Technical Disclosure Commons

Defensive Publications Series

November 07, 2018

Automatic obfuscation of sensitive content on locked devices

Victor Cărbune

Pedro Gonnet

Automatic obfuscation of sensitive content on locked devices

ABSTRACT

The content of notifications from the device operating system and applications is displayed on the screen of a device even when the device is locked. Such notifications make private or sensitive information accessible by other parties without unlocking the device. To counter this risk, device operating systems allow users to hide notification details when the device is locked. However, such hiding results in significant loss of understandability and utility of notifications. Per techniques of this disclosure, a trained machine learning based model is applied with specific user permission to identify sensitive content by analyzing pixels of the device lock screen image that includes notifications. The techniques may be incorporated within the device operating system or web application frameworks. If the inferred level of sensitivity is higher than a threshold value, the corresponding content is obfuscated when displaying the notification on a locked device.

KEYWORDS

- Notifications
- Lock screen
- Notification obfuscation
- Content hiding
- Sensitive content
- User privacy
- Operating system
- Transaction Authentication Numbers (TANs)

BACKGROUND

Users typically lock their devices, such as smartphones, tablets, etc. to prevent unauthorized access by other parties. Although most device functions and content cannot be accessed when the device is locked, the content of notifications from the device operating system and applications is displayed on the screen even in the locked state. Such notifications may contain private or sensitive information, such as financial information, authorization codes used as the second factor of login, bank Transaction Authentication Numbers (TANs), sensitive or private images, etc. Displaying such information on the lock screen creates a privacy and security risk since the content may be accessed by other parties without unlocking the device. To counter this risk, typical device Operating Systems (OSes) include the ability for the user to specify that notification details be hidden when the device is locked. However, completely hiding the details of the notification contents results in significant loss of understandability and utility of the notification.

DESCRIPTION

This disclosure uses a trained machine learning based model to analyze the pixels of the device lock screen or relevant parts of the lock screen. The pixels and/or text content within the display region extracted (e.g., via Optical Character Recognition (OCR) techniques) is provided as input to the model. The output of the model identifies sensitive information within the input along with a numeric score that indicates the inferred level of sensitivity of the content. If the inferred level of sensitivity is higher than a threshold value, the corresponding content is obfuscated within the notification that is displayed on screen when the device is locked.

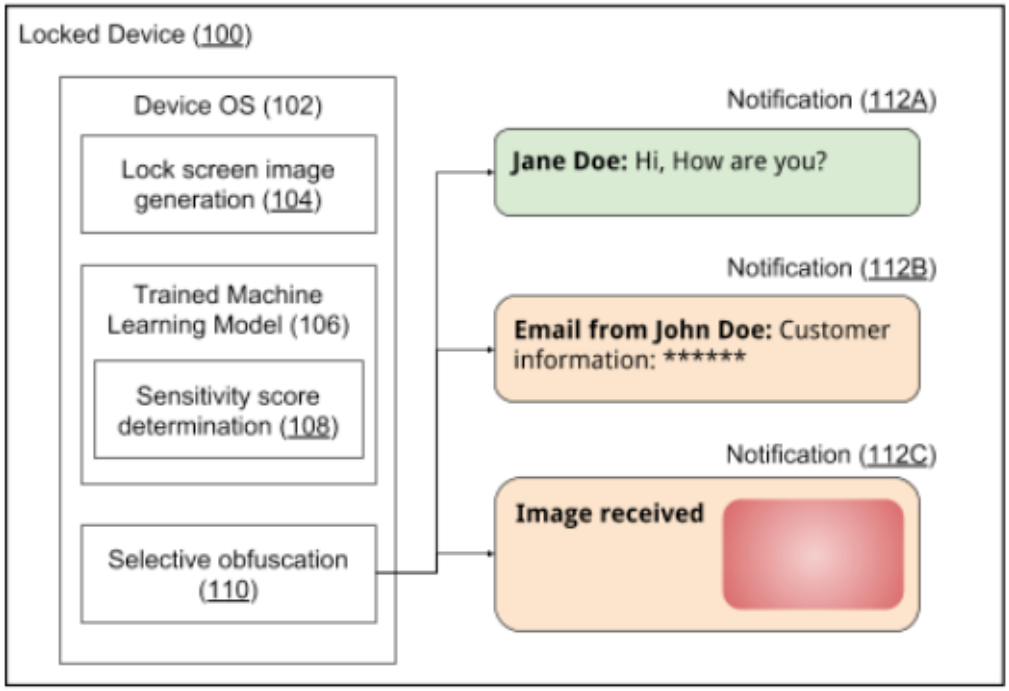

Fig. 1: Detecting and obfuscating sensitive notification content

Fig. 1 shows an implementation of the techniques of this disclosure. The device (100) is in the locked state when the device operating system (102) generates a lock screen image update that includes notifications (104), e.g., based on operating system events, based on a notification generation request received from an application, etc. The generated lock screen image is provided to trained machine learning model (106) prior to display.

The trained model determines a sensitivity score (108) for the lock screen image. The sensitivity score is evaluated to determine whether it meets a threshold, and accordingly, selective obfuscation (110) is performed. The obfuscation can include replacing pixels of the generated image that correspond to the text or image that was deemed sensitive.

In the example illustrated in Fig. 1, three different notifications (112A, 112B, and 112C) are shown. Notification 102A is a notification regarding a text message with a greeting from a user Jane Doe, notification 102B is a notification regarding an email from a user John Doe

regarding a confidential corporate matter, and notification 102C is a notification with an instant message that includes a private image.

The trained machine learning model determines that pixels corresponding to notification 112A do not include sensitive content, and as a result, the notification is displayed as generated. Notifications 112B and 112C are determined as including potentially sensitive content, and are therefore displayed by obfuscating the sensitive text or image content. For instance, the email notification displays the sender and the subject while hiding the message content deemed sensitive (illustrated using "*****" in Fig. 1) and the sensitive image is obfuscated (illustrated in red color in Fig. 1). This enables the user to view the visible non-sensitive information to determine whether the notification may warrant action, while restricting access by third parties to the sensitive content.

The model can be trained via a supervised learning approach using image patches known to contain sensitive information with corresponding labels marking the sensitive parts within the image patch. The labeled sensitive parts may be sequences of words in the text contained within the image patch. The labeled training data can be obtained from users that provide such data for training the model, and can be synthetically generated via simulation. For instance, such synthetic data can mimic typical messages that send codes, TANs, sensitive corporate information, confidentiality markers, etc.

In addition to the initial training, when users permit, the model can be further trained based on user input that labels certain notifications as sensitive. For example, if a user indicates that a notification not classified as sensitive by the model includes a TAN, such user input is incorporated to update the learned parameters of the model such that the next receipt of a TAN is properly detected and obfuscated. Moreover, a user may specify sensitivity preferences to

indicate whether obfuscation should be applied based on various criteria based on sender identity, application that generated the notification, time, location, etc. In addition to enforcing the preferences, the learned parameters of the model can be adjusted based on learning from the specified user preferences. The additional model training based on user input and preference specification are implemented completely within the user device.

The trained model can be applied to analyze the pixels of the entire screen whenever the device lock screen is updated with new content. Alternatively, the operation can be limited only to specific lock screen patches of interest where notifications are displayed. Such patches may be determined and extracted by using the screen rendering logic or an appropriate screen segmentation heuristic based on the operating system layout infrastructure. Further, with user permission, text content within each of the screen image patches may be extracted via OCR. The image and the text content within the patch may be utilized to determine the potentially sensitive image and text content within the patch along with a corresponding sensitivity score for the content. For instance, such a score may be a numeric score that indicates sensitivity on a scale of 1 to 5, with 1 being the least sensitive and 5 being the most sensitive. The sensitivity score is an indication of the likelihood that the corresponding content is sensitive.

The sensitivity score is compared with a threshold level of sensitivity. If the score exceeds the threshold value, the content is deemed too sensitive to display and is appropriate for obfuscation. The threshold level may be provided as a part of the model and/or can be set by application developers, learned from training data, or specified by the user. Multiple threshold values can be employed to support several sensitivity intervals for more granular operation. For instance, the extent of obfuscation can range from full to partial to none depending on whether the sensitivity interval is high, medium, or low, respectively.

By operating on raw pixels, the proposed techniques can detect sensitive content received as images as well as (rendered) text. Moreover, the detection of sensitive content can take into account the context surrounding the content, thus potentially lowering the number of false positives. The model architecture may be a combination of a Convolution Neural Network (CNN) for image processing and sequence based models, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU). Other complex models may also be equally suited to implement the described operation. The training procedure may utilize standard backpropagation techniques and may be augmented by further on-device training for operational fine tuning based on user specification and feedback. Moreover, the techniques may be incorporated within the device operating system or web application frameworks.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

To avoid displaying sensitive information to unwanted parties, device operating systems of smartphones and other devices allow users to hide notification details when the device is locked. However, such hiding results in significant loss of understandability and utility of notifications. Per techniques of this disclosure, a trained machine learning based model is applied with specific user permission to identify sensitive content by analyzing pixels of the device lock screen image that includes notifications. By operating on raw pixels, the proposed techniques can detect sensitive content received as images as well as text. Such detection of sensitive content takes into account the context surrounding the content. The techniques may be incorporated within the device operating system or web application frameworks. If the inferred level of sensitivity is higher than a threshold value, the corresponding content is obfuscated when displaying the notification on a locked device.