DRD15-J. Consider privacy concerns when using Geolocation API

The Geolocation API, which is specified by W3C, enables web browsers to access geographical location information of a user's device.

In the specification, it is prohibited that user agents send location information to web sites without obtaining permission from the user:

> *4.1 Privacy considerations for implementers of the Geolocation API*
>
> *User agents must not send location information to Web sites without the express permission of the user. User agents must acquire permission through a user interface, unless they have prearranged trust relationships with users, as described below. The user interface must include the host component of the document's URI [URI]. Those permissions that are acquired through the user interface and that are preserved beyond the current browsing session (i.e. beyond the time when the browsing context [BROWSINGCONTEXT] is navigated to another URL) must be revocable and user agents must respect revoked permissions.*
>
> *Some user agents will have prearranged trust relationships that do not require such user interfaces. For example, while a Web browser will present a user interface when a Web site performs a geolocation request, a VOIP telephone may not present any user interface when using location information to perform an E911 function.*

A conforming implementation must acquire permission through a user interface before sending the user's geolocation to the web site.

An example Javascript for using Geolocation API is as follows:

```
<script>
navigator.geolocation.getCurrentPosition(
  function(position) {
         alert(position.coords.latitude);
         alert(position.coords.longitude);
  },
  function(){
  // error
});
</script>
```

The Javascript above will show the location of the device on a screen.

To enable geolocation in an application using the `WebView` class, the following permissions and the use of the `webkit` package is necessary:

- permissions
  - `android.permission.ACCESS_FINE_LOCATION`
  - `android.permission.ACCESS_COARSE_LOCATION`
  - `android.permission.INTERNET`
- `webkit` package
  - `WebSettings#setGeolocationEnabled(true)`
  - `WebChromeClient#onGeolocationPermissionsShowPrompt()` implementation

Among these, implementing the `WebChromeClient#onGeolocationPermissionsShowPrompt()` method needs security consideration. There are vulnerable apps and code examples that override this method so that a user's geolocation information is sent to servers without the user's consent. With such an implementation, the user's geolocation location data will leak just by visiting malicious sites.

Noncompliant Code Example

This noncompliant code example sends the user's geolocation information without obtaining the user's permission upon request from a server.

```
public void onGeolocationPermissionsShowPrompt(String origin, Callback callback){
        super.onGeolocationPermissionsShowPrompt(origin, callback);
        callback.invoke(origin, true, false);
}
```

Compliant Solution #1

This compliant solution shows a UI to ask for the user's consent. Depending on the user's response, the application can control the transmission of the geolocation data.

```
public void onGeolocationPermissionsShowPrompt(String origin, Callback callback) {
        super.onGeolocationPermissionsShowPrompt(origin, callback);
        // Ask for user's permission
        // When the user disallows, do not send the geolocation information
}
```

Compliant Solution #2

The following compliant solution is from a real world fix of a previously vulnerable application.

```
public void onGeolocationPermissionsShowPrompt(String origin, GeolocationPermissions$Callback callback) {
        super.onGeolocationPermissionsShowPrompt(origin, callback);
        if(MyPreferences.getBoolean("SECURITY_ENABLE_GEOLOCATION_INFORMATION", true)) {
            WebViewHolder.a(this.a).permissionShowPrompt(origin, callback);
        }
        else {
            callback.invoke(origin, false, false);
        }
}
```

If the user setting of geolocation is enabled, the code will show a screen to ask for the user's permission. If the setting is disabled, it will not transmit the geolocation data.

Risk Assessment

Sending a user's geolocation information without asking the user's permission violates the security and privacy considerations of the Geolocation API and leaks the user's sensitive information.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
DRD15-J	Low	Probable	Medium	P4	L3

Related Vulnerabilities

- JVN#81637882 Information disclosure vulnerability in Sleipnir Mobile for Android

Related Guidelines

Geolocation API by W3C	http://www.w3.org/TR/geolocation-API/

Automated Detection

It is trivial to automatically detect if an app requires the permissions needed for the vulnerability, if the app also uses the `WebView` class, and if the app also implements the `WebChromeClient#onGeolocationPermissionsShowPrompt()` method. Tracing taint flow of sensitive geolocation data between components of one or more Android apps, and eventual transit to a sink, is a complex dataflow analysis.

Bibliography

[Android API 2013]	class `WebChromeClient`, class `WebSettings`
[W3C 2013]	Geolocation API Specification