Group 2 - Android Malware - Security Investments

Introduction

In this report, we consider a security issue and then evaluate the different stakeholders associated to that issue. We highlight how the differences in security metrics reveal different aspects of security performance for different parties. We also identify several strategies for each of the stakeholders. Then, we pick one strategy and evaluate its feasibility by estimating its Return on Security Investment (ROSI).

Dataset

The given dataset contains information regarding games and software that has been uploaded on Baidu and 360, two of the most popular android application stores in China. For every app, it contains information regarding its category, software version, size, software package, number of downloads, download URL, last update date, and download date.

Problem Owner

The security issue under consideration is the presence of malware infected apps on an app platform. There are multiple stakeholders of the security issue at hand. The one particular problem owner that we will evaluate our security metrics on is the ‘‘user of a platform”. Other problem owners are the platform owners and the developers who develop the apps that are hosted on a platform. These parties will be discussed later in the document.

Security Performance Evaluation

Just like the last time, we assume those apps to be malicious which were last updated more than 50 days ago. We make this assumption because there is no security related information in our dataset. We chose 50 days as a suitable number based on the distribution of the last-updated-days values.

Moreover, since this time we consider the problem owner as ‘users of a platform’, we also assume that the probability of a certain user downloading the app from a certain category is uniform, so there is no need to consider individual categories for different types of users - each category is equally likely. Also, the impact of a malicious app on a certain user is also uniform. We use these assumptions because we do not have the data to actually calculate these values, and these assumptions help make sense of the metrics.

Metric 1: Maliciousness of a platform

$$maliciousness = \frac{\#apps_m}{\#apps}$$

Where:

$\#apps_m$ = number of malicious apps

$\#apps$ = total number of apps on a platform

The first metric measures the degree of maliciousness of a platform based on the fraction of malicious apps that are present on that platform. This metric is directly relevant to the platform owners, but can help users of a platform in choosing a suitable platform as well. If security of a platform is a factor based on which different platforms are ranked, then this metric will help form that ranking.

This metric will be 0 if no app is malicious and 1 if all apps are malicious. In this way, it is a normalized metric that can be used to compare multiple platforms.

Table 1 shows the maliciousness of Baidu and 360. It is clear that 360 is much more malicious than Baidu, and the reason is solely the number of malicious apps. This metric highlights the fact that if two platforms have the same number of apps, where one platform regularly patches its apps (or rather the developers of that platform do that) while the other does not, the latter one will be more malicious than the former. It also highlights that if a much smaller platform (in terms of number of apps hosted) does not regularly patch the apps, compared to a much bigger platform that imposes such restrictions of regular app updates, the smaller platform will be more malicious than the bigger one, even though the sizes of the platform suggest otherwise.

Table 1: Metrics evaluation

	# apps	# downloads	# malicious apps	# malicious downloads	maliciousness	impact
baidu	2258	6753740305	1403	2621453730	0.6213463242	0.2411746033
p360	18846	5443398109	17004	3793291588	0.9022604266	0.6287500598

Metric 2: Impact on users

$$impact = \frac{\#apps_m * \#downloads_m}{\#apps * \#downloads}$$

Where:

$\#apps_m$ = number of malicious apps

$\#downloads_m$ = number of times malicious apps were downloaded

$\#apps$ = total number of apps on a platform

$\#downloads$ = total number of times all apps were downloaded

The second metric measures the collective impact on the users of a platform because of malicious apps on that platform. This metric is directly relevant to the users of a platform, but the developers may take it into account if they want to check user satisfaction of their apps, for example. This metric's value also lies between 0 and 1, where 1 means total impact on all users, while 0 means no impact on any user.

This metric highlights the following difference in security performance: The impact on users is directly affected by the number of apps that the users are using (or in other words, a platform is hosting) and the number of times those apps have been downloaded. For example, suppose that a platform 'A' only hosts one app which is downloaded by 10 million people, and another platform 'B' hosts 1 million apps but the total user base of that platform is 5 million people. Platform 'A' will have a much higher impact if that one app that it hosts was to go malicious, as compared to platform 'B', where even if all the apps were to go malicious, it will only have half of that impact.

Table 1 shows the values of impact for the users of Baidu and 360. Clearly, the impact for users of 360 is much higher than the users of Baidu, even though Baidu is a much more popular platform (and has many more downloads than 360). This shows that if the metric's value is 1, it means that all the apps on a platform are malicious so there is a total impact. However, if the metric's value is 0, it can either mean that no app on that platform is malicious, or that nobody downloaded any app from that platform. The latter case remains true even if all the apps on that platform were malicious - If the users do not use the app, there is no impact.

Risk Strategies by Problem Owner (Users)

The are several strategies that customers can take in order to mitigate the risk of downloading/using a malicious app. Since the users lie at the lowest level of the ecosystem, the only strategies that they can take are to protect themselves from direct impact. Strategies for indirect impact and supporting their business (from the slides) are not applicable in their context.

Firstly, depending on the type of OS they are running on their devices, a strategy can be to limit certain permissions while using an app. For instance, there is no reason for a calculator app to get access to the location of a user. In this case, users could deny the access to the location even if it is asked by the app. This, on the other hand, might cause some apps to lose their functionality depending on the type of access. iPhone users have such control much more frequently than android users.

To protect themselves, users can install anti-malware and antivirus on their phones, so that even if the platform hosts malicious apps, the users' phones will be protected. On the other hand, if an app exploits a zero-day, no scanner can save the user.

If the users become sensitive to platforms that provide security, the platform owners will have an incentive to improve the security of their platforms. Otherwise, users can switch platforms. This strategy, however, is highly dependent on the type of application the user uses -- in case the application is only available on one platform, then the user is technically locked-in and cannot make the switch, and the platform owner has no incentive to improve the security either.

Based on the aforementioned strategies, we see that the customers do not have a lot of options in order to mitigate the security issue under consideration, but the stakeholder that does have a lot of options, i.e. platform owner, has no real incentive to resolve the issue because they might be dealing with other aspects of their platform, but the malware might not be a priority for them.

Overall, if the vast majority of the users start prioritizing the security of a platform as the main factor to use a platform and move to a safer platform as a consequence, platform owners would start prioritizing the security of their respective platforms.

Other stakeholders

As previously specified, the security issue concerns the presence of malware infected apps on an android app platform. Other than the main problem owner, i.e. users, there are other actors that are either directly or indirectly associated to the platform and the apps hosted on it. They are discussed below:

Platform owners

Platform owners own the business behind the platform in question, which forms the infrastructure on which app developers sell apps and users buy apps. The main incentive for the platform owners to take any action can either be to protect themselves from harm, protect their customers from harm or to enhance their business. Whether the malware on the platform is considered a problem by the platform

owners or not is dependent on their value proposition. The platform can create internal regulations and organizational policies in order to monitor and control the applications that are being uploaded.
Platform owners' main goal is the reputation of the brand name and to have an edge over their competitors. To do that, the platform will have to be selected by developers and users alike, and to do that, it has to provide a wide selection of apps and to provide security for both developers and users. The latter requirement becomes even more crucial if the developers and users are security sensitive.

Developers

A developer is the one that writes the code of the app in order to provide a service to the users. Developers utilize the platform to sell their apps. A developer is the second most important actor after the problem owner, i.e. user, so his abilities and motives greatly influence his/her app's security. If the app is badly designed, or there are security vulnerabilities in it that can be used by malware to perform illegal activities on a user's device, the developer will get a bad reputation which may result in him/her not being able to increase his/her user base. Thus, it is in the best interest of the developers to keep their apps patched and properly maintained to retain their customers (again, we consider security-aware customers). On the flip side, there are rogue developers that purposefully leave backdoors in apps, or misuse permissions for their ulterior motives. The platform owners do not have a direct incentive to get rid of such developers so there is not much anyone can do about them. (Because of the diverging interests, these latter type of developers are separately considered in the section 'Attackers').

Government

The government of a country forms a very interesting case of security consumers where their motives behind their strategies are based on the indirect impact to their citizens as well as on the direct impact to the whole country's economy. Some countries, like the Netherlands, has created regulations and laws that control the information flow between parties and protects users' right to privacy. Therefore, a fraudulent app's developer will face more serious consequences if caught than other countries with no such regulations. Considering the Chinese market[1], the regulations are very complex as there is no single party in control of the privacy regulations. Also, since the Chinese government has censored play stores like Google Play, the customers only have an option to use local markets to avail android apps.

Researchers/tech advances

Researchers are actors that are remotely associated to the security issue under consideration, and can be considered security providers. Their findings dictate the latest countermeasures that are available to tackle the malware problem. Technological researchers can find new loopholes which can be used to exploit vulnerabilities, and can either inform the other parties about it or provide solutions to fix them. Security researchers are business-driven in the sense that that their growth strategy is to tackle the security issues to gain more insight on the vulnerabilities themselves and to base future research on the findings.

Attackers

Attackers are the stakeholders that monetize on exploiting the vulnerabilities in apps. They can either be a separate entity than the app developers or can be rogue app developers that perform illegal activities using legit-looking apps. Their strategies are often in direct contrast with almost all the other parties because they want to exploit as many vulnerabilities as possible. These actors influence the security issue by being present -- if such actors were absent, there would be no exploitation and hence, no risk.

[1] http://privacylaw.proskauer.com/2017/05/articles/international/a-primer-on-chinas-new-cybersecurity-law-privacy-cross-border-transfer-requirements-and-data-localization/

Risk Strategies by Other Stakeholders

Platform owners

Platform owners are essentially security consumers. So, their reasons for implementing security can either be to protect their business from direct impact, protect their consumers, or to enhance their business. Example strategies for direct impact are as follows:

- Reputation - publish reports about how many apps they blocked in a year,
- Regulations/policies (complying to regulations, proper audit trail maintenance),
- Legally binding the app developers to stay sincere to functional requirement,
- Organizational wide policies (incident response teams, follow best practices).

For indirect impact the following strategies could be adopted:

- Remove apps that are not updated for more than 50 days,
- Limiting excessive permissions - (privacy aspect),
- Detect vulnerability in the apps (either by matching CVEs[2] or using our assumption of 50 days) - (protecting users),
- Alert developers if negative feedback is received or a scanner flags their app

And for supporting the business the following strategies can be implemented:

- Imposing a small monetary requirement to upload apps to back-off malicious developers,
- App scanning before publishing the app,
- Developers upload checksum - checking integrity of app,
- Private/public keys of developer - non-repudiation/identity management

Developers

The strategies that a developer can use in order to tackle malware-infected apps are as follows:

- A developer must monitor security advancements regularly and ensure that his application is not affected. (direct impact)
- Protecting their reputation by regularly updating their apps (direct and indirect impact)
- Complying to regulations and maintaining proper audit trails (direct impact)
- Publishing regular patches that concern new vulnerabilities to alert their users (indirect impact)
- To develop apps with good design and security in mind (enhancing business)
- Also, gathering and analyzing the feedback from current users in order to identify potential issues is a really good and helpful risk strategy from the developer's point of view. User feedback can identify issues that were not identified during the development phase but may be crucial for the security part of the application. This strategy will do all -- protect the developer's business, protect the users as a consequence, and support business strategy by actively involving their customers in the development phase.

Government

A government's strategies in order to tackle the problem in question are limited, but very powerful.

- In order to protect the government itself, it can create laws and regulations under which any action to hurt it would be prosecutable under these regulations, or hefty fines are charged to the offenders. (direct impact)
- In order to protect the civilians, privacy laws and anti-fraud schemes can be introduced. However, the tricky part in these cases would be attribution of a crime to a person or persons involved.

[2] https://cve.mitre.org/

Moreover, the government does not really have an incentive to introduce such mechanism unless it, itself, is security-aware, which in many countries is not the case. (indirect impact)
- Under some circumstances, the governments that are being convinced by researchers or other parties to enforce protective schemes are better off if they do implement them in order to get the chance to be elected again in the next term. Since a government does not have any 'competitors' when it is in charge, this might be the only strategy for them in the context of 'supporting their business'.

Researchers/tech advances

The risk strategies that researchers will use as actors in order to tackle our security issue are as follows:
- Identification and analysis of the signatures from the maliciously infected software to proactively create countermeasures for future vulnerabilities. Researches' main goal is to keep the security industry up to date with the latest security solutions. This strategy applies to both protecting their customers (which are security consumers, in general) and to enhance their business.
- It is unlikely that the researchers are affected by the malware itself so they do not have any strategies to protect themselves from the direct impact.

Attackers

The risk strategies for the attackers are quite different from the other actors that we have considered previously:
- Being vigilant with their identities when exploiting vulnerabilities (Direct impact)
- Gathering intel from different sources and training to evade security mechanisms such as anti-malware (Direct impact)
- To keep probing different apps and to exploit as many vulnerabilities as possible to maximize their benefit (supporting business)
- To become popular among the 'underworld' or among the offenders (supporting business)
- Often times, these attackers are hired by 3rd-parties to perform illegal activities in return for money. In this context, protecting the identity of the clients, if caught, can be one strategy to protect from indirect impact. However, if an attacker is caught, it is most likely that he/she will spend quite some time behind bars, and the investigators exploit this fact to provide incentives for the attacker to give away their clients' names in exchange for leniency. Therefore, this strategy is highly dependent on the circumstances. (indirect impact)

Actors with contrasting strategies

There are actors with different strategies because of the way the malware problem affects them. Here are a few examples:
- Attackers vs. users -- the attackers want more malware on the platform while the users want less of it.
- Government vs. developers -- a government wants to protect the users and their privacy, while the developers are concerned with their profit margins. However, this might not always be the case. For example, when the developers care about their users, they will be on the same side as the government.
- Developers vs. platform owners -- assuming that the platform owners are interested in protecting the end users, they would want to limit the excessive use of miscellaneous permissions while the developers would want to collect as much information as possible to run analytics for market segmentation, for example.

Strategies over time

As the world moves forward in the digital revolution, the risk strategies are changing significantly over time. In the context of the security issue, the strategies that were used many years ago are obsolete in today's world. For example, five years ago mobile phones did not support as many functionalities as today, so privacy regulations were not in place. But now, as more and more sensitive information is on the phones, stricter privacy regulations are in place and strategies to protect against malware that exploits different kind of vulnerabilities have changed.

The attackers themselves are becoming increasingly sophisticated. Previously the focus was on reactive security, but now many companies are pushing for security by design[3] to become streamlined. Also, the importance of killchain[4] and threat intelligence[5] is being realized and incorporated in security strategies. These are completely different ways to think about security that were not present until a few years ago.

Strategy Feasibility - ROSI

Assumption:

- Number of downloads are equal to number of users; each user has installed only on app on the market and no two users have installed the same app.
- Incidents involving less losses are more likely than bigger incidents. Therefore, the probability distribution would be right tailed.

Strategy under consideration: Regularly scanning apps and alerting developers if negative feedback is seen or a scanner raises a red flag

ROSI - A case of Baidu:

For example's sake, we consider Baidu as a running example of the platform and the decisions management will take to improve the security of this platform.

Costs involved

Direct costs

The direct cost is a sum of acquisition, deployment, and maintenance cost of the security solution. This would be as follows:

1. Implementing the scanner
2. Training employees on how to use the scanner
3. Red flag detection framework (sentiment analysis on user comments, running tests and then emailing developers about their app's health)
4. Maintenance of the scanner and detection framework

Indirect costs

This is the cost of productivity loss, opportunity loss and the cost of any other problems that might arise because of the security solution. Such costs are rarely expressed in monetary values. Such costs are:

5. Slowdown of app uploads because of scanning
6. Extra communication overhead with the developers to explain the setup

---

[3] https://aws.amazon.com/compliance/security-by-design/

[4] http://resources.infosecinstitute.com/cyber-kill-chain-is-a-great-idea-but-is-it-something-your-company-can-implement/#gref

[5] https://www.tripwire.com/state-of-security/security-data-protection/cyber-threat-intelligence/

Sunk Costs

There are no sunk costs because whether the malware problem persists or not, the scanner is a good addition to the platform's overall security.

In terms of monetary value, we assume that 1) and 3) are onetime costs equal to $2,000. 2) and 4) are ongoing costs that have to be borne after regular intervals. Let's assume that they equal to $3,000. 5) and 6) cost $15 per app on the platform (assuming every app represents a unique developer). So, for Baidu, total cost would be 2258*15 (number of apps * cost) = $33,870.

Benefits

1. Improved reputation of the platform
2. Increase in revenue because of trustworthy ads - no malware, increased trust of users so more in-app purchases
3. Reduction in malware on the platform

The monetary value for benefits would be the difference of expected losses with and without the control measure in place. The benefits are calculated as follows:

Without implementing strategy

To get the frequency of incidents/infections, we look at the number of users who have installed malicious apps. From the dataset, for **Baidu,** there are 2,62,14,53,730 infections. We assume a uniform distribution of infections over a time period, so we have 21,375 infections per year. We make this assumption because of the lack of frequency information in our dataset. We also assume that each incident of an infection will directly or indirectly cause an impact of $100 to $1000. This impact remains constant regardless of the implementation of the strategy. Therefore, roughly, we would have:

Frequency = 20,000 -- 22,750 /year

Impact = $100 -- $1,000 / incident

Expected losses = $20,00,000 -- $2,27,50,000/year

Implementing strategy

We assume that implementing the strategy will reduce the incidents by 80%. Therefore:

Frequency = 4,000 -- 4,550 /year

Impact = $100 -- $1,000 / incident

Expected losses = $4,00,000 -- $45,50,000/year

$$\text{RoSI} = \frac{ALE_0 - ALE_S - c}{c}$$

Total cost for Baidu platform = $33,870 + $3,000 + $2,000 = $38,870

Benefit for Baidu (ALE_o-ALE_s) = *$1,600,000 -- $18,200,000 /year*

ROSI (lower) = (*$20,00,000 - $4,00,000* - $38,870)/$38,870 = $40.16/year

ROSI (upper) = (*$2,27,50,000 - $45,50,000* - $38,870)/$38,870 = $476/year

In the following chart 1, we plot the ROSI against the assumed probability of incidents occurring. Since our dataset does not contain any information regarding frequency of the incidents, the probabilities are assumed such that more incidents of low impact are more likely than incidents of bigger impact. Therefore, the series has a right tail. The probabilities are such that the sum of all events adds up to 1.

Based on the figure, if a strategy is implemented with an aim to have the highest ROSI value, i.e. $476, not many incidents of that magnitude will take place and most of the investment will go to waste. However, by looking at the threat landscape, if we implement a strategy that gives the minimum ROSI, the maximum number of incidents can be thwarted.

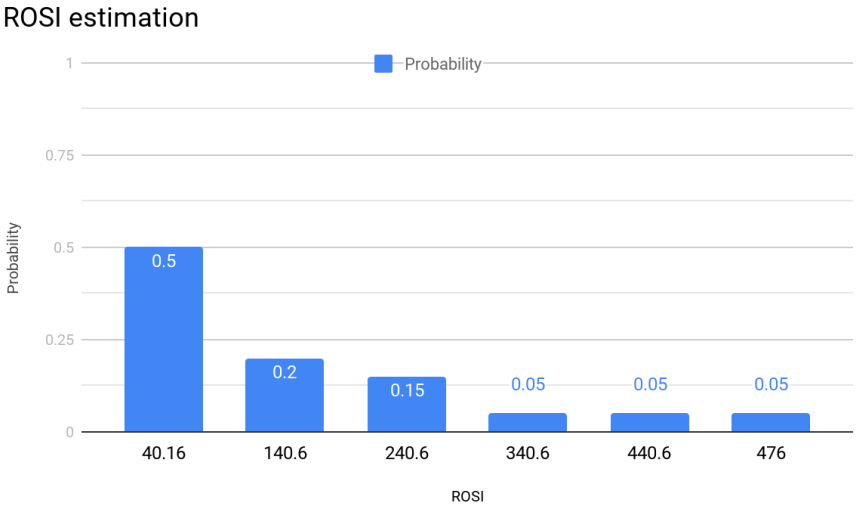

Figure 1: ROSI estimation

Conclusion

There can be multiple stakeholders involved in a security incident. Each stakeholder comes up with a strategy based on their incentives. Not all stakeholders have equal power and/or incentive to take action -- many times, the stakeholder getting most affected by the problem is the one who has the least number of options to resolve the security issue. Return on Security Investment (ROSI) is a scaled value of benefits (scaled by the cost), often used by managers to assess the feasibility of such strategies. In this report, we consider the malware on a platform as a security issue, and consider the security strategies of several different stakeholders. We also pick one strategy and calculate ROSI based on the dataset.