Lifting all boats: getting developers to improve app security
modern design with privacy is possible
we put privacy first
privacy through user experience
Tech savvy citizen journalists and activists in the street use Guardian apps to share updates, photos and videos without interception or monitoring by the authorities.
Reporters in the field can use Guardian apps to stay in touch with their safety networks, while safeguarding information on contacts, story notes and captured digital media, enabling a new, secure "reporter's notepad". In addition, high-resolution cameras of new Android hardware meet the quality standards for broadcast, print and online production.
Business people travel all over the world, using foreign networks, bandwidth and systems. At any point, confidential information can be compromised. While some organizations implement solutions, these are often expensive, difficult to use, and not comprehensive.
An undercover human rights researcher traveling through a remote region without mobile data service is able to use Guardian to document local conditions using secured video, audio and photo capture. Data is stored encrypted on the device, and if necessary, it can be safely and quickly erased.
PARTNERSHIPS

We believe in protocols, not products / in partnerships, not proprietary fiefdoms / in building a community of collaborators, not a cacophony of criticism and unnecessary competition / in practical solutions to perilous problems.
there are many threats
CipherKit libraries

YOUR APP HERE!

- Cache Word
- IOCipher
- NetCipher
- SQLCipher
- OpenSSL
- SQLite
- java.io.File
- Android HTTP, java.net.*
- Orbot: Tor for Android
- android.database.*
SQLCipher for Android

https://github.com/sqlcipher/android-database-sqlcipher
~ sjlombardo$ hexdump -C sqlite.db
00000000 53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 33 00 |SQLite format 3.|
... 000003c0 65 74 32 74 32 03 43 52 45 41 54 45 20 54 41 42 |et2t2.CREATE TAB|
000003d0 4c 45 20 74 20 73 32 28 61 2c 62 29 24 01 06 17 |LE t2(a,b)$...
... 000007e0 20 74 68 65 20 73 74 65 70 70 6f 72 74 20 6d 6e |the show.../..o|
000007f0 6e 65 20 66 6f 72 6d 61 74 20 33 00 |ne for the money|

~ $ sqlite3 sqlcipher.db
sqlite> PRAGMA KEY='test123';
sqlite> CREATE TABLE t1(a,b);
sqlite> INSERT INTO t1(a,b) VALUES ('one for the money', 'two for the show');
sqlite> .quit

~ $ hexdump -C sqlite.db
00000000 84 d1 36 18 eb b5 82 90 c4 70 0d ee 43 cb 61 87 |..?6..?p.?C?a.|n
00000010 91 42 3c cd 55 24 ab c6 c4 1d c6 67 b4 e3 96 bb |.B??..?|
00000bf0 8e 99 ee 28 23 43 ab a4 97 cd 63 42 8a 8e 7c c6 |..?(#C??.?cB..?)|

~ $ sqlite3 sqlcipher.db
sqlite> SELECT * FROM t1;
Error: file is encrypted or is not a database
```java
import net.sqlcipher.database.SQLiteDatabase;

SQLiteDatabase.loadLibs(this);

SQLiteDatabase db = eventsData.getWritableDatabase("mymypassword");
```

[https://github.com/sqlcipher/android-database-sqlcipher](https://github.com/sqlcipher/android-database-sqlcipher)
<table>
<thead>
<tr>
<th>Operation</th>
<th>Normal (ms)</th>
<th>Encrypted (ms)</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Table</td>
<td>61</td>
<td>142</td>
<td>132.8%</td>
</tr>
<tr>
<td>500 Inserts (no transaction)</td>
<td>20832</td>
<td>24414</td>
<td>17.2%</td>
</tr>
<tr>
<td>300000 Inserts (with transaction)</td>
<td>11002</td>
<td>11281</td>
<td>2.5%</td>
</tr>
<tr>
<td>500 Updates (w/o index, w/o transaction)</td>
<td>37986</td>
<td>39164</td>
<td>3.1%</td>
</tr>
<tr>
<td>30000 Selects (w/ index)</td>
<td>5334</td>
<td>5498</td>
<td>3.1%</td>
</tr>
<tr>
<td>2500 Updates (w/ index + transaction)</td>
<td>1214</td>
<td>1373</td>
<td>13.1%</td>
</tr>
</tbody>
</table>

https://github.com/sqlcipher/android-database-sqlcipher
NetCipher

https://github.com/guardianproject/netcipher
NetCipher

- add TLSv1.2 on older devices
- good TLS settings on all devices
- easy Tor support
- simplified proxy support
- used in Facebook Android app

https://github.com/guardianproject/netcipher
build.gradle:

compile 'info.guardianproject.netcipher:netcipher:1.2.1'

Android URLConnection:

HttpsURLConnection connection =
    NetCipher.getHttpsURLConnection("https://mysite.com")

ch.boye Apache HttpClient:

StrongHttpsClient httpClient = new
    StrongHttpsClient(getApplicationContext());
more APIs!

- OkHTTP / Retrofit
  'info.guardianproject.netcipher:netcipher-okhttp3:2.0.0-alpha1'

- Google Volley
  'info.guardianproject.netcipher:netcipher-volley:2.0.0-alpha1'

- Apache HttpClient for Android
  'info.guardianproject.netcipher:netcipher-httpclient:2.0.0-alpha1'

https://github.com/guardianproject/netcipher
more proxying!

new proxies:

• Lantern
• Psiphon
• Meek and Tor Pluggable Transports

https://github.com/guardianproject/netcipher
IOCipher

https://github.com/guardianproject/iocipher
IOCipher: The Stack

- `info.guardianproject.iocipher` - Java/JNI wrapper API
- `LibSQLFS / FUSE` - Virtual Filesystem that maps to SQL schema / structured database
- `SQLCipher` - Encryption layer for SQLite
- `SQLite` - Base storage mechanism

https://github.com/guardianproject/iocipher
**Barebones Example**

```java
import info.guardianproject.iocipher.File;
import info.guardianproject.iocipher.FileOutputStream;
import info.guardianproject.iocipher.VirtualFileSystem;

File dbFile = getDir("vfs", MODE_PRIVATE).getAbsolutePath() + "/myfiles.db";

vfs = new VirtualFileSystem(dbFile);

// TODO don't use a hard-coded password! prompt for the password
vfs.mount("my fake password");

File file = new File(dirPath);
File[] files = file.listFiles();
```

[https://github.com/guardianproject/IOCipherExample](https://github.com/guardianproject/IOCipherExample)
Adding IOCipher to App

- manage the password
- connect to your encrypted disk’s file using new VirtualFileSystem(dbFile)
- mount it with a password using VirtualFileSystem.mount(password)
- replace the relevant java.io import statements within info.guardianproject.iocipher, e.g.:
  - import info.guardianproject.iocipher.File;
  - import info.guardianproject.iocipher.FileOutputStream;
  - import info.guardianproject.iocipher.FileReader;
  - import info.guardianproject.iocipher.IOCipherFileChannel;
  - import info.guardianproject.iocipher.VirtualFileSystem;
  - import java.io.FileNotFoundException;
  - import java.io.IOException;
  - import java.io.InputStream;
  - import java.nio.channels.Channels;
  - import java.nio.channels.ReadableByteChannel;

https://github.com/guardianproject/iocipher
CacheWord

https://github.com/guardianproject/cacheword
The problem with app passwords

(Activity, Service and even App lifespan is unpredictable)

https://github.com/guardianproject/cacheword
CacheWord Solution

Activity
- `onCreate()`
  - prompt user for passwd
  - store in CacheWord
- `onDestroy()`
  - close DB instance (but keep cacheword alive!)

Activity
- `onCreate()`
  - re-open DB instance via cached passphrase in CacheWord

SQLCipher DB

Cacheword (long running, foreground, minimal memory service)

https://github.com/guardianproject/cacheword
public class CacheWordSampleActivity extends Activity implements ICacheWordSubscriber {

    mCacheWord = new CacheWordActivityHandler(this);

@Override
    public void onCacheWordLocked() {}

@Override
    public void onCacheWordOpened() {
        // fetch the encryption key from CacheWordService
        SecretKey key = ((PassphraseSecrets) mCacheWord.getCachedSecrets()).getSecretKey();
    }

@Override
    public void onCacheWordUninitialized() {
        mCacheWord.setCachedSecrets(PassphraseSecrets.initializeSecrets(
                CacheWordSampleActivity.this, “my secret passphrase”));
    }

https://github.com/guardianproject/cacheword/tree/master/sample
do not use device IDs as passwords!

KEY = MD5(IMEI + UIN)[0:7]

IMEI: 357725678854269
UIN: -1881034049

IMEI + UIN = 357725678854269-1881034049

MD5(IMEI + UIN) = 4bc36a03296a8b4fc63e5bb8e74db2a2

KEY = 4bc36a0

https://articles.forensicfocus.com/2014/10/01/decrypt-wechat-enmicromsgdb-database/
PanicKit

https://github.com/guardianproject/panickit
PanicKit

https://github.com/guardianproject/panickit
02 Responders

Defaults

The default panic action of a responder is a non-destructive action such as locking the app or disguising the app icon. This default response is set by the creators of the responder app.

User Actions

A. Tap to choose contacts. Go to 03 ZOM CONFIG: CHOOSE CONTACTS

B. Tap to edit the message. Go to 03 ZOM CONFIG: EDIT MESSAGE.

C. Toggle to enable or disable this action. If enabled the default action within this section would be selected (ex: Disguise app icon). The default action would change based on the most recently selected action by the user.

Add a note about the OR cases
TrustedIntents

https://github.com/guardianproject/trustedintents
is it really you?

• am I directing users to the real Orbot?
• did this file come from the real Google Drive?
• was this Intent from one of our own apps?

https://github.com/guardianproject/trustedintents
public class MainActivity extends AppCompatActivity {

    private static TrustedIntents trustedIntents;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        trustedIntents = TrustedIntents.get(this);
        trustedIntents.addTrustedSigner(GuardianProjectRSA4096.class);

        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        Intent intent = trustedIntents getIntentFromTrustedSender(this);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    Intent intent = new Intent(Intent.ACTION_VIEW);
                    intent.setClassName("info.guardianproject.gpg",
                        "info.guardianproject.gpg.MainActivity");
                    trustedIntents.startActivity(activity, intent);
                } catch (ActivityNotFoundException e) {
                    e.printStackTrace();
                    Toast.makeText(activity, e.getLocalizedMessage(), Toast.LENGTH_LONG).show();
                } catch (CertificateException e) {
                    e.printStackTrace();
                    Toast.makeText(activity, e.getLocalizedMessage(), Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
reproducible builds!
XCodeGhost

- malware version of XCode inserted library
- 10s of millions of users received affected apps
- reproducible builds would have prevented this
- more info at https://reproducible-builds.org
F-Droid

an community run Android app store that distributes verified Free Software
fdroidserver tools

- makes reproducible builds trivial
- drozer scans for vulnerabilities
- libscout scans for old libs
- full, automated, secure build environment
- flexible automated signatures
Lifting all boats: getting developers to improve app security

Hans-Christoph Steiner
Guardian Project
hans@guardianproject.info
The Guardian Project
https://guardianproject.info

Secure Your Mobile Life
Apps & Tools You Can Trust

The Guardian Project creates easy-to-use open source apps, mobile OS security enhancements, and customized mobile devices for people around the world to help them communicate more freely, and protect themselves from intrusion and monitoring.