MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA
FACULTATEA DE AUTOMATICA SI CALCULATOARE

Detectarea Keylogger-elor

Rețele de Calculatoare

Proiect de Semestru

Student: **Timea NAGY**

Profesor îndrumător: **Prof.dr.ing. Vasile Teodor DĂDÂRLAȚ**

2024

Contents

Chapter 1. Introducere

1.1. Ce sunt Keylogger-ele?

Un Keylogger este un program malware plantat pe un device fără acordul utilizatorului care înregistrează tastele apăsate. Poate fi folosit cu scopul de a fura informații sensibile de la utilizator, precum conturi, parole, date de identificare sau datele de pe carduri de credit.[1]

Există două tipuri de Keyloggere, cele de tip Software și de tip Hardware. Cele software pot fi instalate fără știrea utilizatorului, descărcându-se în paralel cu alte aplicații malițioase. Cele hardware sunt dispozitive fizice de dimensiuni reduse care pot fi introduse în calculator, de obicei între tastatură și unitatea centrală de procesare (CPU). Keyloggerele se mai pot categoriza și astfel:

- Keyloggere de utilizator, care înregistrează tastările utilizatorilor folosind API-ul Win32 în sistemele de tip Windows.
- Keyloggere de kernel, care sunt implementate ca drivere de dispozitiv sau drivere de filtru.

1.2. Context, istorie

1.2.1. Istoria Keyloggerelor

Keyloggerele au apărut prima dată la începutul anilor 80. Ele au fost programe simple, folosite în scopuri legitime, precum supravegherea copiilor sau a angajaților. Cu timpul, ele au devenit programe complexe și au fost folosite pentru scopuri ilegale cum ar fi furtul de informații personale și financiare.

1.2.2. Istoria detectării keyloggerelor

Inițial, detectarea keyloggerelor se făcea prin verificarea manuală a fișierelor și proceselor active pe computer. Utilizatorii trebuiau să fie atenți la activități neobișnuite sau la prezența fișierelor suspecte.

Au apărut primele programe antivirus care aveau funcționalitatea de a detecta și elimina keyloggerele. Se foloseau în principiu metode de detecție bazată pe semnături. Când au început să apară cât mai multe programe și cât mai complexe, metoda de detecție bazată pe semnături nu mai era suficientă și a apărut nevoia de a aduce o metodă nouă pe piață. Această metodă a fost metoda bazată pe comportament.

În prezent, detectarea keyloggerelor înseamnă o combinație de metode bazate pe semnături, potrivirea tiparelor și comportament. Programele moderne de securitate folosesc inteligență artificială și machine learning pentru a îmbunătăți eficiența detectării.

1.3. Tehnici de prevenție a keyloggerelor

Este important să conștientizăm problema keyloggerelor și să le evităm, mai ales în cazuri în care utilizatorul folosește calculatoare publice, când introducem date sensibile în mediul online, la descărcările fișierelor de pe internet și la accesarea site-urilor nesigure.

Există câțiva pași care se recomandă pentru a încerca evitarea programelor malițioase și a keyloggerelor: [2]

- Instalarea unui antivirus actualizat și performant
- Utilizarea programelor special concepute pentru a detecta keyloggere
- Verificarea proceselor active care rulează în fundal
- Actualizarea regulată a sistemului software
- Evitarea descărcărilor din surse nesigure
- Unele keyloggere care intră prin intermediul browserelor sunt numite, în general, keyloggere de browser. În acest caz, este indicat să dezinstalăm browserul respectiv și să reinstalăm o versiune nouă.

1.4. Importanța detectării Keylogger-elor

Cea mai mare problemă care apare în detectarea Keyloggerelor este că ele sunt greu de detectat. Keyloggerele pot funcționa fără ca utilizatorul să observe ceva ieșit din comun, și astfel pot rămâne nedetectate. [3]

Detectarea keyloggerelor din timp are o importanță deosebită deoarece acestea pot fi folosite în mod malițios pentru a fura informații sensibile sau pentru a spiona activitatea utilizatorului, încălcându-i intimitatea.

Keyloggerele moderne sunt extrem de sofisticate și din ce în ce mai greu de detectat de către programele antivirus și anti-malware disponibile pe piață. Detectarea și prevenirea keyloggerelor reprezintă o sarcină dificilă pentru managerii de securitate. Spre deosebire de virușii tradiționali, keyloggerele avansate sunt aproape imposibil de detectat.[2]

Deși numărul de fraude care exploatează keyloggerele a crescut în ultimii ani, nu au fost găsite multe soluții eficiente pentru a aborda această problemă.

Chapter 2. Analiza metodelor de detecție

2.1. Starea actuală a detecției

Pentru a detecta programele malware, s-au propus multe modele și tehnici de-a lungul timpului. Pentru detecția malware-ului de tip keylogger, soluțiile existente nu sunt destul de eficiente. Există modele bazate pe semnături, care pot fi evitate cu ușurință. Există și modele bazate pe comportament, care sunt relativ mai eficiente deoarece ele încearcă să diferențieze între aplicațiile periculoase și cele sigure prin analizarea comportamentului programelor legitime sau a programelor malware. Cele mai multe tehnici analizează ce apeluri de sistem sau bibliotecă sunt folosite în timpul funcționării programelor. [4]

Cu toate acestea, eficiența software-ului de securitate actual este limitată deoarece folosește metoda de pattern matching care poate detecta doar keyloggerele cunoscute, dar nu și pe cele noi, necunoscute. [3]

2.2. Mecanisme existente de detectare a keylogger-elor

2.2.1. Detectarea bazată pe semnături

Metoda de detectare bazată pe semnături este o metodă comună în detectarea programelor malițioase. Aceasta se bazează pe înregistrarea semnăturilor programelor malițioase și căutarea acestor semnături într-o bază de date. Deși este o metodă eficientă pentru identificarea amenințărilor cunoscute, prezintă limitări importante deoarece poate recunoaște doar programele deja cunoscute, nu și pe cele nou apărute. Prin urmare, programele noi sau actualizate pot trece neobservate folosind această metodă, lăsând sistemul vulnerabil la atacuri noi. De aceea, este esențial să fie completată cu alte tehnici de detectare pentru a asigura o protecție cuprinzătoare împotriva tuturor tipurilor de malware.

Baza de date poate fi dificil de completat cu semnături ale keylogger-elor existente deoarece nu toate keylogger-ele se răspândesc pe un număr mare de calculatoare, ca și în cazul altor programe malițioase. Unele keylogger-e sunt create doar pentru a fura informații de la o anumită companie sau grup. Din acest motiv, unele keylogger-e pot rămâne neidentificate. [3]

2.2.2. Detectarea de tip pattern-matching

Metoda de potrivire a tiparelor este o altă metodă comună în detectarea programelor malițioase. Se monitorizează comportamentul unui program și se compară cu alte modele cunoscute. Poate identifica amenințările noi care au un comportament similar cu cele deja cunoscute și este eficientă pentru detectarea comportamentelor suspecte.

2.2.3. Detectarea bazată pe comportament

Această metodă monitorizează acțiunile aplicațiilor în timp real și identifică activități suspecte sau neobișnuite, cum ar fi înregistrarea tastărilor. Este mai eficientă pentru detectarea keylogger-elor noi, deoarece nu se bazează pe semnături preexistente. Poate detecta un program malițios cunoscut cât și unul necunoscut prin analiza acțiunilor acestora.

2.2.4. Analiza programelor existente de detecție a keylogger-elor

În conferința International Conference on Network-Based Information Systems 'A keystroke logger detection using keyboard-input-related API monitoring' din 2011, au analizat câteva programe de detectare a keylogger-elor și au ajuns la concluzia că este necesară îmbunătățirea performanței produselor existente, deoarece singurul program care nu a detectat eronat unele programe care sunt legale ca fiind keylogger-e a fost Spybot-Search & Destroy. În conducerea analizei, au folosit și alte programe de detectare precum Keylogger Hunter, Keylogger Stopper și Anti-Keylogger. Criteriile de evaluare includ rata de detectare a keylogger-elor și rata de detectare falsă. [5]

Chapter 3. Propuneri de soluții

3.1. Soluții propuse

Pentru soluțiile propuse de acest paper, s-au analizat publicațiile Bait your Hook: a Novel Detection Technique for Keyloggers de la Stefano Ortolani, Cristiano Giuffrida, and Bruno Crispo [1], KLIMAX: Profiling Memory Write Patterns to Detect Keystroke-Harvesting Malware din Recent Advances in Intrusion Detection - 14th International Symposium [4], respectiv A keystroke logger detection using keyboard-input-related API monitoring de la Kohei Nasaka, Tomohiro Takami, Takumi Yamamoto and Masakatsu Nishigaki [3].

3.1.1. Simularea undei secvenței de tastare și observarea comportamentului aplicațiilor

Metoda este propusă și analizată în conferința Bait your Hook: a Novel Detection Technique for Keyloggers de la Stefano Ortolani, Cristiano Giuffrida, and Bruno Crispo [1]. Metoda se bazează pe ideea că keylogger-ele vor încerca să înregistreze toate apăsările tastelor, inclusiv cele simulate, iar prin observarea scrierilor în memorie, se poate detecta prezența lor.

Metoda propusă funcționează prin utilizarea unei tehnici de simulare a apăsărilor de taste pentru a detecta keylogger-ele:

1. Se injectează o secvență atent aleasă de apăsări de taste (bait) în sistem. Această secvență este concepută pentru a semăna cu activități legitime de utilizator, dar este în mod specific creată pentru a atrage și declanșa comportamentul keylogger-elor.
2. Sistemul monitorizează în timp real scrierile în memorie și alte activități legate de tastatură. Se observă dacă pattern-urile de scriere în memorie se corelează cu secvența de apăsări de taste injectate.
3. Se analizează corelația între secvența de apăsări de taste și activitățile detectate în memorie. Valorile mari de corelație indică faptul că un keylogger ar putea fi prezent, deoarece încearcă să înregistreze apăsările tastelor injectate.
4. Dacă se detectează o corelație mare, sistemul poate concluziona că există un keylogger activ și poate trimite o alertă sau poate lua alte măsuri de securitate pentru a proteja utilizatorul și datele sale.

Arhitectura Sistemului

Arhitectura sistemului se poate observa în Figura 3.1. Sistemul constă din 5 componente principale. Acestea sunt: injector, monitor, pattern translator, detector și pattern generator:

- Rolul injectorului este de a injecta un flux de intrare în sistem, imitând comportamentul unui utilizator simulat la tastatură. Scopul său este de a simula apăsările de taste pentru a observa cum se comportă aplicațiile monitorizate.

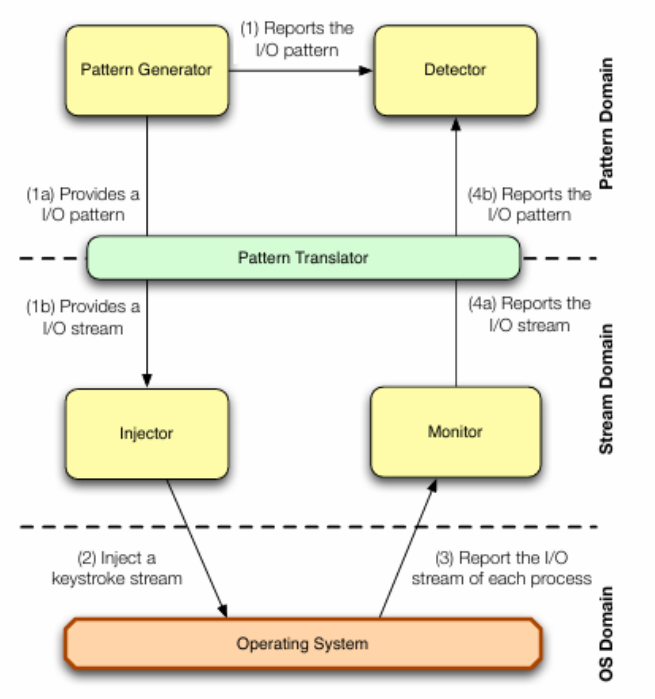

Figure 3.1: Arhitectura sistemului

- Monitorul este responsabil pentru înregistrarea fluxului de ieșire al tuturor proceselor aflate în execuție. Acesta urmărește activitatea în memorie pentru a detecta scrierile asociate cu apăsările de taste simulate.
- Rolul translatorului de patternuri este de a transforma un AKP (Abstract Keystroke Pattern) într-un flux și invers, având în vedere un set de parametri de configurare țintă. Translatorul presupune o corespondență între apăsările de taste și bytes și le tratează în mod egal ca unități de bază ale fluxului de intrare și ieșire, respectiv.
- Rolul detectorului este de a analiza fluxul de ieșire monitorizat și de a identifica prezența keylogger-elor. Acesta folosește corelația dintre fluxul de intrare (apăsările de taste simulate) și scrierile în memorie pentru a detecta activități suspecte. Atunci când corelația este mare, detectorul poate determina că există un keylogger activ și poate alerta utilizatorul sau poate lua măsuri de securitate.
- Rolul generatorului de patternuri este de a crea secvențele de tastare care vor fi injectate în sistem de către injector. Aceste secvențe sunt atent alese pentru a atrage și declanșa comportamentul keylogger-elor, facilitând astfel detecția lor de către sistem. Generatorul de patternuri trebuie să fie capabil să creeze diverse secvențe care să acopere o gamă largă de comportamente posibile ale utilizatorilor pentru a asigura eficiența tehnicii de detectare.

Sunt propuse câțiva algoritmi de generare de pattern-uri:

- **Random (RND):** Fiecare eșantion este generat aleatoriu, fără constrângeri suplimentare. Acesta este cel mai simplu algoritm de generare a pattern-urilor.
- **Random with fixed range (RFR):** Pattern-ul este o permutare aleatorie a unei serii de eșantioane distribuite uniform pe intervalul [0, 1]. Acest algoritm încearcă să maximizeze variabilitatea pattern-ului de intrare.

- **Impulse (IMP):** Fiecare eșantion 2i este asignat cu valoarea 0 și fiecare eșantion 2i + 1 este asignat cu valoarea 1. Acest algoritm încearcă să producă un pattern de intrare cu variabilitate maximă, minimizând în același timp durata perioadelor inactive.
- **Sine Wave (SIN):** Pattern-ul generat este o undă sinusoidală discretă care oscilează între 0 și 1, primul eșantion având valoarea 1. Unda sinusoidală crește sau scade cu un pas fix de 0,1 la fiecare eșantion. Acest algoritm explorează efectul incrementelor constante (și decrementelor) în pattern-ul de intrare.

3.1.2. Metoda KLIMAX: Soluția bazată pe pattern-uri de scriere în memorie

Această metodă este propusă de publicația KLIMAX: Profiling Memory Write Patterns to Detect Keystroke-Harvesting Malware din Recent Advances in Intrusion Detection - 14th International Symposium [4].

Se propune o soluție specifică detectării keylogger-elor, bazată pe analiza comportamentală, dar în loc de analiza apelurilor de sistem, se propune analiza scrierii în memorie. Ideea de bază este analizarea corelației dintre apăsările de taste ale utilizatorului și scrierile în memorie efectuate de malware pentru a colecta date sensibile. Se injectează un flux de taste specifice și se observă modelele de scriere în memorie ale aplicației. Corelațiile mari indică detectarea imediată a programului malițios.

Metoda constă în analiza detaliată a pattern-urilor de scriere în memorie pentru a detecta malware-ul care se comportă ca și un keylogger. Monitorizarea se face la nivel de kernel pentru a nu afecta performanțele sistemului. Scrierile în memorie sunt analizate detaliat pentru a identifica pattern-uri specifice care pot indica prezența unui keylogger cu precizie mare. Aceste pattern-uri sunt caracteristice activităților de recoltare a datelor, cum ar fi înregistrarea apăsărilor tastelor.

Metoda sugerată poate detecta keylogger-ele înainte ca acestea să producă daune semnificative (proactiv), cât și după ce au început să înregistreze tastele (reactiv), asigurând o protecție completă împotriva keylogger-elor.

Pentru a evita alarmele false, metoda compară patternurile de scriere suspecte cu comportamentele legitime ale aplicațiilor. Astfel, se asigură că numai activitățile malițioase sunt identificate și raportate.

Arhitectura metodei KLIMAX

Metoda este implementată în spațiul kernel-ului la un mediu de dezvoltare la nivel scăzut pentru a avea acces la gestionarea memoriei și la informații în timp real. Soluția are o arhitectură de 3 nivele: monitor, injector și detector. În spațiul kernel-ului rulează componentele de monitor și injector. Monitorul expune un contor de performanță a scrierilor în memorie pentru injector și este împărțit în două subcomponente, shadow-er și clasificator. Prima subcomponentă se ocupă cu interceptarea fiecărei scrieri în memorie efectuate de procesul monitorizat. A doua subcomponentă clasifică regiunile de memorie care trebuie monitorizate și scrierile în memorie care trebuie contorizate. Componentele și relația dintre componente se poate observa în Figura 3.2.

Optimizarea acurateței detecției

Pentru maximizarea acurateței detecției se inspectează sursele de negative false și pozitive false.

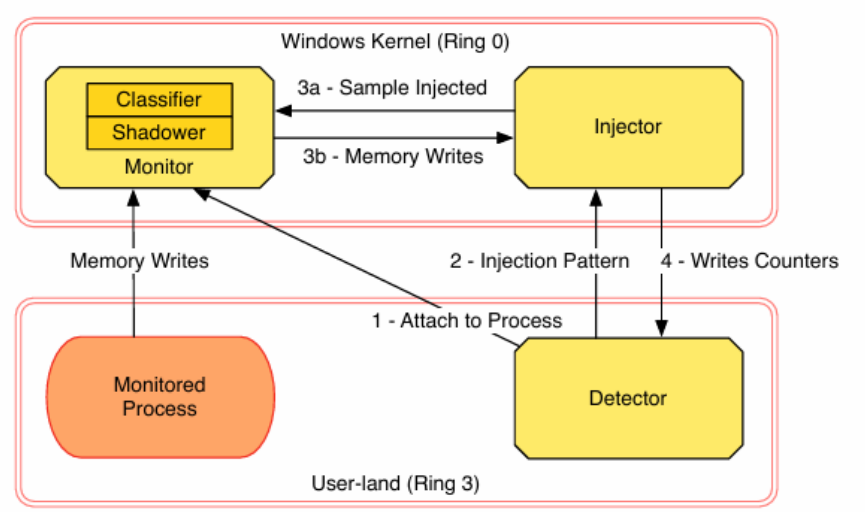

Figure 3.2: Arhitectura metodei KLIMAX

Negativele false apar când un program malițios scapă neidentificat. Aplicația malițioasă poate rula în procese și thread-uri diferite, poate exploata orice canal de scurgere de informații disponibil. Negativele false apar și când o aplicație malițioasă efectuează un atac de tip denial-of-service, producând un număr excesiv de scrieri în memorie. Acest lucru poate face ca monitorul să nu mai poată ține pasul și să rateze apăsările de taste, reducând astfel corelația și evitând detectarea.

Soluțiile pentru aceste negative false includ eliminarea shadowing-ului al regiunilor de memorie doar în citire, monitorizarea scrierilor în memorie de către application code precum și DLL-uri, și suportarea monitorizării simultane a proceselor și thread-urilor multiple.

Pozitivele false apar în situația în care o aplicație legitimă monitorizată arată o corelație mare cu patternul injectat și declanșează detectarea. Astfel de aplicații pot fi editoare de text avansate, IDE-uri de programare sau jocuri care necesită multe comenzi rapide de la tastatură. Ca și soluție se elimină logarea scrierilor de memorie făcute de către USER32.dll.

O altă sursă de pozitive false poate fi scrierile temporare în memorie pe stivă, care sunt folosite frecvent în callback-ul furnizat de programator pentru a implementa logica aplicației. Soluția constă în alocarea unui buffer suficient de mare pe stivă în punctul de intrare al programului și menținerea unui pointer global pentru a accesa buffer-ul din callback.

3.1.3. Sistem bazat pe utilizarea DLL-ului pentru a detecta API-uri legate de tastare

În conferința A keystroke logger detection using keyboard-input-related API monitoring de la Kohei Nasaka, Tomohiro Takami, Takumi Yamamoto și Masakatsu Nishigaki [3] se propune utilizarea unui astfel de sistem. Studiul constată că programele Windows interacționează cu Win32 API, care este o interfață de programare a aplicațiilor pentru Microsoft Windows. Acest lucru înseamnă că keyloggerele care sunt executate în modul utilizator, folosesc API-uri legate de tastatură. Din acest motiv, există posibilitatea ca comportamentul keylogger-elor să fie definit pe baza Win32 API.

În continuare se vor defini 5 comportamente specifice keylogger-elor care folosesc Win32 API. Aceste comportamente vor ajuta la identificarea proceselor de keyloggere.

De obicei, înregistrarea tastărilor se poate face folosind două API-uri: GetAsyncKeyState și SetWindowsHookEx.

Comportament 1

API-ul GetAsyncKeyState determină dacă o tastă este apăsată sau nu în momentul în care este apelat acest API și dacă tasta a fost apăsată după un apel anterior la GetAsyncKeyState. Au ajuns la definirea unui comportament specific keylogger-elor care folosesc acest API.

Comportament rezultat: un proces înregistrează cel puțin apăsările tastelor A-Z și 0-9 folosind GetAsyncKeyState.

Comportament 2

API-ul SetWindowsHookEx funcționează prin instalarea unei proceduri de hook. O procedură de hook monitorizează un proces pentru a obține informații specifice despre proces. Există două tipuri de hook-uri: globale și locale. Cele globale sunt instalate în toate procesele iar cele locale sunt instalate doar într-un singur proces.

De exemplu, pentru a fura un ID de utilizator sau o parolă în Internet Explorer, un keylogger trebuie să înregistreze apăsările de taste din Internet Explorer. Deoarece Internet Explorer este un proces diferit pentru un keylogger, acesta trebuie să folosească nu un hook local, ci un hook global cu o procedură de hook pentru tastatură.

Comportament rezultat: Un proces înregistrează apăsările tastelor folosind SetWindowsHookEx cu un tip de hook pentru tastatură.

Comportament 3

În cazul în care un keylogger înregistrează apăsările tastelor prin comportamentele 1 sau 2, acesta poate fi găsit de Managerul de Activități Windows. Pentru a evita descoperirea, infractorii încearcă să ascundă programul injectându-l într-un program legitim. Acest lucru se face prin API-urile CreateRemoteThread sau SetWindowsHookEx. CreateRemoteThread creează un thread în spațiul virtual al altui proces, iar SetWindowsHookEx poate injecta un DLL în spațiul de adrese al unui proces remote folosind un hook global. De aici rezultă al 3-lea comportament.

Comportament rezultat: Un proces folosește API-ul CreateRemoteThread sau SetWindowsHookEx pentru a nu fi descoperit în Task Manager.

Comportament 4

Comportamentul de înregistrare a apăsărilor tastelor din alt proces: O procedură este injectată într-un proces, iar procedura înregistrează apăsările tastelor din acel proces folosind una dintre cele trei metode descrise mai sus, adică folosind GetKeyState, GetKeyboardState sau SetWindowsHookEx cu hook-ul local.

Comportament 5

Comportamentul de înregistrare a apăsărilor tastelor din toate procesele într-un alt proces: O procedură este injectată într-un proces și metodele prezentate anterior sunt aplicate în acel proces.

Comportament 6

Unii infractori pot încerca să injecteze procedura lor într-un proces care, la rândul său, va injecta procedura în alte procese. Pentru a detecta un astfel de comportament, trebuie să verificăm activitatea de injectare recursivă. Acest tip de activitate, în general, nu se găsește în procesele legitime. Comportament de injectare repetată: O procedură este injectată într-un alt proces, apoi acest proces injectează procedura în alte procese folosind CreateRemoteThread sau SetWindowsHookEx.

3.2. Rezultate

În această secțiune vor fi detaliate rezultatele metodelor propuse în secțiunea anterioară.

3.2.1. Simularea undei secvenței de tastare și observarea comportamentului aplicațiilor

Soluția propusă a fost testată de către autori, implementând un prototip al sistemului propus, pe mai multe tipuri de calculatoare. Soluția a fost scrisă în C și rulată ca o aplicație pentru Windows OS. Prototipul a fost testat cu mai multe keyloggere disponibile public cât și pe un keylogger dezvoltat de către echipă.

Algoritmul RFR a fost folosit ca și algoritm de generare de pattern. Keyloggerele au fost detectate cu succes, în doar câteva secunde, fără niciun răspuns pozitiv fals. Pentru ultimii doi keyloggeri testați, nu s-a reușit obținerea niciunui rezultat de detectare, deoarece nu a fost generat niciun fișier de log consistent în niciunul dintre cele două cazuri, chiar și după experimente repetate.

3.2.2. Metoda KLIMAX: Soluția bazată pe patternuri de scriere în memorie

Sistemul KLIMAX a fost implementat și testat. Strategia de detectare a avut succes în identificarea tuturor tipurilor de malware examinate care au utilizat în mod explicit API-uri de interceptare a tastelor și au prezentat comportament de keylogging.

Strategia propusă este efectivă și în ferestre scurte de observație. Cât despre pozitive false și negative false, în majoritatea cazurilor, detectarea corectă a colectării de date de tastatură generează puține pozitive false. Aceste aplicații detectate pot fi inofensive, precum cele de remapare a tastaturii. Rezultate false negative pot apărea în cazul în care comportamentul de keylogging este declanșat doar la evenimente specifice și acesta nu este declanșat în timpul observației.

3.2.3. Sistem bazat pe utilizarea DLL-ului pentru a detecta API-uri legate de tastare

În paper s-a propus o metodă de detecție a keylogger-elor bazată pe monitorizarea API-urilor folosite. La momentul efectuării studiului, încă nu a fost făcută o analiză complexă a sistemului propus.

În studiu, s-au analizat performanțele altor programe de detectare de keylogger-e, mai precis: Keylogger Hunter, Keylogger Stopper, Anti-Keylogger și Spybot-Search & Destroy. Rezultatele evaluării susțin că niciun produs nu are rata de detecție de 100

Chapter 4. Concluzie

În concluzie, malware-ul de tip Keylogger este încă o cauză principală a incidentelor de securitate. Printre diferitele tipuri de malware, cel care colectează informații private ale utilizatorilor crește în impact și frecvență.

Soluțiile bazate pe semnături sunt limitate deoarece pot fi evitate ușor și necesită o semnătură validă pentru a detecta noi amenințări. Tehnicile de detecție bazate pe comportament sunt mai eficiente, deoarece disting între aplicațiile legitime și cele malițioase prin profilarea comportamentului lor. Majoritatea se bazează pe apelurile de sistem sau bibliotecă invocate în timpul execuției.

Detectarea keylogger-elor folosind apeluri de sistem este dificilă din cauza aplicațiilor legitime care interceptează apăsările de taste, cum ar fi managerii de scurtături sau utilitarele de remapare a tastaturii. Aceste aplicații pot genera pozitive false. Whitelisting-ul nu este o soluție practică din cauza numărului mare de astfel de programe.

Soluțiile de detecție recomandate par să rezolve câteva dintre aceste probleme și să îmbunătățească rata de recunoaștere. Totuși, cu evoluția noilor programe de tip keylogger, este necesară și evoluția continuă a sistemelor de detectare a keylogger-elor.

Bibliography

[1] S. Ortolani, C. Giuffrida, and B. Crispo, "Bait your hook: a novel detection technique for keyloggers," *Proceedings of Recent Advances in Intrusion Detection (RAID 2010)*, 2010.

[2] A. Singh, P. Choudhary, A. kumar singh, and D. kumar tyagi, "Keylogger detection and prevention," *Journal of Physics: Conference Series*, vol. 2007, no. 1, p. 012005, aug 2021. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2007/1/012005

[3] K. Nasaka, T. Takami, T. Yamamoto, and M. Nishigaki, "A keystroke logger detection using keyboard-input-related api monitoring," *Proceedings of the 14th International Conference on Network-Based Information Systems*, 2009.

[4] S. Ortolani, C. Giuffrida, and B. Crispo, "Klimax: Profiling memory write patterns to detect keystroke-harvesting malware," *Proceedings of Recent Advances in Intrusion Detection (RAID 2011)*, 2011.

[5] K. Nasaka, T. Takami, T. Yamamoto, and M. Nishigaki, "A keystroke logger detection using keyboard-input-related api monitoring," in *2011 14th International Conference on Network-Based Information Systems*, 2011, pp. 651–656.