Information Security of Remote File Transfers with Mobile Devices

Term paper for INFORTE.fi event Security of e-Systems and Computer Networks 6-7th of June, Tampere

Matti Vuori
Department of Pervasive Computing
Tampere University of Technology
Student number 47226
matti.p.vuori@tut.fi

***Abstract*-Mobile devices are everywhere and will cause security risks to the whole ICT Infrastructure if not handled properly in the design of the mobile platforms, communication infrastructure and application design for both the mobile device and the server ends. Many forms of communication between local and remote systems include transferring of files between the mobile device and a server – or another device. This report looks into typical such situations, the risks involved, some of the mechanisms for controlling the security and into the testing of mobile applications.**

***Keywords*-information security; mobile devices; mobile phones; mobility; platform security; file transfer; file storage; software testing.**

I. INTRODUCTION

This paper gives an outline of the security issues related to remote file transfers with mobile devices. The paper aims to open up the general situation so we can "see the forest from the trees". Deep discussions about security technologies and solutions are left outside of this paper. In this paper, we first describe the context of file transfers. After that, we look into the special problems that mobility and mobile devices present. Then we look into how application should be tested during their development in order to minimize the risks.

II. BACKGROUND

Mobile devices, such as laptops, tablets and mobile phones are increasingly used for "serious" data handling in work, E-commerce and E-government. The whole global population seems to have a phone that has Internet capabilities and that development provides great opportunities but also new information security risks. There is a growing understanding about the risks [11][3][6][10][12] and device manufacturers take the issues very seriously. However, the application developers not always do not, and the attention of those parties is not sufficient. The overall systems are complex functionally and technologically and there are many actors in various roles. The number of possible cyber-criminals is growing at the same rate as the number of users. That is why the issues of mobile security need more and more attention. There is a danger of getting a false sense of security, because of the seemingly good situation, as described by Becher [11]:

"*Many researchers and practitioners are expecting a major security incident with mobile phones ever since these devices began to become more powerful: with increased processing power and memory, increased data transmission capabilities of the mobile phone networks, and with open and third-party extensible operating systems,*

phones become an interesting target for attackers. However, no major incident has happened as of the time of this writing."

III. Context and Overall Architecture of Remote File Transfers

A. Transfer contexts and types

For security we need to first assess the different contexts in which the file transfers can occur. The following are the most common:

- Social interaction and acting in social media, including uploading photographs to sites such as Facebook.
- Maintaining a web site, for example a personal photography site on a tailored system or a ready-made platform.
- Work. Transferring work documents from a back-end system, including email systems, to and from a laptop, tablet or a mobile phone.
- E-commerce. Receiving tickets and uploading documents, such as photographs of a passport to prove identity.
- E-government. Uploading and downloading documents when dealing with officials. Even this is often mobile activity.

The types of file transfers also varies. Upload and download are most common, but synchronization is becoming more and more important. For example, one might synchronize a folder on a laptop with a folder on a server at workplace, or automatically synchronize a directory of photographs. Lately, file transfers between devices, for example between two mobile phones or between a camera and a mobile phone have become more common (those are outside the scope of this paper).

The applications used for initiating and controlling the transfers may vary from web browser based user interfaces, in which case the application may really be thought to reside on the server (this is a simplification) or special mobile application ("apps").

The file transfers may happen automatically in the case of synchronization or may use a file download or upload dialogue, with any associated security problems.

Usually the files are transferred as such, but we may consider, from the end users perspective at least, any additional processing as being part of the transfer. That might include for example resizing a photograph or extracting metadata from a document and executing some storing action based on those.

The file type can be various, including the following:

- Photographs, often as JPEG files that contain metadata, usually in EXIF format.
- Word processor documents, most commonly in Microsoft Word format.
- Spreadsheets in usual office spreadsheet formats, most commonly in Microsoft Excel format.
- Presentations, most commonly in Microsoft PowerPoint format.
- PDF files, often converted from the aforementioned document types.

- Application installation packages. For tablets and mobile phones, application installation packages are often transferred to the device after being bought from a store.
- Other.

B. Orientating risk assessment

The various usage contexts have varying risk levels when it comes to file transfers. Often, any work related activities have the highest risk, as the information is confidential. For E-commerce related transfers the documents may have very critical information that could be used for criminal purposes. When it comes to any confidential personal information used in E-government context, the same criticality applies. Yet, when it comes to the interest of other parties, the E-commerce is clearly the most important context and criminals will want to capture any information that they possibly can.

Application installation packages have another, special role. If a package should be malware or criminally altered, such a package can compromise the security of any data transfers.

When it comes to the risks to the files transferred, the following are most important:

- The file gets into the hands of a third party.
- The file gets corrupted and data is destroyed or a document cannot be opened.
- Metadata gets removed from the file.
- The file gets infected with a virus.
- The file gets altered otherwise.

However, the list above is at this point of the paper just for orientating the reader. We will get back to the risk assessment later, after addressing some of the issues that influence the risks.

C. The nature of mobility

The remote user may have varying mobility, causing varying problems to the file transfers.

The user may be located stationary in some location during the whole file transfer process, but the remote session with the local system may span various locations and thus may use various networks during the session.

Alternatively, the use may be on the move, perhaps using public transport and for even medium size files the transfer may require roaming and using various networks.

REMOTE FILE TRANSFER SECURITY TECHNOLOGY CHALLENGES

A. System security architecture

Understanding the overall system architecture is the next step in understanding of the security issues.

The system can be said to consist of a host system, which in this context can be said to be “local”, whereas the mobile device is “remote”. It could be an information system, an E-commerce site, a social media system or even a peer to peer (P2P) system, in which case it is another mobile device. However, we shall not pay any attention to the last case in this report.

The remote, mobile system is a mobile device that is used in a mobile manner, most importantly its network access point may vary during use. The device can be considered to be a mobile phone, a tablet computer, or a laptop. In the case of a tablet computer or a laptop, a mobile phone can also be used as a modem or to provide an ad-hoc WiFi station.

Between those there is the network system consisting of any modern network technology. For the purposes of this assessment, we may consider the connection at the remote end to always be wireless. General risks involved in that are plentifully reported in literature, see for example Kizza [7]. Special element is provided to the system by any special proxies that are sometimes offered to speed up mobile connections. Such proxies are provided by network providers or by mobile web browser manufacturers, such as Opera. Sometimes they may cause content to be altered, for example the transferred HTML to be simplified for a given web browser.

Sometimes the connections and thus file transfers can be initiated with various close range technologies, such as NFC connection between the mobile device and another device, or a Bluetooth connection between two remotely located devices. NFC is one interesting are in security. Mulliner [4] has an interesting paper about its vulnerability risks.

One of the defining characteristics of a mobile system is that there are many interfaces in the overall system, which also means that there are many attack points in the overall system! They may include:

- Mobile device file system
- Mobile operating system
- NCF initiation of communication.
- Mobile web browser or app
- WiFi network
- Firewall
- Internet (and anything that the packets might pass during it)
- Server network firewall
- Load balancers and routers
- Server application
- Database for the files

B. Characteristics of the remote system

Obviously, it is important to understand the characteristics of the remote system. Of the device types mentioned previously, the mobile phones are more radically different from traditional computing equipment that we need to take a look at those in appropriate detail.

First of all, a mobile phone is a computer, albeit a small one with limited computing capabilities. Yet, their RAM and mass memory sizes often surpass those of a desktop computer from a decade ago in the case of the high-end smartphones. However, the low-end phones are very different. Yet, the whole range of devices needs to be considered due to the fact that low-end devices are the ones that the world's population really uses, not the high-end devices that are more often mentioned in media.

All mobile phones have a unique platform security system, very unlike the one found in the operating systems of desktop and laptop computers. One important element of that is that is grants “capabilities” (perhaps by another name) to applications that define what data they can access. Unfortunately, in some systems, it may be possible for the user to change the security configuration.

Below that security system there is an operating system, which may even be derived from a non-mobile operating system; for example Android is based on Linux, as is MeeGo, and Windows Phone is very much related to other operating systems in the Windows product family.

The same applies to web browsers. While many browsers may look simple, they may share their technology with desktop browsers, including the security features – yet the range of those may not be available to the user and the default settings must be used.

For an example, we will look into the security of Android based from Shabtai et. al. [2] in Table 1.

Table 1.
SECURITY MECHANISMS INCORPORATED IN ANDROID [2]

Mechanism	Description	Security issue
Linux mechanisms		
POSIX users	Each application is associated with a different user ID (or UID).	Prevents one application from disturbing another
File access	The application’s directory is only available to the application.	Prevents one application from accessing another’s files
Environmental features		
Memory management unit (MMU)	Each process is running in its own address space.	Prevents privilege escalation, information disclosure, and denial of service
Type safety	Type safety enforces variable content to adhere to a specific format, both in compiling time and runtime.	Prevents buffer overflows and stack smashing
Mobile carrier security features	Smart phones use SIM cards to authenticate and authorize user identity.	Prevents phone call theft
Android-specific mechanisms		
Application permissions	Each application declares which permission it requires at install time.	Limits application abilities to perform malicious behavior
Component encapsulation	Each component in an application (such as an activity or service) has a visibility level that regulates access to it from other applications (for example, binding to a service).	Prevents one application from disturbing another, or accessing private components or APIs
Signing applications	The developer signs application .apk files, and the package manager verifies them.	Matches and verifies that two applications are from the same source
Dalvik virtual machine	Each application runs in its own virtual machine.	execution, and stack smashing

Shabtai et. al. [2] also list the Table 2. Security mechanisms applicable to Android.

Table 2.
SECURITY MECHANISMS APPLICABLE TO ANDROID.

Mechanism	Description	Security issue	Existing tools
Antimalware	Scans files, memory, short message service (SMS), multimedia messaging service (MMS), email, URLs, and Java scripts	Viruses, Trojan horses, worms, root-kits, and other malware	SMobile, Mocana, DroidHunter, ClamAV*
Firewall	Can block or audit unallowed connections to or from the device	Services that are exposed to an untrusted network; network attacks	SMobile, Netfilter/iptables
Intrusion-detection / prevention systems	Detects abnormal or known malicious behavior in the system, process, network traffic, or user	Fraud (for example, expensive calls), unusual telephone activity, theft, malicious attacks	Andromaly, DroidHunter
Linux access control	Limits the access of processes and users to resources or services	Damage from malicious or exploited applications	SELinux10
Login	Lets users provide a secret password to use the device	Unauthorized device use	Android screen-lock pattern
Selective Android permissions	Lets users grant only a subset of permissions to an installed application	Unneeded permissions that attackers can maliciously exploit	
Android permissions access control	Hardens Android devices by limiting granted permissions using a predefined policy; relevant mainly to corporate users	Unneeded permissions that attackers can maliciously exploit	Secure Application INTeraction†
Permissions management application	Scans Android's applications' permissions, giving the user a concise summary	Installed unwanted applications, Trojan horses	
Data encryption	Encrypts the device's content	Access to sensitive information when the device is lost or stolen	
Phone call encryption	Provides secured connection (authenticated or encrypted)	Eavesdropping, identity verification	
Spam filter	Blocks MMS, SMS, emails, and calls from unwanted origins	Spam	
Virtual private network	Connects to a remote network over the Internet; relevant mainly to corporate users	Insecure network connections	PPTP, L2TP, and IPSec-based VPN connections enabled on Android release 1.6 (that is, Donut)
Application certification	Allows for signing each application with a certificate authority (CA)	Damage from untrusted applications	Open Mobile Terminal Platform's (OMTP) Application Security Framework‡
Resource management	Enables fairness in resource allocation (CPU for phone application, disk quotas, I/O rate limiting and quotas, network quotas, and traffic shaping)	Denial of service (DoS)	
Remote management	Remotely configures and manages the device (settings, firewall policy, remote "bricking," application tracking); relevant mainly to corporate users	Device theft	

Mechanism	Description	Security issue	Existing tools
Context-aware access control	Dynamically allows and restricts access to resources and services based on a predefined model	Breaches of confidential content and the integrity of services	"Local" application on the Android Market, Andromaly
Integrity checking	Verifies system and application state	Offline tampering	

Enck et. a. [5] also have a nice paper about Android security from the viewpoints of the application components, but we will not go into that here. Enck also has another paper [9] that presents many issues of application certification – a very critical issue in mobile security. For comparison Delac et. al. [12] describe briefly the platform security of iOS operating system, but for space restrictions we will not go to that.

C. Security challenges

Due to the aforementioned facts, the remote systems may have very similar security problems as any desktop system, including:

- Security "holes" in the system.
- Cracking.
- Network traffic monitoring.
- Malware and viruses that could cause any problem from slight annoyance to "bricking" a device.

Viruses are an interesting area. They have not *yet* become as large a problem as was expected when smartphones become available, but they really are a large threat. See Shih et. al. [8] for a nice review of them, although from a couple of years back (2007).

D. User interface related security problems

Especially mobile phones are small and that necessarily causes some problems that have security percussions.

Due to the small size, any file selection dialogues are small or may require scrolling to get to all transfer settings. There is an increased probability to download or upload a wrong file or to select wrong visibility settings for uploaded files, which is a common problem in social networking sites. On some systems, checking the files after downloading is so slow that it will easily get skipped, unlike the desktop systems where one might immediately open a downloaded file to see that it is the desired one and was downloaded correctly. On high-end smartphones these problems are getting smaller due to bigger screens and touchscreens that make navigation in dialogues easier than in previous device generations.

E. Practical observations of security design issues

To show the security platform and device issues more practical, here are some anecdotic observations:

- In Twitter, a user reported that when installing a VoIP application to an Android phone, the installer checked that the user wishes to grant sufficient rights to the application. The sufficiency was: EVERYTHING. That is

not exceptional and users will respond positively to such queries. If the application should be malware, it would have access to every file and every piece of data on the device.

- Some mobile phones only allow installing signed and approved applications. However, that setting can usually be turned off, meaning that there is no limitation to installations. In the same way, some platforms allow installation from an app store, but that default can be changed, raising the risk of installing malware directly from some cracker's site.
- While some applications may have protected directories or databases for their data, downloading and storing files is often done to any public directory on the device, making the files available for any application.
- If an online system uses passwords that need to be typed for each session (as they should), mobile users – at least the author – prefer short and simple passwords, as typing of good passwords can be difficult on some touch screens (due to cumbersome shift key, very small keys, hiding of what is typed with stars and so on).
- Files are often compressed into zip files for various real and imaginary security reasons. Mobile phones often have lacking functionality for creating and opening zip files.
- No mobile phone probably has a tool for calculating a checksum for a file as standard equipment and most perhaps not even as an accessory.

F. Special challenges of remote activity that may compromise security of file transfers

The following include some important challenges to the security of remote file transfers to and from mobile devices:

- Interruptions of network connection. The file transfer may be interrupted and systems at both ends must detect the situation and recover from it. Unlike desktop applications, mobile applications should be able to tolerate network problems and perhaps continue when a connection is restored. This of course depends on the application. Mummert et. al. [1] lists these problems with disconnections: a) updates are not visible to other clients, c) cache misses may impede progress, c) updates are at risk due to theft, loss or damage, d) update conflicts become more likely, e) exhaustion of cache space is a concern.
- Use of unknown or for some reason incompatible networks. While roaming, the networks may change automatically or by the choice of a user and may lead to using insecure connections.
- Slow data transfer rate. The data transfer rate may be so slow that the system may encounter a timeout and cancel the transfer.
- Due to the network problems, file corruption can occur. This is again unlike non-mobile situations where the networks and (general) routing is known and the problems may have been solved as they may have appeared during the first times of using a system.

V. SECOND RISK ASSESSMENT

Now that we have tackled many security issues, it is a good time to present a second risk assessment that collects together many of the findings. We will do this by looking into the flow of a file upload process.

Table 3.
RISK ASSESSMENT OF A FILE UPLOAD.

Phase	Threats	Control measures
File stored in mobile device's file system	Access by intruder: information leak, changing or altering file	Good mobile platform safety (proper choice of platform)
	Access by malware in the system: information leak, file corruption, file modification, virus infection	File storage policies, mandatory application security practices, virus checking
Open a network application to carry out a "business process"	Wrong application chosen, that mimics the real one	Just being very careful with browser based applications Special mobile apps can be more secure
Start a session and execute user actions to start uploading process	User credentials leaked	Use of encryption for all applications
Selection of file to upload	Wrong file selected	Simple user interfaces suitable for small screens that minimize risk of selecting wrong file
File transfer from remote device to server	Risks related to upload session	Using good generic security measures for session security and file encryption
	Upload interrupted	Upload mechanisms designed to be robust for mobile use (speed, handling interruption, roaming during transfer, timeouts, recovery), upload process not modal from business process perspective
	Long upload session prone to problems	Use small file formats, file compression, check and prohibit transfer of large files
Correct file received	Changed, corrupted or infected file received,	Use CRCs to validate received file, automatic checking for malware, viruses
File transferred to information system	File altered during process. This is a generic issue, but for example digital cameras require special attention (resizing, removal of metadata)	Good system design practices that "honor" the user's intentions

It would be interesting to continue with more scenarios like that but due to space restrictions in this report it is not possible.

VI. REFLECTIONS ON TESTING OF REMOTE FILE TRANSFER FUNCTIONALITY IN APPLICATION

File transfers are obviously tested during application development during system testing. Ideally, they should be tested in proper end-to-end manner using any variations imaginable for the files and the configuration for the transfer. In practice, for example very large files or handling of any variations in filenames are not tested sufficiently. Any dialogues or other user functions need to be subjected to functional testing to make their functioning as robust as possible, and usability assessment and testing in order to develop them to handle any human errors. Sometimes, a special human error assessment may be in order.

Mobile platforms usually have mandatory testing requirements that an application must pass for it to get digitally signed and for it to be sold in the app store of the platform. This practice was already used during the change of century with Symbian applications. Such a test suite obviously must be used in testing even when the real acceptance testing would be provided by the store or by a third party.

Notably missing is the systematic testing of how the system handles any roaming and network problems. That requires using a special test network system, which is not available to most application developers though there may

be regional test networks available but their capabilities may still be lacking in this regard. Device manufacturers obviously must have and use such networks, but testing should always be sufficient also at the third party application level. A new area of robustness testing is fuzzing, where the system is tested for its capability for handling any kinds of even "broken" files. That way the receiving end of the system can be assured to be robust and not to for example crash or to lock a session or even lock a user out of using the system with any file handling related problems. This also reduces the risk of denial of service attacks. Of course, any system functions should be tested with proper security testing techniques, including for example testing XSS attack handling in any file transfers. Unfortunately, the general culture of security testing is very weak and at best, very simplistic testing is done in companies.

V. Conclusions

Remote file transfer, using mobile devices, is clearly an area that has many risks that should be understood in application and systems development. The devices need to provide a good platform security system that cannot be compromised by actions of the user and secure the stored files from any malware or intruders. Companies and consumers should prefer safe products. However, back-end systems must not rely on the quality of the mobile terminals but be prepared for anything for the files received.

All elements of the network infrastructure and the network stack need to be robust and to be able to handle any disturbances in connectivity. The network infrastructures need to enforce good security practices and technologies. Server applications that serve or revive the files need to be designed for mobile use and need to consider the special requirements of mobility, such as the problems of problems in connectivity. Attention must be paid to the user interfaces in order to minimize the risk of human error. As for the files themselves, such file formats that are compact – the shorter the transmissions, the less there are problems – and have a minimal risk of, for example, virus infection. Thorough testing of applications is very important and traditional testing methods need to be expanded with new test types, including fuzzing and testing of how any network distractions are handled.

References

[1] Lily B. Mummert, Maria R. Ebling & M. Satyanarayanan. 1995. Exploiting Weak Connectivity for Mobile File Access. Proceeding SOSP '95 Proceedings of the fifteenth ACM symposium on Operating systems principles, p. 143-155.

[2] Asaf Shabtai, Yuval Fledel, Uri Kanonov, Yuval Elovici, Shlomi Dolev & Chanan Glezer. Google Android: A Comprehensive Security Assessment. Security & Privacy, IEEE (Volume:8 , Issue: 2), March-April 2010, p. 35-44.

[3] Max Landman. Managing Smart Phone Security Risks. Proceeding of InfoSecCD '10 2010 Information Security Curriculum Development Conference. p. 145-155.

[4] Collin Mulliner. Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones. International Conference on Availability, Reliability and Security.

[5] William Enck, Machigar Ongtang & Patrick McDaniel. Understanding Android security. IEEE Security & Privacy. Volume 7, Issue 1, p. 50-47.

[6] Anup K. Ghosh & Tara M. Swaminatha. Software security and privacy risks in mobile e-commerce. Communications of the ACM. Volume 44 Issue 2, Feb. 2001, p. 51-57.

[7] J.M. Kizza. A Guide to Computer Network Security. 2009. Springer.

[8] Dong-Her Shih, Binshan Lin, Hsiu-Sen Chiang, Ming-Hung Shih, 2008. Security aspects of mobile phone virus: a critical survey, Industrial Management & Data Systems, Vol. 108 Iss: 4, pp.478-494.

[9] William Enck, Machigar Ongtang & Patrick McDaniel. On Lightweight Mobile Phone Application Certification. Proceedings of the 16th ACM conference on Computer and communications security. P. 235-245. 2009.

[10] Collin Mulliner. Privacy Leaks in Mobile Phone Internet Access. Proceedings of the 14th International Conference on Intelligence in Next Generation Networks (ICIN), 2010. P. 1-6.

[11] Michael Becher, Felix C. Freiling, Johannes Hoffmann, Thorsten Holz, Sebastian Uellenbeck & Christopher Wolf. Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices. Proceedings of the 2011 IEEE Symposium on Security and Privacy. P. 96-111.

[12] G. Delac, M. Silic & J. Krolo. Emerging Security Threats for Mobile Platforms. Proceedings of the 34th International Convention of MIPRO, 2011, p. 1468-1473