SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Štefan Balogh

Autoreferát dizertačnej práce

ANALÝZA OPERAČNEJ PAMÄTE S CIEĽOM ODHALENIA NEZNÁMEHO KÓDU

na získanie akademickej hodnosti philosophiae doctor, PhD.
v doktorandskom študijnom programe
9.2.9 aplikovaná informatika

Bratislava 2014

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Ústave informatiky a matematiky FEI STU v Bratislave

Predkladateľ: **Ing. Štefan Balogh**
ÚIM FEI STU, Ilkovičova 3, 812 19 Bratislava

Školiteľ: **doc. Ing. Pavol Zajac, PhD.**
FEI STU Bratislava

Oponenti: **doc. RNDr. Karol Macák, CSc.**
Ministerstvo Obrany SR
Kutuzovova 8
832 47 Bratislava, SR

RNDr. Marián Novotný, PhD.
ESET, s.r.o
Einsteinova 24
851 01 Bratislava, SR

Autoreferát bol rozoslaný dňa: 5.11.2014

Obhajoba dizertačnej práce sa koná: 3.12.2014 o 9h na Fakulte elektrotechniky a informatiky STU, Ilkovičova 3, 812 19 Bratislava, v miestnosti C502.

Prof. RNDr. Gabriel Juhás, PhD.
Dekan FEI STU v Bratislave

Obsah

Úvod

V odborných kruhoch zaoberajúcich sa problematikou bezpečnosti je jednou zo stále diskutovaných otázok hľadanie účinných metód pre detekciu škodlivého kódu. Pojmom škodlivý kód (ďalej malware z anglického malicious software) označujeme ľubovoľný program, ktorý je vytvorený so škodlivým zámerom. V minulosti malware vyvíjali autori predovšetkým preto, aby sa zviditeľnili ich programy, ale zlomyseľný zámer neobsahovali. S príchodom internetu sa však zmenila aj podstata malware. Pomocou internet bankingu môžeme vykonávať rôzne finančné transakcie, nakupovať veci v elektronických obchodoch, či podávať rôzne dokumenty na úrady. A to sa snažia autori moderného malware využiť, napríklad na odcudzenie osobných údajov, ktoré na dané úkony využívame. Tento fakt prinútil ľudí hľadať účinné metódy, ako sa proti malware brániť. Vývoj malware však napreduje neuveriteľnou rýchlosťou a útočníci sú často o krok vpred.

V roku 1992 bol počet počítačových vírusov odhadnutý na 1 000 až 2 300. V roku 2002 už bolo 60 000 známych vírusov, trójskych koní, červov a ich variácií (Daoud et al., 2008). Podľa Symantec Global internet Security threat report za rok 2008 bolo vytvorených 1 656 227 nových signatúr pre identifikáciu škodlivého kódu. To je 265% nárast oproti roku 2007, keď bolo vytvorených 624 267 nových signatúr škodlivého kódu. A teda Symantec do konca roka 2008 evidovala celkovo 2 674 171 signatúr a z toho viac než 60% bolo vytvorených v roku 2008 (Symantec, 2009). Ešte alarmujúcejšia je výsledná správa od Symantec za ďalší rok. V roku 2009 bolo vytvorených až 2 895 802 nových signatúr pre identifikáciu škodlivého kódu . Takže počet všetkých evidovaných signatúr do konca roku 2009 vzrástol na 5 724 106. To znamená, že zo všetkých signatúr, ktoré Symantec evidoval za dané obdobie, bolo 51% vytvorených v roku 2009 (Symantec, 2010). Za rok 2010 to už bolo 286 miliónov nových hrozieb (Symantec, 2011) a v roku 2011 bolo

vytvorených až 400 miliónov nových variant malware (Symantec, 2012).
Detekcia malware na základe signatúr je v súčasnosti stále jedna z najrozšírenejších metód detekcie. Jej nevýhoda však je, že môže detegovať iba malware, ktoré má v databáze signatúr. Manuálna analýza malware a vytváranie signatúr pri súčasnom trende vývoja a možnostiach modifikácie malware sa však stáva neúnosná. Na to, aby sa predišlo väčším škodám, vznikla potreba odhaliť nové druhy malware v čo najkratšom možnom čase. A práve v tom majú analytikom pomôcť automatizované nástroje, ktoré sa snažia podať čo najpresnejšiu správu o činnosti analyzovaného programu. Všeobecne platí, že tieto prístupy môžeme zadeliť do dvoch hlavných kategórií: statické analýzy a dynamické analýzy. Statická analýza využíva informácie v podozrivých spustiteľných programoch bez toho, aby boli spustené, na základe kontroly binárneho kódu a čítania priamo v assembleri. Tieto prístupy sú sľubné, ale ukazuje sa, že disassemblovanie spustiteľného kódu môže byť ťažký problém. Približne u 90% vírusov nie je možné plne disassemblovať binárny kód a zachytiť tak ich rutinné správanie (Dai et al., 2009). Dynamická analýza sleduje činnosť programu po načítaní do pamäte. Dynamický prístup sa teda zameriava viac na aktivity, ale je ťažké vopred zabezpečiť, aby sa predišlo poškodeniu, pretože vírus, ktorý testujeme, je "spustený".
Výhodou týchto prístupov oproti bežne zaužívaným metódam je schopnosť odhaľovania vírusov, ktoré sa ešte nenachádzajú vo vírusovej databáze, a teda možnosť odhaliť doposiaľ neidentifikované vírusy a ich derivácie.
Malware sa snaží preto hľadať stále nové prístupy pre schovávanie svojej činnosti a prítomnosti v systéme. Stále viac sofistikované techniky využívajú hlavne rootkity. Rootkity predstavujú samostatnú oblasť na poli malware, ktorých cieľom je schovanie svojej činnosti a činnosti iného malware a ktoré využívajú značne pokročilé techniky.

Jednou z používaných techník je aj aktivovanie, existencia a schovávanie sa škodlivého kódu iba v operačnej pamäti počítača. Čím ďalej, tým viac sa takéto techniky schovávania začínajú používať v rôznych modifikáciach. Výhodou týchto malware, ktoré existujú iba v pamäti počítača, je, že sa veľmi ťažko zisťujú a ich činnosť väčšinou bežnému užívateľovi nie je zviditeľnená. Mnohokrát užívateľ ani netuší o existencii malware.

Malú efektivitu súčasných techník na odhaľovanie malware si uvedomujú aj mnohé výskumné laboratóriá a z tohto dôvodu sa snažia zamerať výskum na hľadanie účinnejších techník. Problematika detekcie škodlivého kódu sa stala jednou z najviac vyvíjajúcich sa oblastí vo svete. V mnohých publikovaných prácach sa k detekcii využívajú metódy dataminingu, čo sa na základe prezentovaných výsledkov detekcie javí ako sľubná metóda. Úroveň detekcie malware sa však značne mení, čo je zrejme zapríčinené výberom rôznych vlastností statických aj dynamických, pomocou ktorých identifikujeme rozdiely medzi neškodným programom a malware. Dané práce sa zaoberajú len bežnými formami malware a neriešia otázku malware rezistentných iba v operačnej pamäti.

Z článku, ktorý zahŕňa doterajšie výsledky prác v oblasti využitia datamining techník (Bazrafshan et al., 2013) vyplýva, že výsledky z veľkej miery závisia na nájdení vhodných vlastností, ktoré sa pri klasifikácii použijú. Bolo vidieť, že prístupy sú rôzne, ale všetky platia len pre určitú skupinu malware a nie je možné ich zovšeobecňovať (aj keď niektoré výsledky sú sľubné). Mnohé testy boli úspešné preto, lebo boli navrhnuté na základe známych vlastností vírusov. Ak sa v budúcnosti tvorcovia malware zamerajú na zmenu týchto vlastností, účinnosť detekcie zrejme môže značne poklesnúť. Ako riešenie by sa javilo nájsť také vlastnosti, ktoré by mohli popisovať čo najpresnejšie rozdiely medzi škodlivým a neškodným programom a zároveň boli ťažko (ak sa má funkcionalita zachovať) zmeniteľné. Medzi také môžeme zaradiť aj vlastnosti popisujúce správanie. Ako zaujímavým riešením z tohto hľadiska by mohol byť návrh

vlastností, ktoré môžeme extrahovať priamo z operačnej pamäte systému, a tým zjednodušiť proces získavania a spracovania potrebných údajov pre extrakciu vlastností (čo sa v súčasnosti stáva čoraz zložitejšie z dôvodu existencie rôznych techník ochrany kódu proti disassemblovaniu, šifrovaniu programov a jeho ukrývaniu iba v pamäti OS).

1 Ciele dizertačnej práce

Cieľom tejto práce je hľadanie nových vlastností, ktoré by sa získavali analýzou pamäte OS a dosahovali by čo najlepšiu úroveň detekcie malware. Ďalej otestovať dané vlastnosti pre vybranú metódu strojového učenia. Táto téza zahŕňa nasledovné úlohy:

- návrh a analýza možných vlastností extrahovaných priamo z pamäte operačného systému, ktoré by zabezpečovali čo najlepšiu úroveň detekcie škodlivého kódu prítomného v pamäti OS,
- návrh a implementácia metódy extrakcie vybraných vlastností (pre vlastnosti získavané priamo z pamäte OS),
- návrh modelu pre testovanie a jeho aplikácia,
- testovanie na štatisticky dôveryhodnej vzorke a natrénovanie klasifikátorov
- na základe testovaných vlastností, sledovanie úspešnosti odhalenia škodlivého kódu.

S prvou úlohou hlavnej tézy súvisí nájdenie spôsobu pre efektívny prístup do pamäte a analyzovania získaných dát, ktoré by bolo možné spracovať v reálnom čase.

2 Dosiahnuté výsledky dizertačnej práce

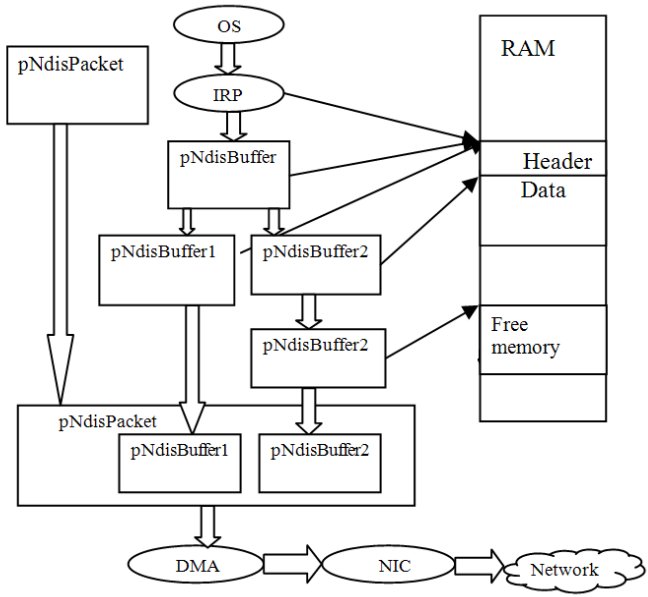
Po našej analýze dostupných riešení pre čítanie obsahu sme zistili, že žiadne riešenie nevyhovuje požiadavkám potrebným pre využitie k detekcii malware.
Základné požiadavky kladené na riešenie:

- čítanie obsahu pamäte nezávisle a bez možnosti filtrovania obsahu dát procesorom a OS,
- obsah pamäte uložiť na neprepisovateľné médium alebo odosielať na iný počítač v sieti, na ktorý sledovaný systém nemá dosah a nemôže dáta modifikovať,
- nástroj bude možné použiť všeobecne bez nutnosti špeciálneho zásahu do systému alebo špeciálneho hardvéru.

Preto bolo potrebné navrhnúť a vytvoriť vlastné riešenie, ktoré by sme mohli využiť aj pre detekciu malware. Aby sme splnili požiadavky zo súčasne dostupných možností pre získanie obsahu pamäte, museli sme vylúčiť softvérové riešenia, ktoré môžu byť malwarom operujúcim na úrovni operačného systému kompromitované. Ako najvhodnejšie sa javilo riešenie s využitím DMA sieťovej karty, pomocou ktorej môžeme čítať obsah pamäte a následne dáta odosielať cez sieťovú kartu na iný (nami určený) počítač. DMA prenos nie je riadený procesorom, preto prenášané dáta nemôžu byť modifikované malware (rootkitom), ktorý získal systémové práva a má možnosť využívať plnú funkcionalitu procesora.
Výsledný program pozostáva z dvoch častí, a to je samotný ovládač a riadiaci program. Riadiaci program vyplní hlavičku paketu potrebnými údajmi, a to zdrojovou a cieľovou MAC adresou a typom paketu (ethertype). Taktiež nastaví všetky potrebné parametre, ako je dĺžka paketu a fyzická adresa pamäte, ktorú chceme kopírovať. Všetky tieto údaje sa pošlú ovládaču ako jeden paket. Ovládač NDIS vytvorí dva nové samostatné buffre ndis_buffer1 a ndis_buffer2 pakety. Do prvého uloží dáta z

hlavičky paketu a do druhého na mapovanú fyzickú adresu na virtuálnu o veľkosti 1500 bajtov. Potom buffer1 a buffer2 znovu zlúčime do jedného paketu, ktorý sa pošle do siete na cieľovú MAC alebo IP adresu z hlavičky (pozri obr. č. 1). Pakety, ktoré sú poslané do siete a majú nastavenú hodnotu 8999 v poli ethertype, odchytáva riadiaci program a obsah druhej sekcie paketu uloží do súboru.

V prípade, ak chceme iba časť pamäte, riadiaci program pošle iba jeden popr. niekoľko paketov (podľa požadovanej veľkosti) ovládaču s adresou začiatku tejto časti. V prípade, že chceme celý obraz pamäte, riadiaci program v cykle posiela fyzické adresy od 0 až po koniec pamäte. Takto postupne získame obsah celej pamäte.



Obrázok 1 Základný princíp čítania a kopírovania obsahu operačnej pamäte

Aby sme overili správnosť nášho riešenia čítať obsah celej pamäte alebo vybranej oblasti, vykonali sme niekoľko testov. Jedným z nich bolo vytvorenie kópie celej pamäte (dump) s použitím nášho riešenia a porovnanie s dumpom pamäte vytvoreného pomocou nástroja "MDD tool". Získané kópie boli zhodné. Po testovaní sme tiež analyzovali nami vytvorený obraz pamäte v programe Volatility Framework. Obraz bol Volatility Frameworkom rozpoznaný a plne funkčný. Tým sme overili, že naše riešenie je použiteľné aj pre digitálnu forenznú analýzu a uložené dáta sú v tvare rozpoznateľnom nástrojmi pre analýzu obsahu pamäte.
Použitie systémového ovládača prináša niektoré výhody pre naše riešenie. Popri jeho využití na riadenie procesu získania obsahu pamäte sme zároveň schopní z pamäte získať adresy dôležitých systémových štruktúr a tabuliek. Po získaní počiatočnej adresy a jej veľkosti dokážeme danú oblasť pamäte odoslať na vzdialený počítač pre podrobnú analýzu. Takým spôsobom môžeme testovať integritu rôznych štruktúr a systémových tabuliek (napr. SSDT). Samozrejme týmto možnosti nástroja nekončia. Možnosť sledovania rôznych systémových štruktúr a objektov alebo obsahu celej pamäte nám dáva možnosť veľkej variability využitia. Pri zohľadnení výsledkov testov a možností využitia môžeme považovať naše riešenie za potenciálne vhodné pre potreby detekcie aj sofistikovaného a pokročilé techniky schovávania používaného malware. Jednou z variant nasadenia môže byť aj integrácia riešenia do agenta pre zber dát z operačného systému pri komplexnejších bezpečnostných riešeniach.

V druhej časti práce sa zameriavame na identifikáciu vlastností, ktoré by boli systémové (teda nie generované zo vzoriek malware) a zároveň by niesli informáciu o správaní malware. Také vlastnosti môžu byť značne nápomocné pri detekcii a mohli by byť použité univerzálne pre všetky typy malware, ktoré by dané správanie vytváralo. Pri výbere týchto vlastností potrebujeme nájsť odpoveď na nasledujúce otázky:

- ktoré vlastnosti zo systému vypovedajú o prítomnosti malware (určitým spôsobom) a sú vhodné pre ich detekciu,
- v akej miere neznámy potenciálny škodlivý kód využíva vybrané vlastnosti,
- aká je správnosť detekcie s využitím vybraných vlastností.

Pre zodpovedanie týchto otázok sme vytvorili metodológiu na testovanie. Pri výbere vlastností sme vychádzali z podmienok uvedených vyššie, teda ide o systémové údaje, ktoré majú určitým spôsobom vypovedať o správaní programu. V našom prípade sme sa zamerali na vlastnosti, ktoré sú dostupné pomocou analýzy pamäte.

Metodológia testovania

Navrhnutá metodológia pozostávala z nasledujúcich hlavných častí:

- Výber vhodných parametrov systému, ako vlastnosti pre metódy strojového učenia (vlastnosti vhodné pre detekciu je možné sledovať pomocou analýzy pamäte).

- Vytvorenie prostredia, kde je možné spúšťať malware a sledovať vybrané vlastnosti spustením každého malware samostatne. Zachytiť (zaznačiť) všetky sledované vlastnosti pre ďalšie spracovanie a uložiť ich do súborov. Prostredie je potrebné nastaviť tak, aby sa čo najviac podobalo reálnemu systému, aby sme minimalizovali počet malware, ktoré sa kvôli detekcii umelého prostredia neprejavia.

- Spracovanie vygenerovaných výstupov získaných pre každý malware do tvaru potrebného pre vyhodnotenie. V našom prípade sa výstupy ukladajú do formátu json alebo formátovaného txt súboru, aby sme mohli výstupy následne parsovať do databázy.

- Vyhodnotenie používania jednotlivých vlastností pomocou SQL jazyka.

- Vytvorenie datasetov s upravenými dátami z testov vhodnými pre algoritmy strojového učenia.

- Vykonanie testovania pomocou modelov strojoveho učenia a testovanie vhodnosti jednotlivých vlastností pomocou feature selection metódy.

Metodika nám umožňuje zistiť pre nás podstatné informácie o vlastnostiach a dať nám odpoveď na otázky formulované na začiatku sekcie.

Na základe poznatkov o v súčasnosti používaných technikách malware sme ako informácie pre sledovanie v pamäti navrhli tieto systémové objekty: VAD štruktúry, Atomy, Mutexy, záznamy v zozname modulov, Spätné volania (Callbacks), Timery a Sokety.
Podľa vytvorenej metodiky sme vykonali testovanie pomocou testovacieho prostredia, ktoré bolo potrebné zabezpečiť tak, aby spĺňalo požiadavky na bezpečnosť a automatické vykonávanie testov. V prvej časti testov sme vykonali zber dát (nami vybraných informácií) z pamäte počítača pri spustení testovacích súborov. V druhej časti testov sme sledovali ich významnosť pre správnosť a presnosť klasifikácie pre vybrané algoritmy strojového učenia a datamining klasifikátory.

Pre overenie významnosti vlastností pre detekciu malware sme vytvorili testovací dataset, ktorý obsahoval testované vlastnosti a niektoré ich parametre. Výpočet významnosti bol realizovaný aplikovaním forward selection metódy na výber vhodných vlastností pre datamining klasifikátory a s využitím výsledkov váhovania pomocou univariantných metód. Na testovanie sme použili IBM bladeCenter HS23 s 8 CPU a 128GB RAM a s inštalovaným OS Debian verzia 7.

Hodnotenie významnosti atribútov na základe váhovania a forward selection metód ukazuje na významnosť atribútov, ktoré ukazujú na použitie DLL injekcie vo väčšine testovaných vzorkách malware.

Z toho usudzujeme, že táto technika je v súčasnosti jedna z najviac používaných, alebo že testovací malware dataset, ktorý sme mali k dispozícii obsahoval vzorky malware, ktoré využívali pri svojej činnosti práve DLL injekciu. To nás priviedlo k myšlienke vytvorenia v ďalšom budúcom výskume testy, ktoré by zisťovali úroveň závislosti významnosti atribútov na typoch malware.

Pomocou vykonaných testov sme vybrali vhodné atribúty (podľa významnosti) pre datamining klasifikátory na základe výsledkov validácie a hodnôt správnosti a presnosti.

Tabuľka 1 Výsledky testov pomocou Forward selection

	Všetky prvky		Výber 100 prvkov z malware	
	Accuracy	Precision	Accuracy	Precision
DTBN	90.78% +/- 2.85	66.44% +/- 7.03	95.02% +/- 3.78	94.27% +/- 6.48
ID3 num	95.31% +/- 1.79	87.05% +/- 2.71	94.57% +/- 3.97	94.16% +/- 6.56
J48	93.81% +/- 1.05	80.34% +/- 6.10	94.15% +/- 4.47	93.11% +/- 5.92
Jrip	94.31% +/- 3.10	83.14% +/- 9.89	95.47% +/- 2.88	95.07% +/- 4.96
Ridor	94.14% +/- 2.91	83.58% +/- 13.87	95.02% +/- 3.19	94.31% +/- 6.00
NBTree	92.80% +/- 3.8	81.08% +/- 16.50	95.02% +/- 3.78	94.27% +/- 6.48

Pre nás sú tieto výsledky (pozri tab.č.1) ukazovateľom, že testované vlastnosti poskytujú potrebné informácie pre klasifikáciu a detekciu malware. Nehovoria však o správnosti detekcie a klasifikácie na základe nich naučeného systému, preto nevieme porovnať naše riešenie v tomto štádiu s inými výsledkami detekčných nástrojov. K tomu by sme potrebovali vytvoriť komplexné riešenie pre zber dát a ich klasifikáciu, čo nebolo cieľom tejto práce, a potom testovať úspešnosť detekcie s použitím navrhnutých atribútov. Z dostupných prác nie je známe získavanie a využitie nami sledovaných atribútov (informácií) získavaných priamo z pamäte alebo podobných informácií, ako vstup pre datamining algoritmy alebo automatické systémy pre detekciu malware.

V tomto je naša práca inovatívna. Výsledok práce môže prispieť k vývoju účinných senzorov na detekciu rôznych vektorov útoku, na systém a jadro systému (zmena integrity systémových tabuliek, zmeny v dôležitých štruktúrach systému, zmena integrity ovládačov atď.). Navrhnuté vlastnosti a výsledky testovania ukazujú tiež na možnosť tieto vlastnosti monitorovať a úspešne využiť pre detekciu napadnutia systému. Tým môžeme pokryť

sledovanie väčšej škály správania sa typického pre malware, a tak zlepšiť úspešnosť jeho detekcie. Obrovský nárast počtu malware núti použiť ako jediný možný prístup automatizované nástroje pre detekciu. Táto práca môže byť prínosom v tomto úsilí.

3 Literatúra

Bazrafshan, H., Hashemi, H., Mehdi, S. 2013. A Survey on Heuristic Malware Detection Techniques. 5th Conference on Information and Knowledge Technology. 2013

Dai, J., Guha, R., Lee, J. 2009. Efficient Virus Detection Using Dynamic Instruction Sequences. In Journal of Computers [online]. May 2009, vol. 4, no. 5, p. 405-414. Dostupné na internete: <doi:10.4304/jcp.4.5.405-414>.

Daoud, E. A., Jebril, I. H., Zaqaibeh, B. 2008. Computer Virus Strategies and Detection Methods. In Open Problems Compt. Math. Vol. 1, no. 2, September 2008.

4 Zoznam publikácií

4.1 Publikované výsledky dizertačnej práce a príspevky na konferenciách

Balogh, Štefan - Mydlo, Miroslav: New Possibilities for Memory Acquisition by Enabling DMA Using Network Card.
In: IDAACS 2013 : 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Berlin, Germany, September 12-14, 2013. - Piscataway : IEEE, 2013. - ISBN 978-1-4799-1426-5. - Vol. 2, p. 635-639

Balogh, Štefan:Memory Acquisition by Using Network Card.
In: Journal of Cyber Security and Mobility (pages 65–76), Vol: 3 Issue: 1, January 2014, ISSN: 2245-1439 (Print Version), ISSN: 2245-4578 (Online Version),

Balogh, Štefan - Mydlo, Miroslav: New Possibilities for Reading Memory Contents by Enabling DMA Using NDIS Driver.
In: Memics 2012. 8th Doctoral Workshop on Matematical and Engineering Methods in Computer Science. Znojmo, October 25-28, 2012. - Brno : Novopress, 2012. - ISBN 978-80-87342-15-2. - S. 12 s

4.2 Ostatné

Balogh, Štefan: Forensic Analysis, Cryptosystem Implementation and Cryptology: Methods and Techniques for Extracting Encryption Keys from Volatile Memory. In: Multidisciplinary Perspectives in Cryptology and Information Security. - Hershey : IGI Global, 2014. - ISBN 978-1-4666-5808-0. - S. 381-396, [1.46 AH]

Lehocki, Fedor - Balogh, Štefan - Kováč, Miroslav - Žákovičová, Eva - de Witte, Bart: Innovative Telemedicine Solutions for Diabetic Patients.
In: 2012 IEEE-EMBS Conference on Biomedical Engineering (IECBES 2012) : Langkawi, Malaysia, 17-19th December 2012. - Piscataway : IEEE, 2013. - ISBN 978-1-4673-1666-8. - S. 203-208

Balogh, Štefan - Pondelík, Matej: Capturing Encryption Keys for Digital Analysis.
In: IDAACS 2011 : 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. Prague, Czech Republic, 15 - 17 September 2011. - Piscataway : IEEE, 2011. - ISBN 978-1-4577-1425-2. - S. 759-763

Balogh, Štefan - Lehocki, Fedor - Ivaniš, Daniel - Kučera, Erik - Lajtman, Miloš - Miňo, Igor: Data Processing from mHealth Patient Data Acquisition Related to Extracting Structured Data from EH Records.
In: Wireless Mobile Communication and Healthcare : 3rd International Conference MobiHealth 2012. Paris, France, November 21-23. 2012. - Berlin Heidelberg : Springer, 2013. - ISBN 978-3-642-37892-8. - S. 255-262

Balogh, Štefan - Chovančák, Dušan: Použitie dataminig metód pri odhaľovaní nových typov malware a pri sieťových útokoch.
In: Znalosti 2011 : 10. ročník konference, Stará Lesná, Vysoké Tatry, 31. ledna - 2.února 2011. Sborník příspěvků. - Ostrava : Fakulta elektrotechniky a informatiky, VŠB - Technical University of Ostrava, 2011. - ISBN 978-80-248-2369-0. - S. 255-258

Balogh, Štefan: Metódy detekcie vírusov a škodlivého kódu.
In: EE časopis pre elektrotechniku, elektroenergetiku, informačné a komunikačné technológie. - ISSN 1335-2547. - Roč. 16, mimoriadne č. : ELOSYS. Trenčín, 5.-8.10.2010 (2010), s. 11-16

Balogh, Štefan - Skokan, Lukáš: Návrh architektúry pre dynamickú analýzu kódu.
In: ELOSYS. Elektrotechnika, informatika a telekomunikácie 2012 [elektronický zdroj] : Trenčín, 9.-12.10.2012. - Bratislava : FEI STU, 2012. - ISSN 1335-2547. - S. 126-130

Balogh, Štefan - Varga, Juraj: Threats in Mobile Security.
In: ELITECH´12 [elektronický zdroj] : 14th Conference of Doctoral Students. Bratislava, Slovak Republic, 22 May 2012. - Bratislava : Nakladateľstvo STU, 2012. - ISBN 978-80-227-3705-0. - CD-ROM, [6] s.

Balogh, Štefan: Using Memory Analysis for Malware Detection.
In: ELITECH´13 [elektronický zdroj] : 15th Conference of Doctoral Students; Bratislava, Slovakia, 5 June 2013. - Bratislava : Nakladateľstvo STU, 2013. - ISBN 978-80-227-3947-4. - CD-ROM, [6] s.

Pondelík, Matej - Balogh, Štefan: Capture of Encryption Keys by Analyzing Memory.
In: ELITECH´10 : 12th Conference of Doctoral Students. Bratislava, Slovak Republic, 26.5.2010. - Bratislava : STU v Bratislave, 2010. - ISBN 978-80-227-3303-8. - CD-Rom

Kubačka, Slavomír - Balogh, Štefan: Electronic Health Records.
In: Meditech - Proceedings of the ESF Project Conference : Innovative Program of Modern Biomedical Technologies. Project No. SORO/JPD-26/2005. Bratislava, Slovakia, 26.5.2008. - Bratislava : STU v Bratislave, 2008. - ISBN 978-80-227-2881-2. - CD-Rom

Pavlovič, Andrej - Lehocki, Fedor - Balogh, Štefan: Security of Electronic Health Records.
In: Meditech - Proceedings of the ESF Project Conference : Innovative Program of Modern Biomedical Technologies. Project No. SORO/JPD-26/2005. Bratislava, Slovakia, 26.5.2008. - Bratislava : STU v Bratislave, 2008. - ISBN 978-80-227-2881-2. - CD-Rom

Balogh, Štefan - Lehocki, Fedor - Juhás, Gabriel: Ehealth na Slovensku a jeho implementácia.
In: SMART S2AI : Workshop SMART systémov a služieb v oblasti aplikovanej informatiky. Bratislava, 18. apríla 2011. - Bratislava : STU v Bratislave FEI, 2011. - ISBN 978-80-227-3513-1. - S. 11-13

Lehocki, Fedor - Oravec, Miloš - Balogh, Štefan - Juhásová, Ana: Vybrané modely diagnostických systémov.
In: SMART S2AI : Workshop SMART systémov a služieb v oblasti aplikovanej informatiky. Bratislava, 18. apríla 2011. - Bratislava : STU v Bratislave FEI, 2011. - ISBN 978-80-227-3513-1. - S. 1-5

Summary

This work focuses on the question how to use information from the volatile memory of a computer for malware detection. To achieve this goal we divide the solution into two parts.

The first part of work deals with the study and implementation of direct memory access that satisfies the requirements for rootkit detection and for forensics analysis. In these cases we need to get the data from memory without a possible intervention of the operating system, or some rootkit installed on the computer. Therefore it is appropriate to implement a memory access in a way which is not controlled by the processor. Thus the rootkit cannot monitor and control the transfer, and the requested data. The obtained data are then stored outside the infected computer. The network cards seem to be the ideal solution in this case, as they have the hardware DMA into memory, and can immediately send the collected data over a network to a remote machine for the data analysis.
Our design follows this idea. We implemented a new method that allows us to dump the memory content using NDIS driver, and the network card (NIC). It would also be possible to replace the NDIS driver used in the attack from protocol driver to intermediate or miniport driver. Then it would be possible to setup the NDIS driver not only by the control program installed on the local machine, but also through the network. The other advantage is that we can write, using NDIS driver, specific functions for getting the physical address of many critical OS components and structures. Using this information we can check the exact part of physical memory corresponding with protected or monitored system components (IAT / EAT / SSDT / IDT/ IRP tables, kernel functions (using integrity checks) or key data structures.
In this way, new possibilities for forensic purposes, as well as

rootkit detection over the network can be enabled. It would still be necessary (and very difficult in practice) to ensure that the modified NDIS driver is not misused by unauthorized entities, and to implement proper security mechanism to protect the driver we implemented new solution for detecting memory access in selected memory area, so we can protect our driver. Our DMA-enabling driver can be implemented in practice as a part of the data collection agent used in a more complex security solution.
Taking into account all advantages of the solution (access to memory using DMA, sending contents of the memory to other computers, control of the driver through network), it seems to be a promising approach for enabling memory access. The ability can be really helpful for forensics analysis, malware and rootkit detection, or as a generic basis for data collection agents.

In the second part, we focus of using the suggested acquired data for malware detection based on machine learning and data mining algorithms. Feature selection was used to test suitability of specific in-memory system data structures to serve as malware indicators. The results of data-mining indicate that analysed features can provide significant detection rates to be used in automatic malware detection software.