CYBER SECURITY TO ANDROID MOBILE DEVICES

Yordan Shterev, Kaloyan Kolev

"Vasil Levski" National Military University, Veliko Tarnovo, Bulgaria

jshterev@abv.bg; kolevkaloqn35@gmail.com

КИБЕР СИГУРНОСТ ЗА МОБИЛНИ УСТРОЙСТВА С ANDROID

Abstract: *This article presents, summarizes, and develops cybersecurity technology concepts for Android mobile devices according to the current state of the problem. Based on the OSI model for android mobile devices, security protocols and applications, and also the main vulnerabilities and threats on the other hand, the current defenses against cyber attacks are revealed.*

Keywords*: Cybersecurity, Android, Mobile Devices*

Въведение

Развитието на хардуера на информационните и комуникационните технологии доведе до създаването на мобилни устройства. Операционна системата (ОС) Android е платформа за мобилни устройства, която е проектирана и разработена с отворен код [2].

Компанията Android Inc. е основана в Пало Алто, Калифорния, през октомври 2003 г. През юли 2005 г. Google купува Android Inc. Екипът разработва в Google платформа за мобилни устройства на основата на Linux. На 22 октомври 2008 г. на пазара излиза първият смартфон с ОС Android. От 2008 г. Android претърпява многобройни актуализации, които постепенно подобряват ОС, добавят нови функции и коригират грешките в предишните версии.

ОС Android е най-разпространена в сравнение с другите ОС-ми за мобилни устройства [11]. За да се гарантира сигурността ОС Android, необходимо е да има мощен защитен механизъм. Затова е необходима мощна и адекватна за сигурността архитектура. Тя има многопластова сигурност, проектирана да осигурява необходимата гъвкавост за ОС с отворен код [1], [2], [5], [8].

Броят на приложенията за Android се увеличават много бързо. Това предполага увеличаване на уязвимостите, насърчава разработчиците на зловреден софтуер, дава възможности за кражба на лични данни, вреди на репутацията на разработчиците, на пазара на приложения и др.

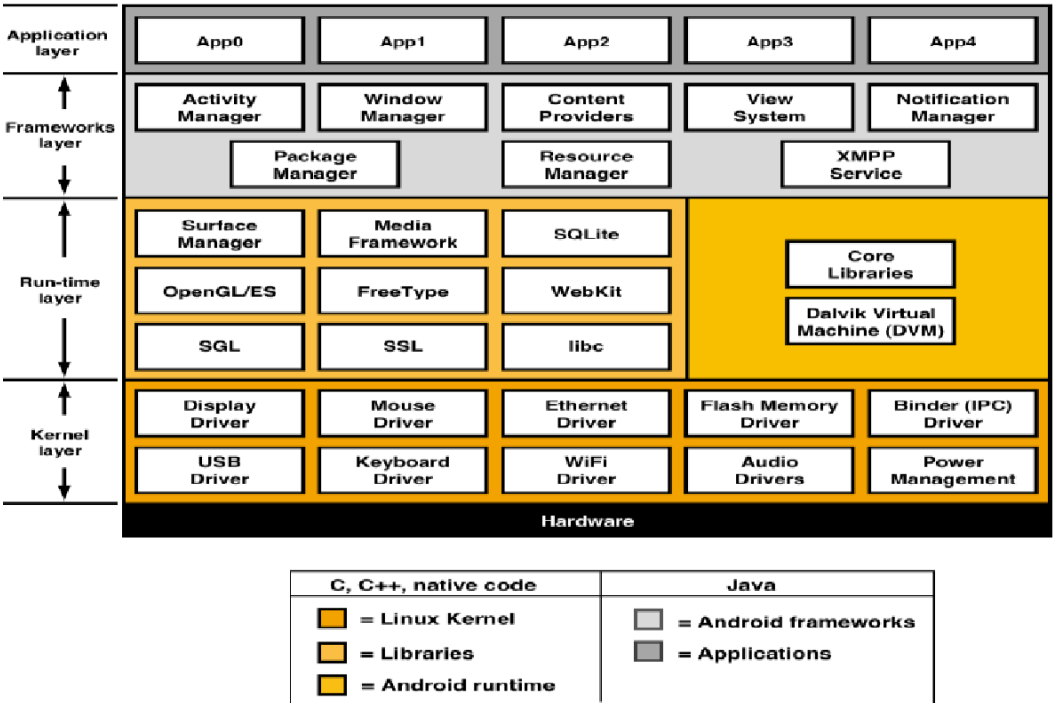
Целта на тази статия е да се извърши преглед на киберсигурността на мобилни устройства с ОС Android, да се изследват архитектурата ѝ, заплахите, уязвимостите и основните принципи на сигурността и съществуващите защити.

Архитектура на операционна система Android

ОС Android има *модулна организация* и *стекова архитектура*. Компонентите на стековата архитектура са разпределени в четири слоя, всеки от които изпълнява специфични функции, свързани с работата на ОС и изпълняваните потребителски приложения [1], [3] – фиг. 1:

- приложен слой (Application layer);
- application framework;
- среда на изпълнение за Java-приложения и библиотеки (Run-time layer);
- слой на ядрото на Android (Linux kernel).

Всеки слой осигурява функционирането на слоевете над него в стека и същевременно ги изолира от детайлите, характеризиращи работата им от по-ниско ниво - фиг. 1.



Фигура 1. Архитектура на операционна система ANDROID.

1. Приложен слой

Разположен е на върха на софтуерния стек, разработен предимно на Java. Включени са по подразбиране системни приложения, използвани от потребителят: Browser, Phone, Contacts, Gallery, Calendar и др., които се разпространяват с Android. Този слой включва и инсталираните в средата потребителски приложения от Google PlayStore или дори разработени самостоятелно. Приложенията от този слой

използват компонентите и услугите, предоставени от слоя Application Framework и се изпълняват чрез виртуалната машина Dalvik (DVM).

2. Слой „Application Framework" (Софтуерна рамка на приложенията)

Включва съвкупност от услуги, които формират средата, в която приложенията за Android се управляват и изпълняват Архитектурата на тези компоненти е разработена за по-лесно многократно използване. Създадени са изцяло на езикът Java. Тук се разполагат основните услуги на Android за управление на жизнения цикъл на приложенията, на пакети, ресурси и др. Всяко приложение може да използва базовите възможности на съставляващите Application Framework слой както директно, така и индиректно - чрез средствата и разрешенията, предоставени от други приложения. Програмистите имат пълен достъп до тези компоненти чрез съответния приложен програмен интерфейс (представен в придружаващите средата Java пакети и класове). Основните услуги, съдържащи се в този слой са:

- *Activity Manager* – отговорна за всички аспекти, свързани с протичането на жизнения цикъл на приложението (стартиране, спиране, възстановяване на изпълнението и т.н.) и поддържания софтуерен стек.
- *Window Manager* – Java абстракция на поддържания в системата мениджър на екрана (surface manager). Window Manager осигурява достъп до функциите на мениджъра на екрана, като дава възможност на приложението да обяви своето клиентско визуално пространство и да използва екранни елементи, например лента на състоянието (status bar).
- *Content providers (Доставчици на съдържание)* – позволяващи на приложенията да публикуват и споделят данните си с други приложения.
- *View System* – разширяема система от визуални компоненти и характеристики, използвани за изграждане на потребителския интерфейс на приложенията и за комуникация с потребителя.
- *Notification Manager* – отговорен за уведомяването на потребителя за настъпили събития (изпълнението на фонови задачи).
- *Resource Manager* – предоставящ достъп до вътрешните ресурси на приложението, които не са програмен код – символни низове, цветове, параметри на оформлението на графичния интерфейс, битови карти и др.
- *Package Manager* - система от услуги за инсталиране и деинсталиране на приложения, за получаване на информация за приложенията, текущо инсталирани на устройството.
- *Telephony Manager* – предоставя достъп и информация за телефонните услуги, достъпни на устройството, като например информация за статуса и абоната, SMS, MMS и др.

- *Location Manager* – предоставя достъп до услугите за местоположение и до информацията, свързана с актуализирането на промените в местоположението – чрез GPS, клетъчни ID или чрез локални Wi-Fi бази от данни.

XMPP (Extensible Messaging and Presence Protocol) услуги основани на набор от отворени технологии за незабавни съобщения, присъствие, многостранен чат, гласови и видео разговори, сътрудничество, олекотен междинен софтуер и обобщено маршрутизиране на XML (Extensible Markup Language) данни.

3. Java-изпълнителна среда (Dalvik VM) и библиотеки, съдържащи и обслужващи средата за изпълнение.

Изпълнителната среда (runtime), осигурява изпълнението на Java-приложенията. Тя включва виртуална машина и необходимите за функционирането ѝ библиотеки (Core Libraries или Dalvik Libraries). Последните се разделят на три категории:

- библиотеки на виртуалната машина;
- Java-библиотеки на оперативната съвместимост (Java Interoperability Libraries);
- Библиотеки на Android OS – тази категория обхваща Java-базирани библиотеки, които са специфични за разработването на приложения в Android. Ползват пакетите *android.app, android.content, android.database, android.graphics, android.hardware, android.text, android.widget, android.webkit, android.provider* и др.

Системни и функционални библиотеки на C и C++.

Според своето функционално предназначение и мястото им в софтуерния стек на Android има две основни групи:

- системна библиотека на C;
- функционални библиотеки на C/C++.

Google разработи системна библиотека Bionic за мобилни устройства на основата на Linux. Размерът ѝ е около около 200 kB и взема под внимание ограниченият изчислителен потенциал на мобилното устройство и е оптимизирана по отношение постигането на максимално бързодействие.

Функционалните библиотеки са разработени на C/C++. Тук влизат библиотеките:

- WebKit – съдържа софтуерен механизъм за представяне и изобразяване на уеб-страници в браузъра. Възможности за подобряване работата на браузърите Apple Safari и Google Chrome;

- Media Framework – медийна библиотека, поддържаща функции за работа с повечето аудио и видео формати: MPEG-4, JPG, PNG, H.264, AAC и др.;
- SQLite – пълноценна „олекотена" версия на система за управление и поддържане на релационни бази от данни, които се използват от приложенията;
- OpenGL – съвкупност от графични библиотеки;
- SGL (Scalable Graphics Libraries) – за изобразяване на двумерни графики;
- OpenSSL – инструментариум с отворен код, реализира *Secure Socket Layer* (SSL);
- Surface Manager - предоставя функционалност за управление на дисплея;
- FreeType осигурява поддръжка на шрифтове.

Повечето от тези библиотеки са с отворен код. За разработчиците достъпът до функциите на тези библиотеки се реализира чрез използването на слоя Application Framework.

4. Слой на ядрото на ОС Android (Linux Kernel)

Базовата подсистема в стековата организация на ОС Android – Linux-ядрото е тази, която осигурява фундаменталните функции на взаимодействие на системния софтуер с конкретната хардуерна конфигурация. Функции на ОС:

- управление на процесите и паметта;
- многозадачност, файлова система;
- входни-изходни канали;
- системни услуги от ниско ниво;
- управление на захранването и т.н.

В ядрото на ОС са реализирани специфичните за хардуера драйвери: Wi-Fi и Bluetooth комуникации, сензорни екрани, камери, акселерометри, GPS-приемници, USB драйвери, управление на захранването, аудио драйвери, клавиатурни драйвери, Binder (InterProcessor Communication) драйвери и др.

Слоят на ядрото е основа на цялата стекова архитектура на ОС Android, предоставя максимална гъвкавост и възможност за работа с разнообразие от хардуерни мобилни конфигурации и модели устройства.

Процеси

Стартирането на всяко приложение в Android е свързано със създаването на отделен процес, който първоначално съдържа една единствена изпълнявана нишка [2], [8]. По подразбиране всички компоненти на приложението се изпълняват в тази нишка, която се нарича „главна" (main). Процесите в ОС Android са по същество Linux-процеси. В Linux - специфичен механизъм на виртуалната памет, предоставя на всеки процес достъп до едно линейно и непрекъснато свързано виртуално адресно

пространство. Това непрекъснато пространство от виртуални адреси се изобразява върху физическото адресно пространство на ОС. Всеки процес работи в своето виртуално адресно пространство и няма достъп до адресните пространства на останалите – т.е. достъпът на процесите до паметта се ограничава до техните собствени виртуални адреси. Само ОС има достъп до адресното пространство, представяно от физическата памет.

Процесите в Android притежават приоритети, базирани на системата от такива, приета в ОС Linux. Всеки процес включва поне една изпълнявана нишка, която се стартира със стартирането на процеса. Отделните нишки в процеса също имат свой собствен приоритет. Йерархията се определя от изпълняваните компоненти и тяхното текущо състояние. Компонентите, които са най-ниско в йерархията се елиминират първи. Според състоянието на компонентите на приложението и приоритета на процесите има пет равнища на йерархията.

- *Активни (foreground) процеси* - изисква се от текущите действия на потребителя. Поддържа активно взаимодействие с потребителя. Прекратява се само в изключителни случаи:
- *„Видими" (Visible) процеси* - те нямат foreground (основни) компоненти, но са видими за потребителя.
- *Процеси - „услуги" (service processes - услуги)* - изпълнява се услуга, която не попада в горните две категории. *Сървисните процеси* не са свързани с визуализация. Те правят това, което потребителят изисква (музикален плеър, сваляне на данни). Системата запазва такива процеси, освен ако не е необходимо да се освободи памет за процеси от по-горните приоритетни равнища.
- *Фонови (Background) процеси* - текущо не видими за потребителя. Тези процеси нямат директно въздействие върху работата на потребителя. Системата ги прекратява всеки път, когато е необходимо да се освободи памет за foreground, visible или service процеси.
- *„Празни" (Empty) процеси* - Процес, който не съдържа никакви компоненти на приложението. Причина да се поддържа "жив" такъв процес е да се ускори времето за стартиране на следващия компонент (процес).

Приложенията в Android се изпълняват в изолирани процеси. За случаите, когато е необходимо да комуникират помежду си чрез обмен на данни или да ползват системните услуги ползва се механизъм "InterProcess Communication".

Организация на адресното пространство

Адресно пространство при ОС Android се разделя на две отделни области:

- потребителско пространство (User Space) - разполагат се и изпълняват потребителските програми и данни. Потребителските процеси се изпълняват в непривилегирован режим - нямат непосредствен достъп до физическата памет или до устройствата;

- пространство на ядрото (Kernel Space) - разполага се и изпълнява кода на ядрото. Разположените в пространството на ядрото процеси имат пълен достъп до физическата памет и до устройствата и се изпълняват в „привилегирован" режим - „режим на ядрото" (kernel mode).

Основни принципи на сигурността в OS Android

Най-важният принцип в подхода на сигурността, приет в Android е да не се допусне случайно или злонамерено увреждане на системни ресурси и на потребителски приложения и данни.

За постигане на сигурност се извършва разделяне на паметта на устройството на два дяла:

- съхранение на системните компоненти;
- съхранение на данни (включително потребителски).

Системният дял се монтира като достъпен само за четене (*read-only*). „Потребителският" дял е мястото, където се запазват приложенията и поддържаните данни в системата.

Android използва многослойната организация на ОС, механизмът на сигурността се реализира на две равнища:

- равнището на ядрото на Linux;
- равнището на слоя Application Framework.

В първия случай всяко приложение се изпълнява в специфична област (Application Sandbox). Ядрото изолира компонентите на приложенията и на ОС. Постига се чрез присвояване на индивидуален идентификатор на всяко приложение. Така то се изпълнява в собствен Linux-процес, напълно изолиран от работата на другите процеси в системата, с ограничен достъп до функциите и услугите на ОС.

Достъпът до файл се определя от създателя или от собственика на ресурса и третира правата на три група потребители:

- самият собственик на файла;
- потребителите от същата група;
- всички останали потребители.

За всеки тип потребители се установяват група от права за достъп във вид „четене-запис-изпълнение" (r-w-x).

Основни заплахи за мобилната сигурност през 2023 г.

Потребителите на настолни компютри се излагат на значително по-нисък риск от измама. Предизвикателство по отношение на мобилните устройства е тяхната свързаност с други услуги и устройства. След като смартфон бъде компрометиран, всяко друго устройство или услуга, свързана с това конкретно мобилно устройство, е изложено на риск. Основните видове заплахи за мобилната сигурност са [3], [4], [5], [7], [10]:

1. Заплахи от мобилните приложения

Мобилните приложения са основна причина, поради която мобилните устройства се оказват уязвими към набор от заплахи. Основна причина за причиняване на много щети е, че *нищо неподозиращият потребител не знае за атаките*. Заплахите от тях са:

- приложенията със съмнителни практики за сигурност са с некриптирани данни. Чувствителна информация и данни за кредитни карти, представляват проблем за сигурността в мобилната система. Повечето от тях не са разработени с първостепенно значение за сигурността. Натискът върху по-бързото излизане на пазара, за да стартира новото мобилно приложение, води до пренебрегване на сигурността;
- злонамерени приложения - обикновено предлагат сделки, които са твърде добри, за да са истина. Другата опция е имитиране на външния вид на легитимни приложения, които имат много потребители. Вместо да получат добра сделка, потребителите в крайна сметка изтеглят вирус, способен да заключи телефоните им и да присвои данните на устройството. За да се предотвратят подобни атаки, приложенията да се изтеглят от доверени сървъри за приложения;
- IoT (Internet of Things) устройства имат пропуски в сигурността още при проектиране. Такива устройства се свързват със смартфона чрез Wi-Fi или Bluetooth, оставяйки отворени врати за атака. Всяко IoT устройство излага на риск цялата мрежа, към която е свързано;
- изтичане на данни чрез приложения. Потребителите на смартфон имат инсталирани около 80 инсталирани приложения на мобилното си устройство – лични и служебни. Заплахите свързани с мобилните приложения са огромни. Например разрешаване на достъп до галерията на устройството, причинява пробиви на данни;
- Drive-by се отнасят за инсталиране на зловреден софтуер на мобилно устройство без разрешение на потребителят. Извършва се при посещение на подозрителен сайт или отваряне на грешен имейл, инсталацията се извършва автоматично. В инсталирания файл може да присъства шпионски софтуер, рекламен софтуер или зловреден софтуер. Ситуацията става по-сериозна, ако файлът съдържа бот, който може да изпълнява злонамерени задачи на телефона.

За Android мобилни устройства е най-безопасно да се ползва за инсталация на приложенията Google play. Там те се проверяват предварително, преди да се предоставят на потребителя.

2. Физически заплахи от мобилно устройство

Те се отнасят до изгубено или откраднато мобилно устройство. Избягването на всякакъв вид мерки за сигурност под формата на ПИН или биометрични данни прави по-лесен достъпа до личните данни. Директният достъп до мобилен хардуер е голям риск за потребителите.

Имайки това предвид, отговорност на компанията е да осигури допълнителен слой защита под формата на VPN (virtual private network) и двуфакторна автентификация.

3. Заплахи в мобилната мрежа

Използването на обществени Wi-Fi мрежи е особено опасно поради възможността за атаки в тях:

- чрез фалшива Wi-Fi точка на достъп създадена от хакери, потребителите изпращат своите идентификационни данни, имена и пароли. Така се извършва насочване към лични акаунти и причиняване на щети;
- атаките *Man-in-the-Middle* включват прихващане на мрежата, последвано от промяна на данните при трансфер или от подслушване. Мобилните устройства са особено уязвими към този вид атаки, тъй като лесно се прихващат SMS комуникация поради липсата на протокол за сигурност около тях. Мобилните приложения, използващи некриптиран HTTP за прехвърляне на чувствителна информация, също представляват голям риск;
- мобилният *ransomware* е основна заплаха. Основава се на криптиране на файловете на мобилния телефон и искане на собственика да извърши плащане, за да бъдат отключени файловете;
- Grayware приложенията (сив софтуер) са изскачащи прозорци или безкрайни пренасочвания. Те биха могли да съдържат шпионски и рекламен софтуер;
- атаките чрез социално инженерство разчитат на фишинг измами. Изпращат се имейли или незабавни съобщения, със злонамерени връзки, или се изискват потребителски идентификационни данни. Съобщенията включват сделки, които са твърде добри, за да се отхвърлят, спешни въпроси, изискващи достъп до акаунт или лична /финансова информация и т.н.;
- когато се пренебрегват системните актуализации, потребителите работят с версии, които съдържат недостатъци и се превръщат в лесна мишена за нападателя;
- пропуските в криптирането оставят потоците от данни незащитени и следователно уязвими за атаки.

4. УЕБ базирани мобилни заплахи

Те са под формата на фишинг имейли и спуфинг (смяна на IP адрес на компютър в мрежата за прикриване на идентичността му).

- фишингът е основна причина за повечето днешни атаки. Мобилните устройства са уязвими за фишинг измами, поради разнообразието от канали за разпространение на зловреден софтуер - имейл, SMS,

платформи за незабавни съобщения, злонамерени реклами, социални медии и др.

- шпионският софтуер обикновено се активира от злонамерена връзка или реклама. Той сканира мобилното устройство за чувствителни данни, потребителски идентификационни данни и всякакъв вид информация полезна за атакуващия.
- Madware е мобилен рекламен софтуер (скрипт). Той преглежда телефона за използване на интернет и интереси. Тази информация по-късно се продава на рекламни компании. Проблемът е, че това се случва без разрешение. Още по-голям проблем е фактът, че в обработените данни може да се намери точното местоположение, списък с контакти, в някои случаи цели галерии и бележки, съхранени на мобилното устройство.
- браузър експлойтите работят върху уязвимости, присъстващи в мобилния браузър, като се възползват от тях. Всяка промяна във външния вид на мобилния браузър указва, че е жертва на атака.
- лоша хигиена на паролата. С много услуги и устройства често се избира една и съща парола. Освен това, потребителите често избират общи пароли. Служителите използват смартфони за достъп до работни и до лични акаунти, застрашавайки себе си и цялата компания.

Уязвимости

Мобилните приложения, злонамерени или доверени, имат голям потенциал да окажат отрицателно въздействие върху сигурността и поверителността на потребителите. Злонамерено приложение може да съдържа код, предназначен да използва уязвимости, присъстващи потенциално във всеки целеви хардуер, фърмуер или софтуер на устройството. Като алтернатива или във връзка с код, злонамерено приложение може да злоупотреби с всяко устройство, лични или поведенчески данни, до които изрично или косвено му е предоставен достъп, като контакти, данни от клипборда или услуги за местоположение. Доверените приложения могат да имат уязвимости или слабости, които злонамерените такива могат да използват. Освен това, те могат да изложат на риск поверителността на потребителя, като събират повече информация, отколкото е необходимо, за да се предостави желаната от потребителя функционалност [5].

Уязвимостите на приложенията за Android са проблем поради отворения му код, а също и защото потребителите могат да зареждат приложения от различни източници. Тестването на мобилни приложения за Android показва, че несигурното съхранение на данни е най-честият пропуск в сигурността на приложенията за Android. Уязвимостите причиняват загуба на данни, споделяне на лична информация и други. Основните уязвимости в ОС Android са [1], [5], [8], [9]:

- уязвимост в двоичната защита/слабост в Root. Руутването е процес, чрез който се получава достъп до кода на операционната система Android. Jailbreaking е премахване на софтуерни ограничения, които са поставени от производителя на устройството. Руутването и/или Jailbreaking заобикалят защитата на данните и схемите за криптиране в системата.

Когато дадено устройство е било компрометирано, всяка форма на злонамерен код може да се изпълни на него. Това може значително да промени планираното поведение на логиката на приложенията;

- недостатъчна защита на транспортния слой - приложенията и ОС често не криптират мрежовия трафик, когато е необходимо да се защитят чувствителни комуникации;
- уязвимост в оторизацията/удостоверяването – получава се, когато дадено приложение не извършва адекватни проверки за оторизация, за да се увери в легитимността на потребителя;
- уязвимост в криптографията/неправилно валидиране на сертификати – приложенията не валидират входящите данни от Secure Socket Layer (SSL) и Transport Layer Security (TLS) протоколи. Те осигуряват криптиране на връзката между сървър (уебсайт) и браузър - потребител;
- уязвимости свързани в Brute Force атаки - това е метод за намиране на неизвестна стойност, заобикаляйки автоматизиран процес, за да се проверят голям брой възможни стойности;
- уязвимости свързани с изтичане на сесията - след като потребителят излезе от приложение, идентификаторите на процесите, които са били използвани по време на сесията, се предполага, че са невалидни. Ако сървърът (ОС) не успее да анулира идентификаторите на сесията, възможно е други потребители да ги използват;
- уязвимости свързани с изтичане на информация и кеш на приложенията - чувствителни данни могат да изтекат от кешовете на приложенията или чрез основния код на приложението.

В [6] са дадени известни експлоатирани уязвимости, включително и на ОС Android. Мобилните устройства представляват уникално предизвикателство по отношение на сигурното съхранение на данни.

Защити срещу кибератаки

Заплахите срещу сигурността на Android мобилни устройства и техните приложения могат да се предотвратяват чрез група приложения като *App Protector*[1]. За да предотврати и открие заплахи за мобилната сигурност в реално време, App Protector се интегрира с приложенията и ги защитава отвътре, според зададената конфигурация.

Друг инструмент за защита е *BullGuard Mobile security*[2]. Включва антивирусна защита, защитна стена, родителски контрол и антиспам.

Wiseid[3] е инструмент за информационна сигурност на мобилните устройства.

На адрес[4] са дадени мобилни приложения за защита и същите са сравнени.

[1] App Protector - Cybersecurity ASEE, https://cybersecurity.asee.co/app-protector/ (last view: 24-03-2023)
[2] Download BullGuard Mobile Security…, https://bullguard-mobile-security.en.uptodown.com/android/download; BullGuard 2021, https://www.bullguard.com/ (last view: 24-03-2023)
[3] WISeID - Protect the data and more…, https://wiseid.com/; WISeID Mobile App Ecosystem – WISeKey, https://www.wisekey.com/products-services/digital-identity-pki/wiseid-mobile-app-ecosystem/ (last view: 24-03-2023)
[4] Lookout Reviews, Ratings & Features…, https://www.gartner.com/reviews/market/mobile-threat-defense/vendor/lookout?marketSeoName=mobile-threat-defense&vendorSeoName=lookout (last view: 24-03-2023)

Aircrack-ng[5] [9] е мрежов софтуерен пакет за анализ на безжични мрежи. Работи в различни области на Wi-Fi сигурността:

- мониторинг - улавяне на пакети и експортиране на данни в текстови файлове за по-нататъшна обработка от инструменти;
- атакуване - повторни атаки, деавтентификация, фалшиви точки за достъп и други чрез инжектиране на пакети;
- тестване - проверка на Wi-Fi карти и възможности на драйвера (улавяне и инжектиране)
- кракване: WEP (Wired Equivalent Privacy) и WPA PSK (WPA 1 и 2, Wi-Fi Protected Access, Phase-Shift Keying).

Използва се от команден ред. Работи под Linux, FreeBSD, macOS, OpenBSD, Solaris и Windows. Също така е пренесена към платформите Android, Zaurus PDA и Maemo. Освен това предварително е инсталиран като инструмент в Linux дистрибуциите Kali, Parrot и други насочени върху сигурността и разработени в рамките на Debian.

Всяка отделна програма от "*Wireless Attacks*" раздела на Kali Linux предоставя възможност за работа с безжичните мрежи. Необходимо е да се провеждат експерименти и да се подхожда внимателно, понеже някои атаки е възможно да бъдат засечени от специални системи, анализиращи трафика пренасян през WLAN.

Заключение

В статията е извършен преглед на киберсигурността на мобилни устройства с Android. Обхванати и разяснени в тяхната цялост съобразно обема на статията са архитектура на ОС Android, нейната структура, основни принципи на сигурността в ОС Android, основни заплахи за мобилната сигурност за 2023 г., уязвимости и защита срещу кибератаки в ОС Android. Указани са множество линкове, които биха могли да се ползват за защита на мобилни устройства с Android и представляват интерес за практически изследвания.

В статията не са засегнати протоколите за защита, тяхното използване, поведение при атаки и резултати след атаките. Те са едно бъдещо поле за изследване в тази област. Освен това анализ на функциите на мобилни приложения за защита и тяхното практическо проявление за 4G и 5G мрежите е също така област за изследване.

References // Литература

[1] Ahmed, O.M.; Sallow, A.B. (2017). "Android Security: A Review", Academic Journal of Nawroz University, 6(3), 135–140, August 2017. DOI: https://doi.org/10.25007/ajnu.v6n3a99

[2] Android, (n. d.). "Android - Secure & Reliable Mobile Operating System", https://www.android.com (last view: 24-03-2023)

[5] Aircrack-ng, https://www.aircrack-ng.org (last view: 24-03-2023)

[3] Brown, C.; Dog, S.; Franklin, J.M.; McNab, N; Voss-Northrop, S.; Peck, M.; Stidham, B. (2016). "Assessing Threats to Mobile Devices & Infrastructure", National Institute of Standards and Technology, September 2016.

[4] Check Point Software Technologies Ltd. (n. d.). "The top 4 cyber security threats to android mobile devices", www.checkpoint.com, March 30, 2015.

[5] Chin, A.; Jones, B.; Little, P. (2021). "A Comparative Analysis of Smartphone Security Behaviors and Practices", International Journal of Education and Development using Information and Communication Technology (IJEDICT), 2021, Vol. 17, Issue 3, pp. 57-80, 2021.

[6] CISA, (n. d.). "Known Exploited Vulnerabilities Catalog | CISA", https://www.cisa.gov/known-exploited-vulnerabilities-catalog (last view: 24-03-2023)

[7] Cybersecurity ASEE, (2023). "Top mobile security threats and prevention tips (2023) - Cybersecurity ASEE" https://cybersecurity.asee.co/blog/top-mobile-security-threats-prevention-tips/ (last view: 24-03-2023)

[8] Franklin, J.M.; Howell, G.; Boeckl, K.; Lefkovitz, N.; Nadeau, E.; Shariati, B.; Ajmo, J.G.; Brown, C.J.; Dog, S.E.; Javar, F.; Peck, M.; Sandlin, K.F. (2020). "NIST SPECIAL PUBLICATION 1800-21, Mobile Device Security: Corporate-Owned Personally-Enabled (COPE)", September 2020. DOI: https://doi.org/10.6028/NIST.SP.1800-21

[9] NC State University, (n. d.). "Securing Your Wireless Connections - Office of Information Technology", https://oit.ncsu.edu/it-security/mobile/android/connections (last view: 24-03-2023)

[10] Shterev, Y. (2022). "Concepts of Cyber Security", Science Series "Innovative STEM Education", volume 04, ISSN: 2683-1333, Institute of Mathematics and Informatics – Bulgarian Academy of Sciences, pp. 79-88, 2022. DOI: https://doi.org/10.55630/STEM.2022.0411

[11] Statcounter Global Stats, (n. d.). "Mobile Operating System Market Share Worldwide | Statcounter Global Stats", https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201601-201601-map (last view: 24-03-2023)

Received: 30-03-2023 Accepted: 29-06-2023 Published: 24-07-2023
Cite as:
Shterev, Y.; Kolev, K. (2023). "Cyber Security to Android Mobile Devices", Science Series "Innovative STEM Education", volume 05, ISSN: 2683-1333, pp. 129-141, 2023. DOI: https://doi.org/10.55630/STEM.2023.0516