Journal of the Korea Industrial Information Systems Research Vol. 20 No.5, Oct. 2015 :37-44
http://dx.doi.org./10.9723/jksiis.2015.20.5.037 ISSN:1229-3741

MWMon: A Software Defined Network-based Malware Monitor

Jo Min Jae[1] and Ji Sun Shin[2]*

Abstract An antivirus is a widely used solution for detecting malicious softwares in client devices. The performance of antivirus solutions in the mobile client environment is critical due to its resource constrains. Many solutions light-weighting client's overhead in the mobile client environment have been developed. However, most solutions require platform modifications or software installations and it decreases their realizations in practice. In this paper, we propose a solution detecting malwares on networks using the Software Defined Network (SDN). Our main goal is designing a solution detecting malwares of mobile client without involving the client into the work. We contribute to provide a solution that does not require client-side installations or modifications and so is easily applicable in practice.

Key Words : Software defined network, mobile security

1. Introduction

As a mobile device is widely used, the operations done in the PC are now possible by the mobile. In addition, smartphone users have increased in number and attacks targeted smartphones have increased as well. In 2014 third-quarter, McAfee[1] found over 0.7 million new mobile malwares. The types of an Antivirus for detecting malwares are a signature-based antivirus, a heuristic-based antivirus and a behavior-based antivirus. The signature-based antivirus, the most basicantivirus, has the increasing number of signature patterns as the number of malwares increased. Hence, it causes a performance degradation issue: detection time of signature-based antivirus increases and then throughput decreases. In this paper, the method we propose is to detect malwares on networks instead of the client environment. In this regard, we use the software defined network [2, 3]. SDN, the software defined network, enables the management and control of the network. Our main goal is providing a solution removing client's involvement so that the solution can be easily realized in practice without software installation or platform modification in the client environment.

The paper is organized as follows: Section Ⅱ and Ⅲ describes related work. In Section Ⅳ , we provide the description of our solution, MWMon(Malware Monitor). Finally, in Section Ⅴ, we conclude.

* Corresponding Author : jsshin@sejong.ac.kr
† 본 연구는 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF- 2013R1A1A3009163).
Manuscript receivrd September 14, 2015 / revised October 1 ,2015 / accepted October 27, 2015
1) 세종대학교 컴퓨터공학과, 제1저자
2) 세종대학교 정보보호학과, 교신저자

2. Related Works 1: The Research for Security Using SDN

Solutions applied to a security by using SDN are classified as three types that Table 1 shows.

On the basis of the purpose of applying SDN to the security, there are three types of solutions: monitoring solutions, detecting solutions for malicious behaviors and network security enhancement solutions using the access control.

The SDN is used for monitoring network traffics and detecting malicious network flows. To overcome that configuring when security equipment is applied is difficult in the cloud, CloudWatcher[4] suggests a network monitoring service. SDN-based forensic system [5] investigates attacks such as various network failures and data leakages in data center networks. For the protection of mobile wireless network environment, the framework [6] that monitors the network and then detects an intrusion in a network Endhost is suggested.*

The SDN is used to detect DDoS (Distributed Denial of Services) attacks and an intrusion of malicious applications. Once the DDoS attack is detected, the SDN application provides a new IP address to the server to defend against the attack and redirection address information that contains CAPTCHA [7] to clients. The method[8] which blocks a bot if it fails to recognize the redirection information is proposed. Monitoring network threats, MalwareMonitor[9] can detect malwares such as a botnet. A solution[10, 11] for detecting malwares of embedded and mobile devices and a framework[6] for detecting intrusions in wireless networks are also proposed. To protect against an attacker tries to discover network vulnerabilities by scanning tools, a solution[12] that changes from a real IP to a virtual IP randomly is suggested and it manages a virtual IP by using the SDN controller. It makes a network access control policy that control and monitor network flows. CloudWatcher guarantees all possible network packets which security equipment can check in the Cloud. It also controls network flows and provides the Simply Policy Script. In a guest WiFi, OpenWiFi[13] uses SDN for the access control and authentication so that it suggests the WiFi system. OpenSec[14] suggests the OpenFlow-based security framework. In this circumstance, a network manager creates security policies for a specific flow. Such security policies are composed of explanation about the flow, security service lists where the flow will be applied and a countermeasure against the case malicious contents are found.

Table 1 The Research of Security Using SDN

Purpose	Related Works
Monitoring	Cloud [4]
	Wireless Network [6]
	Forensic [5]
Detection	DDoS Attack[8, 9]
	Malware [6, 9, 10, 11]
	Vulnerability Scanning [12]
Access Control	Cloud [4]
	Wireless Network [13]
	Access Control [14]

3. Related Works 2: Antivirus Solution In a Lightweight Client Environment

As mobile internet devices are pervasive,

* While many network intrusion solutions already have been researched[28-30], SDN is also used to detect intrusions.

more efficient antivirus solutions for lightweight client environment are necessary: many researches have been studied to reduce client-side overheads of antiviruses [15, 16, 17, 22, 26]. In particular, there have been researches improving client-side performance of signature-based antiviruses by relying on Cloud server[15, 16, 22, 26]. Also, there is a solution designed for Android mobile devices[17]. This solution introduces an agent relaying malicious file detection requests from client to the third-party scanning services and sending the result back to client. In this section, we briefly review major antivirus solutions designed for lightweight client environment.

3.1 CloudAV

CloudAV[15] suggests the signature-based antivirus used to detect malwares. When a file is downloaded or created by a client, CloudAV sends it to a server to analyse. After analyzing the file, the server sends the result to the client, and then CloudAV determines whether to block the file or not on the basis of the result. Because CloudAV, however, sends the files to the server to examine, it may violate the privacy of users who handle sensitive data[15, 16].

3.2 SplitScreen

SplitScreen[16] proposes the cloud-based signature antivirus. To reduce violating the privacy, SplitScreen sends a bit vector of a file which showed the suspicious result from the prematch to the server instead of sending the whole of the file. Then, the server searches for actual signature patterns from the given bit vector and sends the signature data to the client so that the actual test is progressed. SplitScreen offers lightens client environment because it does not save whole signature databases but receives the signatures of the suspicious file determined by the prematch from the server.

3.3 ThinAV

ThinAV[17] proposes the antivirus which detects maliwares on Android based mobile devices. Using the Internet, it uses a number of anti-virus services, and offers the lightened cloud-based anti-virus for Android devices. ThinAV consists of two main components: an Android client and a server. The Android client sends applications to examine to the server, and the server sends the files again to the third-party scanning services (Kaspersky [18], VirusChief[19], VirusTotal[20], ComDroid [21]) so that the client knows the reported result from them. The Android client modifies the Android OS Package Manager and performs periodical checks by the Killswitch module for applications installed.

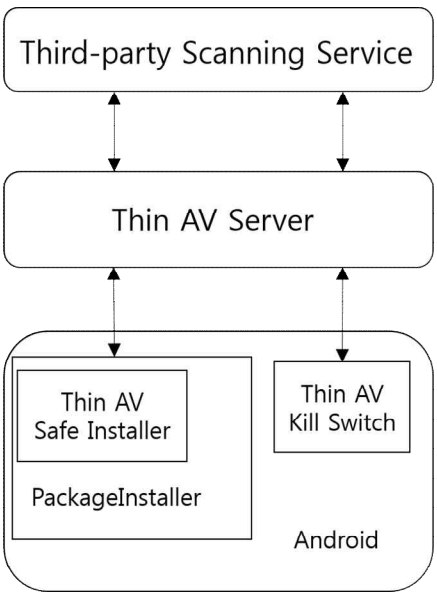

Fig. 1 The System Architecture of ThinAV.

4. MWMon(Malware Monitor)

The main goal of our solution is providing an antirivus solution not requesting client-side work. Consequently, it requires neither client-side platform modification nor client-side program installation. Previously introduced solutions such as CloudAV, SplitScreen and ThinAV reduce client-side overhead but still need client-side work. Therefore, their solutions require client platform modification or client program installation. To realize our goal, we use SDN and monitor traffic to mobile devices.

The essential way MWMon works is as follows: when an Android client downloads an application to install, MWMon detects the application's malice and blocks if it is. Since detection is done on networks, there are no needs to install any applications to the client or modify the Android platforms.

Furthermore, while previous solutions for lightweight environment focus on signature-based detection, our solution provide an integrated detection system including signature-based detection, static analysis and dynamic analysis.

4.1 System Architecture

Fig. 2 is the structure of MWMon. When the client tries to install an application on the mobile device, the OpenFlow switch copies the packets and sends them to the OpenFlow controller. Then, the controller sends the given packets again to the detection module for the result that tells if the application is malicious one or not. The detection module is largely made up of two modules: the signature detection module and the analysis module. The signature detection module detects malware by signatures, but in the case of searching failure, the analysis module is followed to tell whether an application is malicious or not.

4.2 Detection Module

4.2.1 Signature Detection Module

The signature detection module examines malicious application by stored signature patterns. The signature keeps malwares' information detected by signature module and the analysis module. In case the application is not detected by the signature detection module, the detection is proceeded by the analysis module.

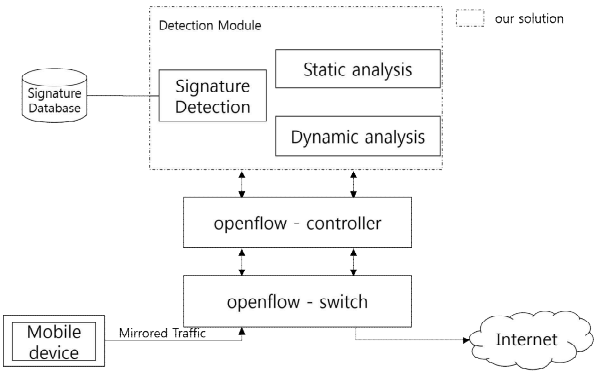

Fig. 2 The System Architecture of MWMon.

4.2.2 Analysis Module

The analysis module works in the static analysis and the dynamic analysis. First, it verifies the authority used in the Android applications using the static analysis. It also analyzes and tracks the called API function so that the malicious patterns can be identified. Then, this makes it possible to check if the user information is leaked and collect the leaked server of an attacker. After it calculates the risk index of the application by the authority and API function used, and the like, it performs the dynamic analysis if the

calculated value is beyond the threshold. The dynamic analysis determines whether the program is malicious or not by executing it in the independent environment such as SandBox and analyzing activity logs of the program.

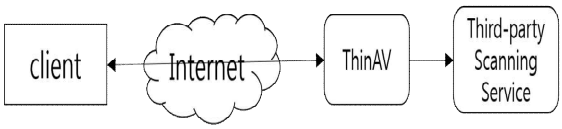

Fig. 3 The Network Strucutre of ThinAV.

Once the detection module is performed and the application is identified as the malware, the information of it is stored in the local signatures database. The package information of the malware, the file signatures of the application and the IP address to download the file are stored, too.

4.3 Solution Comparison and Assessment

We compare the network structure of ThinAV, the existing solution, with MWMon. First, the network structure of ThinAV is shown in Fig. 3. The structure ThinAV has is that the Android client accesses to the ThinAV server, and then the server detects the malwares by using external antivirus services via the Internet and informs the client about it.

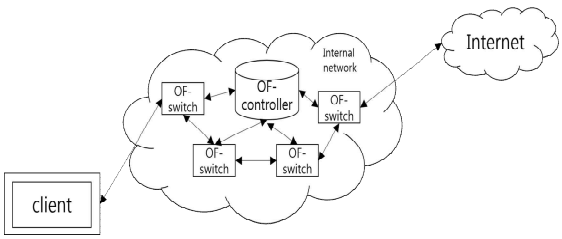

Fig. 4 The Network Strucutre of MWMon.

The network structure of MWMon can be seen in Fig. 4. To detect incoming malicious programs via the Internet in networks, MWMon uses OpenFlow Controller. In ThinAV, the client needs to access to the ThinAV server or the server needs external antivirus using the Internet for the detection whereas MWMon detects malwares using SDN in internal networks without any connections to the Internet. Because it detects and blocks in networks, it is more efficient in the lightweight client environment.

Table 2 shows the comparison MWMon with ThinAV and SplitScreen. MWMon, in detail are as follows:

- There are no needs to install programs on clients or modify platforms because detecting malwares is worked in networks.
- There are no needs to connect to the internet because it works without third-party antivirus scanning services.
- Through the analysis module, it also can detect unknown malwares.

Table 2 Solution Evaluation and Comparison

	MWMon	ThinAV	Split-Screen
Install Client Application or Modify Client Platform	X	O	O
Use Third-party AV scanning	X	O	X
Analysis Module	O	X	X

5. Conclusion

We looked into previous studies focused on detecting and blocking mobile malwares, and

suggested the solutions for them. Also, there were the definition of the SDN which is the up-and-coming paradigm in networks and other studies that applied the software defined networking to the security. For the lightweight client environment, the system which detects mobile malwares and blocks them using the SDN was proposed in this study. The ways the existing studies had researched into have the client-server structure so that the client needs to install programs or modify the client platform. To improve the drawbacks, on the other hand, what MWMon proposed is the network-based mobile malwares detection using the SDN. Because MWMon detects malwares in networks, it is the advantage that users are not asked to install any additional programs. In this regard, detecting and blocking malwares downloaded unawarely are possible and thus the lightened client can be used in the secure network environment.

In this paper, only the system to detect malwares in the lightweight client environment is presented. For the future study, we implement the system to detect them using the SDN. We plan to implement the system based on open source solutions: in particular, ClamAV[23] for signature-based detection, ComDroid[21] and DroidMat[24] for static-analysis, and TraceDroid[25] for dynamic-analysis. We will look into the influence of the malwares detection module on the performance such as the traffic latency in real networks, and also the accuracy of the module through the future research, too.

References

[1] McAfee Report, http://www.mcafee.com/kr/resources/reports/rpquarterly-threat-q3-2014.pdf

[2] SDN wiki, http://en.wikipedia.org/wiki/Software-defined_networking

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” SIGCOMM Comput. Commun. Rev., Vol. 38, No. 2, pp.69 - 74, March 2008.

[4] S. Shin, G. Gu, “CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?),” in 20th IEEE International Conference on Network Protocols (ICNP). IEEE, pp.1 - 6, 2012.

[5] Bates A, Butler K, Haeberlen A, Sherr M, Zhou W, “Let SDN be your eyes: Secure forensics in data center networks,” Proceedings of the NDSS Workshop on Security of Emerging Network Technologies (SENT’14). 2014.

[6] AY Ding, J Crowcroft, S Tarkoma, H Flinck, “Software defined networking for security enhancement in wireless mobile networks,” Vol. 66, pp.94 - 101, 2014

[7] L. von Ahn, M. Blum, N.J. Hopper, J. Langford, "CAPTCHA: Using Hard AI Problems for Security," Lecture Notes in Computer Science 2656, pp.294-311, 2003

[8] S. Lim, J. Ha, H. Kim, Y. Kim, S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks." Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on. IEEE, pp.63-68, 2014.

[9] Abaid, Zainab, Mohsen Rezvani, Sanjay Jha. "MalwareMonitor: An SDN-based Framework for Securing Large Networks," Proceedings of the 2014 CoNEXT on Student Workshop. ACM, pp.40-42, 2014.

[10] R. Skowyra, S. Bahargam, A. Bestavros, “SoftwareDefined IDS for Securing Embedded Mobile Devices,” 2013. [Online]. Available: http://www.cs.bu.edu/techreports

/pdf/2013-005-software-defined-ids.pdf
[11] R. Jin, B. Wang, “Malware detection for mobile devices using software-defined networking,” in Research and Educational Experiment Workshop (GREE), 2013 Second GENI. IEEE, pp.81 - 88, 2013.
[12] J. H. Jafarian, E. Al-Shaer,Q. Duan, “Openflow random host mutation: transparent moving target defense using software defined networking,” in Proceedings of the first workshop on Hot topics in software defined networks. ACM, pp.127 - 132, 2012.
[13] K. Yap, Y. Yiakoumis, M. Kobayashi, S. Katti, G. Parulkar, N. McKeown, “Separating authentication, access and accounting: A case study with OpenWiFi,” Open Networking Foundation, Tech. Rep., 2011.
[14] Lara, Adrian, Byrav Ramamurthy. "OpenSec: A framework for implementing security policies using OpenFlow." Global Communications Conference (GLOBECOM), pp.781-786 2014.
[15] Oberheide, Jon, Evan Cooke, Farnam Jahanian. "CloudAV: NVersion Antivirus in the Network Cloud." USENIX Security Symposium, pp.91-106, 2008
[16] Cha, Sang Kil, et al. "SplitScreen: Enabling efficient, distributed malware detection." Communications and Networks, Vol 13, No. 2, pp.187-200, 2011
[17] Jarabek, Chris, David Barrera, John Aycock. "Thinav: Truly lightweight mobile cloud-based anti-malware," Proceedings of the 28th Annual Computer Security Applications Conference. ACM, pp.209-218, 2012.
[18] Kaspersky, http://www.kaspersky.com
[19] VirusChief, http://www.viruschief.com
[20] VirusTotal, http://www.virustotal.com
[21] E. Chin, A.P. Felt, K. Greenwood, D. Wagner, “Analyzing inter-app lication communication in Android,” In Proceedings of the 9th international conference on Mobile systems, applications, and services, ACM, pp. 239-252 ,2011
[22] Min Jae Jo, "Performance Enhancement of malware detection in the lightweight client environment", MA thesis, Sejong University, 2015
[23] ClamAV, http://www.clamav.net/index.html
[24] Wu. D. J, Mao. C. H, Wei. T. E, Lee. H. M and Wu. K. P, "Droidmat: Android malware detection through manifest and api calls tracing." Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on. IEEE, pp. 62-69, 2012.
[25] V. Van der Veen, "Dynamic Analysis of Android Malware," Master Thesis, VU University Amsterdam, Aug. 2013. [Online]. Available: http://tracedroid.few.vu.nl/thesis.pdf
[26] Min Jae Jo and Ji Sun Shin, “A Performance Enhancement Scheme for Signature-based Anti-Viruses,” Journal of the Korea Industrial Information System Society, Vol. 20, No. 2, pp. 65-72, 2015.
[27] S. Scott-Hayward, G. O’Callaghan, S. Sezer, “SDN Security: A Survey,” IEEE SDN for Future Networks and Services, s pp.1 - 7, November 2013.
[28] Eun Jun Yoon, Hyun Sung Kim and Ki Dong Bu, “An Intrusion Detection System Using Pattern Classification”, Proceedings of the Korea Society for Industrial Systems Conference, 2002.
[29] Hyun Chul Cha, “A Solution for Timing Gap Problems on Network Intrusion Detection Systems”, Journal of the Korea Industrial Information System Society, Vol. 7, No.1, pp. 1-6, 2001.
[30] Jae Min Son, Hyun Sung Kim and Ki Dong Bu, "A Scheme for Protecting Security Rules in Intrusion Detection

System", Journal of the Korea Industrial Information System Society, Vol. 8, No.4, pp. 8-16, 2003.

조 민 재 (Jo Min Jae)

- 학생회원
- 세종대학교 컴퓨터공학과 학사
- 세종대학교 컴퓨터공학과 석사
- 이씨스 주식회사 기술연구소 연구원
- 관심분야 : 모바일 보안, 네트워크 보안

신 지 선 (Ji Sun Shin)

- 정회원
- 서울대학교 컴퓨터공학과 학사
- 메릴랜드 주립대학(University of Maryland at College Park) 컴퓨터과학과 박사
- 삼성SDS 책임연구원
- 세종대학교 정보보호학과 조교수
- 관심분야 : 정보보호, 암호학, 컴퓨터 보안