Android Malware Network Behavior Analysis at HTTP Protocol Packet Level

Shanshan Wang[1], Shifeng Hou[2], Lei Zhang[1(✉)], Zhenxiang Chen[1], and Hongbo Han[1]

[1] School of Information Science and Engineering, University of Jinan, Jinan 250022, China
zhanglei@ujn.edu.cn
[2] Library of Rizhao Polytechnic, Rizhao 276826, China

Abstract. Smart phones, particularly the ones based on Android, have become the most popular devices. The surfing habits of users have been changed from the traditional PC terminal to mobile terminal officially. However, the mobile terminal application exposes more and more problems. Two common ways to analyze malware are source code analysis and dynamic behavior analysis. Researchers pay little attention to the network traffic generated by mobile terminal application. Nevertheless, shell technology makes source code analysis difficult while dynamic behavior analysis consumes too much resource. In fact, normal application and malware perform differently at the network level. We found that the features of HTTP packet are dramatically different in normal traffic and malicious traffic dataset. The application analysis from the perspective of network traffic can provide us a new way to detect malware.

Keywords: Android · Malware · Network traffic · Analyze · Detection

1 Introduction

The vigorous development of smart phones leads us to a new network area, where we have no time limitations, no space limitations and even no hardware limitations. A variety of smart devices and mobile applications flood every aspect of people's lives. But as former computer age, virus, Trojans and other security threats are also predictable. How to protect users' smart devices from invasion and protect users' privacy are more and more important.

Now there are two common mobile malware detection technologies: The one is statistic scanning technology. The other one is dynamic analysis technology. The main idea of statistic scanning is based on the known characteristics of the virus to match with the source code of the application. If the scanning results are consistent with a virus in some aspects, it is considered as a malware. If the result does not contain any virus' feature, it can be considered as a normal application. Dynamic analysis technology is according to the procedure of invoked method. The invoked methods mainly include unusual behavior of the operating system layer, accessing to sensitive data, calling to key system functions, etc.

G. Wang et al. (Eds.): ICA3PP 2015 Workshops, LNCS 9532, pp. 497–507, 2015.
DOI: 10.1007/978-3-319-27161-3_45

This approach requires massive calculation and mobile phones consume too much resource.

Generally speaking, no matter malwares or unwanted applications will affect the network behavior patterns. Just on the application layer of network can we find a number of different network traffic features. The innovations and contributions of this paper are as follows:

- An automatic network traffic generation and collection platform [1] was exploited. Through this platform we obtain abundant traffic data of normal Android applications and Android malwares.
- We analyzed the network traffic features of normal Android applications and Android malwares. In the process of features analysis and comparison, we concluded that there were a great differences between normal traffic and malicious traffic.

2 Related Work

Now a lot of network traffic analysis are aimed at computer terminals and pay little attention to the mobile terminals. But there are still some previous researches provide us with a lot of reference.

Zhou et al. [2] dissected the characterization and evolution of Android malwares. They had managed to collect more than 1,200 malware samples that covered the majority of existing Android malware families. In addition, they systematically characterized malware from various aspects, including their installation methods, activation mechanisms as well as the nature of carried malicious payloads. Cheng et al. [3] designed SmartSiren to collect the communication activity information from the smart phones. But it must run a agent on the smart phones. Hong et al. [4] made some summarizes about smart phone viruses characteristic and detection method comprehensively. They collected the flow of traffic and compared them to the fitting curve in real time. Tenenboim-Chekina et al. [5] described and analyzed a new type of malware which has the ability of self-updating. They also analyzed this malware based on network.

These prior works showed that malwares and normal applications have many different behaviors in many sides. This paper focuses on the network traffic features between malwares and normal applications. In fact we have found that on the network traffic layer the noramal traffic and malicious traffic have different traffic features. For instance, sending and receiving bytes, sending and receiving data, inner and outer time intervals. By comparison and analysis of the same feature in normal traffic and malicious traffic, we can better understand the network behavior patterns of malware. This approach provides a new idea for malware detection.

3 Methodology

Firstly, we obtained abundant normal Android applications and Android malwares (In this paper, we only focus on Android application. So for the sake of

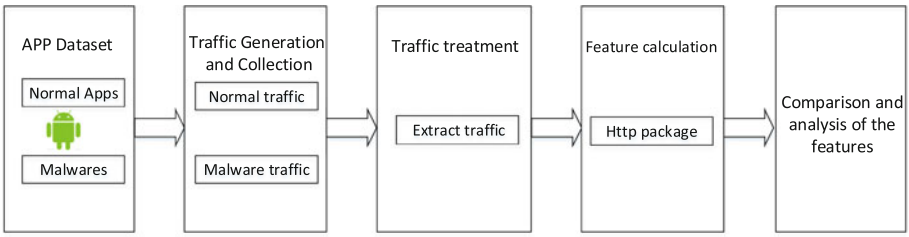

Fig. 1. Flow diagram of the mathodology

simplicity, in the following we called normal Android applictions as normal applications, the malicious Android applications as malwares). Secondly, we collected the network traffic generated by normal applications and malwares respectively using the traffic generation and collection platform. Thirdly, we calculated the same feature in normal traffic and malicious traffic dataset respectively. Fourthly, we compared these features and analyzed the reasons. Figure 1 is a flow diagram of our work. It can describe the work from an overall perspective.

3.1 Normal Application Dataset and Malware Dataset

In experiment, the normal application dataset is obtained from the Android market. We wrote a crawlers using the python language. The crawkers can realize downloading the applications to the PC from the Android market uninterruptedly. In order to make the result more accurate, four virus detection tools (kaspersky [6], avira, Lookout, AVG) concurrently were used to test the applications which we download from the Android market. That is to say, as for every application from the Android market was tested four times. The malious applications were filtered out. The applications whose four test results are benign were selected as our normal application dataset. 5666 applications are divided into 18 categories.

The malwares are obtained from Drebin project [4]. Additionally, it includes all samples from the Android Malware Genome Project [2]. We removed the ads applications out. The remaining 5560 malwares as our malware dataset.

3.2 Traffic Generation and Collection Platform

In order to get the network traffic which the experiment required. We utilize the traffic generation and collection platform to obtain traffic dataset. This platform is made up of four parts: foundation platform, traffic generator, traffic collector and network proxy/firewall.

The foundation platform consists of Android emulator (AVD) and Android debug bridge(ADB). This foundation platform provides a basic Android simulation environment and command line mode of interaction and it could realize some basic functionalities: creation, installation and operation. The traffic generator's

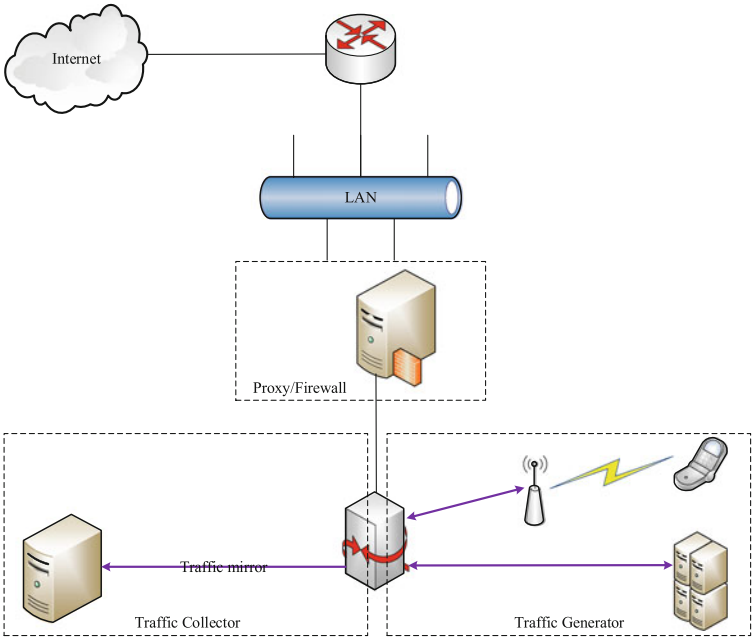

Fig. 2. Traffic generation and collection platform

task is to install and activate the normal applications and malwares to generate traffic. The traffic collector is designed to realize the function of capturing normal traffic and malicious traffic dataset. We utilize traffic mirror technology to mirror traffic data which pass through the gateway of server. The network proxy's/firewall's task is to mirror and control the attack behavior. Figure 2 is the structure diagram of traffic generation and collection platform.

3.3 Extracting Pure Malicious Traffic

The traffic generated by malware are not all malicious. So in order to make the result more accurate, our work joined the step that extracted pure malicious traffic. Firstly, we split every flow in network traffic according to the quintuple form (source IP, source port, destination IP, destination port and protocol). Secondly, we parsed the HTTP packets of each flow and extracted 'Host' fields and then these fields were sent to VirusTotal [7] to test. If the test result is abnormal, we can determine the flow as malicious flow. Then the flow was added to the malicious traffic as the malicious traffic dataset. In this way, finally we got our pure malicious traffic dataset. Table 1 lists the application dataset and network traffic dataset. The $ANumber$ represents application number while the $TNumber$ represents traffic number.

Table 1. The applcation dataset and network traffic dataset

Normal applications			Malwares		
Category	ANumber	TNumber	Family	ANumber	TNumber
Game	1328	320	FakeInstaller	925	79
Productivity	581	350	DroidKungfu	667	193
AntiVirus	385	385	Plankton	625	475
DailyLife	385	350	Opfake	613	89
Reading	343	343	GinMaster	339	3
NewsAndMagazine	332	332	BaseBridge	330	220
HealthAndFitness	328	328	Iconosys	152	39
Finance	324	274	Imlog	52	12
Education	320	320	FakeDoc	132	119
MediaAndVideo	290	290	Geinimi	92	3
Photography	237	237	Adrd	91	13
Input	228	228	Hamob	124	12
Social	208	208	ExplcitLinuxltoor	70	2
Communication	116	116	Glodream	69	12
TravelAndLocal	103	103	MobileTx	69	46
Personalization	62	62	FakeRun	61	52
Tools	52	52	SMSreg	108	7
Browser	44	44	Gappusin	58	89

4 HTTP Packet Analysis

On the basis of above traffic dataset, we began our analysis. Our analysis are at the packet level. We counted the application layer protocol and found that: On the application layer, the number of HTTP protocol packets accounted for 71.69 % and the number of DNS protocols packet accounted for 28.25 %, while only 0.06 % is the SSL protocol packets. Moreover, the most common way to get users' information is through HTTP request. Users' private information were sent to the server and this paper mainly analyzes the features of HTTP packet.

First of all, different time or different applications may affect the result of the experiment. But we obtained normal traffic and malicious traffic under the same network environment. Moreover, we get a large number of traffic data, the result is relatively reliable.

4.1 The HTTP Packet Average Length

HTTP packet length occupies an important position in the analysis of network traffic. In general, for all kinds of malware traffic, the packet length is an important parameter. Especially in data loss and in theft behavior detection

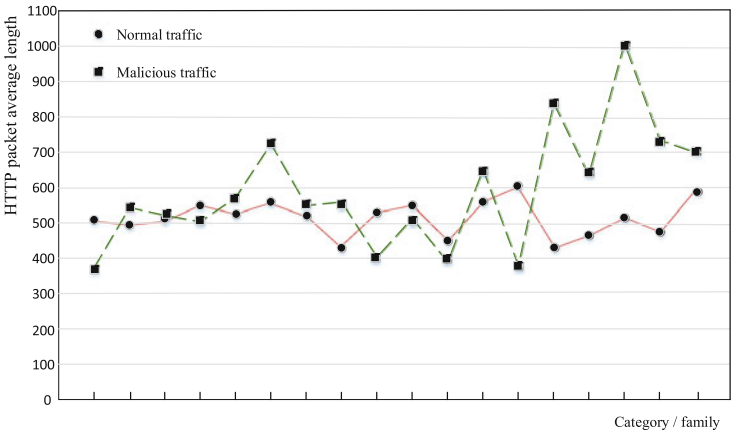

Fig. 3. HTTP packet average length in normal traffic and malicious traffic

technology, HTTP packet length plays an important role. So we calculated the HTTP packet length in normal traffic and in malicious traffic and then two sets of data were contrasted to looking for the rule.

For normal traffic dataset and malicious traffic dataset, the HTTP packet length were analyzed. Figure 3 is the HTTP packet average length in noraml traffic and in malicious traffic. The horizontal axis represents category name in normal traffic or family name in malicious traffic. The ordinate axis represents the HTTP packet average length of every category or every family.

From the comparison of HTTP packet average length in normal traffic and in malicious traffic. Several points were concluded.

- We calculated that the minimal HTTP packet average length is 433 bytes and the biggest HTTP packet average length is 604 bytes in normal traffic samples. while the minimal HTTP packet length is 369 bytes and the biggest HTTP packet length is 1038 bytes in malicious traffic samples.
- According to the formula of standard deviation we calculated that the standard deviation of the normal traffic samples is 50.3 and the standard deviation of malicious traffic samples is 173.5. Because the standard deviation can be used to measure the fluctuation magnitude of a batch of data. Under the condition of the same sample size, the bigger standard deviation shows the greater data volatility. The calculation results indicated that the HTTP packet length in normal traffic is more stable than in malicious traffic.

4.2 HTTP Packet Length Distribution

Figure 4 declares HTTP packet length distribution of normal traffic and malicious traffic. From this figure, we found the following points.

- No matter in normal traffic or in malicious traffic, the HTTP packet length distribution accords with normal distribution in general.

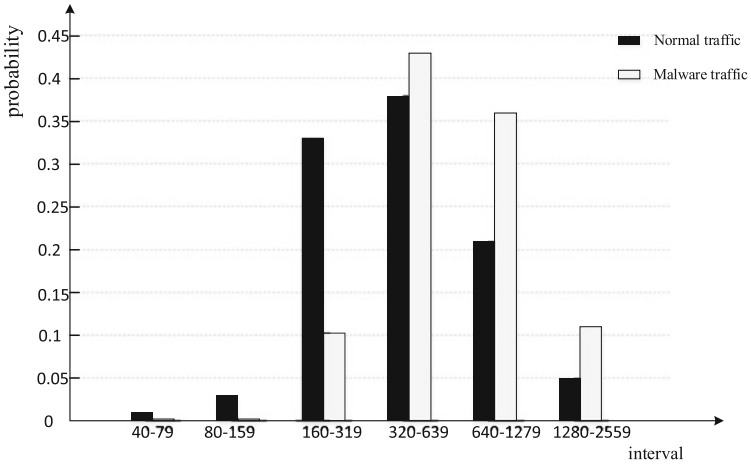

Fig. 4. HTTP packet length distribution in noraml traffic and malicious traffic dataset

- Both normal traffic and malicious traffic, most of data which the proportion of HTTP packet length fall in the interval of 320-619. Specific data of the proportion is 35 % in normal traffic and 43 % in malicious traffic dataset.
- The HTTP packet length is bigger in malicious traffic dataset than in normal traffic generally speaking. Such as HTTP packet length within the interval of 1280-2559, 640-1279, 320-639 accounted for 12 %, 35 %, 43 % in malicious traffic dataset while accounted for 5 %, 20 % and 5 % in normal traffic respectively.

4.3 HTTP Upload Packet Number and Download Packet Number

Table 2 shows HTTP upload and download packet number. For simplicity, the *DPN* is defined as HTTP download packet number and the *UPN* means HTTP upload packet number.

- Both normal traffic and malicious traffic dataset, the number of HTTP upload packet equals with the number of HTTP download packet generally speaking.
- As for normal traffic, In 11 samples (account for 61.1 %), the number of HTTP download packet is greater than the number of HTTP upload packet. There are 7 samples (account for 38.9 %) in which the number of HTTP download packet is less than the number of HTTP upload packet. For malicious traffic dataset, there are 3 samples (account for 16.7 %) in which the number of HTTP download packet is greater than the number of HTTP upload packet. There are 8 samples (account for 44.4 %) in which the number of HTTP download packet equals to the number of HTTP upload packet. In 11 samples (account for 38.9 %) the number of HTTP download packet is less than the number of HTTP upload packet.

Table 2. HTTP download packet and upload packet number in two traffic dataset

Normal traffic dataset			Malicious traffic dataset		
Category	DPN	UPN	Family	DPN	UPN
AntiVirus	6675	5357	Adrd	13	13
Browser	124	123	BaseBridge	344	859
Communication	583	570	DroidKungFu	735	733
DailyLife	2980	2868	ExploitLinuxlotoor	8	8
Education	3222	3517	FakeDoc	469	469
Finance	2005	1929	FakeInstaller	142	142
HealthAndFitness	1484	1457	FakeRun	147	143
Input	1826	1974	Gappusin	197	199
MediaAndVideo	6602	6593	Geinimi	10	20
NewsAndMagazines	3353	3311	GinMaster	6	6
Personalization	747	933	GlodDream	13	13
Photography	1372	1399	Hamob	74	75
Productivity	2030	1976	Iconosys	40	40
Reading	6716	6680	Imlog	6	6
Social	4775	4505	MobileTx	472	464
Tools	372	380	Opfake	195	196
TravelAndLocal	2225	2479	Plankton	1931	2207
Games	3145	3200	SMSreg	10462	21997

After application installed successfully, many applications will load a lot of resources, so the HTTP download packet number is greater than the HTTP upload packet number. In the malicious traffic dataset, the majority of malware is activated [8] after restarting. The activated malwares' network behavior become more active. They not only need to load the application resources, but also need to receive and execute command from a remote server. At the same time, malwares upload a lot of user information to a remote server. So most of the HTTP download packet number is less than or equal to HTTP upload packet number in malicious traffic dataset.

4.4 HTTP Upload Bytes and Download Bytes

Table 3 describes HTTP download bytes and upload bytes of normal traffic and malicious traffic dataset. The *DBytes* is defined as HTTP download bytes and the *UBytes* means HTTP upload bytes. From Table 3 we can conclude several points.

Table 3. HTTP download bytes and upload bytes in two traffic dataset

Normal traffic			Malicious traffic		
Category	DBytes	UBytes	Family	DBytes	UBytes
AntiVirus	4,314,141	1,767,586	Adrd	3445	6147
Browser	69,864	51,817	BaseBridge	182,665	459,353
Communication	366,386	217,205	DroidKungFu	419,541	348,344
DailyLife	1,840,103	1,405,723	ExploitLinuxlotoor	4332	3754
Education	1,914,591	1,651,154	FakeDoc	394,789	151,836
Finance	1,352,451	857,346	FakeInstaller	167,651	38,267
HealthAndFitness	923,474	601,519	FakeRun	61,312	96,120
Input	865,516	788,152	Gappusin	128,213	95,066
MediaAndVideo	3,584,366	3,450,731	Geinimi	4724	7519
NewsAndMagazines	2,237,195	1,443,726	GinMaster	3502	2648
Personalization	458,590	306,132	GlodDream	5768	4401
Photography	806,805	742,071	Hamob	52,298	46,343
Productivity	1,351,631	1,066,844	Iconosys	13,670	15,808
Reading	3,262,301	1,066,844	Imlog	4770	5196
Social	2,367,845	1,954,169	MobileTx	350,172	246,696
Tools	217,712	170,121	Opfake	287,074	118,935
TravelAndLocal	1,089,088	1,153,457	Plankton	1,405,248	1,628,966
Games	2,150,731	1,615,773	SMSreg	10,462	21,997

- For normal traffic, in all the samples, HTTP download bytes are bigger than HTTP upload bytes.
- For malicious traffic dataset, in 10 samples (account for 55.6 %), HTTP download bytes are bigger than HTTP upload bytes. In 8 samples (account for 44.4 %), HTTP download bytes are less than HTTP upload bytes.

In the first a few minutes many applications need to load the network resources they required. So the number of HTTP download bytes are larger in normal traffic. While for malicious applications, malware is activated and then a lot of malwares began to carry out illegal activities, such as stealing users' personal information and then uploading to remote server etc. So in some samples HTTP upload bytes are greater than download bytes.

5 Evaluation

We have analyzed several features of application-layer protocols. Because HTTP protocols account for a crucial part on application-layer, we focuses solely on HTTP protocol packet. The features we analyzed are HTTP packet average length, variance of HTTP packet length, distribution of HTTP packet length,

radio of HTTP upload packet number and download packet number as well as radio of HTTP upload bytes and download bytes. We found every feature performs differently in normal traffic and malicious traffic. But we cant assert every feature can be used to detect malware. Moreover there are large numbers of traffic features, which we didnt analyze. Our next work is to analyze more traffic features and find some specific features which can recognized malware from normal applications.

6 Conclusion

In this paper, we get abundant traffic dataset. A lot of differences at HTTP package protocol level were present of normal traffic and malicious traffic. By analyzing the causes of these differences, we can better understand the network behavior of malicious software. From the aspect of network traffic, normal application and malwares have different features which lays the foundation for the next work. The next step of our work is to deeply analyze the features of traffic and then to detect malware. In this experiment, there are some different features, such as HTTP packet average length, the distribution of HTTP packet length, the number of HTTP upload packet and download packet etc. This paper provides a feasible method for malware detection. Namely, if a feature of network traffic performs great differently in noraml traffic and malicious traffic dataset and then they can be used to detect malwares.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grants No. 61472164 and No. 61203105, the Natural Science Foundation of Shandong Province under Grants No. ZR2014JL042 and No. ZR2012FM010.

References

1. Chen, Z., Han, H., et al.: A first look at android malicious traffic dataset in first few minutes. In: The 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (2015)
2. Zhou, Y., Jiang, X.: Dissecting Android malware: characterization and evolution. In: 2012 IEEE Symposium on Security and Privacy (SP), Conference Proceedings, pp. 95–109. IEEE (2012)
3. Jerry, C., Wong Starsky, H.Y., Hao, Y., Songwu, L.: Smartsiren: Virus detection and alert for smartphones. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services, pp. 258–271. ACM (2007)
4. Yunfeng, H., Chao, X., Dixin, S.: Research of smart phone malware detection based on anomaly data flow monitoring. Comput. Secur. 9(11–14) (2012)
5. Tenenboim-Chekina, L., Barad, O., Shabtai, A., Mimran, D., Shapira, B., Elovici, Y.: Detecting application update attack on mobile devices through network features. In: INFOCOM 2013 (2013)

6. Kaspersky. http://www.kaspersky.com.cn/
7. Virustotal. http://www.virustotal.com/
8. Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., Shapira, B., Elovici, Y.: Mobile malware detection through analysis of deviations in application network behavior. Comput. Secur. **43**, 1–18 (2014)