APPROACH TO PREVENT OF SPOKEN INFORMATION LEAKAGE THROUGH GYROSCOPE AND ACCELEROMETER SENSORS IN MOBILE DEVICES WITH ANDROID OPERATION SYSTEM

Olena Chebanyuk, Anton Yukhymenko,

Abstract: *Paper considers questions of the statistics analysis of the usage of mobile devices. The threat of spoken information leakage through gyroscope and accelerometer sensor in mobile device is considered. The approach for analyzing and evaluating threat level of audio information leakage from mobile device with installed android operation system is proposed.*

Keywords: *Mobile device, threat, Android OS, spoken information, speech, recognition, dangerous application, MEMS, gyroscope, accelerometer, sensor, methodology.*

DOI: *https://doi.org/10.54521/ijita28-03-p02*

Introduction

Nowadays mobile devices have a broad range of capabilities for communication and data processing. On the other hand software and wireless technologies increase attackers' opportunities.

The term mobile device describes great size of electronics, especially ones with wireless technologies, capable of collecting, storing, processing and transmitting data. It includes mobile phones, tablets, smart watches etc (The IT Law Wiki, 2021).

According to the gathered statistics (DataReportal, 2021), there are 5.27 billion of unique mobile phone users – almost 67% of world population. The number of internet users – 4.8 billion of people and 92.1% of them utilize mobile devices to use the internet.

Currently about 98% of the whole amount of mobile devices are powered either by Android or by IOS operation system (Poetker, B., 2021). in the materials of the research (Michalevsky, Y at. el., 2014) were presented practical results of retrieving spoken information from gyroscope signals. The possibility is grounded on the design features of the MEMS sensors used in mobile devices.

Sophisticated spectrum analysis techniques were applied to reconstruct signal from a data sampled with a rate of 200Hz which is far below the Nyquist rate for the speech – 6.8kHz. Then the machine learning techniques were applied to recognize data from the signal. The results show the success percentage for a speaker itself, speaker-dependent and speaker-independent isolated word recognition (Michalevsky, Y at. el., 2014).

Android OS implements a permission control that manages applications' access to a hardware components of mobile devices, such as a microphone, camera etc. The full list can be obtained from the Android API documentation (Android, 2021). Nonetheless, the permission control don't protect users. Furthermore, until Android OS 12 there is no any permission required for the usage of gyroscope data. Considering that Android 12 is currently installed on the less than 1% of mobile devices, there is a high risk of exploiting the threat.

Both MEMS gyroscope and accelerometer sensors are based on the same physical principles, susceptible to an acoustic influence and usually incorporated on a single chip. Despite that possibility of recognition was presented only for gyroscope sensor, in current paper we will consider them both a threat sources.

We conducted a survey to determine users' awareness about the threat. The most important results are:

- more than 75% of asked users use smartphone with Android OS installed, shown in figure 1.
- almost 30% install new software from untrusted sources e.g. downloaded from a website on the internet (figure 1)
- 70% of asked didn't know about the discussed threat before the survey, shown at figure 2.

- almost 80% of asked people would like to have an application that could analyze the other application’s use of gyroscope sensor and notify about possibly harmful activity (figure 2).

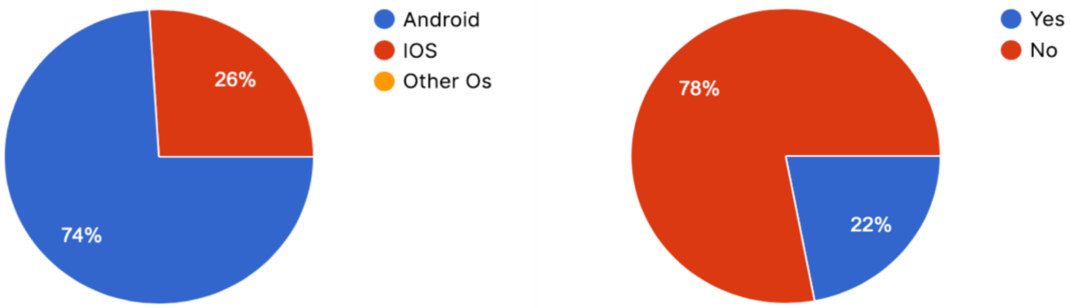
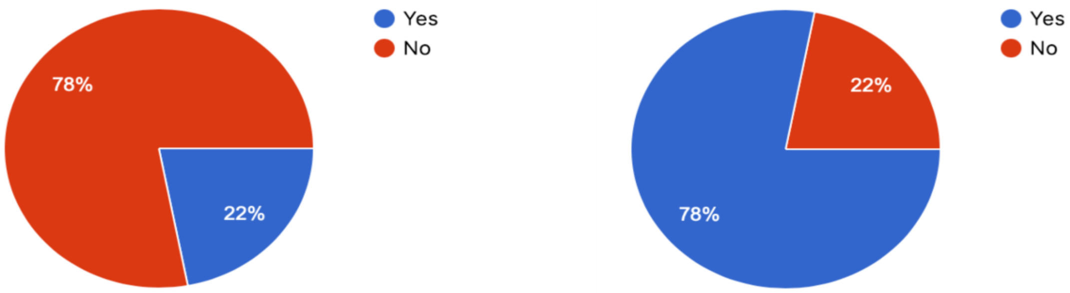


Figure 1 The usage of the mobile OS among the asked users; the percentage of users, that install applications from untrusted sources



Figures 2 Users’ awareness about the threat; users’ need in application capable of detection of dangerous applications

Software for detection gyroscope state on Android operating system

Sampling at higher rates provides more data and will improve percentage of successful recognition. Along with it the threat will appear more dangerous. In this section we explore a maximum available sampling rate for Android mobile devices.

We obtained a list of constant variables defined in SensorDirectChannel class (SensorDirectChannel, 2017), their values are presented in a table 1. These constants are used in configure(…) method call, which allows an application to establish a communication channel with a certain sensor. The constants are used for setting up a sensor event rate which can be read by an application after a new set of data is provided by a sensor. Essentially, the implementation of this mechanism limits the maximum rate at which an application can receive sample of data from a sensor. Nonetheless, the actual sampling frequency for the sensor may be set to even a higher values.

Table 1. Sample rate constants and values

Constant	Typical rate, Hz	Estimated rate, %	Estimated rate, Hz
RATE_STOP	0	—	0
RATE_NORMAL	50	55-220	27.05-110
RATE_FAST	200		110-440
RATE_VERY_ FAST	800		440-1760

In the Table 1 we can see a typical and estimated columns for each constant. In the estimated rate column there are given a ranges of values and an upper limit of the each range is greater than a typical value. This means that the event occurs at a rate within a given range rather than with a constant rate. It is also stated that usually events occur faster than with a typical rate and application can read samples of data at a higher rates that the set up values. For example: if sampling rate is set to a RATE_FAST, the real sampling frequency may appear as 440Hz instead of 200Hz.

Android OS considers these values rather as a guideline than as a requirement. It remains unclear how a maximum rate is set up on a given mobile device while an application is running. To gather more precise information we conducted a test and determined the real maximum rate.

For the measurements we used a SensorData app downloaded from the Google PlayMarket. This application features measure, record and analysis of sensors' data and supports of all the available sensors in a given device. Application was installed on a Samsung Galaxy S8 SM-G950FD with the latest firmware update.

First of all we gather information about all available sensors in the smartphone. This can be done my choosing My Device option that can be selected from the application menu. For the given smartphone embedded incorporates combined gyroscope/accelerometer STM LSM6DSL sensor. sensor models are presented on a screenshot below. As we can see it The typical maximum sampling frequency for the sensor is stated to be 6664Hz (iNEMO, 2017) which is greater than RATE_VERY_FAST constant value.

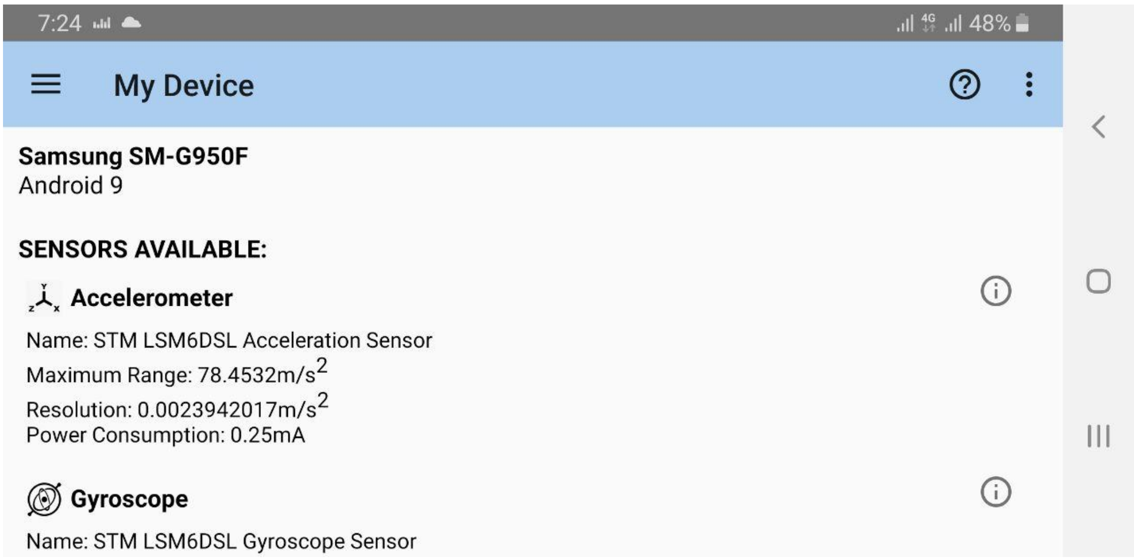


Figure 3 – The available gyroscope and accelerometer sensor model

Then we proceed to setup the application for measurements. From the Settings>Options we selected maximum sampling frequency value which corresponds to the RATE_VERY_FAST constant (shown on a figure 3).

From the *Settings>Sensors* we selected a single sensor to read from. This step to insures that reading from multiple sensors won’t influence measurements. Finally we started to record data on a *Record* screen. The data values and sampling frequency are continuously updated and displayed on the screen while recording.

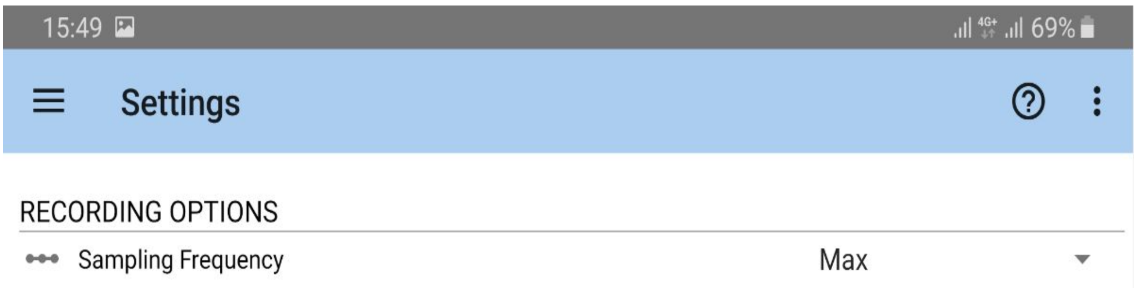


Figure 4 – Setting a sampling rate to a maximum value

Recording was conducted for both calibrated and uncalibrated gyroscope and accelerometer sensors. The results for each sensor are shown below (figures 5).

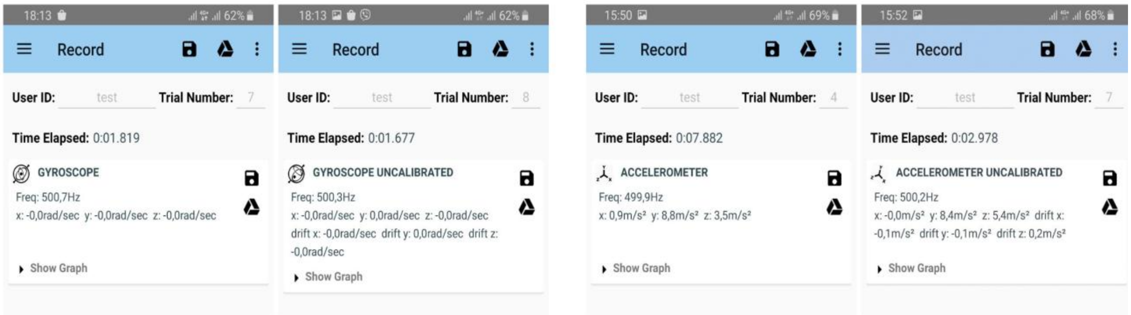


Figure 5 – sampling frequency measuring for the calibrated and uncalibrated gyroscope and accelerometer sensors

During the test the sampling rate for each sensor was about the same. The measured value is nearly 500Hz which is far below the hardware limit. This value is also below the upper limit of the RATE_VERY_FAST constant. This showed us that the sampling rate is limited even more by the given smartphone's firmware. Also we can see that OS doesn’t distinguish between calibrated and uncalibrated sensors in terms of sampling rate limitation.

Nevertheless, the measured value is 2.5 times greater than the presented in the research. Such frequency range is enough to restore the 1-st harmonic for the most male voices using only the Fourier transform and gives attackers more accurate speech recognition.

This problem is not yet properly addressed by the Google and has no a third-party solution so we suggest our own.

Other GameEngine Unity3D also can use gyroscope to support different game actions, for example, when phone is moved through different directions (X,Y or Z axes) corresponding script is given below

Source code of unity C# script processing	Main steps of algorithm of gyroscope recognizing

public class Gyroscope1 : MonoBehaviour {
    public test z;
    public int v;
    private bool gyroEnabled;
    private Gyroscope gyro;	

private GameObject cameraContainer;
    private Quaternion rot;
    TextMesh tm;
	Gyroscope is checked by moving or rotation of mobile device

private void Start()
    {
        cameraContainer = new GameObject("Camera Container");
        cameraContainer.transform.position = transform.position; transform.SetParent(cameraContainer.transform);
        tm = GameObject.Find("ApplyText").GetComponent<TextMesh>();
        gyroEnabled = EnableGyro();
    }	

private bool EnableGyro()
    {
        if (SystemInfo.supportsGyroscope)
        {
	Verification what is supported by current version of operation system

Source code of unity C# script processing	Main steps of algorithm of gyroscope recognizing

<pre>   gyro.enabled = true;
  cameraContainer.transform.rotation =
Quaternion.Euler(90f, 90f, 0f);
            rot = new Quaternion(0, 0, 1, 0);
            return true;
        }
        else
        {
            tm.text = "Gyroscope is not supported";
            return false;
        }
    }   private void Update()
    {
        if (gyroEnabled)
        {
            transform.localRotation = Input.gyro.attitude *
rot;
        }
    }
}</pre>
	Imitation of moving or rotating of mobile device
	
	Output of text into testing component

Questions about organization of User Interface on Unity3D engine are considered in paper (Milovidov, Y., 2017).

Approach for detection and evaluation dangerous threats

Analysis of algorithms that become a cause of leaking voice information from mobile devices are based on

1. Unauthorized access to microphone of mobile device.
2. Using buffers of operating system to collect voice data.
3. Using internet connection to transmit data for hacker.

UML Sequence diagram illustrating the analysis of typical schemas used for voice data leaking is represented in the figure 6.

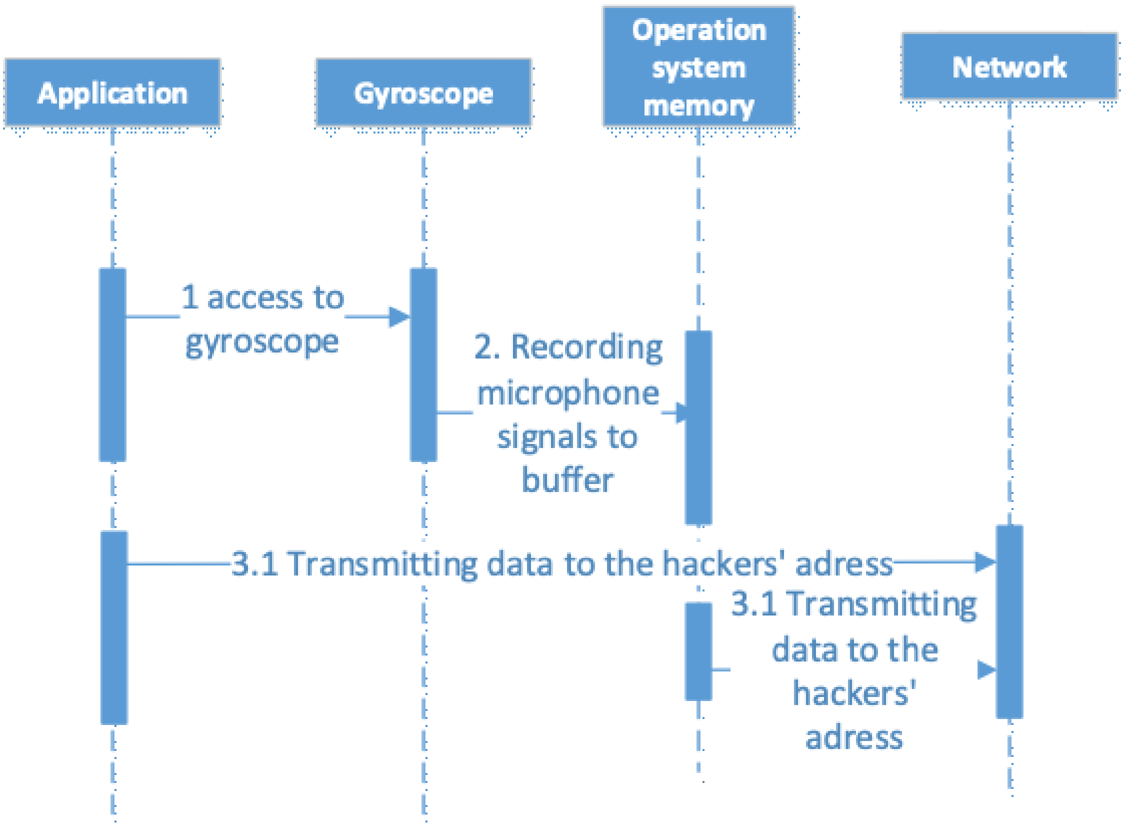


Figure 6. UML Sequence diagram illustrating the analysis of typical schemas

– used for voice data leaking

We propose an approach that allows to distinguish between a different levels of applications’ danger in the context of the discussed threat. This can be done by analyzing a list of applications installed on a mobile device. The analysis utilizes a list of criteria based on the analysis of the process of speech recognition that was presented in the research (Michalevsky, Y at. el., 2021) and on our hypothesis about the behavior of a malicious application as well. The process of evaluation consist of a 3 stages.

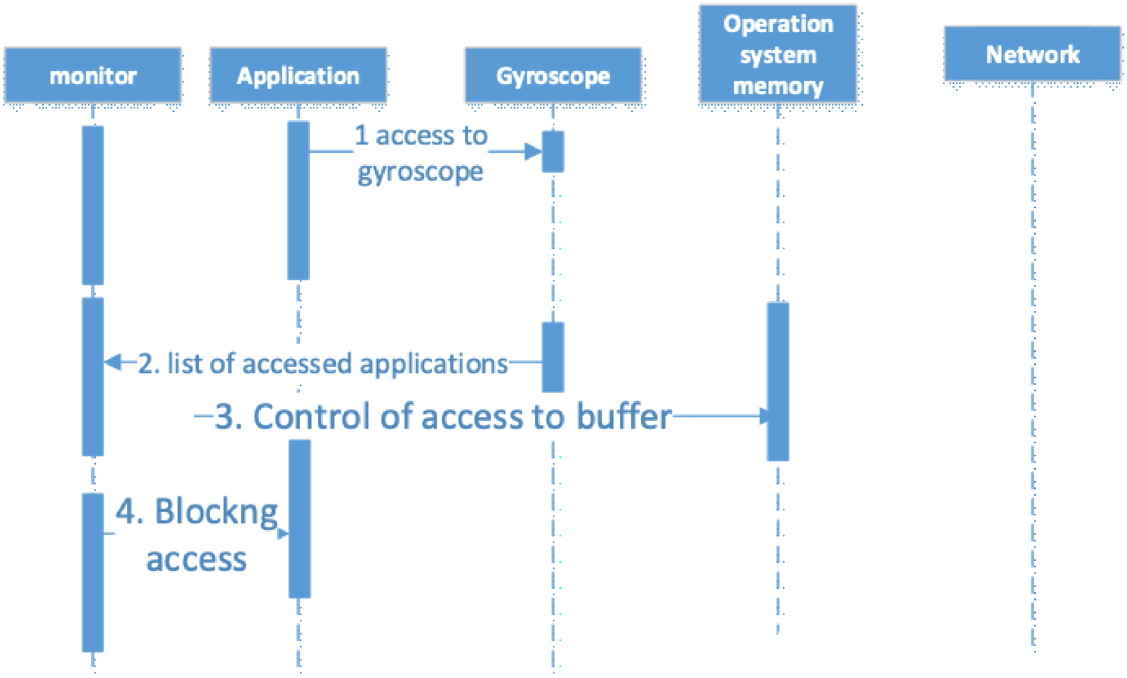
First of all, we need to determine whether a certain application reads a gyroscope/accelerometer data and make a list of such applications. This let us sort a potentially dangerous ones from the whole set of installed apps.

Secondly, we need to define the specific applications` features that will allow to distinguish between a different levels of applications' danger. Depending on our analysis of the technology of retrieving speech from the gyroscope data, we considered a list of application's features required to retrieve spoken information. Each from the list either corresponds to a specific Android permission or reflects an application's behavior. This allows to implement the methodology programmatically. The list of features includes:

- a large amount of heap – to store dynamic variables. Defined by the ApplicationInfo.FLAG_LARGE_HEAP;
- constantly growing amount of memory used to store recorded data before uploading it to a server;
- access to internet connection data and connection itself for analyzing network state and uploading data. Provided by the permissions: ACCESS_NETWORK_STATE, ACCESS_WIFI_STATE and INTERNET.

For each application from the set that was achieved on the first stage we must define which of the above resources used.

On the third stage the set of applications should be divided into a groups by the estimated level of danger. Presented below scheme explains how to evaluate application's danger using the collected data.



– Figure 7. UML Sequence diagram illustrating the gyroscope protecting mechanism

Conclusions

It is shown that sensor sample rate in Android OS appears higher than the value set up by an application. Its maximum value can be limited either by Android OS or by a device firmware.

The proposed methodology utilizes a defined set of criteria that allow to assess application's threat level. Each criterion corresponds either to Android permission type or and can be implement it programmatically.

The results of the conducted survey showed low users' awareness about the threat and the relevance of the implementation of the developed methodology.

Bibliography

(DataReportal, 2021) Digital around the world — DataReportal – Global digital insights. (2021). DataReportal – Global Digital Insights. https://datareportal.com/global-digital-overview

(iNEMO, 2017) LSM6DSL. iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope. (2017, September). STMicroelectronics. https://www.st.com/resource/en/datasheet/lsm6dsl.pdf

(Android, 2021) Manifest.permission. Android Developers. https://developer.android.com/reference/android/Manifest.permission

(Milovidov, Y., 2017) Milovidov, Y. Requirement analysis of user interface components framework for mobile devices. International Journal "Information Models and Analyses" Volume 6, Number 2, 2017 p.146-153

(Michalevsky, Y at. el., 2014) Michalevsky, Y., Boneh, D., & Nakibly, G. Gyrophone: recognizing speech from gyroscope signals. USENIX | The Advanced Computing Systems Association. https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-michalevsky.pdf

(The IT Law Wiki, 2021) Personal mobile device. (2016, August 3). The IT Law Wiki. Retrieved December 3, 2021, from https://itlaw.wikia.org/wiki/Personal_mobile_device

(Poetker, B., 2021) The Mobile Operating Systems That Matter Right Now (+Effects on Development). G2.com. https://www.g2.com/articles/mobile-operating-systems

(SensorDirectChannel, 2017) Android Developers. https://developer.android.com/reference/android/hardware/SensorDirectChannel

Authors' Information

Olena Chebanyuk *– Doctor of Technical Sciences, Professor of Information Security Department, National Technical University of Ukraine, “Igor Sikorsky Kyiv Polytechnic Institute”*

Major Fields of Scientific Research: *Model-Driven Architecture, Model-Driven Development, Software architecture, Mobile development, Software development,*

e-mail: chebanyuk.elena@ithea.org , chebanyuk.olena@gmail.com

Anton Yukhymenko *– student, Department of Information Security, National Technical University of Ukraine, “Igor Sikorsky Kyiv Polytechnic Institute”*

Major Fields of Scientific Research: *Broadband Signals, Information Security,*

e-mail: antonyukhimenko692@gmail.com