Raport Ştiinţific şi Tehnic
Hub inovativ pentru tehnologii avansate de securitate cibernetică (ATLAS)
Contract Nr. 17PCCDI/2018
- Etapa I -

Introducere

Etapa I s-a desfăşurat în perioada 31/03/2018 - 31/12/2018 şi a avut ca obiective principale:

- identificarea abordărilor şi tendinţelor în domeniu, stabilirea tehnologiilor aplicabile, elaborarea specificaţiilor pentru produsele / serviciile ce urmează a fi realizate;
- stabilirea specificaţiilor pentru echipamentele şi aplicaţiile software ce urmează a fi achiziţionate; achiziţia primului lot de echipamente.
- familiarizarea membrilor echipelor de cercetare cu metodologia de lucru a proiectului; armonizarea echipelor de cercetare.
- stabilirea metodelor de comunicare şi colaborare în cadrul proiectului;
- iniţierea de întâlniri cu potenţialii beneficiari din mediul guvernamental şi privat;
- diseminarea rezultatelor cercetării prin publicarea de articole în jurnale şi la conferinţe indexate ISI / BDI;
- realizarea site-ului Web de prezentare a proiectului;
- recrutarea şi angajarea noilor cercetători.

Pentru atingerea acestor obiective, în cadrul celor 3 proiecte componente, au fost planificate 5 activităţi de tipul A1 (Cercetare fundamentală). În continuare, sunt prezentate în detaliu rezultatele ştiinţifice şi tehnice obţinute în urma derulării acestor activităţi.

Proiect Component 1 - Metode formale şi tehnici de învăţare automată pentru securizarea aplicaţiilor şi sistemelor de operare

Activitatea 1.1 - Analiza mecanismelor de securitate folosite în sistemele de operare moderne

Sistemele de operare moderne folosesc multiple mecanisme pentru a oferi o securitate mai buna, atât a diverselor servicii pe care le furnizează, cat şi a aplicaţiilor. În cadrul acestei activităţi s-a urmărit realizarea unei analize a acestor mecanisme având ca scop înţelegerea interoperabilităţii acestora, metodele de utilizare în diverse scenarii şi definirea unei arhitecturii

pentru analiza aplicaţiilor şi modul acestora de folosire corecta a noilor metode de securitate. Raportul tehnic este împărţit în doua componente, în prima etapa am identificat şi analizat principalele metode de securitate, iar în partea a doua am introdus o metoda generica de validare a aplicaţiilor care urmează a fi dezvoltata în etapa a doua a proiectului ATLAS.

Sistemele de operare moderne impun securitatea datelor folosind o abordare ierarhica, în care fiecare nivel al sistemului poate adaugă un anumit privilegiu de securitate. La primul nivel, sistemele de operare folosesc resurse hardware de securitate dedicate pentru a proteja propriile servicii sau pentru a oferi izolarea aplicaţiilor. Majoritatea dispozitivelor mobile folosesc procesoare de tip ARM, care oferă o componenta de izolare a aplicaţiilor denumita TrustZone. Pentru dispozitivele personale (laptop) sau servere dedicate, procesoarele de tip Intel oferă o componenta asemănătoare denumita Intel Software Guard Extensions.

Pentru a înţelege mai bine integrarea dintre componentele de securitate hardware şi sistemele de operare am analizat modul acestora de comunicare. Studiul merge mai departe şi propune o metodă de rulare a programelor GNU / Linux nemodificate în domeniul securizat al ARM TrustZone, obţinând avantajele de execuţie de încredere, păstrând în acelaşi timp accesibilitatea serviciilor sistemelor de operare (cum ar fi accesul la fişierele şi utilizarea reţelei) utilizând un proxy de apel sistem automat. Mai mult, am testat faptul că aplicaţiile de eşantionare care utilizează discul sau reţeaua pot rula nemodificate, având doar o latenţă mică, constantă. Sistemul nostru constă din două componente principale, fiecare corespunzând uneia dintre cele două domenii de securitate oferite de către ARM TrustZone:

- Microkernel-ul de încredere care supraveghează aplicaţiile care rulează în Secure World, trimite apelurile de sistem către domeniul nesigur, şi verifica rezultatele înainte de a le trimite înapoi la program.
- un serviciu care rulează în interiorul sistemului normal, neprotejat de hardware-ul dedicat, şi care interoghează sistemul de operare neprotejat, pentru a-şi furniza serviciile înapoi în domeniul securizat.

Microkernelul de încredere este un sistem minimal de operare care rulează în interiorul Secure World. Când dispozitivul porneşte prima dată, acesta va fi încărcat primul si va iniţializa accesul la memorie şi mecanismele de întrerupere TrustZone, pentru a instala codul de monitor securizat. Memoria fizică este împărţită în două segmente: prima este protejată (accesibilă numai din mediul de încredere), în timp ce partea rămasă va fi gestionata de un sistem de operare posibil maliţios. Microkernelul de încredere îşi rezervă, de asemenea, câteva întreruperi pentru uz propriu sau pentru interceptare, pentru a putea implementa accesul securizat al resurselor.

Microkernel-ul va porni apoi sistemul de operare bogat (GNU / Linux) şi va aştepta Secure Monitor Calls (SMC). Acestea sunt emise pentru cereri de încărcare a aplicaţiilor, răspunsuri la apelul sistemic şi reluarea proceselor de încredere. Pe de altă parte, serviciul creat se integrează cu sistemul de operare normal pentru a programa procesele, pentru a executa apelurile de sistem şi a trimite rezultatele înapoi.

La nivelul sistemului de operare avem biblioteci care oferă diverse resurse pentru dezvoltator, printre care mecanisme criptografice, acces la sistemul de fişiere sau la placa de reţea. Prin intermediul acestor biblioteci se pot implementa diverse sisteme de control al accesului. Mai mult, sistemul de operare oferă o serie de liste de control care pot bloca sau permite accesul la

fişiere pentru aplicaţi. Un alt sistem de securitate oferit de către sistemul de operare este acela de sandboxing, prin intermediul căruia o aplicaţie este limitata la accesarea unor resurse. In cadrul primei etape am studiat implementarea sistemului de sanboxing pentru iOS dar şi funcţionalitatea diverselor biblioteci de criptare existente pentru sistemele de operare moderne.

Pentru sistemul de operare iOS, ne-am concentrat în a răspunde la întrebarea: “Ce metode de securitate şi confidenţialitate NSXPC sunt accesibile pentru aplicaţi?”. Astfel, pentru a răspunde la această întrebare, putem îmbunătăţi cunoştinţele disponibile pentru instrumentele anterioare de analiză a controlului accesului pentru iOS cu semantica IPC şi pentru a identifica defectele de politică necunoscute.

Pentru a determina care dintre metodele NSXPC sensibile la securitate şi confidenţialitate sunt accesibile aplicaţiilor terţe, am identificat trei provocări de cercetare:

- Serviciile NSXPC sunt rezolvate dinamic prin intermediul numelor de servicii. Nu există nicio documentaţie sau fişier de configurare care să cuprindă servicii NSXPC (adică nume de metode) pentru daemonii corespunzători.
- Politica de control al accesului pentru accesarea serviciilor NSXPC este codificată, nu putem consulta o specificaţie a politicii, codificată într-un format proprietar sau altfel.
- Setul de drepturi disponibile pentru aplicaţi nu este cunoscut. În timp ce XCode defineşte un set de drepturi "publice" disponibile tuturor dezvoltatorilor de aplicaţii iOS, rapoartele indică faptul că există un set de drepturi "semi-private" acordate de Apple.

In cadrul raportului tehnic şi prin articolele publicate am rezolvat aceste provocări printr-o combinaţie de analize statice şi teste dinamice. Am identificat toate serviciile, am triat numai acele servicii care sunt accesibile aplicaţiilor şi am folosit euristica pentru a alege servicii accesibile pentru analiza manuală. Testarea dinamică iniţială utilizează valori neiniţializate pentru variabilele transmise ca argumente. În multe cazuri, valorile neiniţializate sunt suficiente pentru a declanşa activitate neobişnuită a sistemului (de exemplu, blochează, solicită utilizarea resurselor de sistem care dezactivează, alerte auditive). Sunt efectuate teste suplimentare pentru a mapa invocările de servicii specifice cu respectarea operaţiunilor sensibile la securitate. De asemenea, folosim numele metodelor ca euristice pentru a prioritiza metodele de investigare manuală. În timpul investigării manuale putem iniţializa variabilele cu valori valide şi putem folosi opţional o activitate a sistemului de monitorizare a dispozitivelor jailbroken (de ex., Jurnale de acces la fişiere) în timpul invocării serviciului.

Un alt mecanism de securitate folosit de către aplicaţii îl reprezinta protejarea datelor, atât la stocare cat şi în tranzit. Am analizat următoarele biblioteci disponibile pentru dispozitive cu resurse hardware limitate:

- mbedTLS (2.9.0 la 3 mai 2018) implementează pe deplin protocolul TLS şi vizează în mod specific sistemele cu resurse limitate. Am folosit doar funcţiile sale criptografice şi nu întreaga stivă (pentru ca analiza să fie comparata cu celelalte biblioteci).
- TomCrypt (v1.18.1, 22 ianuarie 2018): ofera un set de instrumente criptografice modulare. Deşi nu vizează în mod specific platformele încorporate, aceasta biblioteca foloseste si LibTomMath / LibFastMath pentru implementarea operaţiunilor de număr mare necesare pentru RSA.

- wolfSSL (3.15.0 pe 5 iunie 2018): fostul CyaSSL cu o istorie care datează din 2004, wolfSSL implementează, de asemenea, un stack TLS complet cu algoritmi criptografici şi multe altele.

Pentru experimentul nostru, am selectat algoritmi reprezentativi din fiecare categorie:

- AES / Rijndael: Standardul de criptare simetrica, ales dintre cincisprezece modele concurente realizate de NIST ca cifru bloc de securitate simetric recomandat. Are o dimensiune de bloc de 128 de biţi şi acceptă chei de 128/192/256 biţi. Nu s-au descoperit până acum nicio vulnerabilitate critică.
- ChaCha: Un algoritm criptografic de tip stream, realizat de Daniel J. Bernstein, este certificat de eSTREAM ca fiind un algoritm de înaltă performanţă şi sigur, foarte potrivit pentru aplicaţiile încorporate.
- Secure Hash Algorithm (SHA): Proiectat de NSA, acesta a devenit alegerea standard pentru o functie hash criptografica (one way). Am folosit versiunea SHA-2, 256bit.
- Poly1305: De la acelaşi autor ca şi cifrul ChaCha, Poly1305 este un cod de autentificare a mesajului folosit pentru verificarea integrităţii şi autentificare.
- RSA: Primul sistem de criptografie public-cheie, popularitatea RSA nu a dispărut. Se bazează pe problema factorizării întregi, susţinând atât criptarea asimetrică, cât şi semnătura digitală.

Pentru etapa a doua ne propune sa folosim cunoştinţele acumulate şi sa cream un serviciu disponibil online pentru validarea securităţii aplicaţiilor în cadrul sistemelor de operare moderne. Principalele funcţionalităţi identificate şi care se vor regăsi sunt:

- Folosirea unui mod de decriptare a datelor, atât pentru datele stocate locale dar şi pentru cele trimise peste reţea. Am identificat câteva posibilităţi pentru a face asta: folosind un preloader prin care bibliotecile de criptare vor fi înlocuite cu o biblioteca dezvoltata in-house, sau folosind un proxy în care sistemul de operare are încredere şi care va putea realiza un atac de tip MITM.
- Folosirea unei analize complexe a binarelor, atât statica cat şi dinamice.
- Un mecanism de sandboxing prin intermediul căruia se pot analiza toate apelurile de sistem realizate de aplicaţie, incluzând accesul la fişiere, accesul la diversele dispozitive sau altele.
- Posibilitatea de a analiza aplicaţiile atât pe un dispozitiv real cat si pe simulatoarele existente.
- Integrarea cu diversele servicii prin care se pot identifica fisiere malware deja cunoscute.
- Integrarea cu o sursa de threat intelligence pentru a analiza cererile de trafic realizate, cu precădere ne vom uita la traficul de DNS realizat.

Activitatea 1.2 - Analiza problemelor de securitate abordabile cu metode formale si invatare automata

In cadrul acestei activităţi au fost abordate trei mari direcţii:

- Probleme de securitate care se preteaza cu o tehnica speciala de invatare automata numita “dictionary learning” (aplicată în diverse domenii: securitatea reţelelor, fraude financiare etc.);

- Analiza fisierelor binare (executabile) folosind metode formale si automate: analiză statică, dar şi dinamică folosind execuţie simbolică. Fuzz testing îmbunătăţit cu învăţare automată şi algoritmi genetici. implementări distribuite + tool open-source RIVER;
- Analiza dpdv al securitatii al unor protocoale de criptografie moderne care stau la baza unor noi tehnologii precum blockchain.

In continuare vom da mai multe detalii pentru cele 3 directii.

Metode de tip "dictionary learning" pentru securitate

Motivatie. Învăţarea automată ne ajută să abordăm probleme mari de optimizare şi aparent dificile. Chiar dacă reţelele neuronale sunt de departe cea mai populară metoda, noi ne vom concentrăm pe metode de antrenare de dicţionare (DL - Dictionary learning) pentru reprezentările rare (v. B. Dumitrescu and P. Irofti, Dictionary Learning Algorithms and Applications, Springer, 2018). Preferinţa noastră este motivată de modelul mult mai simplu, faptului că vizează metode mai rapide cu un fundal teoretic solid şi interpretabilitate mai bună.

Antrenarea dicţionarelor (DL). Scopul problemei de antrenare a dicţionarului este de a proiecta o matrice $D \in R^{m \times n}$ numit *dicţionar*, astfel încât pentru o clasă de semnale $y \in R^m$ putem obţine o reprezentare rară $x \in R^n$, ce este o aproximare bună a semnalului original $y \approx Dx$. Sparsitatea din x implică faptul că y este combinaţia liniară dintre doar câteva coloane de dicţionar (numiţi şi *atomi*).

Formal, aranjăm un set dat de semnale de antrenament N sub forma coloanelor unei matricii $Y \in R^{m \times N}$ şi continuăm să rezolvăm p problemă de optimizare de tip min dupa D si X pentru ||Y-DX||, unde variabilele sunt dicţionarul D şi matricea reprezentărilor rare $X \in R^{n \times N}$, ale căror coloane au cel mult s elemente nonzero.

Adesea trebuie să specializăm dictionarul la sarcina de îndeplinit. În funcţie de aplicaţie şi de tipul de semnale pe care le procesăm, obiectivul DL de mai sus va trebui să fie modificat pentru a se potrivi mai bine scopului nostru. Spre exemplu, atunci când ne ocupăm de probleme în care lipsesc date, vom dori să adăugăm şi alte constrângeri reprezentărilor rare. Altfel, când problema este una de clasificare, vom dori să includem proprietăţi discriminatorii atomilor din dicţionar.

Adaptarea DL pentru aplicaţii Big Data Aplicaţiile moderne de securitate se bazează pe date de mari dimensiuni. Ca atare, algoritmii de învăţare care trebuie dezvoltaţi necesită o configurare hardware şi software care permite ajustarea problemelor DL la caracteristicile datelor de mari dimensiuni. Pe lângă beneficiile imediate pentru clasa de aplicaţii specifice securităţii cibernetice, această sarcină este de asemenea de importanţă teoretică, deoarece în prezent este neexplorată. Scalarea metodelor DL pentru datele mari nu este neapărat o sarcină trivială. *O parte a cercetării în următoarele faze ale proiectului va fi preocupată de investigarea mijloacelor adecvate de adaptare a algoritmilor DL la cerinţele Big Data sau de eludarea limitărilor de a aplica direct DL colecţiilor mari de semnale.*

Clasificarea DL şi problema identificării aplicaţiilor malware O arie principală de cercetare în ceea ce priveşte aplicarea metodelor de învăţare automată la problemele de securitate cibernetică va consta în dezvoltarea de soluţii de clasificare bazate pe DL pentru detectarea aplicaţiilor malware. Deoarece domeniului îi lipseşte standardizarea în ceea ce priveşte

algoritmii de învăţare automată pentru detecţia malware, precum şi în ceea ce priveşte bazele de date malware, *lucrările preliminare se vor referi la stabilirea cerinţelor şi la obţinerea seturilor de date malware care vor fi folosite în testele ulterioare.*

Se vor testa mai multe complexităţi ale modelului, de la un clasificator simplu, binar malware-nonmalware la diferenţiere între diferite familii malware, precum şi o abordare nesupervizată în care tipurile de programe malware sunt descoperite mai degrabă decât stabilite în prealabil. Toate experimentele vor include o analiză a impactului etapei de feature selection asupra performanţei algoritmului.

Scopul experimentelor este de a aplica clasificarea DL în problema identificării malware-ului, o abordare care, după cunoştinţele noastre, nu a mai fost încercată înainte şi de a testa performanţa algoritmilor în identificarea mai multor clase de familii malware. O abordare nesupravegheată va fi, de asemenea, testată, unde adevărata clasă malware este necunoscută, verificând astfel adecvarea metodelor DL pentru a descoperi structura problemei de detectare de malware.

Detecţia anomaliilor Alegerea anumitor atomi atunci pentru a reprezenta semnale (sau lipsa acestora) ne permite să realizăm detecţia anomaliilor şi poate chiar izolarea acestora, în cadrul unui proces descris printr-un set dat de măsurători, sau semnale. Un punct de plecare al studiului nostru poate fi, din nou, problema de clasificare unde ne concentrăm asupra semnalelor dificil de clasificat. Odată ce un model generic pentru detectarea anomaliilor prin antrenarea dicţionarelor (DL) este identificat, putem continua să îl specializăm continuare pentru diverse aplicaţii, cum ar fi:

- detecţia fraudelor pentru tranzacţiile bancare
- sisteme de detectare a intruziunilor
- detecţia anomaliilor medicale
- detectarea şi izolarea defecţiunilor în reţelele senzorilor (d.ex. distribuţia apei şi reţele electrice)
- identificarea biometrică

Detecţia anomaliilor poate fi considerată ca o sarcină de clasificare în care clasa anomaliilor se deosebeşte de ceea ce este considerat norma sau normalul procesului. Prin urmare, este posibilă o implementare simplă a clasificării DL, cu condiţia ca această normă să fie dată / cunoscută. Pe lângă implementarea unor astfel de metode, cercetarea aplicării detecţiei anomaliilor pentru aplicaţiile de securitate va lua în considerare şi cazurile în care clasa normală nu este furnizată.

Aplicaţii DL pentru securitatea IoT (Internet of Things) Adaptările avute în vedere vor lua în considerare şi aplicabilitatea la problemele din domeniul IoT, în care multe aspecte legate de securitate sunt problematice. Reprezentărilor rare sunt foarte potrivite pentru aplicaţiile IoT, unde puterea de calcul a dispozitivelor este limitată. Simplitatea modelelor DL sugerează că sunt un candidat promiţător pentru aplicaţii de învăţare automată în IoT. Dacă specificaţiile dispozitivului sunt extrem de restrictive, se pot aplica abordări online ale DL. În metodele online, reprezentarea rară este învăţată prin adaptarea dicţionarului la seturi mici de semnale de intrare ce sunt actualizate în permanenţă. Cu fiecare astfel de set, D este actualizat în mod eficient pentru a cuprinde noile informaţii aduse de semnale, fără a fi recalculat în întregime la fiecare moment, reducând astfel resursele necesare.

Problemele la scară largă sunt, de asemenea, potrivite pentru abordările online, deoarece dimensiunea setului de date poate fi prohibitivă pentru metodele DL directe.

Astfel, vom fi efectuate cercetări privind ajustarea metodelor online pentru sarcinile de clasificare descrise mai sus, precum şi crearea cadrului software necesar pentru aplicarea acestui tip de algoritmi pe seturi mari de date.

Metode formale pentru analiza executabilelor

Pentru analiza vulnerabilitatilor fisierelor executabile ne vom concentra pe doua subdirectii: executie simbolica si fuzz testing imbunatatit. Vom prezenta ideile de cercetare pe care le vom urma in cadrul proiectului:

Executie simbolica. RIVER este un cadru de analiză pentru executabil bazat pe execuţia simbolică, care se deosebeste de alte solutii similare prin posibilitatea de a executa un fisier binar înainte şi, mai ales, înapoi prin arborele de execuţie. Acesta a fost initiat intern in cadrul firmei Bitdefender si a fost continuat sa fie dezvoltat si colaborare cu cercetatorii de la Universitatea din Bucuresti (v. *T. Stoenescu, A. Stefanescu, S. Predut, F. Ipate. Binary Analysis based on Symbolic Execution and Reversible x86 Instructions, Fundamenta Informaticae 153 (1-2), pp. 105-124, 2017*). In prezent este un proiect open source la adresa *https://github.com/bitdefender* Cadrul oferă componente cum ar fi un motor de taint analysis (analiza accesului si modificarii datelor), urmărirea unor căi de execuţie de urmărire, dar si un un motor dinamic de execuţie simbolică şi integrare cu toolul Z3 pentru rezolvarea constrângerilor.

RIVER are un mare potential atat din punct de vedere aplicativ dar si de cercetare. Pentru a disemina mai bine functionalitatile platformei, *unul dintre primele obiectivele pentru anul urmator este integrarea soluţiei RIVER într-o platformă online*, în care utilizatorul poate testa executabilul in urmatoarele aspecte:

- detectarea de vulnerabilitati in codul sursă / executabil
- greşelile comune de securitate in programare
- detectarea căilor nefezabile în codul sursă
- detectarea căilor din codul sursă care poate ajunge intr-o stare definita formal de utilizator

Execuţia simbolică (dar si cea concolica) se află la baza detectării problemelor definite mai sus. Ca *exemple de vulnerabilităţi concrete pe care intenţionăm să le arătăm* în instrumentului nostru sunt:

- modificarea frauduloasa a variabilelor stivei
- redirecţionaţi shell-ul şi obţinerea accesului la root
- redirecţionarea apelurilor în interiorul unei aplicaţii prin exploatarea memoriei de tip heap

De asemenea, intenţionăm să verificăm şi să raportam cea mai lungă durată de execuţie între toate căile dintr-un executabil dat. Astfel dorim sa adaptam propuneri de cercetare recente care încearcă să estimeze complexitatea programelor sau să detecteze problemele de performanţă, fără a avea acces la codul sursă.

O versiunea viitoare RIVER va aborda greşelilor obişnuite (dpdv al securitatii) din codul sursă care pot duce la buffer overflow, impartire la zero sau nesatisfacerea unor invarianti predefiniti. Utilizatorul va primi exemplele de date de intrare care demonstreaza existenta unor astfel de

probleme. Din punct de vedere tehnic, cele de mai sus pot fi realizate prin implementarea unor analize de tip *taint* asupra adreselor de memorie utilizate de aplicaţia.

Un alt tip de rezultat folositor un utilizator ar fi detectia de căi nefezabile, care poate fi înlăturate în siguranţă din codul executabil (şi pe care care optimizatorul din compilator nu le-a detectat). Putem rezolva aceste prin folosirea unui demonstrator automat al constrangerilor de pe caile de executie. Astfel, RIVER va trebui sa ofere o modalitate mai bună de a elimina condiţiile de ramificaţie (folosind solverul Z3) şi de a genera noi intrări. O vedere generală asupra fluxului automat care trebuie implementat este redata mai jos:

- Executarea programul împotriva unui set de intrări concrete
- Salvarea temporara a condiţiior logice întâlnite de solverul Z3
- Procesarea condiţiilor şi obţinerea de noi intrări pentru a ajunge la diferite ramuri ale programului în curs de testare.
- Setarea unei intrarea nouă (in spiritul fuzz si concolic testing) şi iterarea procesului de mai sus pana la obtinerea rezultatelor dorite.

Din punct de vedere al programarii, diferitele apecte pentru vulnerabilităţi, greşeli definite mai sus, vor fi rezolvate prin înregistrarea unor “observatori” în acest flux de execuţie simbolică. De exemplu, va exista un observator de tip "vulnerabilitate" care se va ocupa de o anumita submultime de problemelor. Analiza *taint* şi urmărirea evolutiei unor simboluri si variabole vor fi adaptate pentru fiecare observator in parte

Nu in ultimul rand, implementarea fluxului general de mai sus poate fi optimizat din punctul de vedere al performantei prin două mecanisme principale, pe care le vom explora in proiect:

- o platformă distribuită pentru executarea de noi programe împotriva intrărilor.
- folosirea tehnicilor de învăţare automată (cel mai probabil reinforcement learning) pentru invatarea deciziilor pe ramurii de executie, cu efectul selectarii unor cai optime de executie.

Testare de tip fuzzing imbunatatita.

Folosind capabilităţile RIVER care îmbogăţesc instrucţiunile unui executabil cu informaţii suplimentare, avem posibilitatea sa aplicam analiza taint si să urmărim fluxul unui program la nivel de bloc de bază, adică instrucţiuni consecutive care se termină cu un salt. În cadrul organizaţiilor care ţintesc sa aibă o calitate înălţă a codului, este extrem de util să păstreze un corpus larg de date de intrare variate pentru a testa bucăţi de software împotriva diferitelor probleme, cum ar fi cele enumerate mai sus. Unul dintre obiectivele noastre de cercetare din ultimul an a fost de a investiga metode pentru a crea corpus de intrări de testare pentru programe care au o bună acoperire a codului (raport bun între blocurile afectate şi blocurile totale din executabil). Aceasta directie este o imbunatatire a metodei de fuzz testing, care incearca sa descopere probleme incercand un numar mare de intrare (la inceput, date aleatoare).

Prima metodă a folosit algoritmi genetici, în care populaţia era o serie de teste. De-a lungul generaţiilor, populaţia se îmbunătăţea, alegând funcţie de optimizare (fitness function) o funcţie de acoperire a codului; mai precis, oferind scoruri mai mari pentru noua ramură descoperită de un nou test. În al doilea rând, avansând în aceeaşi directie, am grupat categoriile de intrări într-un corpus existent, am învăţat un model generativ pentru fiecare dintre aceste categorii (folosind reţele neuronale recurente), apoi am putut genera un corpus mare de intrări, randomizări care au ajutat la testarea de tip fuzzing obţinand o acoperire mai bună a codului - în mare parte deoarece intrările au fost mult mai bine formatate decât în testarea genetică.

Folosind cea mai recentă versiune a platformei RIVER, am reuşit să realizăm o legătură între condiţiile ramurii şi intrările date. Aceasta ne ofera, în principiu, informatia despre ce bucăţi de intrări au afectat anumite ramuri în timpul execuţiei programului. De la un corpus existent înţelegem apoi formatul pentru anumite zone ale intrărilor şi derivăm o expresie regulată. Folosim colecţia de expresii regulate ca model generativ pentru a crea noi teste de intrare cu o precizie ridicată (adică nu eşuează prin condiţiile precoce datorită intrărilor greşite).

Planul nostru în anul următor pentru metode fuzzing este îmbunătăţirea algoritmului genetic şi a metodelor generative existente prin informatii suplimentare oferite de RIVER pentru fiecare ramură a unui bloc de bază. Ca exemplu, am putea aplica fuzz testing pe zonele intrărilor iniţiale care afectează o declaraţie de tip "for" sau o declaraţie "if".

De asemenea, intenţionăm să folosim execuţia simbolică/concolica pentru a modifica zonele de date de intrare pentru a obţine mai multă acoperire a codului. În cele din urmă, o zonă de interes din lunile anterioare pe care ne-am dori să continuăm să lucrăm este să învăţăm o reprezentare latentă a diferitelor tipuri de intrări (d.ex. autoencoders), apoi să folosim aceasta reprezentare pentru a genera noi intrări. In special aceasta metoda se preteaza pentru situatia in care exista mai multe tipuri de intrări.

Analiza de securitate pentru blockchain

Tehnologiile de tip blockchain se bazeaza pe un protocol de consens, care permite adaugarea tranzactiilor de catre nodurile participante intr-un mod distribuit. Aceste protocoale de consens pot fi de mai multe tipuri, printre care amintim Proof of Work (PoW), Proof of Stake (PoS), etc., fiecare cu proprietati de securitate diferite in coditii diverse de atac. Spre exemplu, PoW este vulnerabil la coalitii mai mari de 50% din nodurile participante. Un aspect interesant il constituie studiul a cat de distribuite sunt de fapt sistemele blockchain. Mai exact, in ce conditii ofera protocolul de consens o functionare corecta a sistemului blockchain, analiza comparativa pentru diferite tipuri de protocoale de consens in diferite scenarii adversariale. Desi prin constructie sunt distribuite, modul de functionare a sistemelor de tip blockchain presupune centralizare la nivel software. Un aspect interesant este deci o analiza a sistemelor client, din punct de vedere al disponibilitatii codului sursa, existenta unor vulnerabilitati de tip trapdoor, monopol de management, etc. Un al treilea aspect interesant de cercetare ar fi legatura intre sistemele de tip blockchain si criptografia de prag (threshold cryptography), analizand spre exemplu primitive deja utilizate in sistemele de tip blockchain (d.ex. threshold signatures).

Platforma de tip blockchain Ethereum pune la dispozitie o masina virtuala Turing-completa, EVM, care executa programe denumite smart contracts. Acestea pot fi scrise in limbaje de nivel inalt, Solidity, Viper, Serpent etc. si compilate in bytecode EVM. Majoritatea contractelor au rol financiar, spre exemplu protofele electronice, sisteme electronice de plati etc., insa exista de asemenea aplicatii non-financiare care pot fi folosite in IoT, sisteme de votare si identificare, gestionarea de token-uri, pariuri etc. Vulnerabilitatile in smart contracts pot aparea din diverse cauze: reentrancy si fallback function (atacul DAO), gas limit (numarul de unitati *gas* alocate pentru efectuarea unei tranzatii) nespecificata sau gas limit DoS (denial of service), de exemplu (atacul Governmental), modul de tratare al exceptiilor (atacul King of the Ether Throne) etc. Pentru a identifica si alte tipuri de vulnerabilitati si pentru detectarea erorilor care pot aparea in smart contracts ne propunem verificarea formala a acestora pe baza definirii unei semantici executabile EVM care modeleaza tranzitia starilor retelei.

In acest context si nu in ultimul rand, vom studia si aplicabilitatea domeniului criptografiei bazate pe latici. Aceasta se bucură de câteva proprietăţi remarcabile, poate cea mai importanta fiind aceea că este considerata rezistenta la atacurile computerelor cuantice. Intenţionăm să extindem colecţia curentă de primitive criptografice bazate pe latici adăugând altele noi care nu au fost studiate anterior în acest context, dar de importanţă practică. Mentionam intai de toate semnaturile digitale care au proprietatea de homomorfism pe chei atat aditiv cat si multiplicativ. Scheme criptografice care au proprietatea de homomorfism peste spatiul mesajelor s-a cercetat destul de mult in ultima vreme. Criptarea complet homomorfa (pe spatiul mesajelor) are aplicatii importante in calculul pe date criptate si este construita cu ajutorul laticilor. In ceea ce priveste semnaturile, prima semnatura complet homomorfa pe spatiul mesajelor dateaza din 2014 si a fost definita tot cu ajutorul laticilor. Activitatea noastra de cercetare se va concentra asupra semnaturilor care au proprietati homomorfice pe spatiul cheilor. Aceasta notiune cere ca avand doua semnaturi pentru acelasi mesaj *m* valide sub cheia pk_1 respectiv pk_2, se poate obtine public o semnatura pe acelasi mesaj m valida pentru o cheie *pk'* obtinuta printr-o operatie aplicata peste pk_1 si pk_2. Aplicatiile acestor tipuri de semnaturi includ atacurile de tipul key relationate (related-key attacks) si protocoale asimetrice de schimb de chei in grupuri de utilizatori (asymmetric group key agreement).

Proiect Component 2 - Arhitecturi de securitate pentru IoT

Activitatea 1.3 - Analiza modelelor arhitecturale şi a protocoalelor de reţea folosite în cadrul IoT

Această activitate a a avut ca scop analiza modelelor arhitecturale şi a protocoalelor de reţea folosite în securitatea IoT. Având în vedere că la nivel de reţea tehnologiile sunt cunoscute şi analizate de o multitudine de cercetări anterioare, studiul prezent s-a concentrat în special pe analiza modelelor arhitecturale existente. Acestea sunt caracterizate de o dinamică ridicată şi înţelegerea lor, a punctelor forte sau a vulnerabilităţilor acestora, permite echipei de cercetare să definească şi să implementeze soluţia optimă.

Arhitectura generală a unei soluţii în care datele colectate de dispozitive sunt colectate şi încărcate în Cloud este descrisă în Figura 1.

Mecanismele de securizare propuse în proiect urmăresc atât îmbunătăţirea dispozitivelor periferice (ex. senzori, actuatori, etc.) cu elemente specifice (ex. SE - Secure Element, eSE – embedded Secure Element, UICC – Universal Integrated Circuite, TPM – Trusted Platform Module), cât şi securizarea protocoalelor de comunicaţie utilizate (ex. HTTP(s)-REST, MQTT, CoAP, etc.).

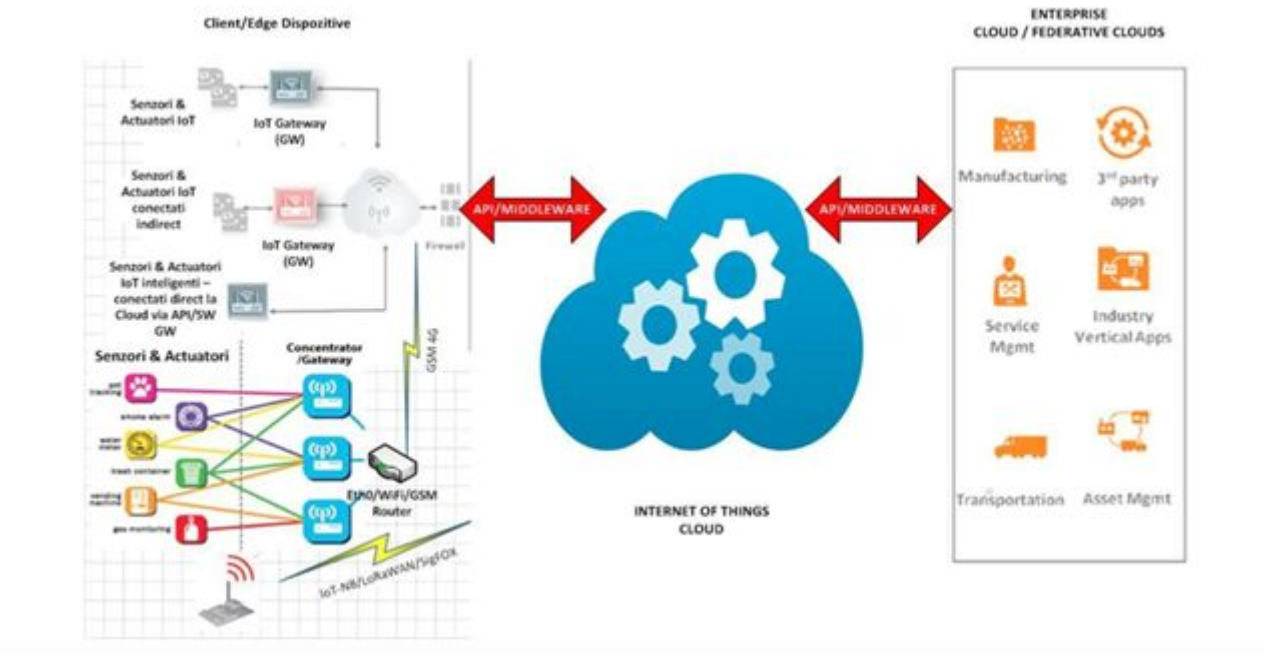

Figura 1. Arhitectura generală a unei solutii IoT

Securitatea este un deziderat ce se obţine prin implementare de software şi hardware în mod transversal la următoarele niveluri:

- Nivel senzori/activatori si IOT noduri/gateway-uri:
  - fizic si de acces la senzori si actuatori
  - firmware si boot securizat la nivel noduri/gateway-uri de tip IoT
  - la nivel de sistem de operare IoT Gateway
  - la nivel de comunicaţie IoT Gateway/IoT Nodes către şi dinspre IoT Cloud.
  - Prin protocoale de comunicaţie specializate (ex. HTTP-REST, MQTT, etc.)
- Nivel IoT Cloud
  - Securitatea infrastructurilor de tip Cloud (IaaS) – securitatea maşinilor virtuale
  - Securitatea containerelor şi imaginilor de containere de aplicaţii
  - Securitatea comunicaţiilor
- Nivel IoT Enterprise
  - Tot ce implica subdomeniul securităţii aplicaţiilor mobile si al aplicaţiilor în Cloud.

Un IoT Gateway este format din sisteme de agregare a datelor senzorilor ce oferă funcţionalităţi, cum ar fi pre-procesarea datelor, securizarea conectivităţii la IoT Cloud, folosind sisteme precum WebSockets, hub-uri evenimente / cozi de mesaje şi „fog computing”. Un IoT Gateway poate fi materializat ca o placa de dezvoltare (ex. Rapberry Pi / Nitrogen) sau ca un Gateway industrial (ex. Matrikon Honeywell sau Intel Windriver) sau chiar ca un Gateway software in Cloud.

Pentru analiză sunt considerate următoarele scenarii de conectivitate de la un dispozitiv direct conectat în două dintre cele mai utilizate IoT Cloud-uri (ambele conectivităţi au fost explorate în cadrul unui Hackthon organizat în cadrul conferinţei www.secitc.eu | http://www.secitc.eu/wp-content/uploads/2018/11/Hackathon_Dev_IoTSecurityJC_2018_v5.pdf realizată în cooperare ASE Bucureşti şi ATM Bucureşti):

1. Oracle IoT CS Cloud (vezi https://docs.oracle.com/en/cloud/paas/iot-cloud/iotrq/QuickStart.html):

- Crearea unui model de dispozitiv: Un model de dispozitiv conţine în format JSON: meta-datele asociate cu tipul dispozitivului, formatele de mesaje asociate cu dispozitivul, resursele web ce pot fi utilizate pentru a trimite comenzi din Cloud către dispozitiv, precum şi capabilităţile dispozitivului în ce priveşte software management. Un administrator poate introduce prin interfaţa web modelul dispozitivului, iar mai apoi îl poate utiliza în format JSON.
- Înregistrarea unui dispozitiv: Acest pas presupune obţinerea pentru fiecare dispozitiv ce urmează a fi înregistrată de la IoT Cloud a datelor pre-înregistrate cu privire la Secretul de Activare (Activation Secret), Identificatorul activării (Actiovation ID) şi a Credentialelor de cont (Account Credentials). Dacă înregistrarea dispozitivului este efectuata cu succes, atunci dispozitivul are asociata în baza de date din Cloud starea de „înregistrat" (Registered) / gata pentru activare. Materializarea unui dispozitiv, virtual sau nu, este data de un fişier de dispozitiv (Device in format JSON) şi identificarea de dispozitiv (Device ID / Endpoint ID).
- Activarea unui dispozitiv: Pasul de activare constă în următorii sub-paşi: 1) obtinerea jetonului/token-ului de activare – JWT (Javascript-object-notation JSON Web Token), 2) obtinerea politicilor de activare in format JSON, 3) generarea cheii RSA (de obicei este e 2048 de biti), 4) Activarea propriu-zisa a dispozitivului. Ultimii doi paşi ce implică semnătură digitală (ex. RSA2048withSHA256) pot fi realizaţi pe un dispozitiv securizat de tip SE (Secure Element), e-SE (embedded Secure Element), i-SE (integrated Secure Element), etc – cu tehnologie JavaCard. Rezultatul hash este obţinut ca aplicare a funcţiei criptografice de dispersie (ex. SHA-256) ce are ca intrare: Secretul de Activare, Identificatorul de Activare, Cheie publică (luată din Certificatul Digital) şi Modelul Dispozitivului (format JSON). Semnătura digitală se aplică pe rezultatul hash obţinut anterior prin aplicarea unui algoritm criptografic de semnătură electronică (ex. RSA cu cheie privată pe 2048 de biţi sau ECDSA – Elliptic Curve Digital Signature).
- Trimiterea de mesaje în Cloud de către dispozitiv: Trimiterea mesajului la dispozitiv consta în obţinerea token-ului de trimitere mesaj şi apoi, transmiterea propriu-zisa a mesajului. Pentru obţinerea token-ului de transmitere mesaj de către dispozitiv către Cloud, dispozitivul trebuie să semneze digital cu cheia privată Identificatorul de dispozitiv (generat de Cloud, Device ID / Endpoint ID). După obţinerea token-ului de mesaj, dispozitivul trimite mesajul întotdeauna alături de token-ul de mesaj şi de Identificatorul de Dispozitiv. În trimiterea mesajului către Cloud este nevoie de semnătura electronică, ce poate fi realizată de un dispozitiv dedicat de tip SE, i-SE, e-SE, etc.

2. Amazon WS IoT Cloud (vezi https://www.nxp.com/docs/en/application-note/AN12131.pdf | https://www.nxp.com/products/identification-and-security/authentication/om3710-a71chard-a71ch-arduino-compatible-development-kit:OM3710-A71CHARD?&tab=Documentation_Tab&linkline=Application-Note):
    - Schema generala utilizând un SE – Secure Element de conectare în IoT AWS Cloud implica utilizarea unui dispozitiv conectat la SE (Secure Element) de tip A71CH. Pentru conexiunea TLS se poate utiliza OpenSSL sau JCA – Java

Cryptography Architecture exclusiv pentru soluţii de securitate software, abordare care este mai puţin fiabilă şi sigură decât utilizarea unui SE – Secure Element dedicat (ex. Element de securitate Java Card NXP A71CH). Dispozitive conectate securizat TLS în IoT AWS Cloud externalizează toate procesele criptografice către acest SE - Secure Element dedicat.

- Conexiunea securizata TLS intre un dispozitiv cu SE – Secure Element şi Cloud implică: generarea de cheie criptografică asimetrică (ex. pentru algoritmul ECDSA – Elliptic Curve Digital Signature) în SE, precum şi generare plus stocare de certificat digital al dispozitivului IoT plus stocare a certificatului digital al Autorităţii de Certificare (AC/CA). Semnăturile electronice şi certificatele sunt utilizate de către SE (Secure Element - ex. NXP A71CH ce rulează o platforma JavaCard) pentru a conecta mai multe dispozitive în IoT AWS Cloud.

Comunicaţia intre SE şi dispozitiv ioT poate utiliza schimb de mesaje standardizat CBOR – Concise Binary Object Representation (http://cbor.io/spec.html | https://tools.ietf.org/html/rfc7049), iar pentru securitate poate fi utilizat COSE - CBOR Object Signing and Encryption (https://tools.ietf.org/html/rfc8152).

Provocarea pentru securizarea dispozitivelor IoT apare pregnant când dispozitivul IoT Gateway nu mai este materializat în hardware, ci rulează software în IoT Cloud.

In ce priveşte securitatea în Cloud a comunicaţiei, deoarece majoritatea soluţiilor de tip IoT Cloud suportă HTTP(s)-REST, atunci elemente de securitate ale comunicaţiei peste stiva de protocoale TCP/Ip trebuie luate în considerare.

Transferul de stări reprezentaţional (REST - REpresentational State Transfer) facilitează comunicarea între sistemele informatice de pe web. Implementarea REST se face folosind următoarele elemente:

- Resursele furnizate de structurile directorului de acces în format URI (Universal Resource Indicator).
- Fişiere structurate (de ex. JSON, XML) ca reprezentare a obiectelor şi atributelor.
- Metode HTTP pentru a trimite mesaje pe web (GET – pentru preluare de resurse, POST – pentru creare de resurse, PUT – pentru actualizări/modificări de resurse, DELETE – pentru ştergeri).
- Starea sesiunii este menţinută numai de clienţi.

REST-ful API oferă o flexibilitate ridicată dezvoltatorilor de software pentru proiectarea, implementarea şi menţinerea aplicaţiilor, datorită principiilor de protocol fără stări şi modularitate ale REST. RESTful API-urile sunt potrivite pentru aplicaţiile web, dar sunt, de asemenea, utilizate cu succes în implementarea de soluţii de tip Cloud Computing şi micro-servicii. Deoarece serviciile REST sunt utilizate pe web, securitatea trebuie să reprezinte principala preocupare şi provocare pentru dezvoltatorii şi integratorii de aplicaţii REST-ful. Conform OWASP (Open Web Application Security Project) şi articolului [35], următoarele tehnologii şi măsuri de securitate pot fi utilizate atunci când API-urile RESTful sunt implementate pentru soluţiile de tip cloud IoT:

- Utilizarea HTTP(S) – HTTP secure – este obligatoriu, deoarece API-ul RESTful transmite prin intermediul informaţiilor web sensibile legate de parole, chei API, tokene (jetoane) Web JSON (JWT) etc. pentru a face autentificarea dispozitivelor IoT sau a

gateway-urilor IoT la infrastructura Cloud. Aceste informaţii trebuie să fie protejate prin criptare pe nivelul de transport al infrastructurii reţelei de calculatoare. HTTPS trebuie să fie implementat atât de dispozitivele client IoT, cât şi de serverele Cloud.

- Controlul accesului – este implementat pentru fiecare punct final REST şi este legat de autentificare şi autorizare. Din motive de eficienţă, deciziile de control al accesului sunt luate la nivel local de către punctul final REST, iar jetoanele/token-ele de acces sunt emise de un server centralizat ce acţionează ca un furnizor de identitate. Există diferite protocoale care trebuie utilizate pentru a gestiona controlul accesului la infrastructura cloud.
- JSON Web Tokens (JWT) – reprezintă structurile de date JSON utilizate de API-ul RESTful pentru controlul accesului. JWT trebuie să fie protejat prin criptare sau cod de autentificare a mesajelor (MAC) pentru a evita lipsa de integritate. JWT este un document RFC care descrie cerinţele, constrângerile şi considerentele de securitate şi oferă exemple pentru acestea. JWT trebuie să fie validată împotriva integrităţii şi revendicărilor conţinute.
- Cheile de acces API – sunt utilizate de către punctul final pentru a crea cereri HTTP către server. Cheile API sunt fluxuri unice de octeţi şi, de obicei, ele sunt incluse în antetul HTTP al solicitării sau în URI-ul propriu-zis. Cu toate acestea, cea de-a doua abordare va expune cheia în istoricul browser-ului şi jurnalele API la nivel de server. Cheile API reprezintă o implementare REST de securitate pentru infrastructura publică cloud unde nu există un control strict al accesului la acesta. Prin urmare, accesările pentru punctul final sunt limitate pentru cei care au chei API. De asemenea, se aplică anumite filtre de acces, în funcţie de categoria finală a clientului.
- Restricţiile aplicate pe metode HTTP-REST – nu toate terminalele (punctele finale / clienţii, dispozitivele software şi hardware IoT) au acces la toate serviciile RESTful furnizate de infrastructura Cloud. Acest lucru este implementat prin restricţionarea unor metode HTTP REST sau prin crearea de liste negre (blacklists) ale punctelor finale / terminalelor / clienţior, dispozitivelor software şi hardware IoT.
- Validarea intrărilor – ar trebui să fie întotdeauna implementată de un serviciu RESTful şi un răspuns de validare este trimis înapoi la punctul final al clientului. Validarea este implementată prin verificarea diferitelor caracteristici ale datelor solicitate: lungimea, intervalul, formatul şi tipul, utilizarea unor tipuri de date puternice, şiruri create de expresii regulate, conţinut neaşteptat sau ilegal, dimensiune cerere HTTP, număr de validări de intrare eşuate pe unitate de timp, având în vedere problemele de securitate ale interpretatorului software/parser-ului utilizat pentru cererile primite.
- Validarea tipurilor de conţinut – tipurile de mesaje REST sunt furnizate de antetul HTTP al solicitării. Conţinutul mesajului trebuie să se potrivească cu tipul furnizat de antetul solicitării. Prin urmare, trebuie implementate măsuri de securitate de către API RESTful atât pentru mesajele de solicitare, cât şi pentru cele de răspuns: respingerea solicitării în cazul în care tipul de conţinut lipseşte sau neaşteptat, nu expune tipul de conţinut neintenţionat, nu sunt incluse în răspunsul tipul de conţinut al solicitării, tipul de conţinut potrivit în funcţie de organismul de răspuns.
- Gestiunea şi managementul punctelor finale / dispozitivelor client IoT ce sunt capabile de cereri HTTP-REST – trebuie să utilizeze mecanisme de autentificare mai puternice pentru a evita expunerea lor la acces necontrolat prin Internet. De asemenea, alte măsuri de securitate pentru a face acest lucru includ utilizarea firewall-urilor,

configurarea adecvată a reţelei de calculatoare, listele de control al accesului. Este preferabil de evitat ac procedurile de management să se realizeze prin Internet.

- Gestionarea erorilor – dacă este implementată necorespunzător, ar putea fi o sursă de informaţii relevante despre serviciul RESTful. Prin urmare, API-ul REST-ful nu trebuie să dezvăluie detalii răspunzând la erorile de mesaje generice. De asemenea, detaliile tehnice nu trebuie returnate clientului.
- Utilizarea jurnalelor de audit – ar putea fi o măsură de securitate pentru a evita posibile atacuri prin salvarea erorilor de validare jeton/token (JWT). De asemenea, jurnalele de audit trebuie dezinfectate de potenţiale atacuri de injectare în log-urile de audit.
- Utilizarea anteturilor de securitate - conţin informaţii corecte despre tipul de conţinut pentru a fi interpretate corect de browser. Astfel de anteturi sunt X-Content-Type-Options şi X-Frame-Options.
- Protejarea informaţiilor sensibile din antetele HTTP - API REST-ful funcţionează cu informaţii sensibile pe Internet (parole, JWT-uri, chei API). Prin urmare, trebuie să împiedice scurgerea credenţialelor de acces, deoarece informaţiile sensibile devin disponibile pe web (istoricul browserului, jurnalele serverului web etc.) în care funcţionează API-ul RESTful. Datele sensibile sunt plasate în corpul solicitării sau în antetul solicitării.
- Codul de returnare HTTP - API-ul RESTful trebuie să returneze codul corect pentru lista de coduri de răspuns disponibile conform protocolului HTTP.

Activitatea 1.4 - Analiza problemelor de securitate şi identificarea de mecanisme de protecţie pentru IoT

Această activitate a avut ca scop analiza problemelor de securitate specifice IoT, definirea arhitecturii şi elaborarea specificatiilor funcţionale pentru framework-ul de securitate bazat pe reputatie pentru IoT.

Studiul tehnic realizat este compus din două părţi. În prima parte a studiului sunt tratate următoarele aspecte:

- ***Arhitecturi generale IoT***. În acest capitol sunt enumerate domeniile în care aplicaţiile IoT pot aduce îmbunătăţiri / transformări substanţiale (industria auto, transporturi, telecomunicaţii, medicină, oraşe inteligente, case inteligente) şi este prezentată structura verticală a acestor aplicaţii (Network, Device, Gateway, Edge/Fog/Cloud);
- ***Probleme generale de securitate în cadrul dispozitivelor IoT***. Acest capitol trece în revistă cele mai importante tipuri de atacuri de securitate specifice IoT, plecând de la nivelul fizic şi terminând cu nivelul aplicaţie;
- ***Securitatea protocoalelor de comunicaţie IoT***. În acest capitol, cele mai importante protocoale de comunicaţie pentru IoT sunt analizate din perspectiva securităţii (LoRA, SigFox, NB-IoT, ZigBee, 6LoWPAN, CoAP, MQTT, LwM2M);
- ***Securitatea platformelor IoT***. În acest capitol sunt analizate o serie de platforme IoT open-source sau comerciale din perspectiva funcţionalităţilor oferite şi a mecanismelor de securitate implementate (Eclipse IoT, OpenIoT, ThingsBoard, Amazon AWS IoT, Microsoft Azure IoT Suite, Google Cloud IoT Core, IBM Watson IoT Platform, Nabto, Bosch IoT Suite);

- ***Implicaţii de securitate ale sistemelor IoT colaborative***. Acest capitol analizează problemele de securitate în scenariile IoT de tipul Smart-City şi subliniază rolul mecanismelor de securitate bazate pe reputaţie şi încredere.

În cea de-a doua parte a studiului este prezentată arhitectura framework-ului de securitate bazat pe reputaţie pentru IoT. Arhitectura generală propusă este o derivată a modelului clasic de infrastructură IoT centralizată, la care au fost integrate elemente de infrastructură descentralizată, la nivelul elementelor de tip Gateway. Descentralizarea infrastructurii este dată de mutarea logicii de control şi administrare la nivelul elementelor Gateway, permiţând astfel un control local al resurselor, modulul Cloud acţionând ca un element de supervizare şi impunere a unor reguli suplimentare în cazul în care configurarea directă a Gateway-urilor nu este posibilă. Dispozitivele IoT acţionează ca noduri terminale, având o singură legătură de comunicaţie, cu elementul Gateway din proximitate, ceea ce presupune că orice conexiune cu un alt nod va fi realizată prin intermediul Gateway-ului, care va gestiona şi monitoriza această conexiune. Modulul Cloud este elementul central al infrastructurii prin care sunt expuse toate serviciile necesare utilizatorilor, atât cele de control a Gateway-urilor şi nodurilor terminale cât şi cele de prelucrare şi stocare a datelor. O vedere de ansamblu asupra arhitecturii este prezentată în Figura 2. După cum se poate observa, arhitectura framework-ului este una modulară, ce integrează diferite componente de securitate la toate nivelurile sistemului IoT şi permite utilizatorului o administrare eficientă a resurselor (dispozitive sau informaţii) prin intermediul modulelor software create pentru acesta.

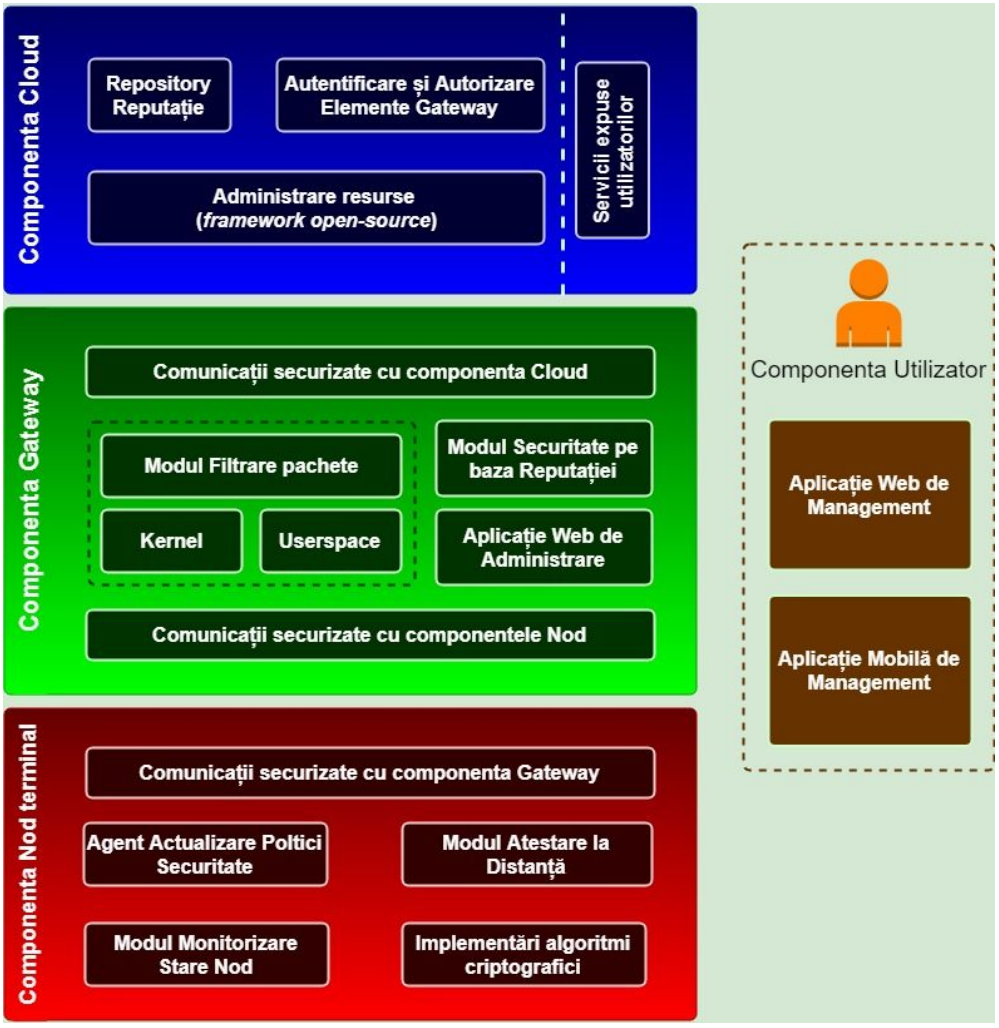

Figura 2. Arhitectura framework-ului de securitate

În continuare sunt detaliate principalele specificaţii şi tehnologii pentru implementarea componentelor software din cadrul framework-ului de securitate.

Componenta Cloud

În cadrul arhitecturii descentralizate propuse, componenta de la nivelul Cloud acţionează ca un element de supervizare şi impunere a unor reguli şi politici de securitate suplimentare. Fiind un element central, componenta Cloud trebuie să asigure mecanismele necesare interfaţării cu echipamentele de tip Gateway, realizând conexiuni securizate, folosite atât pentru administrarea modulelor conectate, cât şi pentru gestionarea informaţiilor primite de la acestea. În acelaşi timp, este necesară asigurarea unor servicii prin intermediul cărora utilizatorii să poată avea acces la datele furnizate de nodurile proprii şi la anumite elemente de control a acestora. Pentru atingerea acestor obiective, arhitectura propusă foloseşte o soluţie existentă, care facilitează comunicaţiile dintre elementele Gateway şi componenta Cloud, şi pe care o completează cu mecanismele de securitate propuse în framework, specifice nivelului Cloud. Descentralizând arhitectura prin mutarea funcţionalităţilor principale la nivelul echipamentelor de tip Gateway, din punct de vedere al securităţii, componenta Cloud constituie un element pasiv al framework-ului, oferind, în afară de crearea legăturilor securizate, autentificarea şi autorizarea Gateway-urilor şi a utilizatorilor, un serviciu de *repository* al nivelelor de reputaţie ale tuturor elementelor din cadrul sistemului IoT conectat la componenta Cloud. Acest *repository* acţionează ca o bază de date, fiind consultată şi actualizată de elementele Gateway, în momentele de recalculare a reputaţiei nodurilor conectate. Prin intermediul interogărilor acestui *repository*, elementele Gateway sunt capabile de a cunoaşte “reţeaua valorilor de reputaţie” ale nodurilor conectate la un alt element Gateway, această caracteristică fiind folosită, fie în cazul stabilirii unor conexiuni între nodurile locale şi nodurile altui element Gateway, fie pentru preluarea unor valori direct de către Gateway de la alte noduri (Ex.: pentru efectuarea unor comparaţii de acurateţe a valorilor preluate de nodurilor locale, atunci când anumite noduri au un comportament suspect).

Componenta Gateway

Gateway-ul este elementul central al framework-ului IoT de securitate propus, acesta având două sarcini principale: implementarea unui modul de calcul al reputaţiei şi un mecanism de filtrare a pachetelor.

Dispozitivele end-point comunică cu Gateway-ul prin intermediul protocolului MQTT, Gateway-ul fiind un server MQTT care preia comenzi de la dispozitivele client. Comenzile dispozitivelor client sunt de două categorii: publicare de informaţii şi cerere de informaţii. Gateway-ul preia aceste comenzi şi execută acţiunea corespunzătoare la nivelul modulului Cloud. Astfel, în cazul cererii de publicare, Gateway-ul preia mesajul MQTT şi publică informaţia la nivelul modulului Cloud. În cazul cererii de informaţii, Gateway-ul caută obiectele eligibile din domeniul Cloud, execută un algoritm care determină cel mai de încredere dispozitiv, preia ultimul mesaj înregistrat şi îl furnizează prin MQTT dispozitivului client care a executat cererea.

Ca mecanisme de securitate, există 3 sub-module:

- *sub-modulul care securizează comunicaţia între Gateway şi platforma Cloud.* Acest modul este responsabil şi cu autorizarea Gateway-ului;
- *sub-modulul care securizează informaţiile de la dispozitivul client la Gateway, prin filtrarea pachetelor:* Framework-ul de securitate propus implementează un modul de filtrare inteligentă a pachetelor la nivelul Gateway-ului, astfel încât aceste atacuri să poată fi minimizate. Protocoalele tratate de către modulul de filtrare a pachetelor sunt MQTT/MQTT-SN şi CoAP, acestea fiind cele mai des întâlnite în cadrul reţelelor IoT. La nivel structural, modulul de filtrare al pachetelor este compus, la rândul său, din două

sub-module: componenta de filtrare a pachetelor în kernel şi componenta de filtrare a pachetelor în *userspace*. Modulul de filtrare al pachetelor tratează nivelul aplicaţie al protocoalelor MQTT/MQTT-SN şi CoAP, prin urmare câmpurile de nivel inferior ale pachetului de date pot fi filtrate cu mijloace bine-cunoscute, precum iptables/ebtables (netfilter). Filtrarea pachetelor de date la nivelul kernelului este implementată prin mecanismul eBPF (Extended Berkeley Packet Filter). Prin intermediul acestui mecanism, modulul de filtrare al pachetelor permite nodurilor IoT autentificate şi autorizate să îşi instaleze propriile rutine de filtrare a pachetelor. Astfel, capabilităţile nodurilor IoT sunt extinse la nivelul Gateway-ului, care alocă resurse (cicli de procesor, memorie) pentru fiecare nod terminal. În cazul dispozitivelor terminale sau a Gateway-urile care nu sunt compatibile cu tehnologia de tip BPF (Ex.: aplicaţii *bare-metal* sau sisteme de operare *lightweight* precum Contiki, care nu fac parte din familia Linux/BSD), framework-ul de securitate propus prezintă şi un modul de filtrare a pachetelor la nivel de *userspace*, care permite proceselor să filtreze pachetele pe baza unor reguli de control al accesului (ACL). Modulul de filtrare al pachetelor din *userspace*, are o interfaţă bine-definită, acesta permiţând instanţierea unui modul de filtrare proprietar care respectă aceeaşi interfaţă. Interfaţa modulului de filtrare din *userspace* a pachetelor constă într-o colecţie de rutine care trebuie apelate de către procesul care gestionează pachetele de date, cum ar fi: rutine de inspectare a pachetului, rutine de interogare a stării de autentificare a emiţătorului pachetului. Astfel, decizia de rejectare a pachetului este delegată componentei de filtrare, modulul care procesează pachetul acţionând pe baza deciziei luate de către modulul de securitate. Modulul de filtrare din *userspace*, precum modulul de filtrare din kernel, utilizează o paradigmă cooperativă între nodurile terminale şi Gateway, care pot veni în întâmpinarea caracterului dinamic al reţelelor IoT.

- *sub-modulul care determină cel mai de încredere dispozitiv:* un mecanism de securitate bazat pe reputaţie aduce diferite beneficii şi presupune costuri relative scăzute pentru anumite secţiuni ale reţelei IoT. Însă, diversitatea sistemelor IoT şi a componentelor din care acestea sunt formate, îngreunează procesul de proiectare şi definire a elementelor unui framework de securitate generic. Cu toate că prin framework-ul propus se încearcă o abstractizare şi modularizare a mecanismelor de securitate, astfel încât acesta să poată fi integrat în reţele IoT variate, pentru a simplifica descrierea mecanismului de calcul a reputaţiei implementat în cadrul framework-ului propus, vor fi prezentate aspecte legate de două scenarii cheie pentru reţele IoT, anume *Smart-City* şi *Smart-Home*, ultimul putând fi considerat o derivată a scenariului *Smart-City* datorită implementării la o scară mult mai redusă. Într-un scenariu de tip *Smart-City*, aspectul principal constă în interacţiunea dintre cetăţeni şi obiectele inteligente, informaţia procesată de către acestea trebuind a fi livrată nu doar către persoane, ci şi către alte obiecte. Livrarea informaţiei înseamnă şi primirea unui *feedback*, atât din partea cetăţenilor, cât şi din partea celorlalte obiecte. Feedback-ul este folosit în calcularea unui grad de încredere al unui obiect inteligent, astfel că informaţia publicată de către un obiect care are un grad de încredere ridicat are şanse mai mari sa fie aleasă şi procesată de către consumatorii de informaţie. Modulul de calcul al reputaţiei utilizează Gateway-ul care preia informaţia de la dispozitivele inteligente terminale şi o trimite mai departe către modulul Cloud. Unul dintre rolurile principale ale modulului Cloud este acela de a furniza o platformă de tipul *S2aaS* şi de a acţiona ca un *repository* pentru datele şi tranzacţiile executate de către nodurile IoT. Dacă un dispozitiv publică o informaţie de încredere, folosită cu succes de către alt obiect, atunci o fracţiune din încrederea ultimului obiect este transferată către primul. Acest transfer de încredere se realizează prin intermediul mijloacelor puse la dispoziţie de către platforma Cloud. În cazul în care platforma Cloud este integrată cu o reţea socială, feedback-ul pozitiv se poate transfera prin mijloace proprietare platformei de reţea socială, cum ar fi acţiunile de *like* sau *comment*. Transferul de încredere este repetat pentru fiecare nod din lanţul

care a contribuit cu date la publicarea informaţiei finale. Alte dispozitive care au nevoie de anumite informaţii îşi pot îmbunătăţi domeniul senzorial prin preluarea informaţiilor necesare de la cele mai de încredere dispozitive din domeniul Cloud.

Un punct extrem de important al arhitecturii de securitate prezentate este acordarea feedback-ului către dispozitivele care furnizează o informaţie. Feedback-ul poate fi acordat atât de către alte dispozitive, cât şi de către utilizatorii umani. Având în vedere aceste informaţii, feedback-ul primit de la utilizatorii umani are o pondere mai mare în determinarea nivelului de încredere.

Componenta Nod terminal

Nivelul nodurilor terminale este format din totalitatea dispozitivelor IoT conectate la sistem, prin intermediul unor elemente Gateway. Datorită resurselor limitate disponibile în cadrul acestor dispozitive, funcţionalităţile componentei de securitate sunt specificate într-un format redus, dar care implementează mecanismele esenţiale asigurării securităţii întregului sistem. Prin urmare, fiecare nod are integrate următoarele modulele de securitate de bază:

- *un agent pentru actualizarea politicilor de transmisie şi recepţie de la nivelul Gateway-ului la care este conectat nodul:* Politicile de securitate de la nivelul Gateway-urilor sunt definite, în principal, ca o metodă de filtrare a traficului la/de la nodurile terminale, însă nu ţin cont de performanţele tehnice ale dispozitivelor care sunt conectate. Prin urmare, este posibil ca un trafic intens în cadrul reţelei să diminueze considerabil rezerva de energie a unuia dintre noduri sau să afecteze rata de procesare şi transmisie a pachetelor de către acesta. Pentru a minimiza procentajul de clasificare falsă a reputaţiei unor noduri, datorită scăderii performanţelor computaţionale sau a inactivităţii prin consumarea bateriei, nodul poate solicita actualizarea politicilor de la nivelul Gateway-urilor, prin transmiterea unor Cereri de Actualizare Politici (CAP) la un anumit interval de timp, până când va fi primit un răspuns. Gateway-ul va transmite un mesaj Cerere de Actualizare Politici Acceptată (CAPA), caz în care se va opri transmiterea CAP de către nod, sau Cerere de Actualizare Politici Respinsă (CAPR), caz în care se va modifica şi retransmite CAP.
- *un modul de atestare la distanţă a nodului:* modulul de atestare vine ca o componentă complementară celorlalte mecanisme de securitate implementate la nivelul nodului, precum folosirea algoritmilor criptografici pentru criptarea datelor transferate, şi nu ca o componentă unică de securitate. În majoritatea variantelor existente la momentul actual, mecanismele de atestare reprezintă module software care rulează prima dată după alimentarea dispozitivului, executând operaţiuni de verificare a existenţei aplicaţiilor ţintă, a anumitor funcţionalităţi ale platformei hardware, precum şi a integrităţii acelor aplicaţii sau a altor zone de date în care sunt stocate informaţii importante pentru dispozitiv/reţea. Aceste verificări sunt executate într-o serie de auto-teste care folosesc, pentru compararea valorilor obţinute cu valorile de conformitate, zone sigure de stocare a unor valori hash sau chei criptografice. Însă, prin proiectarea framework-ului de a se interfaţa cu o gamă variată de dispozitive IoT, există o limitare a costurilor echipamentelor existente la nivelul nodurilor terminale, prin urmare existenţa unor zone protejate de memorie va fi considerată opţională. Pornind de la aceste considerente menţionate anterior, modulul de atestare va fi realizat pentru verificarea rezultatelor la distanţă, prin compararea cu valorile conforme, stocate la nivelul Gateway-ului şi preluate la nivelul nodurilor printr-un canal de control securizat. În plus, modulul de atestare va putea fi apelat de la distanţă de către Gateway, în cazul în care “comportamentul” nodului devine suspect.
- *un modul pentru monitorizarea stării nodului şi stabilirea legăturilor pe planul de control cu Gateway-ul:* framework-ul propus va integra un modul de monitorizare a stării nodului, prin care vor fi măsurate anumite metrici specifice tipului de nod. Acest modul va avea rolul nu doar de colectare a informaţiilor, ci şi de determinare iniţială a situaţiei depăşirii unor praguri de performanţă admise, stabilite la momentul instalării nodurilor. În

funcţie de rezultatele acestor verificări, nodul va transmite anumite alerte (în cazul depăşirii pragurilor valorice stabilite) sau atenţionări (în cazul apropierii de valorile maximale cu un anumit pas stabilit de administrator) către Gateway-ul la care este conectat, în vederea stabilirii unor măsuri de securitate. De asemenea, tot pe baza acestor rezultate, nodul va încerca stabilirea unor actualizări ale politicilor de securitate aplicate asupra traficului de date, sub forma unor Cereri de Actualizare Politici (CAP), definite în capitolele anterioare. Pornind de la datele de stare provenite de la senzori, Gateway-ul îşi poate crea o imagine de ansamblu a stării reţelei, putând adapta politicile de trafic şi gestionarea activităţilor astfel încât să mărească durata de funcţionare a anumitor noduri sau să menţină acelaşi nivel de încărcare, în cazul în care nodurile respective colectează informaţii pentru care latenţa colectării este un element critic.

De asemenea, pentru asigurarea unei securităţi sporite a datelor transmise de nod, framework-ul va pune la dispoziţie diferite implementări ale unor algoritmi criptografici, ce vor putea fi utilizate în aplicaţiile client dezvoltate în funcţie de specificul fiecărui sistem IoT în care este desfăşurat acest framework.

Modulul de administrare, jurnalizare şi alertare

Un framework IoT de securitate tratează o suită vastă de scenarii de implementare, astfel că gestionarea la distanţă a unei reţele IoT este un element extrem de important. Din perspectiva securităţii IoT, un administrator trebuie să aibă vizibilitate la nivelul întregii arhitecturi de dispozitive inteligente, să obţină statistici de securitate în timp real şi să poată controla nodurile IoT, prin aplicarea unor politici de securitate. Modulul de administrare al framework-ului de securitate conţine două sub-module principale:

- *aplicaţia web de management:* este o aplicaţie web care rulează la nivelul componentei Cloud şi tratează, în special, zona securităţii IoT *enterprise* (în mod particular, scenariul *Smart-City*). conţine o interfaţă grafică prin care un administrator de securitate poate să instaleze diferite politici de filtrare ale pachetelor şi parametrii de calcul ai reputaţiei la nivelul Gateway-urilor şi la nivelul nodurilor IoT. Această aplicaţie de management conţine un modul care permite stabilirea unei relaţii de încredere între componenta Cloud şi componenta Gateway, fie prin folosirea certificatelor digitale X.509 (autentificare mutuală), fie prin folosirea cheilor asimetrice de tip *raw* sau a cheilor simetrice pre-partajate în cadrul structurilor de tip JWT. În ceea ce priveşte modulul de calcul al reputaţiei, administratorul de securitate poate să configureze parametrii care ţin de optimizarea consumului la nivel de Gateway (parametrii care controlează compromisul dintre nivelul de securitate şi disponibilitate). De asemenea, administratorul poate să introducă într-o structura de tip *blacklist*, o serie de dispozitive sau categorie de dispozitive, astfel încât acestea să nu participe la procesul de oferire al datelor în structura *SaaS*. În consecinţă, dispozitivele care sunt în structura *blacklist* nu vor fi incluse în graful de calculare al reputaţiei. La nivelul aplicaţiei web de management, administratorul poate să instaleze diferite filtre care permit alertarea şi monitorizarea procesului de calcul al reputaţiei. În ceea ce priveşte monitorizarea modulului de filtrare de pachete de la nivelul Gateway-ului, administratorul poate să instaleze, prin intermediul aplicaţiei web, atât politici de filtrare în *userspace*, cât şi politici de filtrare în kernel. Aplicaţia web de management permite instalarea unor senzori, la ambele module de filtrare, pentru a vizualiza rata cu care sunt procesate pachetele, precum şi configurarea ratei de eşantionare a statisticilor de trafic. De asemenea, aplicaţia web va furniza statistici cu diferite câmpuri din pachetele de date (inclusiv câmpuri din protocoalele MQTT şi CoAP), care să permită unui administrator să evalueze tipul de trafic din reţea şi să instaleze politicile de securitate necesare. Aplicaţia web va avea o structură modulară şi va permite unui administrator să configureze diferite *end-point*-uri de tip REST care să fie apelate în cazul unei alerte de securitate (Ex.: în cazul depăşirii ratei normale de procesare a pachetelor pe secundă). Astfel, framework-ul IoT de

securitate poate fi instalat într-o infrastructură deja existentă şi poate fi extins prin integrarea unor module terţe, de procesare a datelor. Aplicaţia web permite exportarea, în format JSON, a tuturor informaţiilor care sunt afişate în interfaţa grafică, permiţând unui administrator să folosească scripturi proprietare pentru procesarea adiţională a datelor sau chiar integrarea într-un sistem de *machine learning*.

- *aplicaţia mobilă de management:* este o aplicaţie Android care are rolul de a gestiona, la nivel local, dispozitivele IoT şi de a fi folosită de către un utilizator pentru autentificarea şi autorizarea comenzilor primite de către dispozitivele IoT de la modulul Cloud. Această aplicaţie tratează scenariul de utilizare cu un număr redus de dispozitive IoT (în mod particular, scenariul *Smart-Home*). Aplicaţia mobilă de management este un instrument software de securitate care permite utilizatorului să înroleze dispozitivele în domeniul de securitate *Smart-Home* prin stabilirea unei relaţii de încredere mutuală, folosind chei criptografice asimetrice şi ulterior să autentifice şi să autorizeze comenzile primite de către dispozitivul IoT folosind un protocol de tip *challenge-response*. Înrolarea dispozitivul IoT în domeniul de încredere *Smart-Home* se realizează înainte de conectarea acestuia cu modulul Cloud proprietar, folosind reţeaua HAN. Principalul scenariu securizat de către aplicaţia mobilă de management este acela al comenzilor primite de către un dispozitiv IoT de la un modul Cloud terţ. Astfel, aplicaţia de management permite realizarea unui canal de autentificare şi autorizare punct-la-punct, prin executarea a diferite procese specifice sistemului IoT în care este integrat framework-ul. Aplicaţia mobilă permite vizualizarea dispozitivelor înrolate, starea curentă a acestora, precum şi un log de audit al comenzilor executate. De asemenea aplicaţia mobilă permite şi trimiterea de comenzi către dispozitivul IoT, acţionând ca un element securizat de control la distanţă. Pentru această capabilitate, dispozitivul IoT publică către aplicaţia mobilă, o listă de comenzi suportate, care vor fi afişate utilizatorului prin intermediul interfeţei grafice.

Prin structura flexibilă şi mecanismele de vizualizare şi alertare implementate, modulul de administrare permite încorporarea framework-ului IoT de securitate într-o infrastructură deja existentă, adăugarea de noi module şi integrarea cu alte instrumente de analiză de securitate.

Proiect Component 3 - Mediu distribuit pentru exerciţii şi cercetare avansată în domeniul securităţii cibernetice

Activitatea 1.5 - Inventarierea soluţiilor existente, identificarea tehnologiilor aplicabile şi elaborarea specificaţiilor pentru platforma cyber range

Această activitate a avut ca scop inventarierea soluţiilor existente, identificarea tehnologiilor aplicabile şi elaborarea specificaţiilor pentru platforma cyber range distribuită ce va fi folosită la nivelul partenerilor pentru organizarea în comun de exerciţii şi experimentări în domeniul securităţii cibernetice.

Raportul de cercetare realizat este compus din două părţi. În prima parte a raportului sunt tratate următoarele aspecte:

- ***Importanţa şi rolul platformelor cyber range***. În acest capitol este evidenţiată nevoia pregătirii de specialişti având în vedere ameninţările actuale din spaţiul cibernetic şi este prezentat rolul platformelor cyber range în crearea de medii de antrenament care să reflecte cât mai bine condiţiile reale.

- ***Tipuri de exerciţii şi competiţii de apărare cibernetică***. Acest capitol face o analiză a principalelor tipuri de exerciţii de apărare cibernetică folosite în momentul de faţă pentru antrenarea specialiştilor în domeniul securităţii cibernetice:
    - **Network Defense** - echipa îşi securizează şi protejează reţeaua împotriva atacurilor care urmăresc accesul la sisteme, furtul de date sau sabotarea infrastructurii;
    - **Penetration Test** - echipa atacă o infrastructură cu ajutorul unor tool-uri predefinite sau proprii, prin utilizarea unor exploit-uri cunoscute sau noi, pentru identificarea vulnerabilităţilor sistemelor informatice;
    - **Incident Response** - echipa este verificată dacă este capabilă să detecteze anumite tipuri de atacuri, să evalueze şi să raporteze incidentele de securitate;
    - **Digital Forensics / Reverse Engineering** - echipa identifică artefactele digitale în urma unui atac asupra unui sistem, pentru stabilirea modului de acţiune şi al vulnerabilităţilor exploatate;
    - **Secure Programming** - echipa scrie software care trebuie să reziste atacurilor cibernetice sau evaluează securitatea aplicaţiilor deja dezvoltate.

  De asemenea, tot în acest capitol sunt trecute în revistă principalele competiţii organizate în domeniu: U.S. Cyber Challenge, National Collegiate Cyber Defense Competition (NCCDC), CyberPatriot, DefCon CTF şi European Cyber Security Challenge (ECSC).
- ***Exemple de platforme cyber range***. Acest capitol face o trecere în revistă a celor mai cunoscute platforme de tip cyber range, evidenţiind modelul arhitectural, tehnologiile folosite şi avantajele oferite:
    - **XNet.** Platformă dezvoltată de CERT-ul Universităţii Carnegie-Melon, în special pentru agenţiile guvernamentale şi contractorii acestora. Platforma oferă acces în timp real, printr-o interfaţă Web, la evenimente de securitate din cadrul sistemelor şi reţelelor. Platforma suportă implementarea de scenarii diverse, de la apărare până la ameninţări din interiorul reţelei. De asemenea, platforma permite monitorizarea şi evaluarea activităţii cursanţilor şi comunicarea între membrii echipelor.
    - **National Cyber Range.** Platformă dezvoltată de Departamentul Apărării din SUA ce permite testarea securităţii unor reţele simulate la scară mare, realistice şi complexe. Poate susţine peste 40.000 noduri şi topologii complexe, inclusiv servicii de Web şi e-mail, similare cu cele întâlnite în Internet. Platforma poate testa rezilienţa sistemelor în faţa atacurilor cibernetice în medii similare cu cele reale, sau evalua capabilităţile unor soluţii ofensive sau defensive. Platforma este destinată doar agenţiilor guvernamentale, contractorilor şi companiilor care administrează infrastructuri critice (energie, apă, gaz, combustibil, etc) şi care îşi pot evalua astfel securitatea sistemelor, la costuri extrem de mici.
    - **SANS Netwars.** Reţea interactivă care simulează mediul Internet, utilizată pentru analiza atacurilor cibernetice. Utilizează un mediu virtual pentru a permite „jucătorilor” să deruleze atacuri cibernetice. Prin interacţiunea cu acest poligon, studenţii învaţă tehnici şi tactici ofensive, pentru a identifica şi exploata vulnerabilităţi ale infrastructurii virtuale, dar şi metode de securizare a sistemelor şi de derulare a investigaţiilor digitale. Scenariile sunt dezvoltate pe diverse

niveluri de dificultate şi permit acumularea de cunoştinţe şi creşterea nivelului de expertiză al participanţilor.

- **EduRange.** Este o platformă care funcţionează pe Amazon AWS Cloud care asigură stocarea şi rularea exerciţiilor. Scenariile prezente pe această platformă acoperă: analiza traficului de reţea, detecţie malware, analiză malware, blue team, social engineering, securitate Web. Majoritatea exerciţiilor necesită cunoştinţe minimale de Linux şi de înţelegere a conceptelor de securitate.
- **Emulab.** Platformă academică dezvoltată de Universitatea din Utah, SUA, care utilizează software open source. Platforma este distribuită la peste 20 de contribuitori, în special alte universităţi şi institute de cercetare. Platforma are ca scop dezvoltarea şi cercetarea în domeniul sistemelor distribuite. Accesul este gratuit, iar cercetătorii îşi pot crea propriul laborator Emulab. Platforma este utilizată pentru desfăşurarea de experimente riguroase, cu o fidelitate cât mai buna, repetabile şi asigură acurateţea evaluării şi măsurătorilor. Permite generarea şi captarea de trafic de reţea, cu ajutorul unor tool-urilor open source.
- **OCCP (Open Cyber Challenge Platform)**. Platformă dezvoltată de Universitatea Rhode Island pentru antrenamentul echipelor de securitate cibernetică. Platforma este dezvoltată pe tipicul exerciţiilor de securitate şi permite antrenamentul în echipe de tip Blue Team (apărători), Red Team (atacatori), Gray Team (trafic normal), White Team (management şi scoring). Acţiunile unei echipe pot fi înlocuite, în funcţie de obiectivul exerciţiului, cu scripturi automate.
- **DETERLab**. Platformă pentru testare şi cercetare în domeniul securităţii reţelelor. Printre capabilităţi se regăsesc virtualizarea multi-nivel, federalizarea, experimentele cu mai mulţi actori. Permite rularea de exerciţii realistice, la scară largă. Platforma este orientată mai mult pe cercetare şi permite testarea unor soluţii noi sau care nu sunt verificate suficient în condiţii reale de funcţionare.
- **GENI (Global Environment for Network Innovations).** Una dintre cele mai versatile şi mature platforme federalizate este GENI. Platforma permite cercetătorilor şi specialiştilor să acceseze o infrastructură distribuită, creată prin partajarea resurselor din peste 80 de Universităţi din SUA şi Canada.
- **FIRE (Future Internet Research and Experimentation).** FIRE este o iniţiativă similară cu FIRE, care a pornit în 2008 şi este finanţată de Uniunea Europeană. FIRE are drept scop conectarea resurselor existente la diverse universităţi şi organizaţii, pentru realizarea unui mediu standardizat pentru cercetare şi testare în domeniul securităţii cibernetice.

- ***Unelte pentru specificarea şi deployment-ul exerciţiilor de apărare cibernetică***. Acest capitol este dedicat uneltelor care permit definirea scenariilor şi deployment-ul automat al maşinilor virtuale necesare derulării unui exerciţiu:
  - **RADICL (Re-configurable Attack-Defend Instructional Computing Laboratory).** RADICL este primul limbaj destinat laboratoarelor de securitate cibernetică, dezvoltat la Universitatea din Idaho în 2004. Obiectivul acestui limbaj este de a permite studenţilor şi cercetătorilor să proiecteze, implementeze şi să ruleze tutoriale, exerciţii şi experimente într-un mediu virtual, izolat, sigur şi controlat. RADICL simulează reţele reale prin intermediul tehnologiei de virtualizare VMWare vSphere. Utilizatorii pot configura şi rula reţele formate din

sute de sisteme virtuale, care pot fi accesate simultan de grupuri de studenţi pentru derularea exerciţiilor.
    - **ADLES (Automated Deployment of Lab Environments System)**. ADLES este format dintr-un limbaj formal care permite definirea mediului virtual educaţional şi un set de instrumente pentru popularea automată a acestuia cu maşini şi reţele virtuale. Cu aceste unelte, se poate crea în mod automat un mediul virtual de antrenament pentru un curs, laborator sau competiţie de securitate cibernetică.
- ***Unelte pentru provizionarea şi managementul configuraţiei maşinilor virtuale***. În acest capitol sunt descrise uneltele ce pot fi folosite pentru managementul infrastructurilor de tip cyber range / Cloud:
    - **Terraform**. Aplicaţie open source dezvoltată de HashiCorp pentru managementul infrastructurilor de tip Cloud. Cu ajutorul unor fişiere de configurare se creează sau modifică mediul sau infrastructura dorită a se realiza. Terraform poate fi utilizat pe o mare varietate de infrastructuri de tip cloud, inclusiv Docker, Amazon AWS, MS Azure, VMWare vShpere;
    - **Vagrant**. Vagrant este, de asemenea, o soluţie open source, utilizată în manipularea mediilor virtualizate, implicit şi a maşinilor virtuale. Funcţionalităţile oferite sunt expuse utilizatorilor prin intermediu unei interfeţe de tip linie de comandă şi a unui interpretor pentru parsarea fişierelor cu definiţii specifice configurării unui mediu virtualizat, identificate ca Vagrantfile. Vagrant poate fi văzut ca o completare a soluţiilor de virtualizare existente, prin introducerea unui layer suplimentar de control, care simplifică sarcinile utilizatorului în ceea ce priveşte folosirea unor soluţii de virtualizare variate în diferite tipuri de sisteme;
    - **Ansible.** Ansible este o soluţie open source aparţinând Red Hat Inc. (dezvoltată iniţial de AnsibleWorks Inc.) care permite automatizarea managementului configuraţiei unei infrastructuri Cloud. Similar majorităţii soluţiilor din această categorie, Ansible este construit pe modelul element de control - noduri, unde elementul de control reprezintă punctul în care sunt definite etapele automatizării nodurilor;
    - **SaltStack**. SaltStack este o soluţie open source ce oferă funcţionalităţi pentru managementul configuraţiei unei infrastructuri (menţinerea unor noduri într-o stare definită) şi pentru execuţia distribuită şi de la distanţă a unor comenzi sau interogări ale nodurilor conectate;
    - **CHEF** este o platformă pentru managementul şi automatizarea configurării maşinilor virtuale. Platforma este dezvoltată de OpsCode şi are componente open source dar şi licenţiate. CHEF poate fi utilizat la configurarea platformelor care rulează pe Cloud-uri publice, cum sunt Amazon, OpenStack, Rackspace dar şi a celor care rulează local pe platforme de virtualizare;
    - **Puppet.** Puppet este o platformă open source dar şi licenţiată, orientată spre managementul infrastructurilor. Platforma este compusă dintr-un server de management al nodurilor (Puppet Server), agenţi software (Puppet Agent) instalaţi pe fiecare maşină, care comunică şi se sincronizează cu serverul de management, consola Web (Puppet Enterprise Console) pentru raportare şi controlarea infrastructurii şi serviciul de stocare a datelor (PuppetDB).
    -

- ***Tehnologii de virtualizare şi emulatoare de reţea***. După prezentarea conceptelor specifice virtualizării, în acest capitol sunt trecute în revistă cele mai importante tehnologii de virtualizare disponibile în momentul de faţă: VMware (ESXi & Workstation), Xen, Microsoft Hyper-V, Oracle VirtualBox, Linux KVM & QEMU, Linux Containers (LXD & Docker), LibVirt, OpenStack. De asemenea, este prezentat emulatorul de reţea GNS3 care permite combinarea de dispozitive virtuale cu echipamente hardware pentru simularea de infrastructuri de reţea complexe.
- ***Platforme e-Learning***. Acest capitol este dedicat platformelor e-Learning utilizate la nivelul universităţilor partenere în proiect (Moodle şi Sakai) prin intermediul cărora poate fi livrat conţinutul educaţional către studenţi. Aceste platforme sunt analizate din perspectiva integrării cu laboratoare virtuale create şi rulate prin intermediul platformelor cyber range.

În cea de-a doua parte a raportului sunt prezentate arhitectura şi specificaţiile funcţionale ale platformei cyber range distribuită ce urmează a fi implementată la nivelul universităţilor partenere în proiect.

Un poligon cibernetic (cyber range) este o platformă fizică sau virtuală ce poate fi folosită pentru pregătirea specialiştilor sau pentru experimentări în domeniul securităţii cibernetice. În acest mediu, pe baza unor scenarii, personalul poate fi instruit să folosească instrumente, tactici şi strategii defensive sau de atac. Un cyber range poate fi, de asemenea, utilizat pentru dezvoltarea de noi tehnologii de securitate cibernetică.

Pentru a reduce costurile de implementare, se preferă folosirea tehnologiilor de virtualizare pentru implementarea sistemelor sau reţelelor. Acest lucru nu este însă posibil întotdeauna, mai ales atunci când scenariul implică folosirea de sisteme embeded sau de sisteme pentru control industrial (ICS / SCADA).

Pentru a folosi la maxim resursele existente şi pentru a beneficia de expertiza acumulată, o tendinţă firească este de a interconecta infrastructurile aparţinând diferitelor organizaţii pentru a crea poligoane cibernetice distribuite (federated cyber range). Conceptul de cyber range distribuit este prezentat schematic în Figura 3.

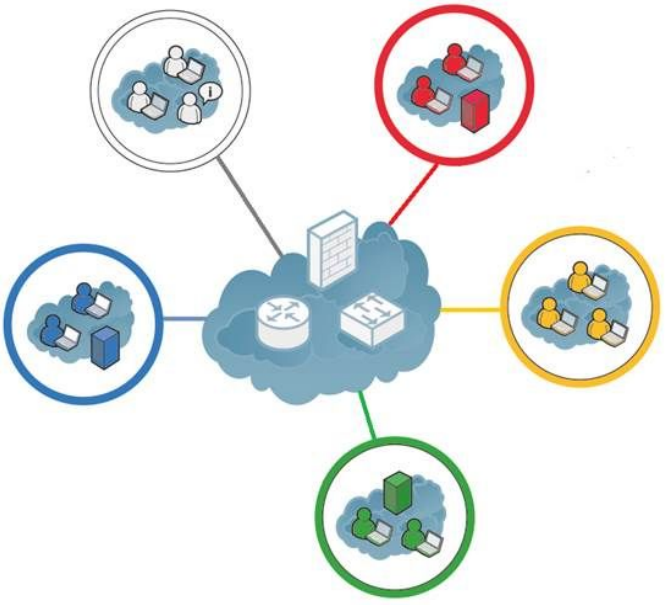

Figura 3: Poligon cibernetic distribuit

Avantajele federalizării platformelor cyber range sunt evidente:

- Partajarea resurselor - pe lângă resurse de procesare şi stocare, se pot partaja scenarii, conţinut educaţional (cursuri online) sau echipamente specializate (generatoare de trafic, sisteme SCADA, etc);
- Organizarea în comun de exerciţii - funcţie de competenţe, fiecare organizaţie poate juca roluri diferite (red team, blue team, white team, etc);
- Evaluarea produselor şi aplicaţiilor software dezvoltate şi testarea securităţii produselor se va face într-un mediu controlat, care simulează cât mai fidel mediul real.

Arhitectura logică a platformei cyber range propusă a fi implementată la nivelul universităţilor partenere este prezentată în Figura 4. Componentele principale ale acestei infrastructuri sunt următoarele:

- Infrastructura de bază - servere, SAN, echipamente de reţea (SW, routere), FW, IDS / IPS, echipamente specializate (ICS / SCADA);
- Nivel virtualizare / izolare - tehnologii pentru crearea şi rularea de maşini şi reţele virtuale;
- Catalog de scenarii - descrierea modului de configurare al platformei pentru diferite exerciţii (network defense, penetration testing, digital forensics, etc);
- Generator de trafic - simularea activităţii standard dintr-o reţea sau a atacurilor de diverse tipuri (exploits, malware, DoS / DDoS);
- Frontend (Portal Web) - Interfaţa Web folosită de utilizatori pentru acces în platformă;
- Sistem e-Learning (LMS) – sistem pentru administrarea, urmărirea, raportarea şi livrarea moduleleor de curs către studenţi;
- Instrumente de management şi monitorizare - interfeţe de administrare pentru controlul resurselor utilizare pe timpul unui exerciţiu;
- Mecanisme de federare - soluţii IPSec VPN şi API-uri pentru acces la sisteme şi servicii.

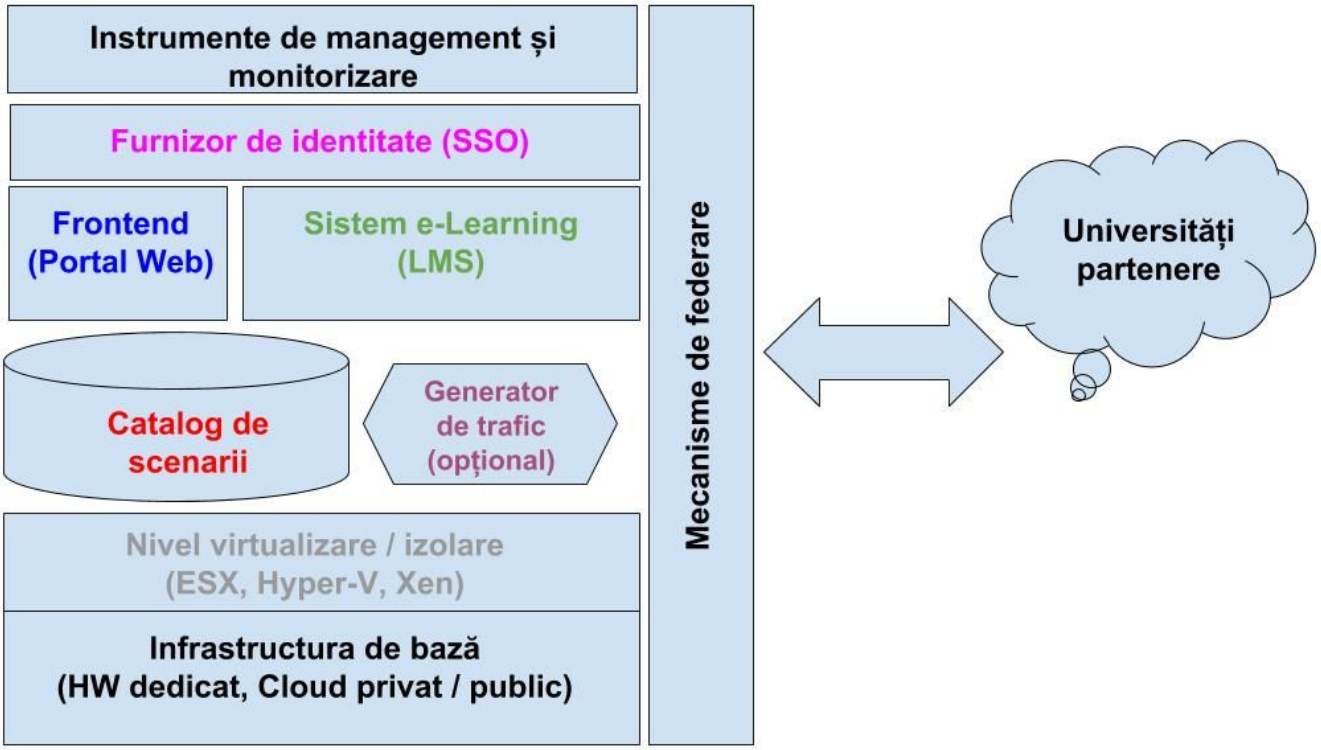

Figura 4: Arhitectura logică poligon cibernetic

Arhitectura propusă oferă flexibilitatea necesară organizării de exerciţii complexe şi experimentări în domeniul securităţii cibernetice. De asemenea, soluţia permite unui partener să se specializeze într-un anumit domeniu (de exemplu: SCADA, dispozitive mobile, IoT) iar ceilalţi parteneri să acceseze resursele dezvoltate de partenerul respectiv. În plus, scenariile şi exerciţiile dezvoltate de un partener vor putea fi portate şi folosite de către ceilalţi parteneri pentru instruirea propriilor studenţi.

Locuri de munca susţinute prin program

Pe parcursul acestei etape au fost organizate concursuri pentru ocuparea posturilor libere de noi cercetători. Pentru promovarea acestor posturi, au fost postate anunţuri pe site-urile www.euraxess.gov.ro şi www.jobs.ancs.ro. Concursurile de ocupare a posturilor s-au desfăşurat în conformitate cu reglementările în vigoare şi metodologiile specifice fiecărei universităţi. La final, toate cele **8 posturi de noi cercetători** asumate prin contract au fost ocupate.

Diseminarea rezultatelor proiectului

În data de 09 Noiembrie 2018, la Academia de Studii Economice din Bucureşti, Sala 2416, a fost organizată **o masă rotundă** care a avut ca temă colaborarea interuniversitară în domeniul securităţii cibernetice. Pe parcursul întâlnirii au fost analizate oportunităţile şi provocările actuale şi s-au identificat o serie de acţiuni comune privind pregătirea specialiştilor şi activităţile de cercetare în domeniul securităţii cibernetice. De asemenea, în cadrul întâlnirii au fost prezentate obiectivele proiectului complex de cercetare „Hub inovativ pentru tehnologii avansate de securitate cibernetică (ATLAS)”, derulat de Academia Tehnică Militară în parteneriat cu Universitatea Politehnica din Bucureşti, Academia de Studii Economice din Bucureşti şi Universitatea Bucureşti. La activitate au fost invitaţi să participe reprezentaţii ai mediului academic şi specialişti din partea agenţiilor guvernamentale şi companiilor private care activează în domeniu.

Rezultatele activităţilor de cercetare derulate în cadrul proiectului au fost publicate în jurnale şi în volumele unor conferinţe internaţionale indexate ISI/BDI. În total, au fost publicate **15 articole ştiinţifice**, după cum urmează:

1. A. Pauna, A.C. Iacob, and I. Bica, “QRASSH – A self-adaptive SSH Honeypot driven by Q-Learning,” in Proceedings of the 2018 International Conference on Communications (COMM), Bucharest, Romania, 2018, pp. 441 – 446, DOI: 10.1109/ICComm.2018.8484261;
2. C. Grumăzescu, V.A. Vlăduta and A. Timofte, “Hybrid identity based cryptographic scheme optimization using machine learning in WSN,” in Proceedings of the 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania, 2018;
3. I. Aciobăniţei, R.Ioana. Guinea and M.L. Pura, “AVISPA versus AVANTSSAR in the Model Checking of Secure Communication Protocols,” in Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 1:

SECRYPT, Porto, Portugal, 2018, pp. 520-525, ISBN: 978-989-758-319-3; DOI: 10.5220/0006887906860691
4. I. Aciobăniţei, I. C. Buhuş and M.L. Pura, “Using Cryptography in the Cloud for Lightweight Authentication Protocols Based on QR Codes,” in Proceedings of the 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2018, pp. 539 - 542, DOI: 10.1109/SACI.2018.8440949;
5. I. Aciobăniţei, I. C. Buhuş and M.L. Pura, “Lightweight version of SQRL authentication protocol based on cryptography in the cloud,” in Proceedings of the 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2018, pp. 173 -178, DOI: 10.1109/SACI.2018.8440957;
6. F. Stancu, D. C. Trancă, M. Chiroiu and R. Rughiniş, “Evaluation of cryptographic primitives on modern microcontroller platforms,” in Proceedings of the 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet), Cluj-Napoca, Romania, 2018, pp. 1-6, DOI: 10.1109/ROEDUNET.2018.8514127;
7. A. Bican, R. Deaconescu, W. N. Chin and Q.T. Ta, “Verification of C Buffer Overflows in C Programs,” in Proceedings of the 2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet), Cluj-Napoca, Romania, 2018, pp. 1-6, DOI: 10.1109/ROEDUNET.2018.8514126;
8. V. Velciu, F. Stancu and M. Chiroiu, “HiddenApp - Securing Linux applications using ARM TrustZone,” Innovative Security Solutions for Information Technology and Communications, Lecture Notes in Computer Science, Springer, 2019 (acceptat spre publicare);
9. D. Sporici, M. Chiroiu and D. Ciocîrlan, “An Evaluation of OCR Systems against Adversarial Machine Learning”, Innovative Security Solutions for Information Technology and Communications, Lecture Notes in Computer Science, Springer, 2019 (acceptat spre publicare);
10. C. Toma, B. Talpiga, C. Boja, M. Popa, B. Iancu and M. Zurini, “Secure IoT Supply Chain Management Solution using Blockchain and Smart Contracts Technology”, Innovative Security Solutions for Information Technology and Communications, Lecture Notes in Computer Science, Springer, 2019 (acceptat spre publicare);
11. C. Toma and M. Popa, “IoT security issues for Industry 4.0 - Oil & Gas Solution,” in Proceedings of the 17th International Conference on Informatics in Economy (IE 2018) Education, Research & Business Technologies, Iasi, Romania, 2018, pp. 113 - 118, ISSN 2284-7472;
12. C. Boja and M. Zurini, “IoT security architecture based on device fingerprinting,” in Proceedings of the 17th International Conference on Informatics in Economy (IE 2018) Education, Research & Business Technologies, Iasi, Romania, 2018, pp. 157 - 162, ISSN 2284-7472;
13. C. Paduraru, M.C. Melemciuc, “An automatic test data generation tool using machine learning,” in Proceedings of the 13th International Conference on Software Technologies - Volume 1: ICSOFT, Porto, Portugal, 2018, pp. 472-481, ISBN 978-989-758-320-9, DOI: 10.5220/0006836605060515;
14. G. Ducoffe, “A new application of Orthogonal Range Searching for computing Giant Graph Diameters,” in Proceedings of 2nd Symposium on Simplicity in Algorithms (SOSA’19), San Diego, SUA, 2019 (acceptat spre publicare);

15. C. Toma and M. Popa, “IoT Security Approaches in Oil & Gas Solution Industry 4.0”, Informatica Economică vol. 22, no. 3/2018, pp. 46 - 62, DOI: 10.12948/issn14531305/22.3.2018.05.

Concluzii

Având în vedere cele expuse mai sus, apreciem că au fost atinse toate obiectivele manageriale şi ştiinţifice prevăzute pentru această etapă şi există create premisele pentru continuarea în bune condiţii a proiectului.

Indicatorilor de rezultat prevăzuţi pentru această etapă au fost realizaţi în întregime şi chiar depăşiţi la unele capitole. Toate cele 8 posturi de noi cercetători asumate prin contract au fost ocupate. Rezultatele cercetărilor efectuate au fost publicate în jurnale şi la conferinţe de prestigiu în domeniu. În total, au fost realizate în cadrul proiectului 15 articole faţă de 3 estimate iniţial. De asemenea, a fost organizată o masă rotundă la care au fost prezentate serviciile şi produsele ce urmează a fi dezvoltate în cadrul proiectului.

Data: 04.12.2018

Director Proiect Complex
Prof.univ.dr.ing.
Ion BICA