GoogleECOM

Khizar Naseer Butt, Ibtisaam Butt, Atif Aziz Malik
i15-0024, i15-0027, i15-0057

Abstract

Improvement of current state-of-the-art algorithm by introducing new heuristics and/or ensemble methods for identifying Personally Identifiable Information in data packets transmitted over the network. In addition to it, we aim to classify the transmission as intended to collect information about the user and necessary for the operation of application or intended to collect information about the user but of curious nature.

Finally, we plan to calculate the consumption of energy associated with the transmission of unintended and intended data packets that will assist in creating awareness and giving further insight to users about their data.

1 Problem Statement

Around 4.57 billion* Mobile Phones in the world frequently fail to provide users with satisfactory control over and visibility into how third-party applications use their private data. These applications have easy access to huge amounts of sensitive, personally identifiable information and there are several applications which transmit this information over the network and use them for their own endeavours and benefits. Ample amount of research has been done in past 7 to 8 years to gain control of fine-grained information being transmitted over the network and therefore, we aim to gain knowledge from recent findings and to move ahead in this area by enhancing and improving state-of-the-art algorithm for detecting PII leakage and calculating Energy Losses that users have to bear.

2 Introduction

The International Telecommunication Union (ITU), a United Nations body, predicts that 4 billion people will be online by 2020. Cisco, the networking company that supplies much of the world's internet infrastructure, says that the total amount of traffic will reach around 1,060 exabytes, or just over one zettabyte. This huge amount of data may be at risk of leakage and our ultimate goal is to empower people as we end this project in 2019, and optimizing the amounts of energy consumed by all these packets.

Mobile devices have access to personal, potentially sensitive data, and there is a growing number of applications that transmit this personally identifiable information (PII) over the network. In this project, we present the GoogleECOM system that performs on-device packet-level monitoring and detects the transmission of such sensitive information accurately and in real-time along with energy consumed for the transmission of such data packets. A key insight is to distinguish PII that is predefined and is easily available on the device from PII that is unknown a priori but can be automatically detected by classifiers. We demonstrate the real-time performance of our prototype as well as the classification performance using a dataset that we collected from an organization and also collected using different networking protocols.

Our intentions are to empower the users to take control of their sensitive data by interacting with them through an Application that will demonstrate the to-and-fro transfer of information from source to destination and vice-versa and it shall permit the operator to stop that leakage or transfer. Another major motivation is to assist community by prolonging the lifetime of their devices which will be possible by calculating the amount of Joules being consumed for data transfer and prompting the user so that he or she may save the battery drainage which will eventually prolong the lifetime of devices battery.

3 Literature Review

3.1 Using the Middle to Meddle with Mobile [1]

This paper was published in CCIS, Northeastern University, Tech. Rep., in December 2012 by Ashwin Rao, A Molavi Kakhki, Abbas Razaghpanah, Amy Tang, Shen Wang, Justine Sherry, Phillipa Gill, Arvind Krishnamurthy, Arnaud Legout, Alan Mislove and David Choffnes. All of them have plenty of work to their name when it comes to monitoring the traffic that a mobile phone generates when it is connected to a network.

3.1.1 Summary

The purpose of this study was to capture entire set of a mobile-devices traffic over the internet: permitting them to characterize network flows, to interrupt them using software middleboxes, and to facilitate research into new middlebox applications for mobile traffic. The final goal of Meddle was to enable all mobile Internet users to monitor and control their Internet traffic.

Meddle [1] was the first approach to provide visibility and control over network traffic for all access network and most access decides Operating Systems, when introduced in 2012. It remarkably outwitted the work previously done [5][6][7][8] which had a lot of limitations associated with it such as: compromising network coverage, portability and deployability. Meddle compromised none of these and enabled visibility and control of network traffic access carriers, devices and access technologies.

Meddle [1] is a framework that combines virtual private networks (VPNs) with middleboxes to provide an experimental platform that aligns the interests of users and researchers. Meddle relies on VPN tunnels to access the mobile traffic regardless of the device, OS, wireless technology, and carrier. Meddle can thus provide a continuous and comprehensive view of how mobile devices interact with the Internet. Once packets arrive at a Meddle server, they use a variety of middlebox approaches to interpose on mobile-device traffic. Meddle [1] offers new opportunities for measuring and characterizing mobile traffic, and designing new in-network features to improve the mobile experience. For example, by accessing network traffic regardless of the wireless technology we can analyze how different operating systems and apps offload their traffic from cellular networks to Wi-Fi. To improve the user experience and attract more users to install it, they implemented packet filters to block ads; unlike existing packet filters for mobile devices, the packet filters provided by Meddle do not require jail-breaking the mobile device. Just like Ant-Monitor or Ant-Shield, which are part of our study during this project, Meddle worked on the layer where granularity of information is very high along with high number of users (which means that the idea is applicable and practical as compared to ISP Traces and User Traces where either you can get high number of users or fine-grained information at an instance, but not both).

After successfully capturing a packet, they use SSL bumping to decrypt and access the plain text of encrypted flows. Meddles VPN server can be configured to use a self-generated root certificate used to sign all subsequent certificates issued to participating mobile devices and this allows them to perform SSL traffic decryption using Squid proxys SSL bumping feature [13].

For capturing Personally Identifiable Information, they have introduced ReCon [10] which uses Meddle [1] on the top to capture and decrypt the packets. ReCon [10] has been separately reviewed by us during the literature Review. ReCon [10] used WEKA data mining tool to train their classifier. They utilized bag-of-words to separate features from their dataset. To reduce their number of features, ReCon [10] set a specific threshold of frequency for each word to be considered as a feature since when PII occurs it rarely occurs just once.

3.1.2 Critical Analysis

There are few limitations associated with Meddle [1], their approach of capturing packets fails for apps that do not trust certificates signed by unknown root authorities. A better approach (SandroProxy library to intercept secure connections and to decrypt) has been used by Ant-Monitor (Reviewed separately).

3.1.3 Relationship with our Work

Their work is related to ours when it comes to intercepting the traffic over the network of a mobile phone when it is connected. In the same way, we aim to capture data-packets on the device and to identify if its a PII or not by first decrypting it and then calculating energy dosage/consumption during the transmission.

3.2 ReCon: Revealing and Controlling PII Leaks in Mobile Network Systems [10]

This paper was published in MobiSys, in June 2016 by David Choffnes, Jingjing Ren, Ashwin Rao, Martina Lindorfer and Arnaud Legout. They have number of publications to their names in field of Privacy leak detection over the network. The purpose of this study was to capture the personal information leaks from smartphones over the internet and give user control on it.

3.2.1 Summary

ReCon [10] is a cross platform system that provides user with the view of their Personally Identified Information (PII) leaks and give command over it. User can block, substitute or permit PII. It works for Android, iOS and Windows phone. ReCon [10] used Machine Learning to identify PII leaks. They Performed a Controlled Experiment for collection of their dataset. In the Experiment they tested 100 Apps of Android, iOS and windows phone. They used each application for 5 minutes. ReCon used man in the middle approach to capture data packets. Recon achieved following high level goals:

- Identify PII in network flow
- Improve awareness by presenting this information to user
- Improve Classifier Accuracy using user feedback
- Enable user to modify block or change PII leaks

ReCon used WEKA data mining tool to train their classifier. They utilized bag-of-words to separate features from their dataset. To reduce their number of features, ReCon set a specific threshold of frequency for each word to be considered as a feature. They build a per domain classifier for those domains which send huge amount of data over the network. Machine learning classifier distinguishes information to be PII or not but rather does not show which content out of the data packet is PII. To resolve this issue, ReCon have a PII type (IMEI, andriodid e.t.c) and key assigned to it. ReCon estimates likelihood of each key to be PII or not by dividing the number of times the key shows up in PII flows with the number of times key show up in overall network flows. ReCon chose to utilize Decision Trees as classifier since DTs maintain great balance among accuracy and time it takes to train them. To additionally enhance their classifier ReCon took feedback from users whether PII was recognized accurately or not and retrain their classifier as needs be.

3.2.2 Critical Analysis

ReCon gathered extensive and clean dataset of 100 Applications form each OS (Android, iOS and Windows) which makes their classifier efficient. Their methodology of using Decision Trees gives slightly less accuracy than Blending Decision Trees and K Nearest Neighbors yet it is 7.24 times more affordable in time complexity. On the other hand a better methodology can be used, as ReCon utilize classifier each time it needs to identify PII, it makes process computational costly. They can have predefined strings of PII which can be utilized to distinguish PII by basic string matching as in case of AntMoniter. In the event that it fails they can use the Machine learning Model. Their methodology of identifying PII does not recognize whether PII was intended or not because of which any application which needs to send PII for instance, MAPS uses location (which it is supposed to send), will be identified as PII leak. There can be several approaches to identify whether PII was intended or unintended. One of them is to train a model on descriptions of application and then it can be predicted whether the leaked PII is needed for the working of the particular application or not. Second and a rather simpler approach requires us to check whether the users last interaction with that application is up to the particular threshold or not to mark the PII leak as intended or unintended.

3.2.3 Relationship with our Work

ReCon's study is closely related with our work in identification of PII leaks. We likewise need to identify PII in the first position to additionally give client capacity to control them and a visualization of energy being consumed. ReCon is utilizing machine learning classifier DTs to distinguish PII leaks we likewise need to build any ensemble method to recognize those PII leaks and further arrange them as expected or unintended PII flows.

3.3 AntMonitor A System for On-Device Mobile Network Monitoring and its Application [3]

This paper was published in arXiv preprint arXiv:1611.04268 in November 2012 by Anastasia Shuba, Anh Le, Emmanouil Alimpertis, Minas Gjoka, Athina Markopoulou. All of them have plenty of work to their name when it comes to monitoring the traffic that a mobile phone generates when it is connected to a network.

3.3.1 Summary

The key emphasis of this paper was on AntMontior system for on-device passive mobile network monitoring. They have introduced a system for collecting large-scale, yet fine-grained network measurements from mobile devices, and for detecting and preventing leakage of private information in real time.

Major goals of their research were:

1. Privacy Leaks Detection and Prevention Module (PII detection)
2. Monitoring Network Performance
3. Learning Network Behavior

Their team thoroughly studied the work already done in the same area by (Haystack [2], Privacy Guard [9], Meddle [10]) and worked very well to produce better and efficient result in the following ways:

1. Their application works without root-access or administrator privileges
2. It runs smoothly as a service app in the background
3. It scales well with the increasing number of users
4. It provides fine-grained control data (high-granularity of information being intercepted)
5. It supports real-time analysis on the device
6. Their application is energy-friendly, along with the best throughput. (It achieves throughput of over 90 Mbps downlink and 65 Mbps uplink, which is 2x and 8x faster than mobile-only baselines and is 94% of the throughput without VPN, while using 212x less energy.)

For capturing the traffic (packets), they have used mobile-only VPN approach in contrast to client-server VPN, custom OS or Rooted Phone methodologies which have several disadvantages associated with them. After capturing the encrypted packets, they have used TLS proxy that uses SandroProxy library[11] to intercept secure connections, to decrypt the packets and then re-encrypt them before sending to their intended hosts. For the identification of personally identifiable information (PII), they have a list of strings, in which users can also add any potential PII (such as Mobile Number, Citizenship Number etc.) which further helps them in Deep Packet Inspection (It is an advanced method of examining and managing network traffic. It is a form of packet filtering that locates, identifies, classifies, reroutes or blocks packets with specific data or code payloads that conventional packet filtering, which examines only packet headers, cannot detect). This list of strings is then used to check if the packet being sent is a PII or not by string matching algorithms in real-time. It also notifies the user and intakes his/her action to allow/block/hash the information being transmitted. As they had all the packets being transmitted, they used them to measure performance of the network to correlate network-level metrics with other information available on device with added advantage of fine-grained measurements at device or destination level.

3.3.2 Critical Analysis

The only problem with AntMontior is that they used string matching only to identify the packet as PII or not which is not as accurate as if we use Machine Learning techniques. Otherwise, the approach of Mobile-Only VPN to gain access to the data packets gives them the best result as far as the performance of the algorithm and application is concerned.

3.3.3 Relationship with our Work

AntMonitors research is closely linked with what we plan to work on in the coming academic year. In the same but an enhanced way, we aim to capture data-packets on the device (without sending it to separate logger/server) and to identify if its a PII or not. However, there is a problem with their identifier (string matching) which can be easily dodged by a refined attacker; therefore, we need to deploy machine learning techniques by building an ensemble method to recognize and classify PII leaks. Another concern that has been ignored by AntMonitor is distinguishing if the PII being leaked is intended or not (Location needed by Google Maps is an intended PII leak). Therefore, in order to detect the intended or unintended transmission, another classifier will be built. To conclude, their work will surely assist us in several ways during our research.

3.4 AntShield On-Device Detection of Personal Information Exposure[4]

This paper was published in arXiv:1803.01261v1 [cs.NI] on 3 March 2018 by Anastasia Shuba, Evita Bakopoulou, Milad Asgari Mehrabadi, Hieu Le, David Choffnes, and Athina Markopoulou. Every one of them have a lot of work to their name with regards to checking the activity that a cell phone produces when it is associated with a network.

3.4.1 Summary

Modern mobile phone devices have provided us with opportunities and facilities we could never think of. But with goodness comes a price. Our mobile phone devices have ingress to potentially sensitive, personal user data. With the boom of world wide web internet, a huge number of mobile applications transmit huge amount of data over the network. This data can be subject to as personally identifiable information which this particular paper has referred to as PII. AntShield- On-Device Detection of Personal Information Exposure, integrates the usefulness of on-device packet monitoring along with the recognition of identifying unknown PII. This particular system is the best state of the art in terms of system performance and accuracy, even out performing Recon [10].

Their research contributions as follows:

- A complete system to detect PII exposure via remote Virtual Proxy Network server in real time (1 ms). They have used AntMonitor [3] library and its resource plug-ins to build this system which assimilate the draw backs that are developed due to other techniques such as customized OS or rooting a phone.
- The PIIs are identified by passing a network packet through 2 protocols (classifiers/ models). To identify a packet containing a potential PII, they use a string-matching algorithm heuristic to classify the packet. Also, their second classifier, multi-label classification model uses binary relevance with decision tress to identify unknown PIIs. The modular system of their application allows them to scale this feature if a better method is introduced.
- Their team built upon the dataset provided by Recon [10]. This richer than previously available dataset constitutes of information over UDP and TCP leaks as well along with HTTP leaks. They collected and analyzed this dataset with manual and automatic testing.

The core functionality of their system is PII exposure detection. This is done by maintaining calls to AntMonitor Library as depicted in the diagram below. Each intercepted packet sent over the network is analyzed with predefined leaks and then passed to the classifier for feature extraction. Their application gives the user the ability to choose whether to block the packet or allowed to continue to its remote destination.

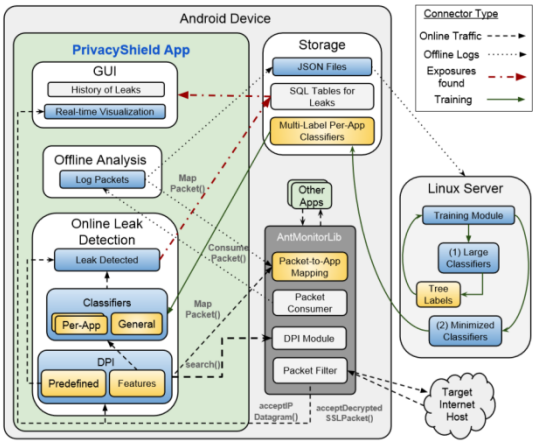


Figure 1: System Architecture of AntShield

They also worked in giving interested and motivated users the ability to run the module offline after logging consumer packet over a period of time. Obviously, this would be computationally expensive but this can help the user reveal the ground truth.

The AntShield application comes with pre-trained classifiers and does not log any data unless the user wishes to. Therefore, this application is space efficient when it comes to storage on device/or server. Along with it their application uses 100MB RAM which is acceptable as compared to many popular applications e.g, Facebook uses 200 MB of RAM.

3.4.2 Critical Analysis

AntShield packet capture mechanism has several advantages compared to other state of the art mechanisms: (a) They were able to map packet to its source and destination accurately; (b) They picked up understanding into TLS, UDP, and general TCP trafc, along with HTTP.

	ReCon Public dataset(s)		**Ant-Shield dataset(s)**	
	Auto	**Manual**	**Auto**	**Manual**
# of Apps	564	91	414	149
# of packets	16761	13079	21887	25189
# of destination domains	450	368	597	379
# of leaks detected	1566	1755	4760	3819
# of *unknown* leaks	4	78	483	516
# of leaks in encrypted traffic	-	-	1513	1526
# of packets with **multiple leaks**	50	224	1506	790
# of background leaks	-	-	2289	639
# of HTTP packets	16761	13079	13694	13648
# of HTTPS packets	-	-	6830	8103
# of TCP packets	-	-	867	2264
# of leaks in TCP (other ports)	-	-	38	7
# of UDP packets	-	-	496	1174
# of leaks in UDP	-	-	17	12

Figure 2: Description of data-set

Their evaluation matrix consists of binary classification: F-measure, specificity and recall.

		(1) Recon on All PII	(2) Recon on *unknown*	(6) Complete AntShield
Per-Domain Avg	accuracy	**89.1% ± 22.1**	91.8% ± 17.7	**99.5% ± 3.99**
	precision	**90.0% ± 21.5**	91.8% ± 17.7	**99.5% ± 3.99**
	recall	**89.2% ± 22.0**	91.8% ± 17.7	**99.8% ± 1.60**

Figure 3: Evaluation results

3.4.3 Relationship with our Work

AntShield is the latest advancement in the research area we wish to pursue. They are nearest to our work, in this manner and we will utilize it as our standard for correlation all through our work. The key difference would lie in the classification methodology and the eventually the energy report we wish to achieve. Their technique creates new routes for distributed learning of individual data leakage which presents one of a kind framework challenges and learning opportunities.

3.5 Distilling the Knowledge in a Neural Network [17]

This paper was published in arXiv:1503.02531 on 9 Mar 2015 by Geoffrey Hinton, Oriol Vinyals, Jeff Dean.

3.5.1 Summary

Complex models generalize the data much better than small models but they are difficult to deploy because of their computational complexity. It is possible to compress the knowledge in a complex (very deep network or ensemble of different networks) model to a small neural network. During distillation, they use soft targets (softmax results of complex model) as ground truth to train a smaller network. Using soft targets instead of hard targets provides much better information of relative probabilities among different classes. For example, in case of image classification it gives you information that one type of fish is more closely related to another type of fish than different types of dogs.

They use modified version of softmax when training a model a shallow network) for distillation. It is given as:

$$S = \frac{e^{zi/T}}{\sum e^{zj/T}}$$

Where zi is the resultant logit of neuron i and T is temperature constant. Increasing the temperature increases the affect of classes having very low probabilities by producing reasonably soft targets. Fig (a) shows the result of different values of T on probabilities given by softmax.

After computing softmax, they use modified cross entropy loss by using pi to be equal to the target that complex model produces on that specific data point.

$$L = \sum pi * log(qi)$$

Then they take derivative of this function and put it equal to zero. It turns out that they are trying to make qi = pi. Training ensemble of neural networks might be very costly so one thing that we can do is to train specialist models on subsets of data and then use distillation loss to transfer their knowledge to the small network. This method also acts as a regularizer and reduces over-fitting on data.

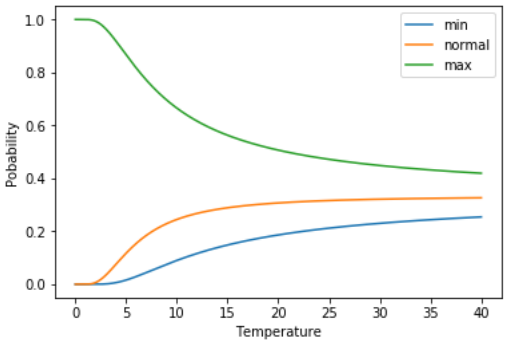


Figure 4: Effect of temperature on softmax scores

3.5.2 Critical Analysis

This method works very well for classification. On MNIST dataset small network of two layers with 800 RELUs on each gives 146 errors on test set but these errors go to 67 after distilling the knowledge from a deep network. Omitting the digit 3 from training set and distilling the knowledge from a deep network, with bias adjusted, gives 14 test set errors. This paper does not discuss the variant of loss that can be used for regression purpose. Also, this paper introduces 3 new hyper-parameters in the models (T, weight for cross entropy and distillation, and bias correction for missing training data).

3.5.3 Relationship with our Work

Most of state-of-the-art algorithms and application face the problem of having to settle between either accuracy or computational cost. In this regard, we aim to train our classifier be it Decision trees or Convolutional Neural Networks, and then convert it into a smaller neural network using distillation loss and deploy it in our final product. This would significantly help us in saving computational resources and at the same time maintaining, if not increasing, the accuracy/ output of the system.

4 Our Approach

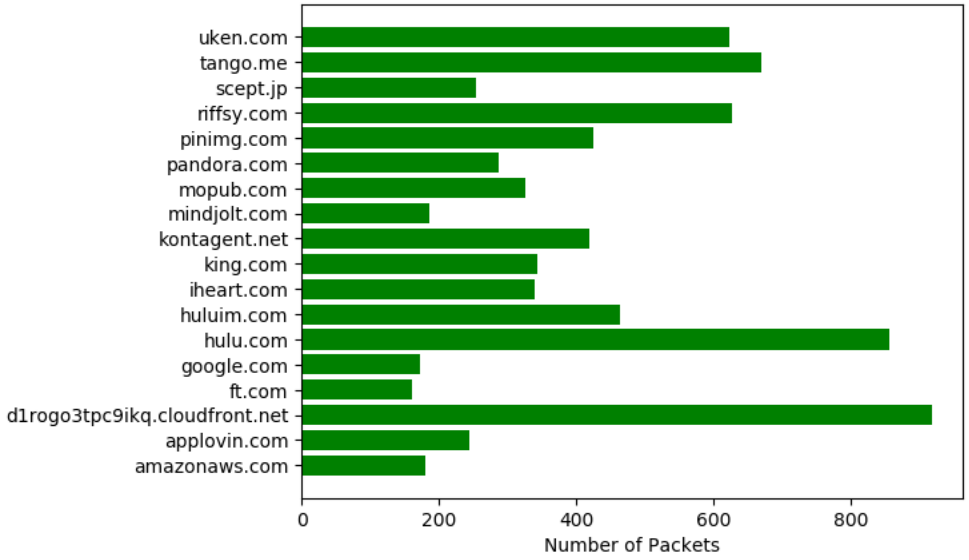
Our proposed approach is an on-device network monitoring paradigm which uses a Virtual Private Network setup on the device to route the packets from the application to their server domain. This precludes the necessity for trust agreement and gives full autonomy to the user.

4.1 Data Visualization

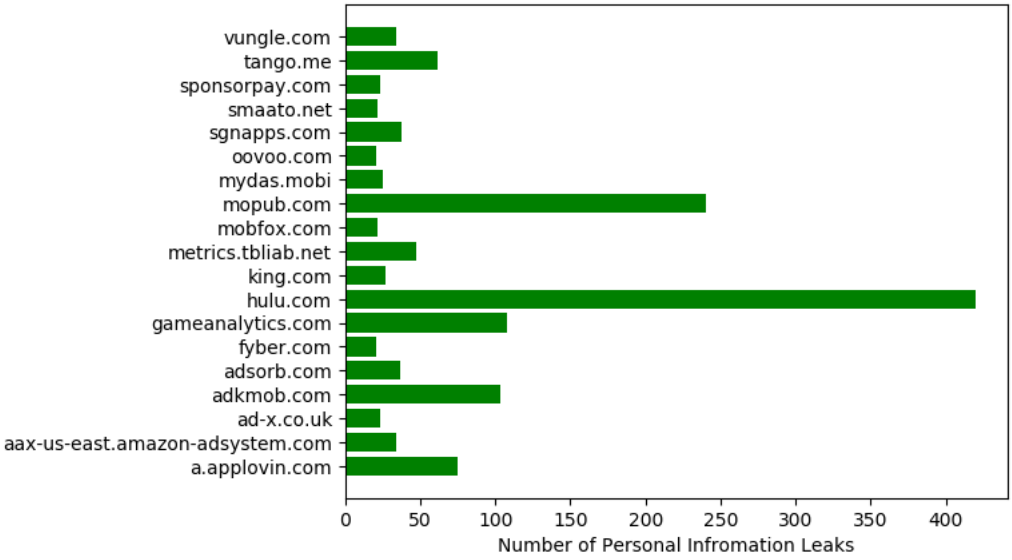
To evaluate GoogleECOM accuracy, we need app-generated traffic and a set of labels indicating which of the corresponding flows leak PII. The data-set from our controlled experiments are open-source and publicly available at http://recon.meddle.mobi/codeanddata.html.

The data-set consists of details of network packets generated from top 100 applications on Play Store, manually annotated along with monkey generated (scripted) packets.

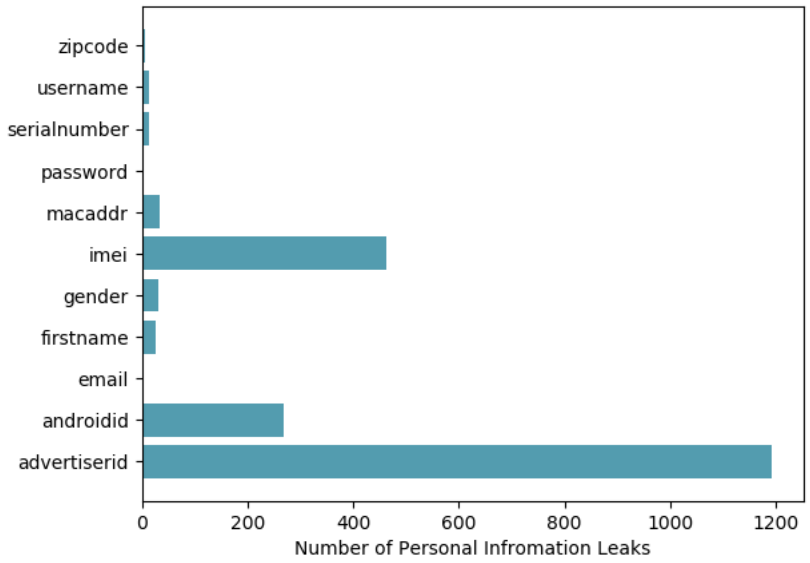
Following are some of the data visualization techniques we applied:



This graph represents the number of network packets that are sent over a particular domain.



This graph represents the number of Personal Information Leaks that are sent over a particular domain.



This graph represents the number of Personal Information Leaks per network packets.

4.2 Feature Extraction

The accuracy of the classifiers described above largely depends on correctly identifying the subset of features for training. Further, the training time for classifiers increases significantly as the number of features increases, meaning that an efficient classifier requires culling of unimportant features. A key challenge in GoogleECOM is determining how to select such features given the large potential set derived from the bag-of-words and one hot encoding approach.

Picking the right number of features is also important for classifier accuracy, as too many features may lead to over-fitting and too few features may lead to an incomplete model. We use an in-built library i.e., select k best features, for picking the right number of features that have high contribution in the determining the original output label. A total of **13079 features** were obtained as a result of one hot encoding approach.

4.3 Extra-trees Classifier

The Extra-Trees algorithm builds an ensemble of unpruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree based ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees.

This class implements a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the data-set and uses averaging to improve the predictive accuracy and control over-fitting. As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminating thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule. This usually allows to reduce the variance of the model a bit more, at the expense of a slightly greater increase in bias.

Table 1 Extra-Trees splitting algorithm (for numerical attributes)

Split_a_node(S)
Input: the local learning subset S corresponding to the node we want to split
Output: a split $[a < a_c]$ or nothing
– If **Stop_split**(S) is TRUE then return nothing.
– Otherwise select K attributes $\{a_1, \ldots, a_K\}$ among all non constant (in S) candidate attributes;
– Draw K splits $\{s_1, \ldots, s_K\}$, where $s_i = \textbf{Pick\_a\_random\_split}(S, a_i), \forall i = 1, \ldots, K$;
– Return a split s_* such that Score$(s_*, S) = \max_{i=1,\ldots,K}$ Score(s_i, S).

Pick_a_random_split(S,a)
Inputs: a subset S and an attribute a
Output: a split
– Let $a^S_{\max}$ and $a^S_{\min}$ denote the maximal and minimal value of a in S;
– Draw a random cut-point a_c uniformly in $[a^S_{\min}, a^S_{\max}]$;
– Return the split $[a < a_c]$.

Stop_split(S)
Input: a subset S
Output: a boolean
– If $|S| < n_{\min}$, then return TRUE;
– If all attributes are constant in S, then return TRUE;
– If the output is constant in S, then return TRUE;
– Otherwise, return FALSE.

5 Evaluation Methodology

This section evaluates the validity of GoogleECOM in terms of accuracy and F-measure. The reason of using F-measure is because the data-set is highly skewed due to the presence of huge number of non-PII(s). F-measure is a combined metric of precision and recall.

First we compare the accuracy, F-measure and time taken to train our model as compared to other classification models and measure the outcome.

	Naive Bayes	String Matching	ADA Boosted Classifier	Gradient Boosted Decsion Tress	Extra Trees
Accuracy	0.748704	0.959018	0.983792	0.982569	0.987768
F1-measure	0.485643	0.824607	0.935601	0.932704	0.952381
Time taken to train	0.068815	2.005605	28.164657	273.710726	3.556483
Precision	0.310784	0.733170	0.894627	0.883665	0.919833
Recall	0.870246	0.717949	0.881007	0.903890	0.915332

Summary of Extra trees performance versus rest of the classifiers

We also calculated confusion matrices on the basis of outcome of all the classification models. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

	Non-PII	PII
Non-PII	2066	766
PII	58	389

(a) Naive Bayes

	Non-PII	PII
Non-PII	11283	41
PII	495	1260

(b) String Matching

	Non-PII	PII
Non-PII	2830	3
PII	37	400

(c) Extra Trees

Figure 5: Confusion matrices

6 Analysis and Results

Extra Trees perform well than the state of the art algorithm in terms of accuracy and time to train the model.

References

[1] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes. Recon: Revealing and controlling pii leaks in mobile network trafc. In *In In ACM MobiSys), volume 16, 2016.*