PLATFORMA SOFTWARE INTEGRATĂ PENTRU ANALIZA MALWARE A TERMINALELOR MOBILE

INTEGRATED SOFTWARE PLATFORM FOR MALWARE ANALYSIS OF MOBILE TERMINALS

PLATE-FORME LOGICIELLE INTÉGRÉE POUR L'ANALYSE MALWARE DES TERMINAUX MOBILES

Lt.col.dr.ing. Dragoș BĂRBIERU*
Col.dr. Ștefan-Antonio Dan ȘUTEU**
Conf.univ.dr. Elena ȘUȘNEA***

Dincolo de marketingul companiilor IT, în contextul escaladării atacurilor cibernetice, care afectează organizațiile din întreaga lume, soluțiile de securitate cibernetică devin elementul principal în protejarea infrastructurilor și dispozitivelor IT. Diversitatea dispozitivelor mobile inteligente și apariția tehnologiilor cloud, Internet of Things necesită noi soluții tehnologice, implementate atât la nivel hardware, cât și la nivel software în scopul combaterii amenințărilor.

Acest articol prezintă rezultatele parțiale din proiectul de cercetare care are ca obiectiv realizarea platformei software integrate pentru analiza programelor malware ale terminalelor mobile. Platforma integrează diverse tehnologii software pentru protejarea dispozitivelor mobile.

Beyond the marketing of IT companies, în the context of escalating cyber-attacks that affect organizations around the world, cyber security solutions become the primary element în protecting IT infrastructures and devices. The proliferation of Intelligent Mobile Devices and Cloud Technologies, the Internet of Things requires new technological solutions, implemented both at hardware and software, to combat threats.

This paper summarizes the Integrated Software Platform for Malware Analysis of Mobile Terminals which aims to integrate various software technologies to protect mobile devices.

Dans le contexte d'une augmentation d'attaques informatiques qui touchent les organisations du monde entier, les solutions de cybersécurité, au-delà du marketing des entreprises de TI, deviennent l'élément principal de la protection des infrastructures et des dispositifs TI. La diversité des appareils mobiles intelligents et l'émergence des technologies de cloud computing, Internet of Things exigent, pour lutter contre les menaces, de nouvelles solutions technologiques, mises en œuvre tant au niveau du hardware, que du software.

L'article présente les résultats partiels lors du projet de recherche dont le but est la création d'une plate-forme logicielle intégrée pour l'analyse des programmes malware des terminaux mobiles. La plate-forme intègre une variété de technologies logicielles pour protéger les appareils mobiles.

Cuvinte-cheie: analiză malware; securitate cibernetică; terminale mobile.
Keywords: *malware analysis; cyber security; mobile terminal.*
Mots-clés: *analyse malware; cybersécurité; terminaux mobiles.*

Universitatea Națională de Apărare „Carol I”
e-mail: *dragos.barbieru@adlunap.ro*
****Universitatea Națională de Apărare „Carol I”***
e-mail: *dan-suteu.antonio@unap.ro*
*****Universitatea Națională de Apărare „Carol I”***
e-mail: *esusnea@yahoo.com*

Analiza aplicațiilor malware destinată terminalelor mobile este un proces dificil, din cauza diversității platformelor mobile și a mecanismelor de securitate existente, frecvenței de apariție a noilor versiuni ale sistemelor de operare și utilizării tehnicilor de protecție a codului malware. În contextul situației naționale și internaționale, modelată de tendințele din domeniul securității, s-a simțit nevoia realizării unei platforme software care să integreze, într-un mod unitar, diferite soluții de analiză malware atât open-source, cât și comerciale, dedicate telefoniei mobile. Majoritatea actorilor din spațiul cibernetic se adaptează mediului existent, dar supremația informațională și tehnologică se obține prin inovare, așa cum susține Vice Admiral Arthur K. Cebrowski: ”I realized that military competition wasn’t about how fast one could align with reality, but how fast one could leap over it and create a new reality”[1].

Securitatea și securitatea cibernetică sunt într-o strânsă legătură, din securitate izvorând majoritatea metodelor și tehnicilor de atac și de apărare în mediul cibernetic.

Cele șapte etape ale modelului Cyber Kill Chain[2], prezentate de corporația Lockheed-Martin, sunt identice cu etapele unui atac elaborat asupra unei persoane sau unui grup de persoane. Analiza statică a aplicațiilor malware poate fi comparată cu o investigație privind stabilirea unui profil psihologic al unei persoane. Deși este mult mai rentabilă economic față de analiza dinamică, în cazul acestei analize, un program poate ascunde cod malware prin criptare sau prin diferite alte metode, la fel cum o persoană poate completa cu date false un chestionar privind personalitatea sa.

Analiza dinamică presupune execuția unui program și urmărirea tuturor parametrilor pentru a identifica activitățile suspecte, într-un mediu controlat.

Conceptul *honeypot* (borcan cu miere) și tehnicile utilizate pentru verificarea siguranței mediului în care acesta se manifestă au corespondențe în viața reală, cum ar fi, de exemplu, persoana care se află sub lupa unui detector într-un mediu sigur sau proiectat ca fiind aparent sigur de către cel care dorește să urmărească anumite evenimente. Atacurile de tipul ”distributed denial of services” sunt similare intoxicării unui adversar cu informații false, acesta consumând timp și resurse până la epuizare. Dezvoltarea rapidă a tehnologiei informației și comunicațiilor și „accesul ușor la Internet nu numai că au adus beneficii incontestabile, dar, de asemenea, aduc unele vulnerabilități mediului de securitate”[3]. Războiul hibrid, dus prin diferiți terți, se oglindește, astăzi, în lumea Internetului prin utilizarea diferitelor tehnologii proxy și a grupărilor specializate de hacking.

Atât în cazul tehnologiilor actuale, cât și în cazul celor viitoare, se vor putea identifica tipare, care, deși sunt într-un număr limitat, modurile de manifestare sunt inepuizabile. Aceste tipare nu sunt caracteristice prezentului, ci își au rădăcinile în istoria speciei noastre și reprezintă forme de atac și apărare, multe dintre ele împrumutate din biologie. Camuflajul și mimetismul sunt arme din arsenalul animalelor și pot asigura victoria împotriva unui posibil adversar[4]. Considerăm că, în spațiul cibernetic, unde manifestările de intruziune și de protecție sunt mult mai diversificate, acționează un set limitat de tipare pe care le regăsim și în biologie, acestea fiind rezultatul unui lung proces de evoluție.

Aplicațiile malware utilizează diferite tehnici de camuflaj. Acestea pot fi instalate în lanțul de distribuție, astfel un utilizator nu va putea observa nicio modificare a activității dispozitivului, care apare adesea după instalarea unui program. Compromiterea procesorului de semnal duce inevitabil la interceptarea apelurilor telefonice și a mesajelor, însă, prin folosirea corelărilor existente între DSP și CPU, atacatorii pot obține capabilități extinse asupra aplicațiilor care rulează pe terminalul mobil. Prin oferirea de aplicații gratuite sau de aplicații din magazinele neoficiale, persoanele rău intenționate pot insera cod malware. Procedura de tipul „control flow obfuscation” împiedică analiza dinamică a aplicațiilor malware. Utilizarea algoritmilor de criptare va conduce la imposibilitatea de a dezasambla și de a decompila codul unei aplicații.

Detectarea comportamentului malițios al terminalelor mobile presupune trei tipuri de analiză: analiza statică, analiza dinamică și analiza hibridă. Analiza statică presupune dezasamblarea și decompilarea unei aplicații pentru a identifica un cod malware. Analiza dinamică urmărește diferiți parametri și evenimente într-un mediu controlat de tip *sandbox*, pentru a identifica acele comportamente suspecte. Analiza hibridă combină cele două tipuri

de analiză prezentate succint anterior. Detectarea codului malware, de regulă, implică existența unei liste de semnături, dar în cazul în care acest proces eșuează, se pot utiliza algoritmi de inteligență artificială sau se poate realiza o analiză manuală. Din perspectiva învățării automate, abordarea comportamentului malițios are în vedere o serie de etape, precum: „alegerea setului inițial de date (set de antrenament), de regulă, un număr egal de aplicații sigure și de aplicații malware din care se extrag anumite caracteristici”[5].

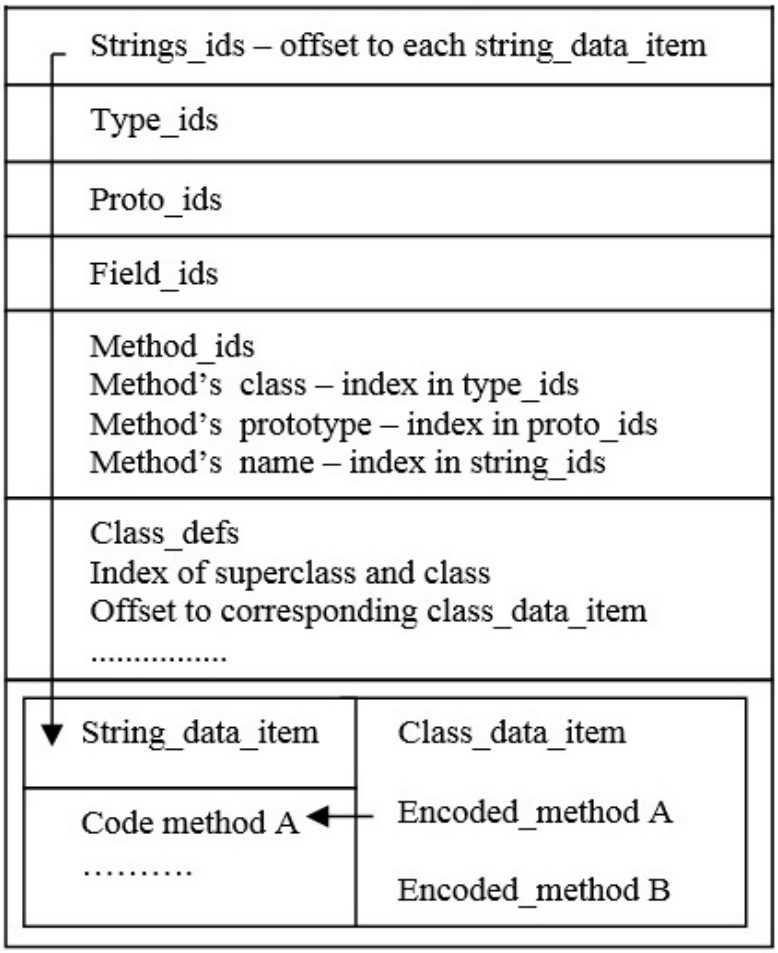

Figura 1 *Structura unui fișier cu extensia .dex*

Sursa: Xiaolu Zhong, Frank Bratinger, Ibrahim Baggali, ”Rapid Android Parser for Investigating DEX files”, *Digital Investigation Journal*, vol. 17, June 2016.

Aplicând metode de selecție a caracteristicilor și algoritmi de clasificare, se obțin caracteristicile cele mai relevante pentru construirea modelului. În faza de testare, se utilizează diferite metrici pentru a evalua acuratețea modelului. Alegerea caracteristicilor nu este un proces aleatoriu. Gama algoritmilor de clasificare este diversă, abordările bazându-se pe statistică, pe rețele neuronale artificiale și pe metode axate pe nucleu. Cei mai folosiți algoritmi de clasificare sunt Naive Bayes, k- Nearest Neighbors și Suport Vector Machine.

Vulnerabilitățile pot fi de două tipuri, preinstalate sau generate de complexitatea mediului Internet. Este aproape imposibil să poți verifica și testa fiecare bucată de cod.

Platformele de dezvoltare software, cum ar fi GitHub, ar putea oferi în viitor instrumente pentru verificarea diferitelor erori apărute în cod, evitându-se astfel exploatarea lor de către atacatori.

Metodele pentru ascunderea codului malware sunt diverse și depind de facilitățile sistemului de operare al terminalului mobil[6]. De exemplu, structura unui fișier (Fig. 1) cu extensia .dex are zone alocate pentru Header, String_ids, Type_ids, Proto_ids, Fields, Methods, Classes and Data.

O clasă se regăsește în matricea *class_defs* sub forma unui index care pointează către un alt index din matricea *strings_ids*, acesta din urmă fiind conectat cu *string_data_item*, care poate returna numele clasei. Fiecare clasă definită în zona de cod este descrisă de o structură *class_data_item* care conține variabilele și metodele ei. Metodele sunt declarate sub forma unei structuri cu numele *encoded_method*. Această structură se compune din: *acces flag* – specifică cum este metoda (publică, privată, protejată etc.), *offset code* – indică adresa unde se găsește codul metodei față de începutul fișierului .dex și *method_idx_diff* – constituie un index de incrementare pentru fiecare metodă din structura *method_ids*. Primul pas pentru ascunderea unei metode constă în manipularea structurii *encoded_method* pentru a referenția o altă metodă, recalcularea sumei de control SHA1, modificarea headerului fișierului .dex și împachetarea aplicației. Manipularea structurii *encode_method* presupune o valoare pentru *method_idx_diff*, care poate fi 0 și modificarea adresei care accesează codul metodei.

Tehnica camuflajului utilizată în spațiul cibernetic poate fi recunoscută prin metode precum: criptare, oligomorfism, polymorfism și metamorpfsm[7]. Aplicațiile malware de tipul semi-polimorfic sau oligomorfic utilizează diferiți algoritmi de criptare la fiecare infecție. Diferența majoră dintre oligomorphism și polimorfism este că acesta din urmă poate utiliza un număr nelimitat de algoritmi de criptare. Metamorfismul schimbă complet codul aplicației malware și nu utilizează algoritmi de criptare. Tehnicile de mimetism se pot identifica în metodele de obfuscation a codului. Cele mai comune metode de obfuscation sunt utilizarea unui junk code (cod gunoi), substituția de variabile și regiștri, permutarea și înlocuirea instrucțiunilor, transpoziția codului și bucle infinite. Atunci când aplicațiile malware rezidă în componentele hardware, procesul de analiză este mult mai dificil.

Platforma software integrată pentru analiza malware a terminalelor mobile

Dezvoltarea platformei software pentru analiza malware a terminalelor mobile constă în parcurgerea unei succesiuni de etape. Astfel, prima etapă presupune identificarea modalităților uzuale și mai puțin uzuale de infectare a terminalelor mobile cu ajutorul diferitelor rapoarte de securitate și definirea unei taxonomii după anumite caracteristici, cum ar fi: vectori de atac, sursă, obiective, vulnerabilitate exploatată, tip amenințare etc.

Următoarea etapă constă în testarea aplicațiilor open-source sau comerciale pentru alegerea soluțiilor care să satisfacă cerințele de securitate. În acest sens, sunt studiate diferite proiecte de cercetare și lucrări științifice, care se referă la detectarea comportamentului suspicios și este propus un concept tehnic de firmware customizat pentru îmbunătățirea sistemului de operare. Printre soluțiile testate menționăm: Cellebrite UFED Pro Series, Cellebrite UFED Field, Cellebrite UFED Analytics, Oxygen Forensics, BlackBag Technologies, Forensic Toolkit, EnCase Forensic Software, Belkasoft Evidence Center, Autopsy, Computer Aided INvestigative Environment, Mobile security testing live environment, MOBILedit etc. Detecția aplicațiilor malware a impus instalarea și/sau verificarea unor framework-uri, cum ar fi: MODELZ[8], Andromaly[9], MADAM[10], ComDroid[11], ProfileDroid[12]. Aceste framework-uri analizate utilizează diferite caracteristici ale terminalelor mobile. De exemplu, MODELZ analizează puterea consumată de baterie atunci când rulează diferite aplicații și pe baza acestei caracteristici identifică o semnătură. În opinia noastră, principalul dezavantaj al acestei analize este necesitatea implementării unui dispozitiv extern care să achiziționeze istoricul consumului de energie într-un mod precis. În acest sens, pe perioada testării se utilizează un osciloscop extern, Agilent Infinium 54851-A, iar având în vedere rezultatele obținute, propunem construirea unui circuit extern ieftin bazat pe un microcontroler Atmel AVR.

Un alt framework utilizat este Andromaly, care necesită instalarea unei aplicații pe dispozitivul mobil pentru monitorizarea unor parametri, cum ar fi consumul CPU, numărul de pachete trimise prin Wi-Fi, numărul de procese care rulează, nivelul bateriei. Pe baza datelor colectate, sunt deduse informațiile referitoare la funcționarea normală a dispozitivului. Numărul maxim de parametri care pot fi monitorizați este de optzeci și opt.

Utilizarea algoritmilor de inteligență artificială pentru clasificare pe un număr mare de caracteristici extrase din terminalul mobil, unele dintre ele redundante sau irelevante, generează mai multe probleme, cum ar fi: infectarea algoritmului de învățare, suprasolicitarea, reducerea generalității, creșterea complexității modelului și a timpului de execuție. În opinia noastră, aplicația care implementează algoritmii de clasificare nu trebuie sa ruleze pe dispozitivele mobile, deoarece acestea sunt adesea restricționate de capacitățile de stocare și de prelucrare a datelor, precum și de puterea bateriei.

Procesul detectării comportamentului malițios devine anevoios, când unele activități malițioase sunt de scurtă durată și nu oferă date suficiente pentru detecție sau pentru antrenarea modelului. Prin urmare, nu există posibilitatea de a accesa un număr mare de baze de date cu aplicații rău intenționate pentru a crește acuratețea algoritmilor. Comportamentul malițios al unei aplicații poate fi generat de mai mulți vectori de atac, astfel clasificarea devine dificil de realizat, iar numărul mic de aplicații malware folosite ca date de intrare generează un dezechilibru. Frameworkul MADAM, deși folosește treisprezece caracteristici și a fost testat pe dispozitive mobile reale, are dezavantajul că impune drepturi de administrare asupra sistemului de operare. Acest framework

monitorizează apelurile de sistem, procesele care rulează, memoria și nivelul de utilizare a procesorului, numerele de telefon apelate, funcționarea Bluetooth și Wi-fi, mesajele SMS primite sau recepționate, perioadele de inactivitate și activitate, apăsarea tastelor.

Frameworkul Droid Detective[13] propune, pentru detectarea aplicațiilor malware, o analiză pe baza grupării permisiunilor. După extragerea permisiunilor, se va calcula frecvența de apariție a acestora când sunt grupate (gruparea permisiunilor pornește cu o permisiune și continuă până la un grup

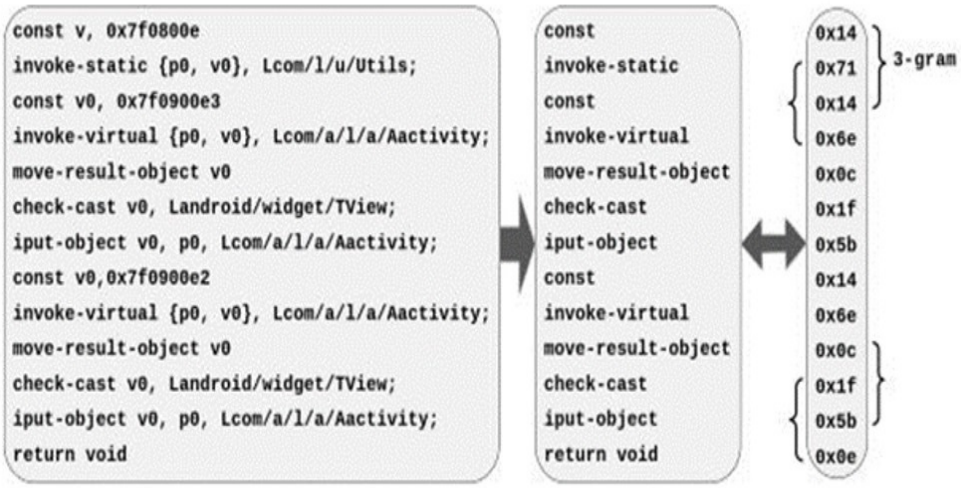

Figura 2 *Extragerea instrucțiunilor dintr-un fișier .smali și generarea vectorului 3-gram*

Sursa: BooJoong Kang, Suleiman Y. Yerima, Sakir Sezer, Kieran McLaughlin, *International Journal on Cyber Situational Awareness,* vol. 1, No. 1, 2016.

de șase permisiuni) atât pentru aplicațiile sigure, cât și pentru aplicațiile malware. Grupul de permisiuni care indică un comportament malițios este identificat pentru aplicațiile care folosesc caracteristicile: ACCESS_NETWORK, INTERNET, READ_PHONE_STATE, READ_SMS și WRITE_SMS. O serie de autori[14] propun utilizarea mai multor algoritmi de clasificare pentru a îmbunătăți acuratețea detecției de malware. Se utilizează mai multe seturi de caracteristici în faza de învățare, cum ar fi: funcții API, permisiuni și comenzi ale sistemului de operare. Algoritmii utilizați sunt: Decision Tree, Simple Logistic, Naive Bayes, Partial Decision Tree și Ripple Down Rule. Numărul total de caracteristici selectate este de 179, din care 125 de permisiuni și 54 de funcții API și comenzi OS. Ca set de date de intrare, se folosește baza de date McAfee cu 2.925 de aplicații malware și 3.938 de aplicații sigure. Pentru evaluarea performanțelor algoritmilor de clasificare, se utilizează metoda 10-fold cross validation (presupune împărțirea setului de date inițial în zece părți, antrenarea pe nouă dintre ele și testarea pe unul, repetarea procedurii și verificarea acurateței). Clasificatorii RIDOR și PART au cea mai bună rată de detecție. Această abordare este completă, deoarece utilizează seturi de caracteristici diferite simultan cu algoritmi de clasificare variați. Nu se specifică modul de selecție a caracteristicilor relevante.

O abordare interesantă[15] este analiza permisiunilor cerute de aplicație în timpul execuției și cele existente în fișierul manifest. O permisiune care nu este cerută în faza inițială, poate fi cerută utilizatorului mai târziu. Ideea este că poate exista o diferență între permisiunile cerute și cele folosite de aplicație. Concluzia este că aplicațiile malware cer mai multe permisiuni decât aplicațiile sigure.

Există posibilitatea construirii unui clasificator pe baza setului de instrucțiuni de nivel jos, utilizând modelul N-gram[16]. Ca procedură de lucru, se dezasamblează aplicația pentru a genera fișiere de tip .smali. Fiecare fișier conține o clasă cu metodele aferente în formatul Dalvik bytecode. Dezasamblarea unei aplicații se face cu utilitarul apktool (Fig. 2). Din fișierele rezultate, se extrag instrucțiunile din fiecare metodă într-un șir și se calculează frecvențele lor de apariție. Fiecare instrucțiune în format Dalvik bytecode are dimensiunea de 1 byte. Numărul de instrucțiuni este de 256 la puterea 130, din care sunt folosite 218 instrucțiuni. Există 218 la puterea n posibilități de a aranja aceste instrucțiuni.

Numărul unic de n-opcodes se calculează după formula:

N = X – (N - 1),

unde X este numărul de instrucțiuni din aplicație și N reprezintă numărul de instrucțiuni dintr-o pereche. Astfel, o metodă cu 10 instrucțiuni are 10 perechi de câte o instrucțiune, nouă perechi de două instrucțiuni, opt perechi de trei instrucțiuni etc.

Clasificarea aplicațiilor malware se poate face și după un set redus de instrucțiuni[17], respectiv șase instrucțiuni. Acestea pot fi: move, jump, packed-switch, sparse-switch, invoke și if. Premisa inițială pleacă de la două întrebări: „Caracteristicile alese sunt în măsură să facă distincția între aplicațiile malware și cele sigure?”, respectiv, „Combinarea caracteristicilor alese aduce plusvaloare comparativ cu cazul în care acestea sunt tratate individual în analiza malware?”. Contribuția științifică se poate rezuma la unicitatea caracteristicilor alese cu rezultate bune, folosind resurse puține pentru analiza malware. Diferența semnificativă în identificarea aplicațiilor malware este dată de instrucțiunile *move* și *jump*. Instrucțiunile *if* și *invoke* nu aduc diferențe semnificative. Ideea de bază este că aplicațiile malware nu implementează o logică de aplicație la fel de complexă ca aplicațiile sigure.

Deoarece sistemul iOS este unul închis, provocările la adresa securității sunt mai puține. Astfel, permite revocarea/acceptul permisiunilor în mod dinamic, execută cod binar ARM, care este dificil de dezasamblat, împachetează conținutul printr-un proces anevoios față de fișierele .dex. Unul dintre vectorii de atac este reprezentat de utilizarea apelurilor API private în aplicații.

Arhitectura platformei este proiectată modular astfel încât să poată integra instrumente software criminalistice fără probleme de compatibilitate (Fig. 3). Fiecare modul este specializat pe îndeplinirea anumitor sarcini, după cum urmează:

• *Modulul interfață utilizator* – acest modul desfășoară activități de management pentru cazurile de investigație, producând atât rapoarte de securitate dinamice, cât și statice și alocând scoruri de risc pentru terminalele mobile, pe baza unei analize și evaluări specifice.

• *Modulul de autentificare/autorizare* – gestionează privilegiile de autentificare pentru utilizatorii definiți, precum și accesul la platforma web centrală.

• *Modulul parametrizare* – gestionează nomenclatoarele și oferă mijloacele de configurare a parametrilor platformei web centrale.

• *Modulul de colectare a datelor* – colectează datele și diseminează rezultatele analizelor, de asemenea calculează nivelul de infectare a aplicațiilor specifice terminalelor mobile.

• *Modulul de analiză forensic* – gestionează instrumentele și procedurile criminalistice de lucru, asigură identificarea caracteristicilor tehnice ale terminalelor mobile pentru a include aplicațiile instalate, colectează informații din terminalele mobile și susține procesul de analiză criminalistică pentru servicii web.

• *Modulul de monitorizare* – funcționează ca un agent Push, în sensul că analizează și evaluează toate aplicațiile instalate pe terminalele mobile, generând liste cu aplicații suspecte, alerte, stări și indicatorii cheie de performanță. Acest modul este conceput pentru a extinde spectrul de identificare a amenințărilor prin monitorizarea comportamentului aplicațiilor instalate pe terminalul mobil și prin transmiterea rezultatelor obținute către aplicația centrală responsabilă de colectarea și analiza datelor.

• *Modulul de inginerie inversă* – oferă capabilități de inversare prin efectuarea de încărcări și descărcări ale programelor specifice, care urmează să fie analizate.

• *Modulul de integrare a comportamentului online* – este conectat direct la modulul de analiză a comportamentului online către care transmite algoritmii de inteligență artificială și învățare automată (AI/ML) actualizați și de la care primește rezultatele analizei comportamentale online pentru o prelucrare ulterioară.

• *Modulul de analiză a comportamentului online* – dispune de o interfață web administrativă, care oferă diverse funcții, precum: Proxy, SSL, VPN, Wireless, USB și Ethernet. Modulul înregistrează datele de trafic produse atunci când terminalul mobil este conectat la platforma web prin Wi-Fi. Prin analiza rețelei, modulul oferă servicii de prevenire a intruziunilor, execută algoritmi AI/ML, detectează anomaliile cauzate de malware și transmite aceste anomalii de trafic către modulul de integrare online a comportamentului, pentru o prelucrare ulterioară. Modulul creează un profil al terminalului mobil, în corelație cu configurațiile implicite și cu traficul înregistrat, achiziționând

liste cu site-uri web clasificate drept periculoase, accesând și integrând informații furnizate de surse online despre amenințări.

• *Modulul cu instrumente criminalistice* – afișează o interfață prietenoasă cu utilizatorul, care permite extragerea și analiza paralelă, de la mai multe terminale mobile, și îndeplinește diferite sarcini, cum ar fi: extragerea datelor, analiza ulterioară a datelor, configurarea raportului pentru extragerea datelor, analiza avansată a aplicațiilor mobile, precum și funcționalitățile de recuperare a parolelor și fișierelor.

• *Agentul pentru modulul de colectare a datelor* – adună datele despre aplicațiile mobile, monitorizând mai multe caracteristici de securitate.

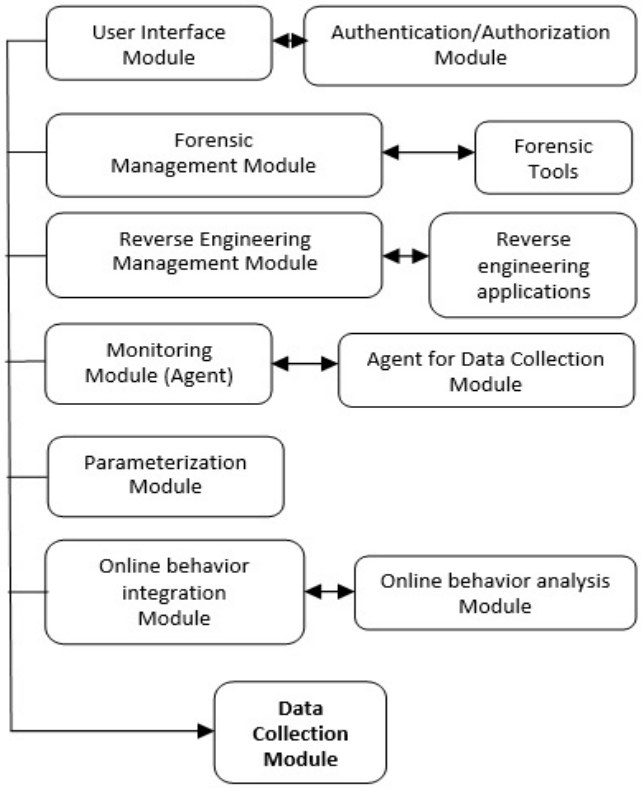

Figura 3 *Arhitectura pentru platforma software integrată*

Astfel, colectează indicatori cheie de performanță, comparând hashes-urile primite de la magazinele de aplicații cu hashes-ul aplicațiilor mobile instalate, gestionează alertele, sprijină detectarea de malware cu ajutorul listelor de semnături, profilează terminalul mobil, corelează configurația implicită cu aplicațiile instalate.

• *Modulul de aplicații de inginerie inversă* – este proiectat să realizeze analiza statică și dinamică a terminalelor mobile printr-un sistem de tip Sandbox. Acesta trimite rapoarte JSON/HTML către platforma centrală web, evaluează rezultatele analizei și identifică automat comportamentul unei aplicații malware.

Pentru a asigura funcționalitatea necesară, este folosit un server de virtualizare Proxmox KVM, pe care sunt create mașinile virtuale care susțin diferite servicii și aplicații. Mașinile virtuale folosesc containere Docker, care integrează Kubernetes pentru a coordona planificarea și execuția containerelor. Pentru administrarea containerelor, se folosește LXD, care este un sistem de generație nouă și oferă servicii de tipul API REST.

Utilizarea containerelor permite crearea de microservicii, aplicațiile fiind astfel decuplate pentru a putea fi instalate și administrate în mod dinamic.

Mașina virtuală VM3 este responsabilă de baza de date PostgreSQL pentru rularea serviciului torsim-database. Tot în această mașină virtuală sunt instalate instrumentele Apache Kafka și Elasticsearch. Utilitarele ADB folosite pentru asigurarea funcționalităților forensic sunt: Android Debug Bridge și MOBILedit. Serviciul torsim-adb înglobează clientul de ADB, care comunică cu serverul ADB instalat pe laptop, iar acesta comunică cu deamonul ADB de pe terminalul mobil. Momentan, integrarea cu MOBILedit este realizată la nivel procedural, rularea MOBILedit se face manual și se obține raportul dorit, care se încarcă apoi în platforma centrală.

Pentru interceptarea traficului generat de dispozitivele mobile, se folosește utilitarul Bro. Acesta trimite pachetele interceptate într-o coadă de mesaje Kafka, care sunt apoi preluate de serviciul torsim-messageprocessor și trimise în Elasticsearch. Pentru analiza traficului și determinarea comportamentului malițios, se folosește un sistem de detecție a traficului malițios, denumit Maltrail. Aplicația Mailtrail utilizează listele publice cu site-uri de încredere și cu site-uri malițioase, informațiile din rapoartele diferitelor produse antivirus, listele particularizate, unde semnăturile pot fi nume de domenii, IP-urile, valoarea din header a HTTP User-Agent și mecanismele euristice, care pot ajuta la descoperirea de malware încă necunoscut.

Tabelul 1

FLUXURI ÎNTRE APLICAȚII ȘI SERVICII

Nr.	Table Column Head		
	Serviciu/aplicație	*Mașină virtuală*	*Rol*
1	nginx	VM 1	Frontend-ul web preia cereri de la clienți
2	torsim-proxy	VM 1	Interfața grafică a aplicației și serviciile securizate primesc cereri de la frontend-ul web (nginx)
3	torsim-bro-logtail	VM 2	Preia traficul înregistrat de mașină și îl salvează într-o coadă din Kafka
4	torsim-message-processor	VM 1	Preia cereri de la torsim-proxy
5	MobSF, CuckooDroid	VM 1	Preia cereri de la torsim-proxy
6	torsim-adb	VM 1	API-ul torsim-adb preia cereri de la torsim-proxy
7	Torsim-database	VM 3	API-ul torsim-database preia cereri de la torsim-adb și torsim-proxy
8	PostgreSQL	VM 3	Preia cereri de la containerul torsim-database
9	Elasticsearch	VM 3	Preia cereri de la torsim-message-processor
10	Kafka	VM 3	Preia cereri de la torsim-message-processor

Mașina virtuală VM 1 are instalat CuckooDroid, o extensie a Cuckoo Sandbox. CuckooDroid este un software open source, utilizat în analiza fișierelor suspicioase, cu capabilități în analiza statică și dinamică a aplicațiilor Android. De asemenea, framework-ul MobSF permite analiza statică și dinamică a aplicațiilor mobile. S-a folosit un container Docker pentru aplicația MobSF, care face analiza statică, iar integrarea cu platforma centrală s-a făcut prin API-ul MobSF. Astfel, aplicațiile pot fi trimise spre a fi analizate, obținându-se raportul atât în format PDF, cât și în format JSON. Acesta din urmă este folosit pentru a stoca datele scanării în baza de date și pentru a le afișa în interfața web.

O etapă importantă în analiza malware o constituie testarea platformei prin verificarea tuturor parametrilor introduși și obținerea de rapoarte corecte. Modulul agent este încă în faza de dezvoltare și va fi suportat de Android și IOS. Agentul trebuie să ruleze pe telefoane nerootate și urmărește permisiunile solicitate de aplicații, instalate înainte și în timpul rulării aplicațiilor. Se pot urmări respectivele caracteristici: accesarea rețelei și a datelor sensibile (precum lista de contacte și locația), recepționarea și transmiterea de SMS-uri, datele din clipboard, accesul la diferite componente hardware, numărul de click-uri în perioada de activitate intensă a utilizatorului, corelată cu perioada de inactivitate. În plus, agentul poate fi integrat cu API-ul public, pus la dispoziție de către VirusTotal, pentru a verifica autenticitatea pachetului apk, prin compararea hash-ului aplicației cu baza de date a site-ului. Se verifică fișierele de configurare a aplicațiilor, pentru a identifica versiunea aplicației, resursele hardware care vor fi solicitate, permisiunile care urmează să fie alocate, componentele, lista de permisiuni periculoase. Existența unor șiruri suspecte de caractere în aplicație poate fi un indicator al prezenței unei infectări cu malware. Cu ajutorul entropiei se identifică dacă există zone de cod criptate.

Concluzii

Noile progrese în materie de inteligență artificială și de învățare automată au permis apariția unei noi etape în evoluția securității cibernetice. Literatura studiată pentru dezvoltarea arhitecturii platformei include doar algoritmi de învățare supervizată. Au fost testate diverse soluții software de securitate a terminalelor mobile și a fost construită infrastructura hardware și software. Proiectul de cercetare nu este finalizat, urmează

faza de testare a aplicațiilor malware selectate de experții proiectului.

Această lucrare a fost posibilă cu sprijinul financiar al UEFISCDI/Ministerului Educației Naționale din România, proiectul PN-III-P2-2.1-SOL-2016-05-0070, cu titlul „Platformă software integrată pentru analiza programelor malware pe terminalele mobile”.

NOTE:

1 James R. Blake, *Transforming military*, Praeger Security International, May 2007, accesat la 12 feb. 2019.

2 https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, accesat la 12 feb. 2019.

3 Elena Șușnea, Adrian Iftene, ”The Significance of Online Monitoring Activities for the Social Media Intelligence (SOCMINT)”, *Conference on Mathematical Foundations of Informatics MFOI’2018, Institute of Mathematics and Computer*, Chișinău, Moldova, pp. 230-240, 2018.

4 Reza Hedayat, Lorenzo Cavallaro, ”The Devil’s Right Hand: An Investigation on Malware-oriented Obfuscation Techniques”, *Computer Weekly*, August 2016.

5 Dragoș Bărbieru, Alexandru Stoica, ”Malware Analysis on Mobile Phone”, *The International Scientific Conference eLearning and Software for Education*, vol. 4, ”Carol I” National Defence University, Bucharest, 2018, pp. 11-15.

6 https://fortiguard.com/events/755/2013-10-25-playing-hide-and-seek-with-dalvik-executables, accesat la 12 feb. 2019.

7 Babak Bashari Rad, Maslin Masrom, Suhaimi Ibrahim, ”Camouflage in Malware: from Encryption to Metamorphism”, *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, No. 8, August 2012.

8 Hannsang Kim, Member IEEE, Kang G. Shin, Padmanabhan Pillai, ”MODELZ: Monitoring, Detection and Analysis of Energy-Greedy Anomalies in Mobile Handsets”, *IEEE Transactions on mobile computing*, vol. 10, July 2011.

9 Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, Yael Weiss, *Andromaly: a behavioral malware detection framework for android devices.*

10 Gianluca Dini, Fabio Martinelli, Andrea Saracino, Daniele Sgandurra, ”MADAM: a Multi-Level Anomaly Detector for Android Malware, Computer Network Security”, *6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012*, St. Petersburg, Russia, October 17-19, 2012.

11 C E.hin, A.P. Felt,. K Greenwood, D. Wagner, ”Analyzing inter-application communication in Android”, *Proc. 9th Int. Conf. On Mobile Systems, Applications, and Services (MobiSys ‘11). ACM*, Washington, DC, USA, June 2011, pp. 239-252.

12 X. Wei, L. Gomez, I. Neamtiu, M. Faloutsos, ”ProfileDroid: multi-layer profiling of android applications”, *Proc. 18th Int. Conf. On Mobile Computing and Networking (Mobicom ‘12). ACM*, Istanbul, Turkey, August 2012, pp. 137-148.

13 Shuang Liang, Xiaojiang Du, *Permission-combination-based scheme for Android mobile malware detection*, IEEE International Conference on Communications (ICC), Sydney, June 2014.

14 Suleiman Y. Yerima, Sakir Sezer, Igor Muttik, ”Android Malware Detection Using Parallel Machine Learning Classifiers”, *Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, December, 2014.

15 Xing Liu, Jiqiang Liu, ”A Two-Layered Permission-Based Android Malware Detection Scheme, *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, April, UK, Oxford, 2014.

16 BooJoong Kang, Suleiman Y. Yerima, Sakir Sezer, Kieran McLaughlin, *International Journal on Cyber Situational Awareness*, vol. 1, No. 1, 2016, pp. 231-255.

17 Gerardo Canfora, Francesco Mercaldo, Corrado Aaron Visaggio, ”Mobile malware detection using op-code frequency histograms”, 12th International Joint Conference on e-Business and Telecommunications (ICETE), Paris, July 2016.

BIBLIOGRAFIE

Babak Bashari Rad, Maslin Masrom, Suhaimi Ibrahim, ”Camouflage in Malware: from Encryption to Metamorphism”, *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, No. 8, August 2012.

Bărbieru Dragoş, Stoica Alexandru, ”Malware Analysis on Mobile Phone”, *The International Scientific Conference eLearning and Software for Education*, vol. 4, ”Carol I” National Defence University, Bucharest, 2018.

Blake R. James, *Transforming military*, Praeger Security International, May 2007.

Canfora Gerardo, Mercaldo Francesco, Visaggio Corrado Aaron, ”Mobile malware detection using op-code frequency histograms”, *12th International Joint Conference on e-Business and Telecommunications (ICETE)*, July 2016.

Chin E., Felt A.P., Greenwood K., Wagner D., ”Analyzing inter-application communication în Android”, *Proc. 9th Int. Conf. On Mobile Systems, Applications, and Services (MobiSys ‘11). ACM*, Washington, DC, USA, June 2011.

Dini Gianluca, Martinelli Fabio, Saracino Andrea, Sgandurra Daniele, ”MADAM: a Multi-Level Anomaly Detector for Android Malware”, *Computer Network Security: 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security, MMM-ACNS 2012*, St. Petersburg, Russia, October 17-19, 2012.

Hannsang Kim, Member IEEE, Kang G. Shin, Padmanabhan Pillai, ”MODELZ: Monitoring, Detection and Analysis of Energy-Greedy

Anomalies în Mobile Handsets”, *IEEE Transactions on mobile computing*, vol. 10, July 2011.

Hedayat Reza, Cavallaro Lorenzo, ”The Devil’s Right Hand: An Investigation on Malware-oriented Obfuscation Techniques”, *Computer Weekly*, August 2016.

Kang BooJoong, Yerima Y. Suleiman, Sezer Sakir, McLaughlin Kieran, *International Journal on Cyber Situational Awareness*, Vol. 1, No. 1, 2016.

Shabtai Asaf, Kanonov Uri, Elovici Yuval, Glezer Chanan, Weiss Yael, *Andromaly: a behavioral malware detection framework for android devices*.

Shuang Liang, Xiaojiang Du, ”Permission-combination-based scheme for Android mobile malware detection”, *IEEE International Conference on Communications (ICC)*, June 2014.

Șușnea Elena, Iftene Adrian, ”The Significance of Online Monitoring Activities for the Social Media Intelligence (SOCMINT)”, *Conference on Mathematical Foundations of Informatics MFOI’2018*, Institute of Mathematics and Computer, Chisinau, Moldova, 2018.

Wei X., Gomez L., Neamtiu I., Faloutsos M., ”ProfileDroid: multi-layer profiling of android applications”, *Proc. 18th Int. Conf. On Mobile Computing and Networking (Mobicom ‘12). ACM*, Istanbul, Turkey, August 2012.

Xing Liu, Jiqiang Liu, ”A Two-Layered Permission-Based Android Malware Detection Scheme”, 2nd *IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, April, 2014.

Yerima Y. Suleiman, Sezer Sakir, Muttik Igor, ”Android Malware Detection Using Parallel Machine Learning Classifiers”, *Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*, UK, Oxford, December, 2014.

https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

https://fortiguard.com/events/755/2013-10-25-playing-hide-and-seek-with-dalvik-executables