Insights into Rooted and Non-Rooted Android Mobile Devices with Behavior Analytics

Yun Shen
Symantec Research Labs
Dublin
Ireland
Yun_Shen@symantec.com

Nathan Evans
Symantec Research Labs
Herndon
USA
Nathan_Evans@symantec.com

Azzedine Benameur
Symantec Research Labs
Herndon
USA
Azzedine_Benameur@symantec.com

ABSTRACT

In the Android community, a commonly held belief is that rooted devices are more at-risk than their standard counterparts. Many applications include “protections” that disable certain functionality in the presence of a “rooted” device. In this paper, we carry out a detailed quantitative analysis on 6.14 million Android devices comparing rooted and non-rooted devices across a broad range of characteristics. Specifically our study focuses on measuring and analyzing various behavioral differences between rooted and non-rooted devices, and how such differences impact the security of rooted devices, e.g. making them more susceptible to malware or at higher risk. We further leverage our insights and implement a machine learning system to determine whether a device is rooted based on a comprehensive set of features with an accuracy of approximately 90%.

CCS Concepts

•Security and privacy → Mobile platform security;
•Computing methodologies → Machine learning;

Keywords

mobile security, rooted mobile device, machine learning

1. INTRODUCTION

It is undeniable that Android has become the most important operating system for mobile devices including smartphones and tablets. As of the end of 2014, Android was installed on more than 78% [3] of smartphones sold globally and over 70% of tablets sold. The success of Android is interesting partly because of its open nature. The majority of the Android source code is released under an open source license, and is derived in large part from Linux. This has allowed the Android ecosystem to grow and prosper. However, its open source nature has also led to extensive customization of the operating system (fragmentation), unlocked phones and a desire from some users for the ability to update and maintain their software as well as their operating system as they see fit. This has led to unlocking bootloaders, flashing custom ROMs and rooting of devices becoming more commonplace.

It is often mentioned [19, 15] that “rooting” an Android mobile device opens it up to more dangerous malware and security risks. Intuitively, this makes sense: “rooting” a device means allowing access to the lowest layers of the operating system, possibly circumventing built-in security measures such as application sandboxing and limiting software to known sources. However, some users choose to take on this apparent risk for various reasons, including the freedom to install a wider range of applications, remove so-called “bloatware” that is factory or carrier installed and update/customize their operating system.

Due to the real or perceived impact on security, many mobile applications also check for the presence of root access and may alter their functionality if detected. Some of the reasons for this are “Bring your own device” (BYOD) policy enforcement and security evaluation. Examples of software including these checks are banking applications, media streaming applications, mobile device management software and security software. In [10], the authors demonstrate that nearly all of the most popular security and MDM application for Android contain some functionality for detecting whether a device is rooted or not.

Knowing a device is rooted is important for certain applications, but can easily be tricked, is one of the motivations for this study, raising the question of whether or not we can detect that a device is rooted solely based on application behavior and/or mobile device characteristics. To that end, we believe this paper to be the first to analyze a large population of real-world Android mobile devices with the specific focus on detailing the differences and similarities between rooted and non-rooted devices. In addition to analyzing a large amount of data gathered on mobile devices, we also raise the bigger question of whether rooted devices are, in general, more “dangerous” than non-rooted devices.

Using data from 6.14 million unique Android devices, we

SAC 2016,April 04-08, 2016, Pisa, Italy
©2016 ACM. ISBN 978-1-4503-3739-7/16/04…$15.00
DOI: http://dx.doi.org/10.1145/2851613.2851713

demonstrate that contrary to our initial hypothesis, non-rooted devices tend to have more applications installed than rooted, even when excluding system and pre-installed applications; and non-rooted devices a greater average number of application updates than rooted. We also find that non-rooted devices on average send and receive more data than rooted devices, possibly due to the higher number of applications installed. Surprisingly, we find that rooted devices do not in general have more malware installed than non-rooted devices.

Building on top of these insights and mobile behavior properties, we create an ensemble machine learning [9] classification model that allows us to take our dataset (with a total of 73 features) and train a model for determining whether a device falls into the rooted category or the non-rooted category. Our classifier is able to achieve an AUC score of 0.96 with an accuracy of approximately 90%.

The rest of the paper is divided as follows: Section 2 includes a summary of the data and the collection method. Section 3 then provides a deep dive analysis of the data, from the perspective of whether or not rooted devices exhibit different characteristics. Next, Section 4 describes a machine learning model we developed in an attempt to automatically classify a device as rooted or not based on features extracted from our data set. Section 5 covers related work in the Android malware and device analysis space. Finally, Section 6 concludes the paper and discusses possible future work based on the results of our analysis.

2. DATA SET

The data used for our analysis is a subset of the CompanyX Mobile Intelligence mobile data set. The Data is collected as an opt-in feature of CompanyX's mobile security solution "CompanyX Mobile Security", and is comprised of user data from more than 6.14 million unique Android devices. Per device data is collected, after explicit user consent, on the device in an event driven manner, and uploaded to the back end approximately twice a month (variations depend on network connection state, device usage, etc.). Data is stored in a SQL-like database and is retained for 3 years.

In order to keep our analysis tractable we limited our experiments to a sample of the data set. The sample is made up of the latest data available to us at the time of writing; 4 months of data running from the beginning of January, 2015 to the end of April, 2015. Since our analysis focuses on the possible differences between rooted and non-rooted devices, we randomly selected 100,000 of *each* device type. The type differentiation was done using an application that runs on the device and looks at various parameters to classify them.

We also used an ancillary data set for our categorical analysis of Android apps. For this we used another CompanyX database made up of APK's collected from multiple Android application markets and websites. This data includes the application name, hash, cost, category, etc. and is comprised of over 3 million total Android applications collected from 358 different android application stores and websites providing apps. This database is maintained by crawling app stores and websites on a regular basis and has apps from May 2012 to present.

3. ROOTED VS. NONROOTED

In this section we present our analysis and highlight the differences and similarities between rooted and non-rooted devices observed in our data set. We have narrowed down the scope of the data to attempt to answer questions that we think are interesting, or results that may be surprising as they uphold or refute preconceptions about the behavior of rooted device users. Prior to beginning our exploration of the mobile data set, we created a list of some characteristics we expected to observe, as well as some commonly held beliefs surrounding the reasons that a user would root their phone (see Section 5). Below, we briefly list two of the questions we posed: 1) *do users of rooted phones install more software because they are more technical, have the ability to add app stores, etc.?* and 2) *do rooted devices have different networking behavior than non-rooted (more data sent/received, more connections, etc.)?* We provide data and explore the answers to these questions in Section 3.1. Then, in Section 3.2 we touch on perhaps a more important question: 3) *do rooted devices exhibit more "risky" behavior than their non-rooted counterparts?*

3.1 Device Behavior

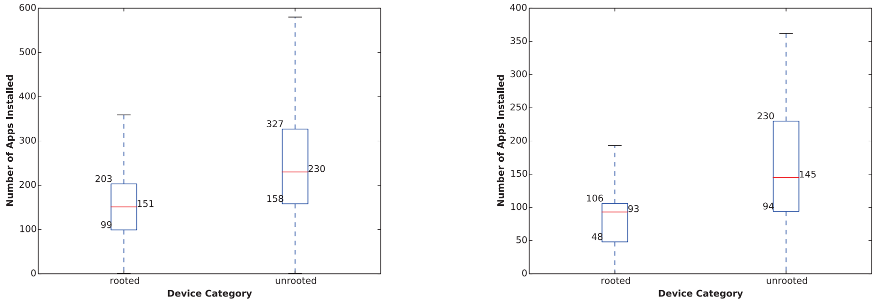
3.1.1 Application Installation

For this experiment, we set out to determine if the number of apps installed on mobile phones is correlated with whether or not the phone is rooted. Our hypothesis was that phones that were rooted would have more apps installed on average than phones that were not rooted. There are a few themes that led us to this hypothesis: more technical users root their phones and also install more apps, some apps are only available to rooted devices (or in third party app stores), and finally developers may root their devices to test apps.

To test this hypothesis, we first began by analyzing the average number of apps installed on devices in our sample set. The results of this analysis are given in Figure 1a. From this figure, we observe that our initial hypothesis about rooted users installing more apps on average is not supported (and further confirmed by Welch's t-Test with p-value=0.00001). In fact, non-rooted devices have on average almost twice as many apps installed.

To further investigate this finding, we limited the plot to include only "system" apps[1] in Figure 1b, which reveals the likely source of the discrepancy in the increased number of apps for non-rooted devices. Clearly, we see that system apps account for a large portion of those installed both on rooted and non-rooted devices, but that the average number present on non-rooted devices is higher. We believe the mostly likely explanation for this observation is that non-rooted devices are locked into whatever base system is provided by their carrier or manufacturer. These pre-built system images commonly contain vendor specific applications for setup, custom user interfaces, communication tools and suites to differentiate them from competitors. For example a T-Mobile Samsung device likely includes the Samsung

[1] To put it simply, any app not found in an app store and shared by greater than 0.25% of devices is labeled a "system" app. Those shared by less than 0.25% of devices (and not present in mobile app stores) are considered "unknown".



(a) (b)

Figure 1: Boxplots comparing counts of (a) *all* and (b) *system* apps between rooted and non-rooted devices.

Health suite of applications, T-Mobile account apps, LookOut security, etc. Conversely, custom Android distributions targeted at rooted devices (CyanogenMod, Paranoid Android, etc.) generally eschew the inclusion of unnecessary apps ("bloatware") for performance and size reasons (some even reduce the footprint by removing stock android apps).

Lastly, we created a plot of the average number of apps installed after excluding system applications (Figure 2a). This reveals that the disparity in the average number of apps is insignificant. Our conclusion is that our original hypothesis is unsupported by the data, and that rooted users and non-rooted users in our data set tend to install roughly the same number of apps on their phones.

Based on the total number of applications installed, it seems that rooted devices on average have fewer installed apps, both system and non-system that non-rooted devices. This seems to validate one common assumption about *why* users choose to root their phones: to remove preinstalled software that they do not need and/or optimize their system by keeping the total number of applications to a minimum. However, we can further investigate application installation behavior.

As the final part of our investigation into mobile device installation history, we looked into the average number of application updates between rooted devices and non-rooted devices, shown in Figure 2b. Somewhat surprisingly, here we see perhaps the greatest difference thus far between rooted and non-rooted applications. On average non-rooted devices have a greater number of application updates than rooted devices. Our intuition tells us there are a couple of reasons behind this observation. First, the larger base number of applications installed on average on non-rooted devices means there are simply more things to be updated in the first place. Second, rooted phones are more likely to have custom ROMs that don't support or enable auto-updates by default, and their users may be more likely to disable automatic updating for more control over their devices.

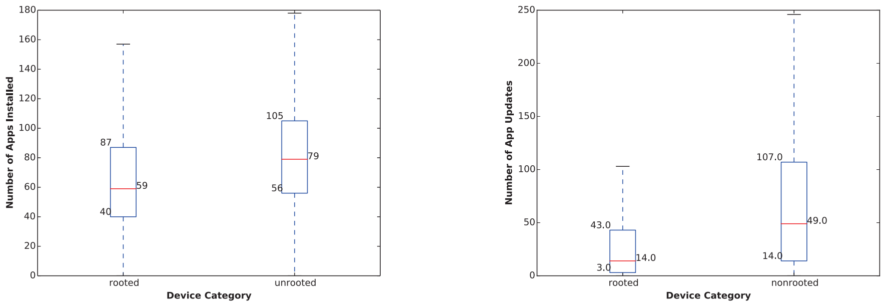
In summary, we have found that rooted devices do differ from non-rooted devices in terms of applications, but not necessarily in the way we original anticipated. Contrary to our initial hypothesis, non-rooted devices tend to have more applications installed than rooted, even when excluding (to the best of our ability) system and pre-installed applications. The second difference we observed was in the average number of application updates, again with a greater average number for non-rooted than rooted. For our extended analysis (Section 4), these two observations prove useful in classifying rooted/non-rooted devices automatically.

3.1.2 Network Behavior

Another interesting part of our data set is related to networking information from devices. We have data covering per connection information (IP address/port per connection and per app) as well as metadata per app (e.g., total bytes sent/received via wifi and/or mobile). By analyzing this data we hope to determine if there is any noticeable difference which sets rooted devices apart. If users root their devices to enable low level networking and system administration applications, we would expect to see an increase in the amount of data usage.Alternatively, if rooted devices are indeed more exposed to malware we would expect a greater number of outgoing connections to suspicious IP addresses and non-standard ports. These and other possiblities are reason to believe network metadata may be a key differentiator between rooted and non-rooted devices.

Table 1 gives data usage statistics for devices. We provide quartile boundaries for the average data usage per device. As expected for asymmetric communication, more data is received than sent for both categories of device. Furthermore, non-rooted devices on average send and receive more than rooted devices, possibly due to the higher number of applications installed on non-rooted devices. Median values for sending and receiving data (across mobile and WiFi) are approximately twice as large for non-rooted devices; however, when narrowing it down to mobile networks, non-rooted devices show nearly three times the data usage. This may indicate that users of rooted devices are more cognizant of the difference between mobile data (which typically is more expensive) and WiFi data usage. It could also be the result of a hidden difference in the population of which we are unaware.

We also examined metadata related to network connections at IP address/port level. Specifically, we calculated averages of the count of total connections, connections per application and port usage across devices. These results are summarized in Table 2. As with data usage, the total num-



(a) (b)

Figure 2: Boxplot comparing (a) counts of non-system apps and (b) counts of updates between rooted and non-rooted devices.

Data	Category	Median	Q3 - Q1
Avg. Bytes	rooted	28524.38	79948.33
Sent	non-rooted	58952.26	184628.19
Avg. Bytes	rooted	281179.41	635089.73
Received	non-rooted	491289.50	1220367.15
Avg. Bytes	rooted	4531.12	16203.95
Sent (mobile)	non-rooted	**12782.55**	41728.94
Avg. Bytes	rooted	25157.27	104442.37
Recv'd (mobile)	non-rooted	**73891.74**	259626.62

Table 1: Comparison of data usage between rooted and non-rooted devices. Across the board, non-rooted devices send and receive more data, especially on mobile networks, than rooted devices.

ber of connections for non-rooted devices are roughly twice that of rooted devices. The average number of connections per application are nearly the same, indicating commonality of applications across device categories. The count of distinct ports stands out slightly, in that the median number of non-standard ports[2] used by rooted devices is slightly higher than for rooted devices. Since more connections are used on average by non-rooted devices, we would have expected the number of distinct ports to be higher for them as well. This supports our speculation that rooted users are more "tech-savvy", either connecting to non-standard ports for development and testing, or using less common system utility apps. We did not find a specific trend towards certain non-stadard ports, which is not surprising, since they are by definition "non-standard", and typically chosen randomly.

3.2 Malware Susceptibility

One of the primary concerns around rooted devices is that they pose a riskier environment for users. It is certainly true that rooted devices could give malware access to more sensitive information and full system control. We wanted to see whether this increased level of system access attracts attackers specifically to rooted devices, or if users who root their devices also happen to be more likely to install malware, given their greater access to mobile application stores and side-loading application methods.

Data	Category	Median	Q3 - Q1
Avg. Total	rooted	251.00	884.00
IP Conns	non-rooted	427.00	1330.00
Avg. Distinct	rooted	3.00	2.00
Std. Ports	non-rooted	3.00	1.00
Avg. Distinct	rooted	3.00	5.00
Non-Std. Ports	non-rooted	2.00	3.00
Avg. Conns	rooted	20.00	25.00
Per App	non-rooted	23.00	29.00
Avg. Conns	rooted	206.00	749.00
Std. Port	non-rooted	361.00	1145.00
Avg. Conns	rooted	32.00	117.00
Non-Std. Port	non-rooted	53.00	161.00

Table 2: Comparison of IP address and port metadata between rooted and non-rooted devices. non-rooted devices have higher average connections, but rooted devices use a slightly wider range of ports.

3.2.1 Malware

For our analysis, we utilized a "mobile ping" database comprised of reported malware detections collected from April, 2015. This data is based on a set of signatures downloaded by end-user devices. Whenever a signature match is found on a device, the backend database is "pinged" with details about the malware detected. Malware is classified into five broad categories based on application behavior: trojan, infostealer, backdoor, hacktool and spyware. Due to technical limitations, we were only able to access one month's worth of mobile ping submissions, so we not only analyzed the devices from our 2015 sample, but also *all* devices seen in the mobile ping submissions for a single month.

Table 3 shows the results of our analysis of the mobile ping submissions. The first column gives the malware type, followed by category, followed by the percentage of devices in our sample in the mobile ping data, followed by the percentage of total devices seen in the mobile ping data. As shown, the percentages in each category are roughly the same between rooted and non-rooted devices, with a *slight* increase in each category for non-rooted devices. Comparing between the devices seen in our 2015 sample, and the population of all devices, again the percentages are on par for each malware type and category. This leads us to two conclusions based on our data: first, rooted devices do not in general have more

[2] non-standard ports refer to those not maintained in the canonical list at [2]

Malware Type	Category	Sample Pct.	Pop. Pct.
Trojan	rooted	0.57%	0.50%
	non-rooted	0.68%	0.71%
InfoStealer	rooted	0.12%	0.10%
	non-rooted	0.13%	0.14%
Backdoor	rooted	0.33%	0.30%
	non-rooted	0.42%	0.43%
HackTool	rooted	0.16%	0.14%
	non-rooted	0.20%	0.21%
Spyware	rooted	0.31%	0.27%
	non-rooted	0.35%	0.37%

Table 3: Percentages of malware present on devices from our 2015 sample as well as across all known devices from 2015.

malware than non-rooted devices and two, insofar as malware installation is concerned our sample is representative of the larger population.

As another attempt at determining if rooted devices are more vulnerable or show more dangerous behavior, we compared the IP addresses visited by each device against a list of known-bad IP addresses. These IP addresses are those known to be associated with malware, for reasons such as distributing exploits, participating in botnets or being the source of a large number of spam emails. As with the data from installed malware, we see virtually no difference between rooted and non-rooted devices. For rooted devices, 730 distinct devices were identified as connecting to a blacklisted IP address during our sample duration, or a total of 0.80%. For non-rooted devices, we saw 782 connections, for a total of 0.78%. Most importantly we do not see a significant difference between the two classes of devices. If rooted devices were more susceptible to malware we would expect to see a greater percentage of rooted devices connecting to known bad IP addresses.

3.2.2 System Updates

Another possibly risky behavior for mobile devices is not keeping them up to date [1]. As shown in Section 3.1.1, rooted devices tend to have far fewer updates to applications in general than non-rooted devices. Furthermore, as part of the data collected, we see the Android API versions in use, which correlate to specific Android releases that we can map to critical vulnerabilities [1]. Using an outdated version of Android with known vulnerabilities is risky behavior, and updating to the latest version is often the safest course of action. Older devices often do not get the latest patches, because mobile carriers have to roll out customized OS updates for each phone. We first analyzed our device data to see how many devices actually updated their API version (and thus their Android OS). Alarmingly, only 646 rooted devices and 11290 non-rooted devices in our sample of 100,000 ever changed API version (0.64% and 10.3%, respectively). While the percentage of devices that were upgraded is low for both, the much lower number of rooted devices updating indicates they may actually be more at risk than their non-rooted counterparts. Next we looked into which OS versions are actually installed on devices, summarizing the data as a count of the latest API version seen on each device, provided in Figure 3.

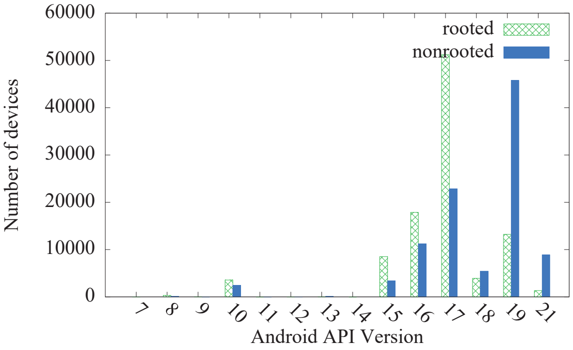


Figure 3: Total count of latest API versions seen on devices.

This figure further shows that rooted devices generally have older Android releases installed, with more than half running API version 17 (which corresponds to Android 4.2). Non-rooted devices are slightly better, with the majority running API version 19 (Android 4.4). While exploitable vulnerabilities exist in many Android versions [1], running an older version increases risk because the longer a vulnerability is known the more likely an exploit has been developed/released.

4. A MODEL FOR ROOTED DEVICE IDENTIFICATION

Up to this point, we have provided analysis of specific high-level differentiators between rooted and non-rooted devices. However, we would like to analyze *all* the data we have in aggregate to determine if there are other differences in the data that are not obvious (or not intuitive). We would also like to see if the data taken in whole gives a clearer picture than only looking at specific parts. For this we turn to ensemble machine learning [9], a method of classification that allows us to take all of the data we have available (a total of 73 so-called "features") and train a model for determining whether a device falls into the rooted category or the non-rooted category.

4.1 Mobile Behavior Features

We define 4 types of features: application profiles, installation behavior, device behavior, and communication behavior, to characterize a mobile device.

Application profiles. These features describe the app profile of each device. We count the number of apps installed under each app category[3] per device. We devise an *Unknown* category to encompass non-system (see Section 3) apps that have no categorical information associated in the market places (or do not exist in any known market places). We also created a special category of *suapps* which are those that perform device rooting or privilege escalation to root. We also utilize the total number of apps installed per device. Lastly, we count the number by type of *known malware* installed per device.

[3] Our categorization conforms to Google Play store application types at https://support.google.com/googleplay/android-developer/answer/113475?rd=1

App installation/update/uninstallation behavior. These features represent how often an app is installed, updated or uninstalled. We count the number of times that a given device *installs*, *updates* and *uninstalls* apps during the observation period. We also calculate the *average gap* (time delta) between two consecutive app installations/uninstallations.

Device behavior. These features capture various aspects of device usage information.

i) Power consumption. We calculate the *total power consumption* of a given device during the observation period. We also take the number of apps installed in that device into account and calculate the *average power consumption* per app.

ii) Status information. We calculate the *average device up time* and *average battery level* of a given device.

iii) App healthiness. We calculate the *total number of app crashes*, and *the total number of distinct apps that crashed* during the observation period.

Communication behavior. We calculate the *total network usage* (in bytes) sent and received via WiFi + cellular, and celluar alone. We also count the *total number of unique IPs visited* and *average count of IPs visited per app* within the observation period. In addition to IP address features, we use the *total number of visits to* and *distinct number of* both standard ports and non-standard ports so as to model various network traffic types. Finally, we use the *total visits to* blacklisted IP addresses per device.

4.2 Root Detection Model

The problem of identifying a rooted device is an instance of binary classification. For a given device, the data point $\overrightarrow{x} \in \mathbb{R}^d$ represents its feature vector in a high dimensional space with d features as detailed in Section 4.1. We label the data point with an accompanying class label $y \in \{+1, -1\}$. $y = -1$ represents a non-rooted device, and $y = +1$ represents a rooted device. A prediction model is used to predict the class label $\hat{y}$ of a previously unseen example $\overrightarrow{x_u}$. If the predicted label $\hat{y} = +1$ but the actual label $y = -1$, then the result is a false negative. If $\hat{y} = -1$ but $y = +1$, it is a false positive.

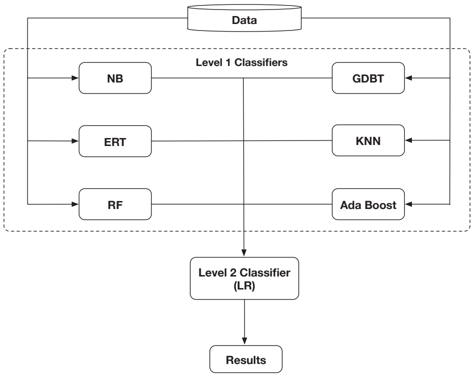


Figure 4: Ensemble Root Detection Model.

In this paper, we opt to use supervised learning methods to do classification as we have a large number of labelled data points. However, due to subtle behavior differences between rooted and non-rooted devices shown in Section 3, we build a meta-learner $\mathcal{M}$ to boost detection accuracy. We employ 6 classifiers, Random Forest (RF) [17], Extremely Randomized Trees (ERT) [14], Naive Bayesian (NB), Ada Boost [12], KNN [4], and Gradient Boosting Decision Tree (GBDT) [13], to form the level 1 classifier. We use logistic regression (LR) to combine the results from each classifier to generate the prediction results. Figure 4 shows the overall structure of the proposed method.

Formally, we have a list of level 1 classifiers as $C_1, C_2, ..., C_k$ where each C_j is an instance of one of the aforementioned classification methods and level 2 logistic regression classifier as C_{LR}. The training data is defined as $\mathcal{T} = \{\overrightarrow{t_i}\}$, where $\overrightarrow{t_i} \in \mathbb{R}^d$, and the test data is defined as $\mathcal{D}$. The prediction algorithm is detailed in Algorithm 1.

Algorithm 1: Root Detection

```
input  : $C_1, ...C_k$, $C_{LR}$, $\mathcal{T}$, $\mathcal{D}$, $m$
1  Partiton $\mathcal{T}$ and $y$ into $m$ fractions, $\mathcal{T}_1, ..., \mathcal{T}_m$ and $y_1, ..., y_m$
2  Initialize $[L1_T]_{|T| \times k}$ and $[L1_D]_{|D| \times k}$
3  for $i \leftarrow 1$ to $m$ do
4      initialize $[P]_{|D| \times k}$
5      for $j \leftarrow 1$ to $k$ do
6          $C_j$.fit($\mathcal{T}/\mathcal{T}_i$, y/$y_i$)
7          $L1_T[i, j]$ = $C_j$.predict($\mathcal{T}_i$)
8          $P[:, j]$ = $C_j$.predict($D$)
9      $L1_D[:, i]$ = avg($P[:, 1...k]$)
10 $C_{LR}$.fit($L1_T$, $y$)
11 Res $\leftarrow$ $C_{LR}$.predict($L1_D$)
```

The classification algorithm works as follows. At line 1, it partitions both training data $\mathcal{T}$ and its associated labels y into m fractions. In our paper, m is empirically chosen to 10. At line 2, we initialise $L1_T$ and $L1_D$ for storing level 1 classification results of $\mathcal{T}$ and $\mathcal{D}$ respectively. Line 6 trains the C_j using $\mathcal{T}/\mathcal{T}_i$ (all $\mathcal{T}$ but $\mathcal{T}_i$), line 7 predicts the label $\hat{y}$ for $\mathcal{T}_i$ and stores the results in $L1_T$. Note that line 6 and 7 resemble k-fold validation. Our approach stores the predicted class probability and does not carry out validation. Line 8 predicts the label $\hat{y}$ of $\mathcal{D}$ and stores the results in P. After the execution of lines 5-8, we average the predicted probability P and store the results in $L1_D$ at line 9. Note that line 9 can be treated as an averaged verdict from each classifier regarding the probability of a device being rooted or non-rooted. Line 10 trains the level 2 logistic regression classifier C_{LR} using $L1_T$ and y, and line 11 predicts $\hat{y}$ of $\mathcal{D}$ represented by $L1_D$.

Classification Evaluation. In total we obtain 90,973 rooted devices and 99,439 non-rooted device's data for our classification model. We randomly select 70,000 devices in each category as the training data $\mathcal{T}$ and combine the rest as the test data $\mathcal{D}$ (50,412 in total). Since our model is a probabilistic classification, we use the following metrics to evaluate its performance.

The *Brier score* measures the mean squared difference between the predicted probability that a device is rooted or non-rooted to its actual status. The Brier score range is between 0 (best) and 1 (worst). The lower the Brier score is for a set of predictions, the better the predictions are calibrated. The formulation is show in Equation 1.

$$BS = \frac{1}{N}\sum_{i=1}^{N}(\hat{y} - y)^2 \qquad (1)$$

The Brier score of our method given $\mathcal{D}$ is **0.078**. It proves that our model is well calibrated.

AUC (Area under the ROC curve) score estimates the probability that a randomly selected positive (rooted device) is ranked before a randomly selected negative (non-rooted device). Its value is in between 0 and 1. The closer AUC is to 1, the better the classifier is able to discriminate between two alternative outcomes. Our method's AUC is **0.96** which proves that the ensemble model is good at differentiating between rooted and non-rooted devices. We omit the ROC curve due to limited space.

Error Rate, Precision, Recall and F1 Score. These classic metrics are employed to measure our model's suitability to perform binary classification, i.e. assigning binary label '-1' (non-rooted) or '+1' (rooted) to data points in $\mathcal{D}$ instead of probabilistic values. We choose 0.5 as the cut-off threshold, i.e. a device is 'rooted' if its probability is greater than 0.5, otherwise it is classified as non-rooted. The scores are shown in Table 4.

The results show that our method of binary classification works well for determining whether a device is rooted or not, with high rates of precision and recall and a low error rate. This provides further evidence that rooted devices in our dataset are sufficiently different from non-rooted devices (based solely on the features we described).

5. RELATED WORK

In our search of related work, we were unable to find any studies directly related to root classification or device hygiene based on non-app analysis. As such, we discuss here work covering techniques for minimizing the impact of rooting a device, common techniques for malware analysis and research into device behavior and creating data sets describing device characteristics.

Root Mitigation In [19] the authors present an enhanced root-management system for protecting rooted Android phones. It does so by providing by providing a fine-grained policy and more contextual information about applications that are requesting root privileges. Similarly in [18] the authors are concerned with applications that could escalate their privileges to root for malicious purposes. Their solution consists of enforcing the principle of least privilege along with a whitelist of programs authorized for root level access. In [11] the authors analyze the incentives of mobile malware. The paper suggests that smartphone users root their phones to install custom versions of the Android operating system, access other app stores, remove bloatware and finally do complete backups. However, these suppositions are not backed by any data.

Malware In [5] machine learning classifiers are used to detect malware. The paper evaluates multiple classifiers along with a data set of over a $1,000$ applications. However the data collected corresponds to emulated user input and not to real user behavior. In [7] the authors proposed a crowdsourcing mechanism for obtaining real traces of application behavior coupled with a back-end that analyzes it to identify malware. The scale of their experiment was limited to 20 clients running the Crowdroid application. In [16] the authors propose to analyze the power consumption of mobile devices to identify the presence of malware. While in theory the idea is sound, in practice the difference is marginal and would require very precise measurements that are unavailable on current devices.

Behavioral Analysis In [6] the authors study the mobile application usage behavior of over $4,100$ users. This study shows that news applications are most popular in the morning, games dominate at night and communication applications are used equally throughout the day. Similarly in [8] the authors investigate the link between automatically extracted behavioral characteristics derived from smartphone data and self-reported personality traits with the use of machine learning techniques. The features extracted are related to messaging and phone call duration. None of the features used overlap with the ones presented in this paper.

As presented above we can see that Android security has been studied intensively in the last few years from security enforcement for rooted phones to various approaches to characterize malware. However, to the best of our knowledge we believe this to be the first paper to propose an approach to determine root status solely from data analysis using a large scale data set.

6. DISCUSSION AND CONCLUSION

To the best knowledge of the authors, this paper represents the first quantitative analysis of mobile devices from the perspective of comparing rooted devices to non-rooted devices. We have attempted to map high level thoughts about the characteristics of users who root their devices to the low-level data at our disposal. To that end, we began by analyzing different aspects of our data set. We started by looking into application characteristics, where we discovered that rooted devices have fewer total apps installed, but the majority of the difference is made up by system apps. We also investigated app installation, removal and updates, finding that while users of rooted and non-rooted devices install and uninstall at about the same rate, rooted phones get far fewer updates, including OS level updates. This is probably the best supporting evidence that rooted devices are more vulnerable to attack than non-rooted, solely because they update less frequently. However, we did not find a higher prevalence of malware on rooted devices, nor did they connect to known malicious IP addresses more than non-rooted devices. So while they may be more vulnerable, it does not appear that attackers are able to leverage that into a higher rate of infection or malicious activity.

While much of the data we have does not stand out on its own as starkly different between rooted and non-rooted devices, by combining all of the data into a set of features we were able to train and test a model for device classification using ensemble machine learning. This method of classification yielded promising results, achieving roughly 90% ac-

		Precision		Recall		F1 Score	
Name	Error Rate	rooted	non-rooted	rooted	non-rooted	rooted	non-rooted
Ensemble Model	0.10	0.86	0.91	0.88	0.90	0.87	0.91

Table 4: Error Rate, Precision, Recall and F1 Score of Ensemble Root Detection Model.

curacy in device classification. This model is useful in and of itself, as it showcases the subtle differences in characteristics between devices, which cannot be easily gleaned from manual analysis.

Our overall conclusion after performing an extensive large scale analysis is that while rooted devices and non-rooted devices exhibit different characteristics (enough to be correctly distinguished with high probability), rooted devices are not apparently more dangerous to users and the ecosystem.

For future work, we plan to extend our machine learning model to indicate level of risk for mobile devices instead of simply classifying whether they are rooted or not. Since we have discovered that malware and risky behavior seem to be similarly distributed across devices, a risk profile across all devices seems a logical next step. We also plan to extend and refine our feature set to improve our detection results.

7. REFERENCES

[1] Android vulnerability statistics. http://www.cvedetails.com/product/19997/Google-Android.html.

[2] Service name and transport protocol port number registry. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[3] Smartphone os market share. http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[4] An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3):175–185, 1992.

[5] B. Amos, H. Turner, and J. White. Applying machine learning classifiers to dynamic android malware detection at scale. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1666–1671, July 2013.

[6] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. Falling asleep with angry birds, facebook and kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 47–56, New York, NY, USA, 2011. ACM.

[7] Iker Burguera, Urko Zurutuza, and Simin Nadjm-Tehrani. 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, pages 15–26. ACM, 2011.

[8] Gokul Chittaranjan, Jan Blom, and Daniel Gatica-Perez. Mining large-scale smartphone data for personality studies. *Personal Ubiquitous Comput.*, 17(3):433–450, March 2013.

[9] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15. Springer-Verlag, 2000.

[10] Nathan Evans, Azzedine Benameur, and Yun Shen. All your root checks are belong to us. In *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '15, New York, NY, USA, 2015. ACM.

[11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '11, pages 3–14. ACM, 2011.

[12] Yoav Freund and RobertE. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In Paul Vitányi, editor, *Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. 1995.

[13] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[14] Pierre Geurts. Extremely randomized trees. In *MACHINE LEARNING*, page 2006, 2003.

[15] Tsung-Hsuan Ho, Daniel Dean, Xiaohui Gu, and William Enck. Prec: Practical root exploit containment for android devices. In *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*, CODASPY '14, pages 187–198. ACM, 2014.

[16] Johannes Hoffmann, Stephan Neumann, and Thorsten Holz. Mobile malware detection based on energy fingerprints - a dead end? In *Research in Attacks, Intrusions, and Defenses*, volume 8145 of *Lecture Notes in Computer Science*, pages 348–368. Springer Berlin Heidelberg, 2013.

[17] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[18] Yeongung Park, ChoongHyun Lee, Chanhee Lee, JiHyeog Lim, Sangchul Han, Minkyu Park, and Seong-Je Cho. Rgbdroid: A novel response-based approach to android privilege escalation attacks. In *Presented as part of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, Berkeley, CA, 2012. USENIX.

[19] Yuru Shao, Xiapu Luo, and Chenxiong Qian. Rootguard: Protecting rooted android phones. *Computer*, 47(6):32–40, June 2014.