ISSN 1512-1232

DEVELOPMENT OF A TWO-TIER DETECTION OF DENIAL OF SERVICE ON ANDROID OPERATING SYSTEM

[1]Ogunleye G.O, [2]Akinlamilo O.

[1]Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria
[2]Department of Computer Science, Redeemer's University, Ede, Osun State, Nigeria

Abstract

The main reason for Denial of Service (DoS) is to disallow or block unauthorized users from accessing a particular network. This paper work proposed a solution to the issue of DoS attack Android Operating system. An application is developed to perform the detection of Dos attack on Android OS. In the developed application, a list of malicious Internet Protocol (IP) access the network, the application first checks the IP against this list and if it is not among the malicious addresses, it is accepted, if not it is denied access to the required service. A packet size limit is set for authorized addresses, when this limit is exceeded they get terminated. In developing the DoS application, HTML5/Javascript (JQuery and Angular JS)- Development Language is used.

Keywords: *DoS, DDoS, Operating Systems, Android*

1.0 INTRODUCTION

In Computing, Denial-of –Service (DOS) attack, is when an attacker attempt to keep legitimate clients from getting data or services (David and Ruby, 2004). Moreover, DoS attack, is an unambiguous effort to make a computer resource inaccessible by either injecting a computer virus or flooding the network with of no use traffic. This means that one computer and one internet connection is used to flood a server with packets (TCP/UDP) to the point that such DoS is targeted to consume the server's bandwidth and other resources, this will make the server inaccessible to other and thereby blocking the website or whatever else is hosted there (Özçelik and Brooks, 2014). The Worldwide Infrastructure Security Report in 2010 (Dobbins and Morales 2010) bring into being that DoS attack had gone standard, and network operators were confronting with bigger and more endless DDoS attacks. Attackers might get acknowledgement in the underground group by means of bringing down popular sites. Since simple to-utilize DoS devices, for example, Trinoo (Dittrich 1999), can be downloaded from the web effortlessly, and as such ordinary computer users can become DoS victims. A Distributed Denial-of-Service (DDoS) remains the place where the attack source is more than one-and frequently thousand- of single IP addresses. These packets arrive in a very large quantity that the resources at the victim like bandwidth, buffers or CPU is quickly exhausted (M.OmairShafiq, 2000).

Illegal perpetrators of Dos attacks regularly aim at websites or organizations facilitated or noticeable web server, for instance, banks, charge card; however thought processes of payback, extortion or contribution can be behind different attacks.

DoS attack remains a danger to the web. They diminish the service quality of the internet service subsequently it is vital to anticipate or possibly decrease them. To apply countermeasure against DoS attacks, or even break them down, the main significant step is to recognize such attack.

There are many risks that can be faced when being attacked by the malicious code. This is problematic as it can lead to the following:

- The prevention of certain mobile services that the users of the android device need to fully operate the system

- The inability of the users to utilize the android operating system to its full capacity.

The proposed system will make use of a two-step filtering process to filter out possible source of DoS attacks. The first tier traffic filter will make use of a data bank of known DoS IP addresses and match them against the incoming traffic. If there is a match then the incoming packet is denied access the system. The second-tier traffic filter approach will employ the use of the drop tail algorithm. A data packet size queue will be allocated for incoming traffic. When the queue size is filled, all incoming traffic is dropped. This will prevent over congestion of the incoming traffic, thus preventing the possibility for any over-flooding of the system network with unwanted traffic.

The main aim of this study is to develop a system that can detect denial of service attacks on Android based devices.

2.0 LITERATURE REVIEW

A DoS attack, is an occurrence in which a client or organization is denied of the service of the benefit they would ordinarily hope to have. In DDoS, huge numbers of compromised systems (sometimes known as botnet) attack a single target. In spite of the fact that a DoS attack does not ordinarily bring about the theft of data or other security loss, it can cost the objective individual or organization a lot of time and money. Normally, the loss of service is the inability of a specific system service, for example, email, to be accessible or the interim loss of all system availability and service. A denial of service attack can likewise destroy programming and files in affected computer system. At times, DoS attacks have constrained Websites accessed by a large number of individuals to incidentally stop operation (Rous, 2007). With more than 45% of the US offers of cell phones, Android is apparently one of the best samples of overcoming affliction of the product business of the latest couple of years (Gartner Group, 2011).

ANDROID

Android is a versatile operating system (OS) as of now created by American organization; google, based on Linux, and open source operating system for PC and design essentially for touch screen cell phones, for example, smartphones and tablets. Android's customer interface relies on direct control, using touch flags that uninhibitedly identify with real exercises, for instance, swiping and tapping, to control on-screen object, nearby with a virtual console for content information. This suggests that you can without quite a bit of a stretch quest for information on the web, watch recordings, chase down headings and make messages on your phone, exactly as you would do on your PC, yet there's more to Android than these essential examples. Google has even gone further to make Android auto for automobiles, androids wear for wrist watches, and android TV for TVs, each with a specific client user interface. Varieties of Android are similarly used on, game console, notepads, electronic cameras, and different hardware. As at 2015, Android had the highest introduced base of every single operating system (Manjoo, 2015). As at July 2013, the Google Play store had more than a million Android applications discharged, and more than 50 billion applications downloaded (Open Handset Alliance, 2007). As at April-May 2013, a study conducted by compact application architects found that 71% of engineers make applications for android, and a 2015 evaluation found that 40% of full-time proficient specialists consider Android to be their major target stage, which is proportional to Apple's iOS on 37% with both stages far above the other. Android's source code is released by Google under open source licenses, even though most android gadgets at last ship with a mix of open source and restrictive programming, including selective programming fundamental for getting the chance to Google administration (ArsTechnica, 2013).

3.0 METHODOLOGY

The proposed system makes use of a two-step filtering process to filter out possible source of DoS attacks.

The first tier traffic filter will make use of a data bank of known DoS IP addresses and match them against the incoming traffic. If there is a match then the incoming packet is denied access the system.

The second-tier traffic filter approach will employ the use of the drop tail algorithm. A data packet size queue is allocated for incoming traffic. When the queue size is filled, all incoming traffic is dropped. This will prevent over congestion of the incoming traffic, thus preventing the possibility for any over-flooding of the system network with unwanted traffic.

The flowcharts showing the sequence of activities for the Two Tier Approach is illustrated below.

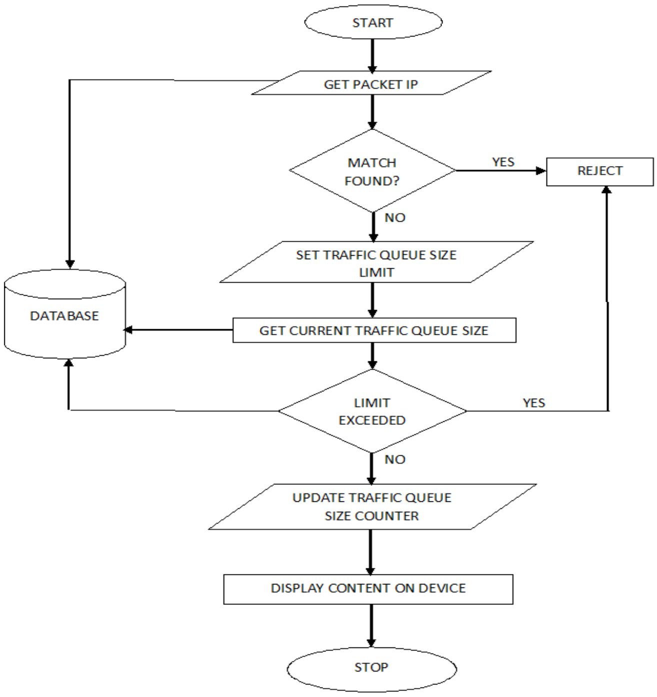

START
GET PACKET IP
MATCH FOUND?
YES
REJECT
NO
SET TRAFFIC QUEUE SIZE LIMIT
DATABASE
GET CURRENT TRAFFIC QUEUE SIZE
LIMIT EXCEEDED
YES
NO
UPDATE TRAFFIC QUEUE SIZE COUNTER
DISPLAY CONTENT ON DEVICE
STOP

Figure 1: Flowchart for the two tier approach of the proposed system

Figure 1 shows the flowchart of the proposed system, it function begins by getting IP addresses from users who request to access the system. When an IP address is detected by the system, the system checks the IP address with the malicious IP addresses stored in the database and if there's a match, the IP address is denied access. And if there's no match, the system checks the traffic queue size available and also checks if the queue size has exceeded the limit given by the system, if the limit has been exceeded it should terminate, and if not, it should update the queue size counter and display it on the device.

ALGORITHM

The algorithm of the proposed two tier approach to detect denial of service attacks is presented below:

1. *Start*
2. *Get incoming packet IP address*
3. *Check database for match*
4. *If found then*
5. *Deny packet access to system*
6. *Else*
7. *Get current traffic queue size*
8. *If current traffic queue size > queue size maximum then*
9. *Reject incoming packet*

```
Else
Allow traffic
Update traffic queue size counter
End
```

SYSTEM ARCHITECTURE

The system architecture for the proposed system is illustrated below:

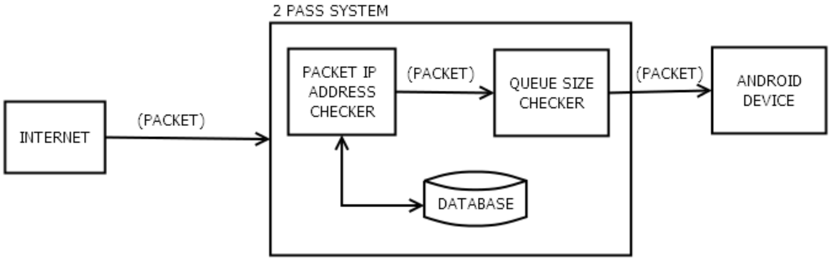


Figure 2: System Architecture

From figure 2, it can be seen that the following components make up the system:

IP Address Checker: Which is used to check if the IP address of the incoming packets can be found in the database of IP addresses of know malicious content

Queue Size Checker: This is used to check if the queue size has been exceeded

Database: Use to store data on the of IP addresses of know malicious content.

4.0 SYSTEM IMPLEMENTATION

The researcher proposed a two-tier coordination approach for detecting and mitigating Denial of Service attacks on android operating systems. The first tier approach will make use of a traffic that filters suspicious traffic for possible flooding.

The implementation of this approach was demonstrated using a software simulation to show the workings of the approach when applied. The software program used to develop the simulation was the HTML5/JavaScript (JQuery and Angular JS)- Development Language. The database that was utilized for the storage of the traffic packet data was stored using the firebase database system.

4.1 LOGIN INTERFACE

The login interface is the first interface of the software that the user will interact with. It will be used to provide a security layer for the intrusion detection system. The user will have to provide security credentials such as the correct username and password before being allowed to gain access to the system. The login interface of the system is displayed in figure 3.

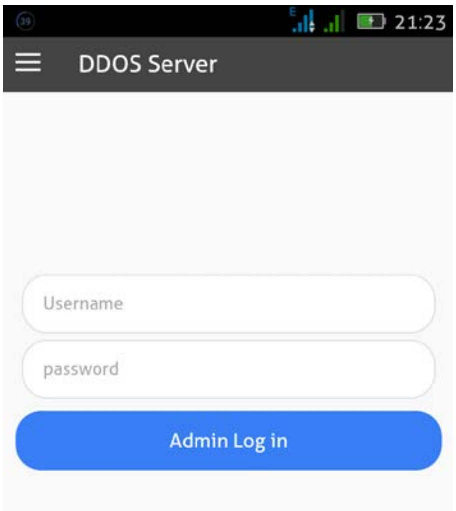
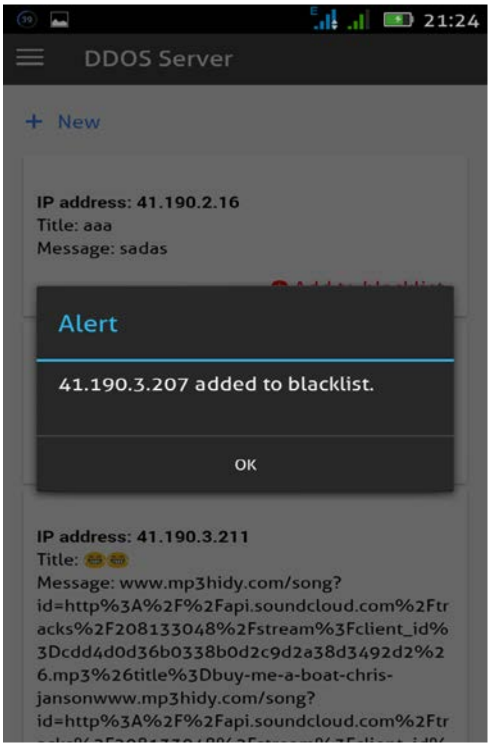


Figure 3: Login Interface

4.2 ADD IP ADDRESS

This module adds the IP addresses of known malicious sites into the system database. These IP addresses will be checked by the system against incoming traffic. If a match is found, then the system will reject the incoming packet.



IP address: 41.190.2.16
Title: aaa
Message: sadas

Alert

41.190.3.207 added to blacklist.

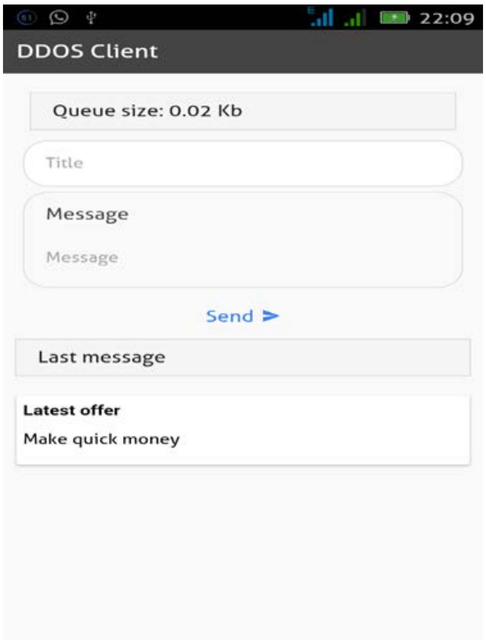
IP address: 41.190.3.211
Title: 😂😂
Message: www.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftracks%2F208133048%2Fstream%3Fclient_id%3Dcdd4d0d36b0338b0d2c9d2a38d3492d2%26.mp3%26title%3Dbuy-me-a-boat-chris-jansonwww.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftr

.

Figure 4: IP Address Module.

4.3 PACKET SIZE LIMIT

The packet size limit was used to define the size of the packet data that will be coming to the android device. The limit, once set, will check the traffic packet counter to check if the size of the incoming packet has surpassed the packet size limit. In the event that the limit is exceeded, the system will terminate traffic so as to avoid traffic congestion which can lead to denial of service.



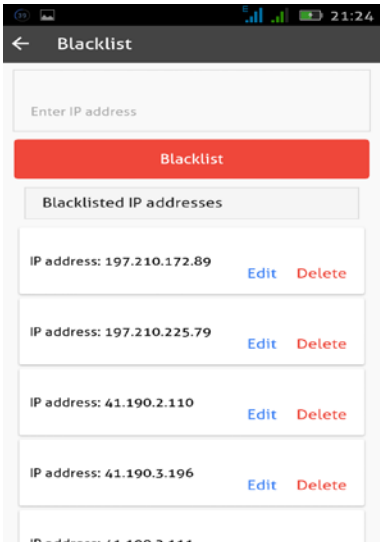
Queue size: 0.02 Kb

Latest offer
Make quick money

Figure 5: Packet size limit interface

4.4 VIEW IP ADDRESS MODULE

This module is used to view the information of the IP addresses of malicious sites that have been added to the system database.



IP address: 197.210.172.89

IP address: 197.210.225.79

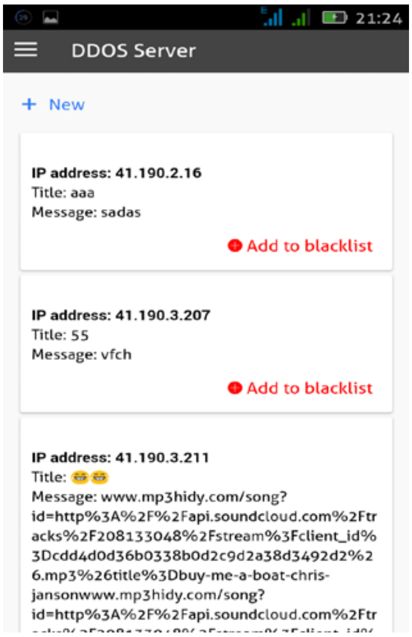
IP address: 41.190.2.110

IP address: 41.190.3.196

Figure 6: View IP Address interface

4.5 TRAFFIC LOG MODULE

This module is used to view the log of the IP address that has been received via the network by the Android Device.



IP address: 41.190.2.16
Title: aaa
Message: sadas

IP address: 41.190.3.207
Title: 55
Message: vfch

IP address: 41.190.3.211
Title: 😂😂
Message: www.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftracks%2F208133048%2Fstream%3Fclient_id%3Dcdd4d0d36b0338b0d2c9d2a38d3492d2%26.mp3%26title%3Dbuy-me-a-boat-chris-jansonwww.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftr

Figure 7: Traffic Log Module interface.

4.6 TRAFFIC MONITOR SIMULATION MODULE

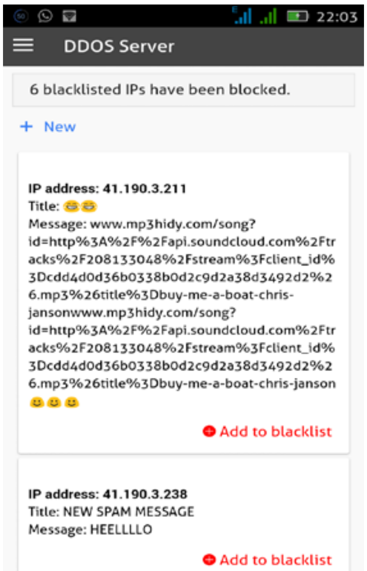
This is the simulation window for the Anti- DOS (Denial of service) System. The window shows the following information:

i. Incoming IP address
ii. Status of the incoming traffic
iii. Current packet size
iv. Packet size limit
v. Traffic information log showing

The interface was used to demonstrate the researcher's approach to eliminating Denial of Service Attacks on Android Based Systems.

4.7 RESULT AND EVALUATION

The result of the simulation is shown in the traffic information window displayed in figure 8.



6 blacklisted IPs have been blocked.

IP address: 41.190.3.211
Title: 😂😂
Message: www.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftracks%2F208133048%2Fstream%3Fclient_id%3Dcdd4d0d36b0338b0d2c9d2a38d3492d2%26.mp3%26title%3Dbuy-me-a-boat-chris-jansonwww.mp3hidy.com/song?id=http%3A%2F%2Fapi.soundcloud.com%2Ftracks%2F208133048%2Fstream%3Fclient_id%3Dcdd4d0d36b0338b0d2c9d2a38d3492d2%26.mp3%26title%3Dbuy-me-a-boat-chris-janson 😃😃😃

IP address: 41.190.3.238
Title: NEW SPAM MESSAGE
Message: HEELLLLO

Fig 8: Blocked IP list

It can be seen that six malicious IP addresses were detected and the system rejected them based on the traffic log information.

When the packet size limit was reached, the network connection was terminated and no more traffic was sent to the system.

In essence this simulation demonstrated the two tier coordination approach for detecting and mitigating DOS attacks on android operating systems.

The suspicious IP address was filtered by checking the incoming IP address against a database of know DOS based IP addresses. When found, the traffic packet of that IP address is terminated. This constituted the first tier approach

The traffic queue length was controlled by setting a limit to the size of the incoming packets that were to be received from the network. Once the size was exceeded, the network connection would be terminated. This constituted the second tier approach.

With this approach, the researcher has demonstrated a way for the prevention of DOS attacks on mobile devices which can be employed by the mobile software developers.

5.0 CONCLUSIONS

This research project was aimed at proposing a two-tier coordination approach for detecting and mitigating DoS attacks on Android operating systems. The objective of the research was to implement and demonstrate the approach system using a mobile device simulation. The simulation program was developed using the HTML5/JavaScript (JQuery and Angular JS) - Development Language to provide the user interface and implement the algorithm of the approach. Firebase was used to implement the database that was used for storing data about the malicious DoS IP (Backend as a service cloud database).

The simulation program was tested and was found to be satisfactory as it met the objective set out by the researcher.

As this paper was targeted towards the development of an approach to prevent Denial of Service attacks on Android Based devices, it has provided a framework with which future researchers can use to implement an actual solution that will run on an android device.

For further improvement to the system, the database containing signatures of the known DoS based sites should be auto-updatable so as to provide up to date information that can be used to protect the Android Device.

6.0 REFERENCE

1. *"Android's Google Play beats App Store with over 1 billion apps, now officially largest"*. Phonearena.com. Retrieved February 28, 2016.
2. *"Industry Leaders Announce Open Platform for Mobile Devices"* (Press release). Open Handset Alliance. November 5, 2007. Retrieved February 17, 2016.
3. *"Microsoft Security Advisory (975497): Vulnerabilities in SMB Could Allow Remote Code Execution"*. Microsoft.com. September 8, 2009. Retrieved 2016-01-14.
4. *"RFC 4987 - TCP SYN Flooding Attacks and Common Mitigations"*. Tools.ietf.org. August 2007. Retrieved 2016-02-12.
5. *"Types of DDoS Attacks"*. Distributed Denial of Service Attacks (DDoS) Resources, Pervasive Technology Labs at Indiana University. Advanced Networking Management Lab (ANML). December 3, 2009. Archived from the original on 2010-09-14. Retrieved 11 February, 2016.
6. Adam, P., Fuchs, Chaudhuri, A., & Jeffrey, S., Foster, Y. *(2009)Scandroid: Automated security certification of android applications.*
7. Adrienne Porter, F., Erika, C., Steve, H., Dawn, S., & David, W. (2011*). Android Permissions Demystifie.*In Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11. 627- 638.
8. Armando, A., Merlo, A., Migliardi, M., &Verderame, L. (2012).*"Would You Mind Forking This Process? A Denial of Service Attack on Android (and some countermeasures)"* in IFIP SEC 2012 27th International Information Security and Privacy Conference.
9. Bishop, F. (2002). Computer Security: Art and Science. Chapter 1, 3-6
10. Burguera, I., Zurutuza, U., &Nadjm-Therani, S. (2011).*Crowdroid: Behavior- Based Malware Detection System for Android* in Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'11).
11. Dagon, D., Martin, T., &Starner, T. (2004).*Mobile Phones as Computing Devices: The Viruses are coming*! *IEEE Pervasive Computing*. 3 (4) 11-15.
12. Di Cerbo, F., Andrea, G., Florian, M., & Svetlana, V. (2011).*Detection of Malicious Applications on Android os*, in Hiroshi, S., Katrin, F., & Shuji, S., (ed.) Computational Forensics. (6540) 138-149.
13. Erika, C., Adrienne, P., Felt, K.,Greenwood, &David,W. (2011). Analyzing Inter-application Communication in Android, in Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, MobiSys. 239-252.
14. Franceschi-Bicchierai, L. (2015)*"Goodbye, Android"*. Motherboard. Vice. Retrieved January, 2016.
15. Gartner Group Press Release (2011) http://www.gartner.com/it/page.jsp?id=1848514*"Android Security Overview"*. Android Open Source Project. Retrieved February 20, 2016.

16. Gritzalis, D., Furnell, S., &Theoharidou, M. (2012). (Eds.), pp. 13-24, Heraklion, Greece, IFIP Advances in Information and Communication Technology (AICT), Vol. 376, Springer.
17. Lucas, D., Alexandra, D., Ahmad-Reza, S., & Marcel, W. (2011).*Privilege Escalation Attacks on Android*, in Mike, B., Gene, T., Spyros, M. and Ivan, I. (ed.). Information Security. (6531) 346-360.
18. Manjoo, F., (2015).*"A Murky Road Ahead for Android, Despite Market Dominance". The New York Times.*ISSN 0362-4331. Retrieved January, 2016.
19. Margaret, R., (2007): Denial of Service (DoS) Definition whatis.com [online] Available at http://searchsoftwarequality.techtarget.com/definition/denial-of-service
20. Mohammad, N., Sohail, K., &Xinwen Z. (2010). Apex: *Extending Android Permission Model and Enforcement with User-defined Runtime Constraints*. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS '10, pages 328-332. New York, NY, USA. ACM, Machigar, O., Stephen, M.
21. Protalinski, E., (2012).*"Android Malware Numbers Explode to 25,000 in June 2012"*. ZDNet. Retrieved December 19, 2015.
22. Robert, Lemos.,(2007). *"Peer-to-peer networks co-opted for DOS attacks"*.SecurityFocus. Retrieved 2016-02-22.
23. Schmidt, A., Bye, R., Schmidt, H., Clausen, J., Kiraz, O., Yuksel, K., Camtepe, S. &Albayrak, S.(2009).*Static Analysis of Executable for Collaborative Malware Detection on Android.* IEEE International Conference on, 1 -5.
24. Sven, B., Lucas, D., Alexandra, D., Thomas, & Ahmad-Reza, S (2011).*Xmandroid: A new android evolution to miti- gate privilege escalation attacks.* Technical Report TR-2011-04, Technische Univ. Darmstadt.
25. Taghavi, Z., Saman, (2013). "*A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks"* (PDF) 15 (4). IEEE COMMUNICATIONS SURVEYS & TUTORIALS. pp. 2046-2069. Retrieved 2016-03-07.
26. Technica, A., (2013) *Google's iron grip on Android: Controlling open source by any means necessary"*. Retrieved January 8, 2016.
27. Technica, A., Amadeo, R. (2015) "*Waiting for Android's inevitable security Armageddon"*. Technica, A., Retrieved 17 March 2016.
28. William, E., & Mc-Daniel, P. (2009).*Semantically Rich Application-Centric Security in Android.* In ACSAC '09: Annual Computer Security Applications Conference.
29. William, E., Damien, O., McDaniel, P., &Swarat, C. (2011)*A study of android application security*. In Proceedings of the 20th USENIX conference on Security, SEC'11, pages 21-21, Berkeley, CA, USA, 2011. USENIX Association.
30. Yajin, Z., Xinwen, Z., Xuxian, J., &Vincent,W. (2011). *Taming Information-stealing Smartphone Applications (on android).* 4th International Conference on Trust and Trustworthy Computing, 93-107.

Article received: 2016-10-27