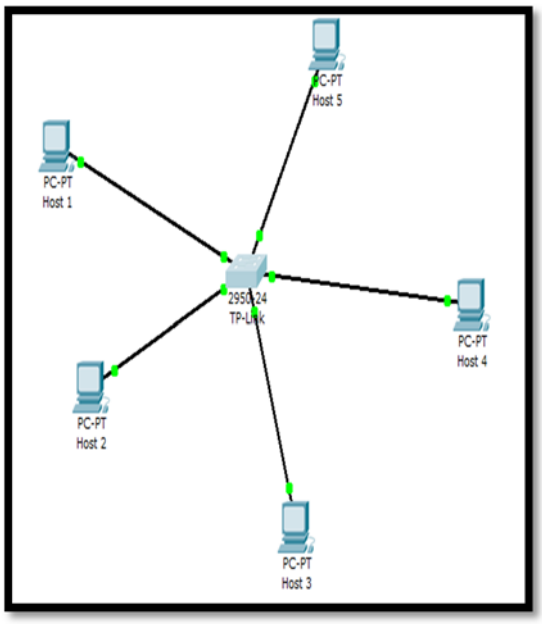


Figure 8: Testing DDPR On A Real Network Of A Star Topology.

The protocol has been implemented on a local network by connecting five devices through the switch as a star topology as shown in figure (8).

The virus has been launched on the first machine with address 10.0.0.2; the device has launched executable files that attacked other devices on the network. It sends too many packets at the same time causing the other four hosts to stop working and exhausted the network resources in processing a fake traffic. The attack packets were captured using the Wireshark[21] application as shown in figure (9).

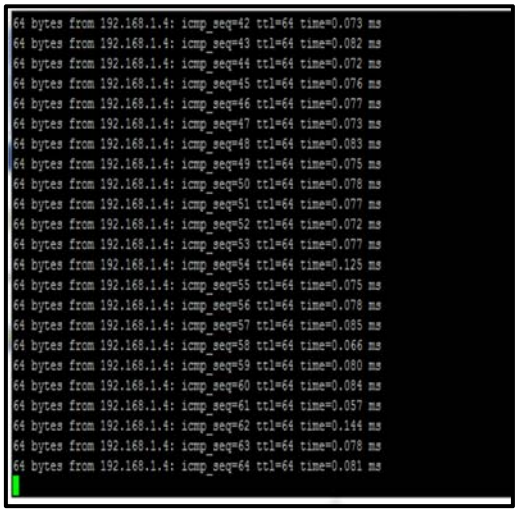


Figure 10: Host Of Address 192.168.1.4 Is Under ICMP Flooding Attack.

Case study 2: Single host attacking a server.

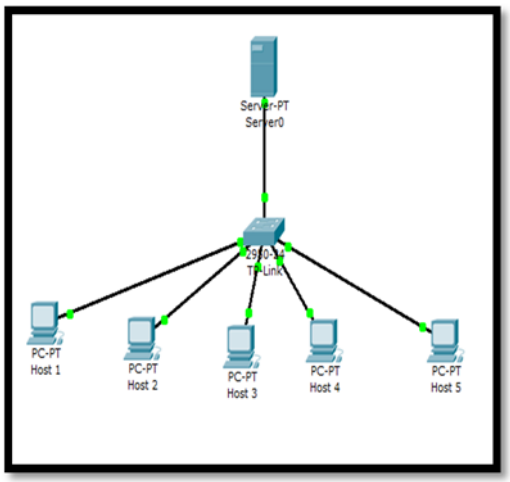


Figure 11: Network Topology Of A Host Attacking A Web Server.

In the above case the host of IP address: (192.168.1.2) is attacking the server of IP address:(192.168.1.254). The attack is launched by an autorun file developed to launch malicious

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

executable files. It sends too many requests to the server causing the system resources to be exhausted and the server to break down. That type of attack called denial of service attack. Wireshark was used to capture the attack packets as seen in figure (12).

Figure (13) shows a DDOS attack on a real server of IP address (193.227.29.20) and the wire shark was used to capture the attack packets sent to the server to exhaust its resources and break it down. As noted, there exist large numbers of packets in which the server is attacked because of an executable file launched by the autorun virus. This executable file launched by the autorun virus carries malicious instructions that make a denial of service attack, which could cause the server to break down. On the other hand, figure (14) shows the network status of DDPR was running with stability and normal packet traffic.

3.3 Implementation Of The Recovery Phase Of The DDPR Protocol

DDPR has run on different devices and obtained the same result with a rate of accuracy of 100% in recovering the files after the virus damage as shown in the figure. DDPR Protocol Application has detected three hidden files named (Applications folder, Multimedia.txt file and Explanatory video folder). They can't be retrieved by the operating system tools as listed in Figure(15). DDPR Protocol has recovered them in just 11 seconds. The DDPR is a fast protocol in recovering files and folders compared with other applications like "USB FlashDrive" application or "USB SHOW"[22] application used to recover files as shown in figure 16.

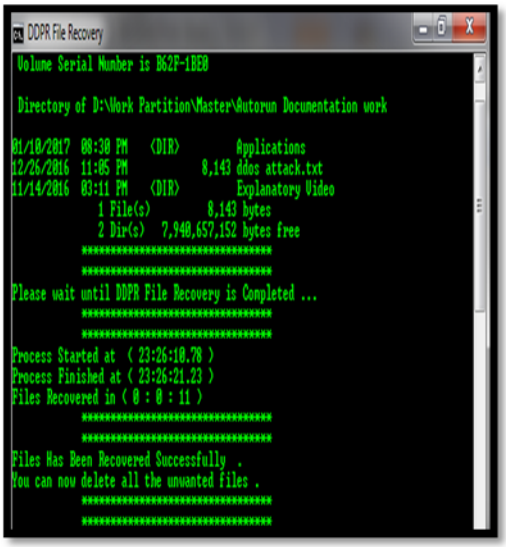


Figure 15: The Recovery Phase Of The Proposed DDPR Protocol.

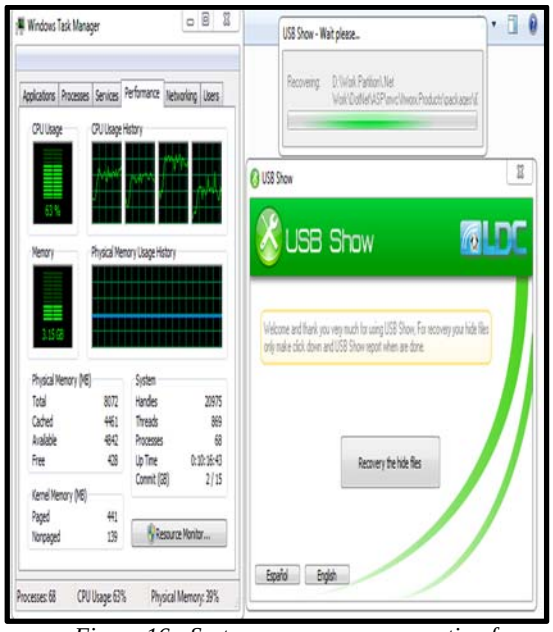


Figure 16: System resources consumption for some commercial applications.

3.4 Comparison With Other Commercial Applications

The DDPR Protocol has recorded very low consumption of system resources with 0.0001% memory usage and no CPU consumption as shown in figure (17) compared with other commercial applications such as "Autorun Remover " which recorded a very high CPU consumption with 39% and memory usage of 33.7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

% as shown in figure (18) and a descriptive graph is given in figure (19).

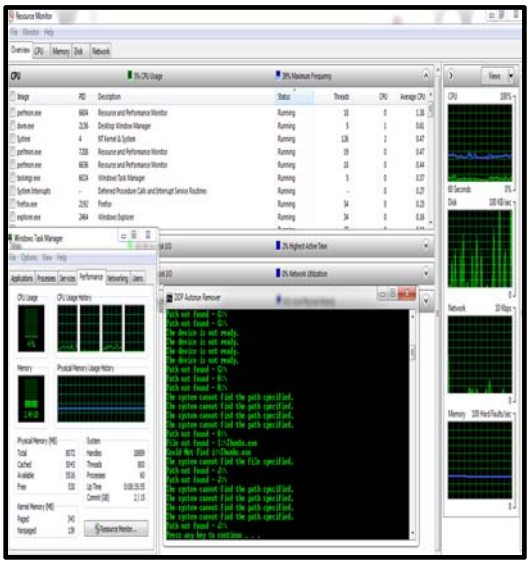


Figure 17: System Resources Consumption By DDPR Solution.

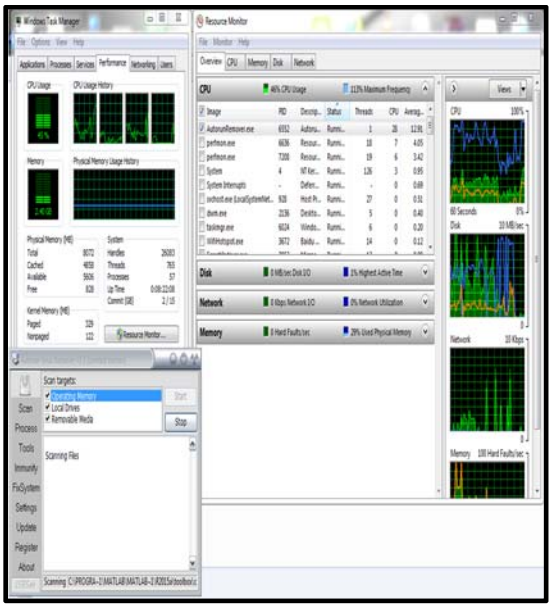


Figure 18: System Resources Consumption By Some Commercial Programs "Autorun Remover".

It is clearly shown that DDPR record a minimum and optimum CPU and memory consumption compared to previous solutions. The proposed protocol had been evaluated through a variety of criteria, namely detection accuracy, protection accuracy, scan time, resource consumption (utilization), average recovery time, single drive recovery option, application freeware, application size and application setup/portability. The criteria of evaluation and general evaluation results are listed in table (4) below.

And there exist some important notes on the test results as mentioned below:

- (Autorun Remover) application shows error messages "Failed to access drivers" in the deletion process and therefore the application accuracy is very weak.
- (USBflash drive) application has run 25 times. It successfully detected and deleted the virus for 16 times and failed in deleting the virus 9 times with accuracy of 64 %.
- Both the (Autorun Remover) and the (USBflash drive) applications failed in the protection phase. They insert a fake file named "autorun.inf". By experiment, it is found that this file can be easily deleted by the virus while DDPR protocol inserts a file with a system privileges that can't be deleted by the virus. That way the virus cannot access hard drives because there is an existing file with the same name and so it cannot publish its malicious files to the system and so the full protection process is achieved.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- DDPR has been tested on different environments. In each environment tested for 25 times run after launching attack and successfully detected and deleted the virus with detection and deletion accuracy of 100 %
- DDPR has a minimum scan and detection time of (1-2) seconds based on the size of data. The runs were done on 500 GB hard disks of different types and different platforms.
- DDPR has the feature of single drive recovery that enables the system administrator to make the recovery process on a specific drive not on all of the system unlike previous applications that do not have this feature.
- DDPR has a minimum and optimum recovery time. It has been tested on different sizes of data and recorded a maximum of 13.4 seconds on 300 GB drive while (USB flash drive) took a very long time to recover the same storage space and data types. There was no output seen for 39 minutes of continuous scan.
- DDPR consumes the system resources lightly as it has only 0.003 % CPU load and 0.0001% memory usage.

4. CONCLUSION AND FUTURE WORK

One limitation in the proposed solution is that it is applicable only on the traditional networks not on the software defined networks. In this paper, the problems caused by the autorun virus have been identified and a new hybrid four phase detection, deletion, protection and recovery (DDPR) protocol has been developed targeting protecting the computing system including both the client and the server. These problems are the core of system and network security. The proposed solution was efficient in detection with accuracy of 100%. The proposed solution provides the advantage of protection with accuracy of 100%.

The proposed solution does not increase CPU load or memory usage compared with previous solutions. The proposed protocol's application is portable (low size / no setup). The proposed solution has the advantage of recovering damage done by the virus file (data recovery).Data recovery process has low processing and recovery time compared with current methods and applications. The proposed protocol has the feature of single drive recovery enabling the system administrator to make the recovery process on a specific drive not on all of the system. In this research, it has focused on a solution that works particularly on the future prevention and protection and recovery for the client, the server and the network from the autorun virus as well. The future work will be presenting the autorun and DDoS detection and prevention scheme in the software defined networks .

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1]Nir Nissim , Ran Yahalom, Yuval Elovici ," USB-based attacks " , Computers & Security , vol .70 , pp. 675-688 , Sep 2017 .

[2] JeremyFaircloth, " Chapter 10: Building penetration test labs - Penetration Tester's Open Source Toolkit (Fourth Edition) “, pp. 371-400 , 2017.

[3] Nihad Ahmad Hassan, Rami Hijazi ,"Chapter 6: Data Hiding Forensics - Data Hiding Techniques in Windows OS ", pp. 207-265 , 2017 .

[4] Jeremy Faircloth, " Chapter 8: Client-side attacks and social engineering - Penetration Tester's Open Source Toolkit (Fourth Edition) ", pp. 273-318,2017.

[5] John Pirc, David DeSanto,Iain Davison,Will Garrido, " Chapter 3: Security Intelligence -Threat Forecasting, “, pp. 29-45,2016.

[6] Eric Conrad,Seth Misenar,Joshua Feldman," Chapter 8: Domain 7: Security Operations (e.g., Foundational Concepts, Investigations, Incident Management, Disaster Recovery) - CISSP Study Guide (Third Edition), pp.347-428, 2016.

[8] Eric Conrad, Seth Misenar, Joshua Feldman, " Chapter 4: Domain 3: Security Engineering (Engineering and Management of Security) -CISSP Study Guide (Third Edition) ", pp.103-217,2016.

[9] Eric Conrad, Seth Misenar, Joshua Feldman, " Chapter 2: Domain 1: Security and Risk Management (e.g., Security, Risk, Compliance, Law, Regulations, Business Continuity) - CISSP Study Guide (Third Edition) ", pp.11-79,2016.

[10] AzizMohaisen,Omar Alrawi, ManarMohaisen," AMAL: High-fidelity, behavior-based automated malware analysis and classification ", Computers & Security,Vol.52,pp. 251-266,Jul 2015.

[11] Eric D. Knapp,Joel Thomas Langill, " Chapter 3: Industrial Cyber Security History and Trends - Industrial Network Security (Second Edition), " pp.41-57,2015.

[12] HarlanCarvey, " Chapter 6: Malware Detection - Windows Forensic Analysis Toolkit (Fourth Edition), " pp.169-209,2014.

[13] Lawrence J. Fennelly, Marianna A. Perry, " 150 Things You Should Know about Security (Second Edition), " , pp. 1-218 , 2018 .

[14] José Luis Fernández-Alemán, Ana Sánchez-Henarejos, AmbrosioToval, Ana Belén Sánchez-García, Luis Fernandez-Luque, " Analysis of health professional security behaviors in a real clinical setting: An empirical study ", International Journal of Medical Informatics, Vol .84, Issue 6,pp.454-467, June 2015.

[15] Eric D. Knapp,Joel Thomas Langill, " Chapter 7: Hacking Industrial Control Systems - Industrial Network Security (Second Edition)," pp.171-207,2015.

[16] AndreasMoser, Michael I. Cohen, " Hunting in the enterprise: Forensic triage and incident response ", Digital Investigation, vol.10, issue 2, pp. 89-98,Sep 2013.

[17] PatrickEngebretson, " Chapter 5: Social Engineering - The Basics of Hacking and Penetration Testing (Second Edition), " pp.127-140,2013.

[18] Aditya K. Sood, Richard J. Enbody , RohitBansal , " Dissecting SpyEye – Understanding the design of third generation botnets " ,Computer Networks , vol. 57 , Issue 2, pp.436-450 , February 2013 .

[19]EliasBou-Harb,Nour-EddineLakhdari,Hamad Binsalleeh,MouradDebbabi,"Multidimensional investigation of source port 0 probing " Digital Investigation,Vol.11, Supplement 2, pp. s114-s123, August 2014.

[20] Marcelo Ayres Branquinho, Jan Seidl, Leonardo Cardoso de Moraes, Thiago Braga Branquinho, Jarcy de Azevedo, " Capítulo 6: Malware e armascibernéticas - Segurança de Automação Industrial e SCADA, ", pp.91-116, 2014.

[21] Wireshark application at the application website "https: //www.wireshark.org/download.html ".

[22]Paolo Palumbo, LuizaSayfullina, DmitriyKomashinskiy, Emil Eirola, JuhaKarhunen , " A pragmatic android malware detection procedure " , Computers & Security, Vol. 70, pp. 689-701 , Sep 2017 .

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] Muhammad Shamraiz Bashir,Muhammad Naeem Ahmed Khan, " A triage framework for digital forensics ", Computer Fraud & Security, Vol. 2015, Issue 3, pp .8-18, March 2015.

[24] Paul Black, IqbalGondal, Robert Layton,"A survey of similarities in banking malware behaviours", Computers & Security, in press, accepted manuscript, Available online 9 October 2017.

[25] Dung Vu Pham, Ali Syed, Malka N. Halgamuge, " Universal serial bus based software attacks and protection solutions ", Digital Investigation, Vol.7, Issues 3–4, pp.172-184 , April 2011 .

[26] Brian Anderson, Barbara Anderson, " CHAPTER 3: USB-Based Virus/Malicious Code Launch- Seven Deadliest USB Attacks ", pp.65-96,2010.

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

List of Large Figures/Tables :

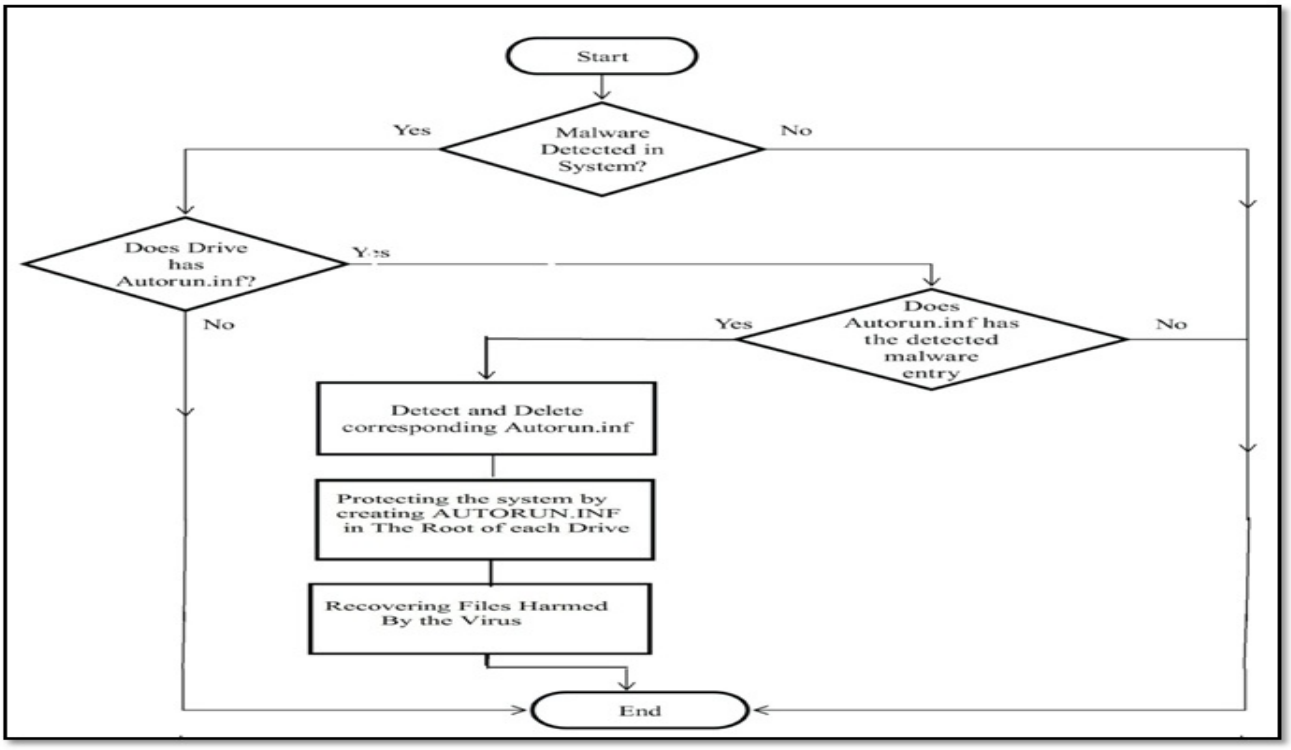


Figure 1: Flowchart Of The Proposed DDPR Protocol.

Table 1: The Pseudo Code Of The Proposed DDPR Protocol.

```
If (malware detected in drive) then
{
If(drive has autorun.inf) then
{
If(autorun.inf has the malware entry) then
{
Detect and delete autorun.inf
}
{
Protect(create a fake AUTORUN.INF file in each drive root )
Recover  files that the virus has been damaged
}
```

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2 - Specifications of platforms used to test DDPR protocol.

Specifications	Platform		
	Desktop Computer (Dell)	Laptop (Dell Latitude E-6420)	Server (Dell)
CPU	Liquid-Cooled Intel i7-6700K 4.2GHz Boost Quad-Core	Intel –core i5-2520M CPU @ 2.50 GHz (4 CPUs), ~ 2.5 GHz	Dual Xeon E5372 Quad Core (Clover town), 2000MHz
RAM	16GB DDR4	8GB DDR3	64096MB DDR Registered ECC
Hard Drive	400GB SSD + 2TB HDD	500 GB	More than 6000GB.
Graphics Card	Dual NVIDIA GeForce GTX 980 Ti	Intel(R)HD Graphicd 3000	NVIDIA® graphics. HD Graphic 3000
Operating System	Windows 10 Home 64-Bit	Windows7Ultimate 32-bit (6.1, build 7601)	Windows server 2003

Table 3: Network Components, Specifications And Configurations Used To Test DDPR Protocol.

Device / Settings	Display name	IP Address	IP Allocation Method	Mac Address	Connection Topology Type	Subnet mask
Switch	TP-Link	None	None		Star	None
PC	Host 1	192.168.1.1	Static	00:01:C7:2A:4E:5E	Star	255.255.255.0
PC	Host 2	192.168.1.2	Static	00:10:11:C7:EC:78	Star	255.255.255.0
PC	Host 3	192.168.1.3	Static	00:0D:BD:77:70:C4	Star	255.255.255.0
PC	Host 4	192.168.1.4	Static	00:60:70:2B:43:63	Star	255.255.255.0
PC	Host 5	192.168.1.5	Static	00:0B:BE:0C:B2:45	Star	255.255.255.0
Server	Server0	192.168.1.254	Static	00:90:0C:B6:D6:42	Straight-through	255.255.255.0

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

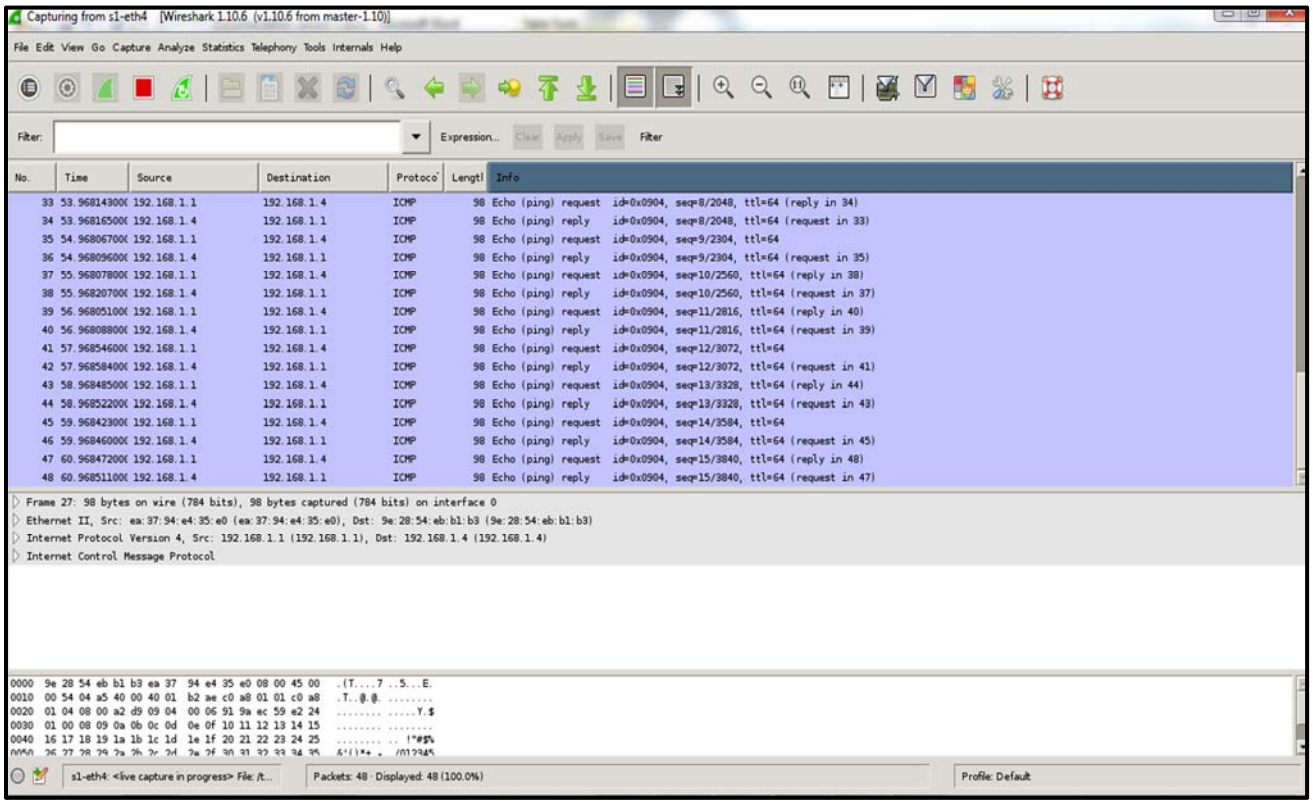
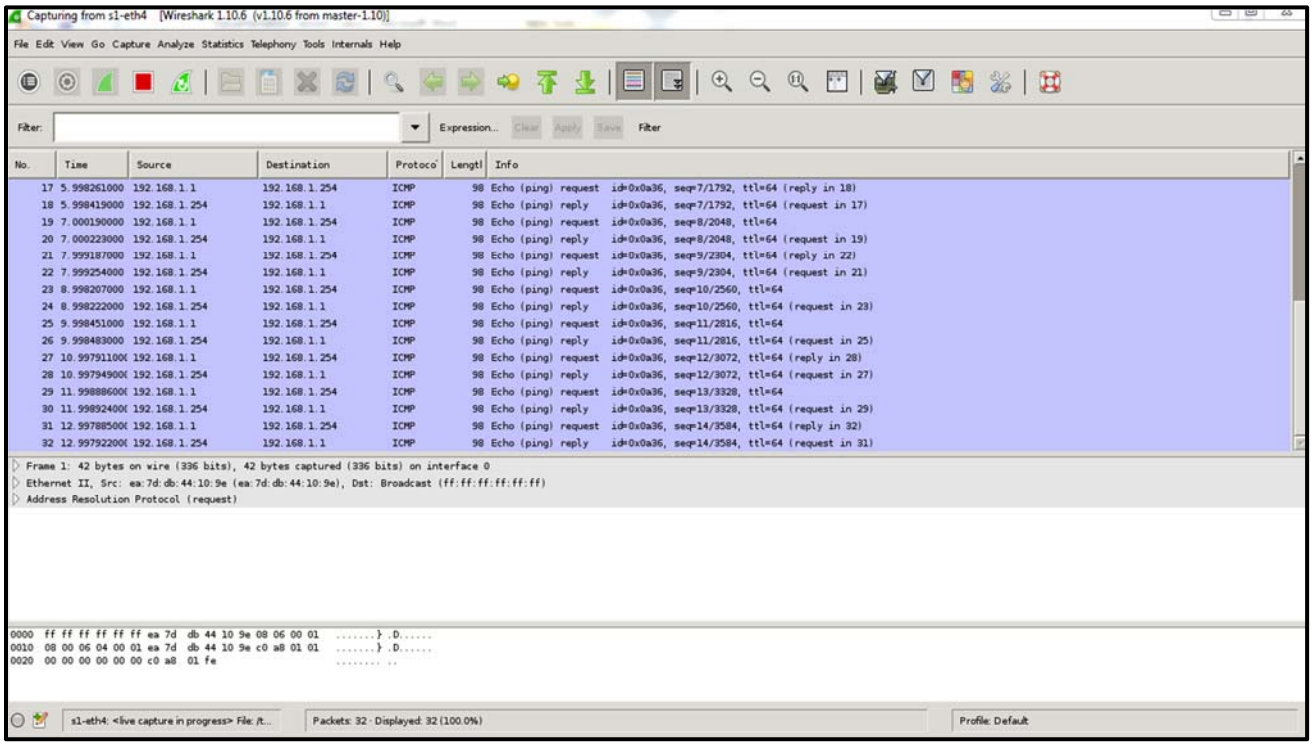


Figure 9: Host Of Address 192.168.1.4 Is Under ICMP Flooding Attack.



ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 12: Web Server Of Address 192.168.1.254 Is Under ICMP Flooding Attack.

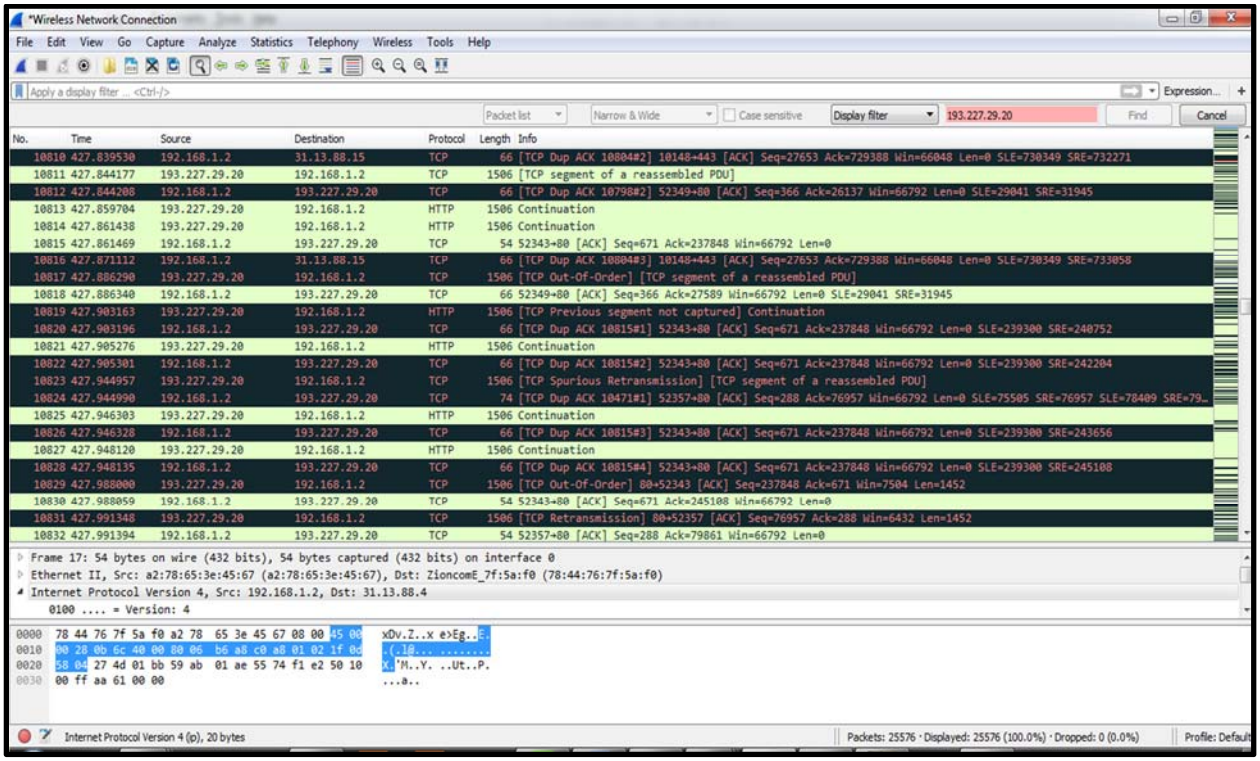


Figure 13: Wireshark Captures DDOS Attack Packets On A Real Server.

Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

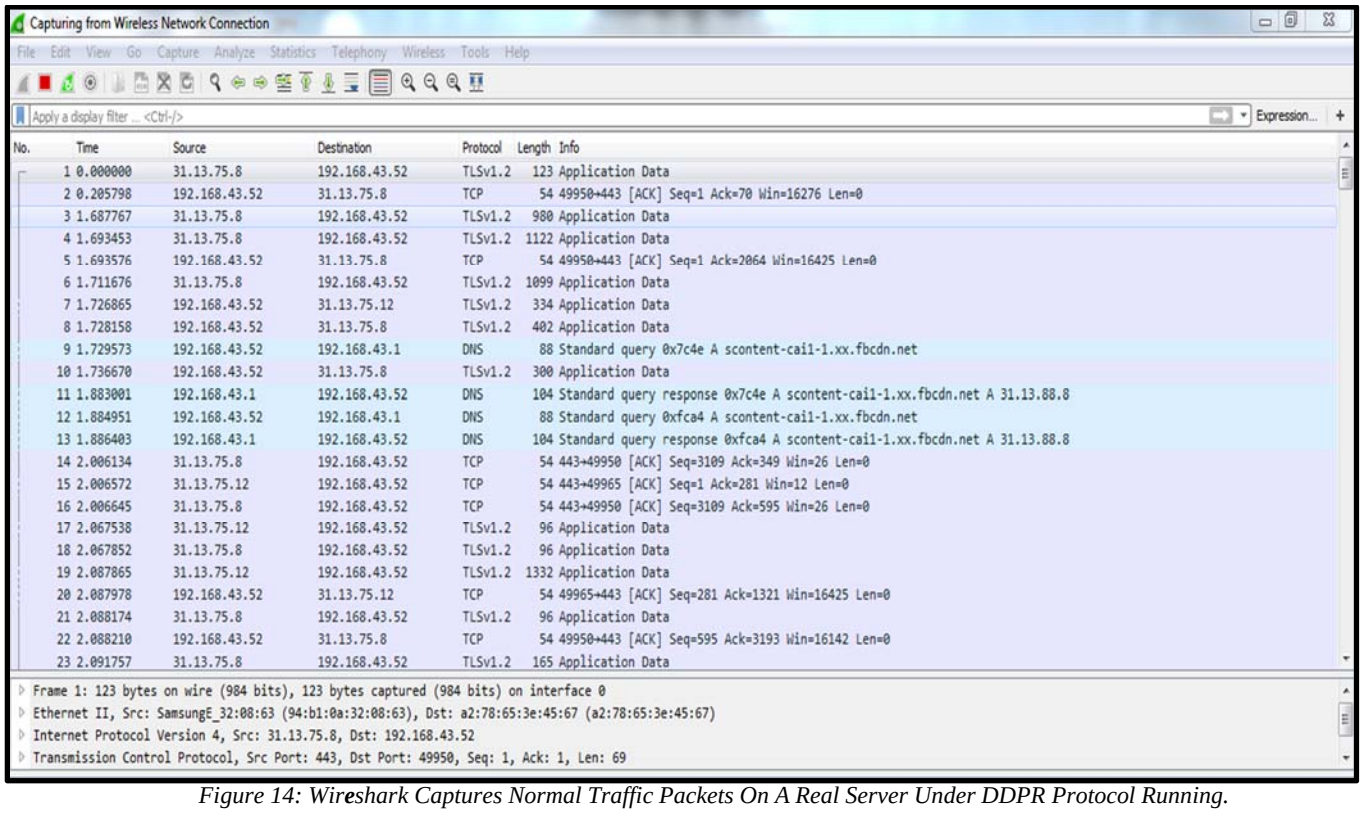


Figure 14: Wireshark Captures Normal Traffic Packets On A Real Server Under DDPR Protocol Running.

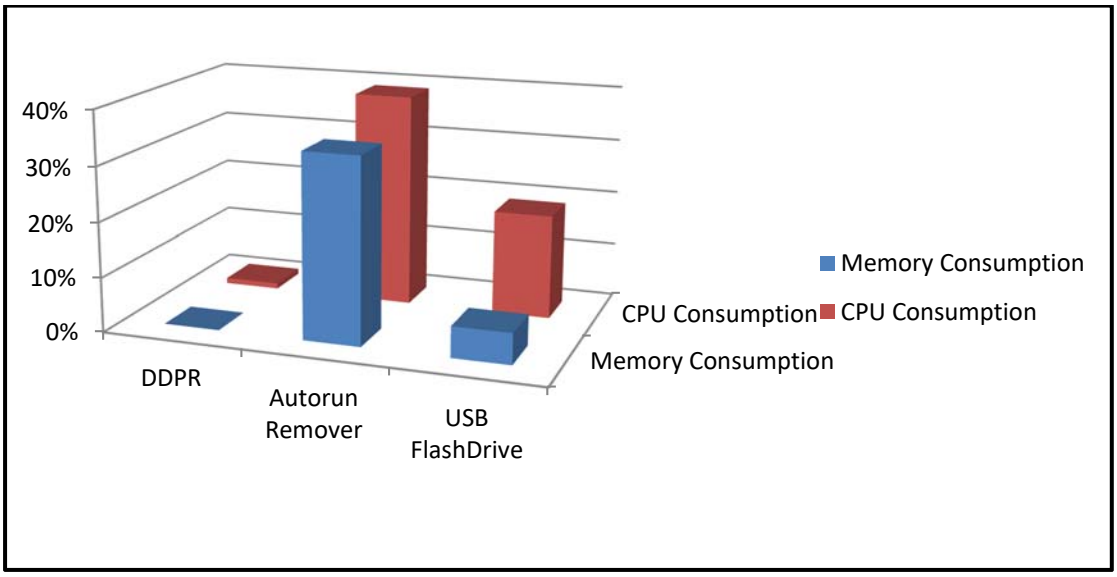


Figure 19: System Resources Consumption By DDPR And Other Commercial.

Table 4: Criteria Of Evaluation And General Evaluation Results

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Protocol/Application / Evaluation criteria	DDPR Protocol	AutorunRemover	USB flash drive	Usb Show
Detection accuracy	100 %	100 %	100 %	None
Deletion accuracy	100 %	Failed with test	64% of all tests	None
Protection accuracy	100 %	Failed with test	Failed with test	None
Scan time	1-2 seconds	Takes large time	3-7 Seconds	None
CPU consumption	0.003 %	39 %	2%	63%
Memory consumption	0.0001 %	33.7 %	5.7 %	39%
Recovery option	Available	Failed with test	Failed with test	Available
Average recovery time	13.4 seconds	None	Takes large time	Takes large time
Single drive recovery	Available	None	Not available	Not available
Application freeware	Yes (Free)	No (Not free)	No (Not free)	Yes (Free)
Application size	300 kb	3866 kb	658 kb	114 kb
Application setup	portable	Requires setup	Requires setup	portable