July 2021

# Utilizing Web Data to Detect Harmful Application Software

Badthody Anil Shenoy

# Utilizing Web Data to Detect Harmful Application Software

**Abstract:**

This publication describes techniques that enable a developer of an operating system (OS) to protect a user(s) and/or an electronic device(s) (e.g., a smartphone) by detecting, disabling, and/or removing a third-party application(s) that may be a potentially harmful application (PHA). The PHA may cause signature symptoms on the smartphone of the user. The signature symptoms may be related to a performance of the smartphone (e.g., the smartphone crashes), billing fraud, unwanted applications, a battery of the smartphone, a temperature of the smartphone, and so forth. Using a search engine, the user may submit a search query regarding the signature symptoms the user may be experiencing with their smartphone. In an anonymized way, the developer of the OS may utilize nowcasting of web data to detect, disable, and/or remove the PHA from the smartphone and/or the application marketplace.

**Keywords:**

Application marketplace, application store, potentially harmful application, PHA, malware, operating system, search engine, search query, nowcasting, forecast, estimate, prediction, log data, web data, search session data, installation log data

## Background:

An operating system (OS) of an electronic device (e.g., smartphone) supports numerous third-party application software (applications) that a user may download through an application marketplace. A developer of the OS and/or an operator of an application marketplace (collectively

"the OS developer") may require developers of the third-party applications to follow various policies, terms, and conditions of the application marketplace, laws, regulations, ethical norms, and/or technical best practices intended to protect the user, the smartphone, and/or society at large (collectively referred herein as "policies"). Unfortunately, some third-party applications may intentionally and/or unintentionally fail to follow the application marketplace policies of the OS developer.

For example, a third-party application may be a "copycat" or an impersonation application of another third-party application or a first-party application. In such a case, a developer of the third-party application may intentionally mislead users by using an icon, description, title, and so forth similar to the other third-party application or the first-party application. In another example, a third-party application may contain and/or facilitate publishing of content in violation of the application marketplace policies. In yet another example, a third-party application may be a potentially harmful application (PHA). The PHA may be difficult to detect and may contain a virus, trojan, malware, and so forth intended to harm the user and/or the smartphone. Assume the third-party application appears to be an application valued by the user (e.g., a free video game). The video game, however, may contain malware that may utilize phishing to trick a user into revealing sensitive information, inappropriately exfiltrate data off the smartphone, conduct fraud (e.g., ransomware, billing fraud), use a messaging system to generate multiple unsolicited messages ("spam"), damage the smartphone (e.g., corrupt the OS), inappropriately use resources of the smartphone (e.g., processors), and/or conduct other harmful activities.

**Description:**

This publication describes techniques that enable an OS developer to protect a user and/or an electronic device (e.g., a smartphone) by detecting, disabling, and/or removing third-party applications that do not follow application marketplace policies (the "policies"). Consequently, the OS developer strives to detect, disable, and/or remove third-party applications that violate the policies, for example, applications that are "copycat" applications, applications that contain and/or facilitate publishing of inappropriate and/or illegal content, and/or applications that may be a PHA. Due to the complexities involved in the detection of PHAs, this publication focuses on detecting, disabling, and/or removing PHAs. Nevertheless, the OS developer can use the techniques described herein to detect any third-party application that does not follow the policies.

In one aspect, the OS developer tests and/or evaluates third-party applications before and after the third-party applications enter the application marketplace and determines whether the third-party applications are PHAs. In addition, the OS may be configured to enable the user to report any issues they may have with a third-party application to the OS developer. The OS developer then may further test and/or evaluate the third-party applications reported by the user(s). Therefore, the OS developer proactively, actively, and reactively protects the user and their smartphone from PHAs.

Due to the complexity of detecting the PHAs, the user may be unable to determine that a third-party application is a PHA. Consequently, the user may not report issues with the third-party application. The user, however, may notice symptoms of the PHA and, using a search engine, may submit a query regarding the symptom, as is further described below. To decrease a reaction time of protecting some users and proactively and actively protect other users, the OS developer may use "nowcasting." Generally (e.g., in economics, meteorology), the term "nowcasting" includes a

prediction of a present, a near future, and a recent past of an indicator (e.g., gross domestic product (GDP)). Similarly, the OS developer may utilize nowcasting to predict a present, a near future, and a recent past of the symptoms caused by the PHAs by analyzing web data of multiple users in an anonymized way, for example, by analyzing one or more of search session data or application installation log data. The search session data includes queries of the multiple users utilizing a search engine supported by the OS.

To protect the user and their smartphone from the PHAs and, at the same time, protect a privacy of the user, the OS developer provides the user with controls allowing the user to make an election as to both if and when the OS, first-party application, and/or third-party application can collect user information (e.g., information about applications used by the user, a user's social network, social actions, social activities, profession, a user's preferences, a user's current location), and if the user is sent content and/or communications from a server. In addition, the web data may be treated in one or more ways before it is stored and/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. In another example, 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.

Figure 1 illustrates a user submitting a query in the search engine regarding an example issue that may arise due to the user unknowingly installing and/or executing (using) a PHA.

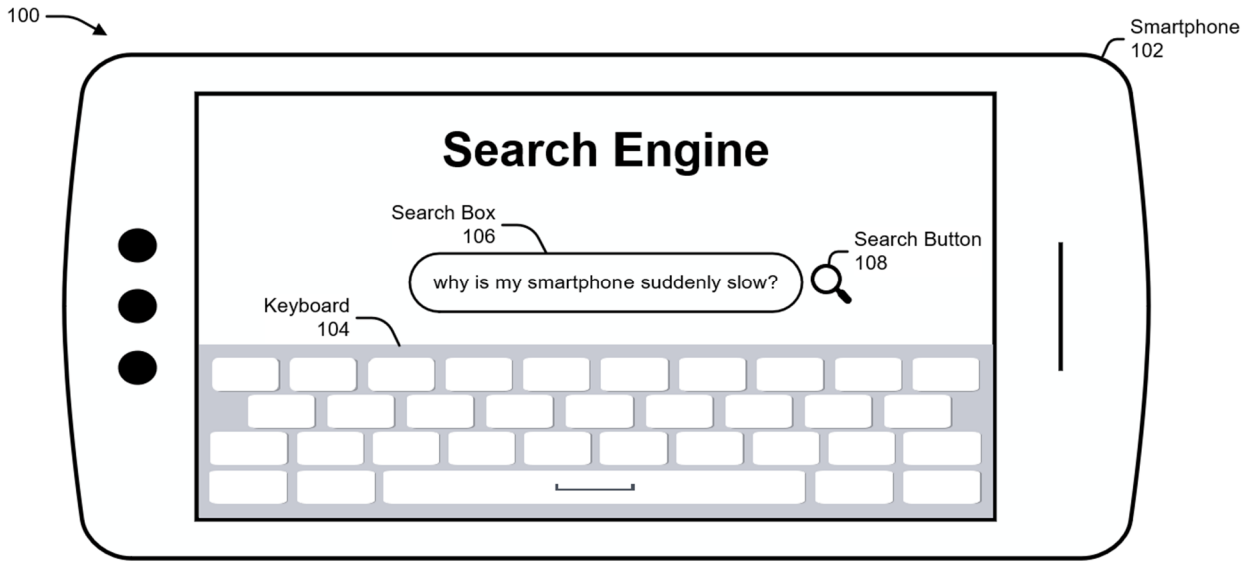


**Figure 1**

Figure 1 illustrates an example environment 100 of a user having a smartphone 102. Assume the user notices that the smartphone 102 suddenly responds slowly to commands (e.g., slow processing times) due to a PHA, which the user unknowingly may have installed (e.g., downloaded from the application marketplace) on a previous date. Initially, the user may try rebooting (e.g., turn off, turn on) the smartphone 102 to determine whether the smartphone 102 can resume responding faster to the commands. Suppose the user determines that the smartphone 102 persistently responds slowly to the commands. In that case, using a keyboard 104, the user may type into a search box 106, for example, "why is my smartphone suddenly slow?" The user then may tap a search button 108 to read search results of their query compiled by the search engine of the smartphone 102 with the described OS. Alternatively, the user may submit the query using a voice-activation and/or voice-assistant feature of the smartphone 102.

The example environment 100 illustrates that sometimes the user may not understand a root cause of a problem with the smartphone 102. Still, the user may observe a symptom(s) of the

root cause of the problem due to the installation of the PHA. Although different symptoms of the smartphone 102 may be indicative of various issues, the PHAs often trigger signature symptoms.

Some signature symptoms may be performance-related, for example, slow loading times, freezing of first-party and/or third-party applications, longer reboot times, and/or other performance-related symptoms. For performance-related symptoms of the smartphone 102, in addition to the query illustrated in Figure 1, the user may also query, "why is the internet of my smartphone slow all of a sudden?" "Why is my smartphone 'model X' slow all of a sudden?" "Why is it taking so long to launch 'application Y?'" "Why does 'application Z' keep crashing?" "Why does my smartphone take so long to reboot?" For clarity and consistency, this description includes queries in the form of a question. The OS developer, however, may analyze search session data (web data) of queries that are not questions, complete sentences, or contain all the symptoms related to the PHA.

Some signature symptoms may be billing-related, where the user may receive unexpected charges from their cellphone carrier or other service provider. The cellphone carrier may be adhering to an agreed contract with the user (e.g., a cellphone plan). Still, the cellphone carrier may charge the user for calls, text messages, data, and so forth that the user did not intentionally use. In another example of signature symptoms, the smartphone 102 may receive unsolicited text messages and send text messages that the user did not initiate. In yet a further example, the PHA may use a cellphone data plan that the user purchased through their cellphone carrier to generate fraudulent advertisement revenue. In this case, the user may also notice an increase in pop-up advertisements while using the smartphone 102.

Some signature symptoms may be related to installations of third-party applications without knowledge or permission from the user. The installation of these unwanted third-party

applications may also contain other PHAs. The other PHAs, for example, may exfiltrate user information data for nefarious purposes.

Some signature issues may be related to a battery and/or a temperature of the smartphone 102, where the battery drains faster than usual and/or the temperature of the smartphone 102 increases. In this case, the PHA may use resources (e.g., processors) of the smartphone 102, for example, to mine for a cryptocurrency. For battery-related symptoms, the user may query, "why is the battery of my smartphone lasting only a few hours?" For temperature-related symptoms, the user may query, "why is my smartphone heating?"

In addition to the above-mentioned symptoms, the PHAs may also cause additional symptoms on the smartphone 102. Further, the various symptoms may be unrelated to each other and/or interrelated with each other. In a case of interrelated symptoms due to a particular PHA, a first group of users may notice a first symptom, a second group of users may notice a second symptom, a third group of users may notice a third symptom, and so forth. Consequently, the first, second, and third groups of users may submit queries related to the first, second, and third symptoms, respectively. Regardless, the OS developer may determine that the first, second, and third symptoms are due to the particular PHA.

Figures 2A, 2B, and 2C illustrate how the OS developer utilizes the web data in an anonymized way to detect the PHAs and protect the user and their smartphone.

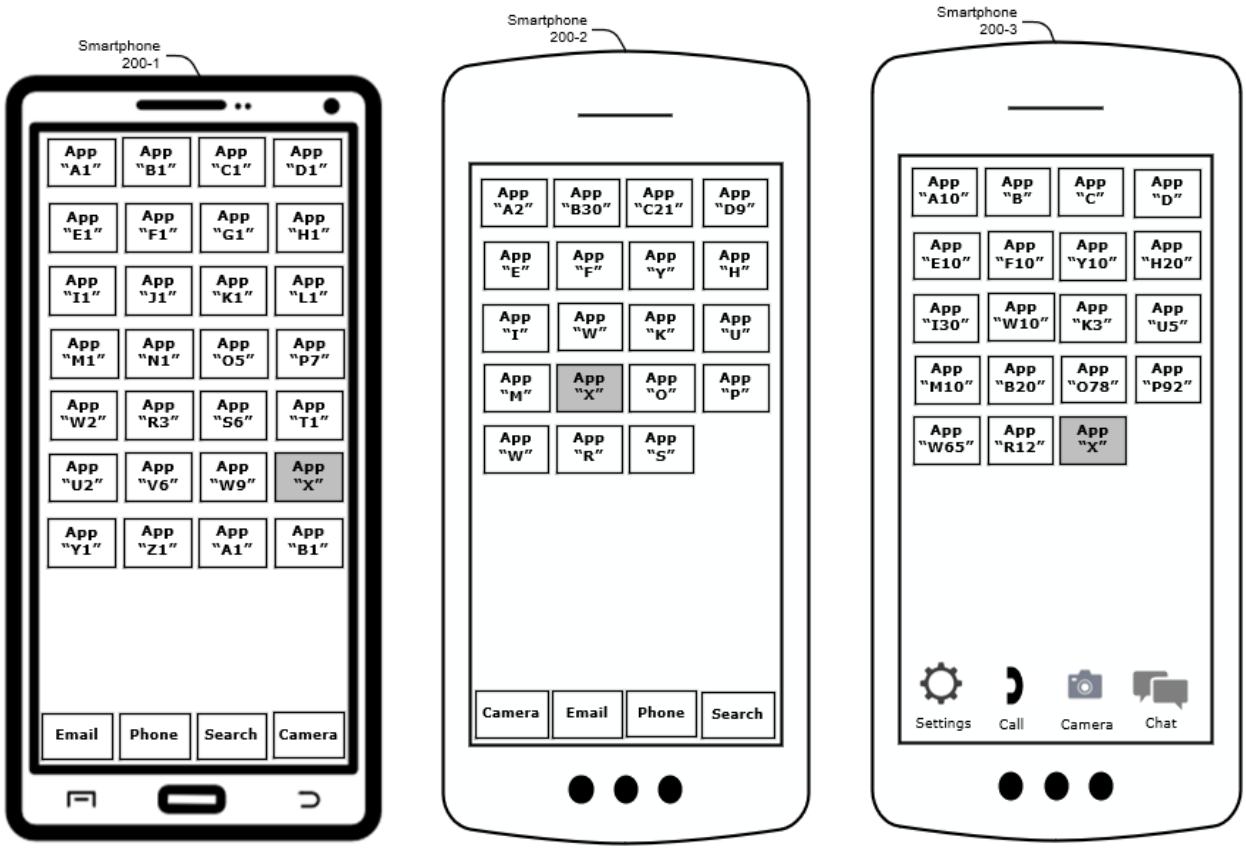


**Figure 2A** **Figure 2B** **Figure 2C**

Figures 2A, 2B, and 2C illustrate example smartphones 200-1, 200-2, and 200-3, respectively. The OS developer may utilize web data of a first user with the smartphone 200-1, a second user with the smartphone 200-2, and a third user with the smartphone 200-3. Suppose the first, second, and third users submit queries on their respective smartphones regarding a similar symptom. The OS developer may compare third-party applications installed in the smartphones 200-1, 200-2, and 200-3 and narrow down that the first, second, and third users have unknowingly installed a same PHA, for example, Application "X" (App "X"). As is illustrated in Figures 2A, 2B, and 2C, the App "X" is the only common third-party application installed on smartphones 200-1, 200-2, and 200-3.

Further, the smartphones 200-1, 200-2, and 200-3 may be different models of smartphones and/or may utilize different versions of the OS. Thus, the techniques described herein are not

limited to a particular model of a smartphone and/or a particular version of an OS, in part because the users may use a same application marketplace regardless of the model of the smartphone and/or the version of the OS. For example, a user with an older smartphone model may receive a same protection from the OS developer compared to a user with the latest smartphone model. In addition, the user may submit a query in the search engine regarding a symptom the user may have with their smartphone by logging in and using a different device (e.g., a laptop), regardless of whether the user unknowingly installed the PHA on the laptop, the smartphone, and/or any other electronic device. In a case where the user logs in on a different electronic device using a same profile (e.g., a same username and password), the OS developer may infer that the user may be observing symptoms of a PHA installed on the smartphone instead of the laptop (and vice versa) by comparing the third-party applications installed on the smartphone to the third-party applications installed on the laptop.

Figure 3 illustrates a technique to proactively, actively, and reactively detect, disable, and/or remove PHAs.

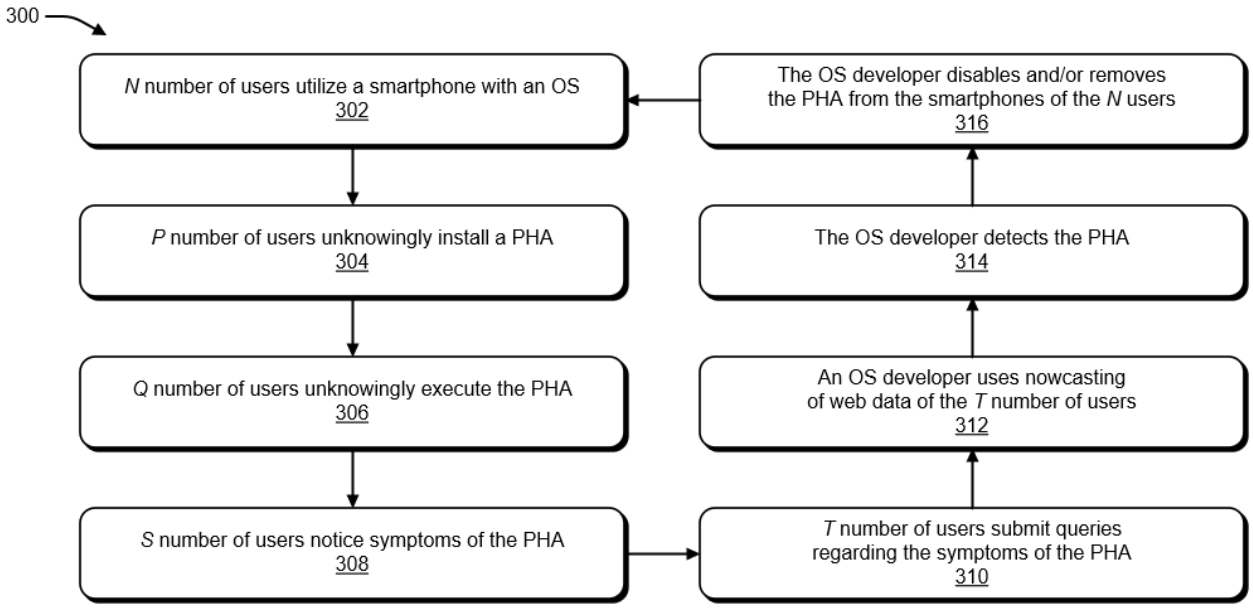


**Figure 3**

Figure 3 depicts a block diagram illustrating a technique 300 implemented on a smartphone. The OS developer can use the nowcasting of web data (e.g., search session data, application installation log data) to detect, disable, and/or remove a PHA from the smartphone and/or an application marketplace. In Figure 3, $N$, $P$, $Q$, $S$, and $T$ are integers greater than or equal to one (1), where $P$ is less than or equal to $N$, $Q$ is less than or equal to $P$, $S$ is less than or equal to $Q$, and $T$ is less than or equal to $S$. In mathematics, the relation between the integers $N$, $P$, $Q$, $S$, and $T$ may be expressed as $1 \leq T \leq S \leq Q \leq P \leq N$.

Suppose at stage 302, $N$ number of users utilize an electronic device (e.g., a smartphone). Unfortunately, even though the OS developer proactively analyzes and/or evaluates a third-party application before allowing the third-party application in the application marketplace, at stage 304, $P$ number of users unknowingly may install a PHA from the application marketplace. For example, a nefarious developer may use a software package of the third-party application that may appear harmless. Still, the third-party application may execute the PHA on a third-party server. Therefore, initially, the OS developer may be unable to detect the PHA in the application marketplace.

Then, at stage 306, $Q$ number of users unknowingly execute the PHA, for example, by using their respective smartphones and/or the third-party application (e.g., the PHA). Consequently, at stage 308, $S$ number of users notice symptoms (e.g., pop-up advertisements) of the PHA. Fortunately, at stage 310, $T$ number of users submit queries in the search engine of Figure 1 regarding the symptoms of the PHA. The queries of the $T$ number of users enable the OS developer, at stage 312, to utilize the nowcasting of web data of the $T$ number of users. Continuing with the example of Figure 2, the OS developer may determine that the $T$ number of users submitted a similar query shortly after installing and/or executing the App "X." In that case, the

OS developer may further evaluate the App "X." After further evaluation, at stage 314, the OS developer detects the PHA. Finally, at stage 316, the OS developer disables and/or removes the PHA from the smartphones of the *N* users (all users of the OS) and/or the application marketplace. The OS developer may also take further action, for example, barring the nefarious developer of the PHA from using the application marketplace.

**References:**

[1] Patent Publication: US20120240236A1. Crawling multiple markets and correlating. Priority Date: October 21, 2008.

[2] Patent Publication: US20190174319A1. Detection and identification of potentially harmful applications based on detection and analysis of malware/spyware indicators. Priority Date: December 01, 2017.

[3] Yoones A. Sekhavat, "Nowcasting Mobile Games Ranking Using Web Search Query Data," International Journal of Computer Games Technology, vol. 2016, Article ID 9859813, 9 pages, 2016. https://doi.org/10.1155/2016/9859813.

[4] Patent Publication: US20150312271A1. Application Spam Detector. Priority Date: April 28, 2014.