O operaţiune cu stil – The Flame

Lect. univ. dr. Cătălin VRABIE
SNSPA, Facultatea de Administraţie Publică

vrabie.catalin@gmail.com

Rezumat:

În acest capitol voi vorbi despre Flame*, un malware deosebit de sofisticat care a atras foarte mult atenţia specialiştilor în informatică şi securitate digitală la momentul descoperirii lui – 28 mai 2012. Aplicaţia, pentru că în esenţa asta este, a fost considerată a fi cea mai laborios dezvoltată armă cibernetică descoperită până la momentul lansării ei (Hanlon, 2012). Fără îndoială, după 2012 au existat şi alte – să le spunem – „iniţiative", dar niciuna nu a reuşit să atingă nivelul de complexitate de care s-a bucurat* Flame *(cunoscută şi cu numele de* Flamer *şi* sKyWIper*).*

Flame – numele şi l-a luat de la linia de comandă care ordona computerului vizat executarea *malware*-ului (Figura 1, stânga) – [FROG.DefaultAttacks.A-InstallFlame] (Zetter, 2012) – s-a dovedit a fi un instrument foarte sofisticat de atac cibernetic, înzestrat cu extraordinare capacităţi de penetrare şi infestare a sistemelor, de culegere a datelor şi, poate mai presus de orice, de a se ascunde pe sine şi efectele „muncii" lui. Este suficient să ne gândim că cei de la Kaspersky Lab au spus că, până în momentul descoperirii lui, opera deja de peste doi ani – din februarie 2010 estimează ei, şi ne dăm seama cât de abil era în a se adapta metodelor şi tehnicilor de protecţie, precum antiviruşii şi *firewall*-urile (Gostev, 2012).

Din fericire, dacă putem spune aşa, ţinta lui pare să fi fost Orientul Mijlociu (Figura 1, dreapta), astfel că este foarte puţin probabil ca europenii sau americanii să fi văzut „în libertate" vreun exemplar al lui Flame. Totodată, s-a observat că unicul scop pe care l-a avut aplicaţia a fost acela de a culege informaţii şi nu a ascuns nici o activitate de natură financiară cu scopul de a se realiza un profit de pe urma dezvoltării şi lansării lui (Clark, 2012), aşa cum fac astfel de aplicaţii atunci când sunt dezvoltate de companii private.

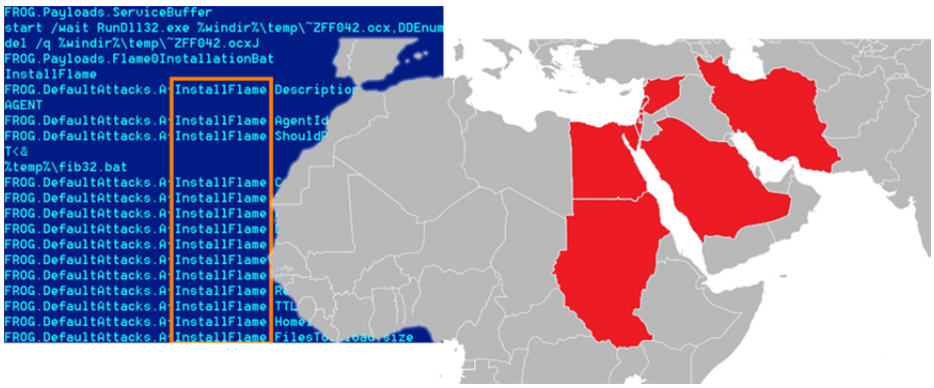

Figura 1. Flame – (1) originea numelui și (2) răspândirea geografică.
Sursa: (1) https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/, (2) http://www.wired.com/2012/05/flame/ (după date preluate de la Kaspersky Lab)

Când ne uităm la aceste caracteristici, precum: (1) ținta – care, la fel ca în cazul Stuxnet, a fost Orientul Mijlociu; (2) faptul că a fost o aplicație software deosebit de complexă – lucru menționat, dar care va fi dezvoltat, mai pe larg, în continuare; și (3) indiferența față de realizarea unui profit – deși, în mod clar, o asemenea aplicație era mai mult decât capabilă să facă și acest lucru, nu poți să nu te gândești că dezvoltarea ei a fost finanțată de unul sau mai multe dintre guvernele lumii. Washington Post a publicat, pe 19 iunie 2012, un articol în care spunea că Flame a fost finanțat de NSA și CIA împreună cu organizații militare israeliene, ca parte a operațiunii Olympic Games – de care am mai vorbit în acest volum (Washington Post, 2012). Un argument în plus pentru acest lucru ar fi că Flame pare a avea același izvor precum DuQu (Symantec, 2011a), Gauss (ComputerWorld, 2012) și deja discutatul Stuxnet, care au fost primele arme cibernetice și de spionaj serioase, dezvoltate pentru a periclita stabilitatea unei (sau unor) națiuni și despre care se știe că au fost finanțate din surse guvernamentale. În plus, Eugene Kaspersky, fondator și CEO al Kaspersky Lab, a declarat, în mai 2012, pe profilul personal de Twitter, următoarele: *The complexity of The Flame, geography & targets leave no doubt this malware was state-sponsored* (Kaspersky, 2012).

***

Câteva cuvinte despre fiecare din *malware*-urile menționate, dar nediscutate până acum:

DuQu – malware-ul care a precedat Stuxnet. Deși a fost construit diferit de acesta, a avut rolul de a culege informații despre controlerele industriale care urmau să fie ținta lui (IEEE Spectrum, 2011).

Gauss – a fost dezvoltat cu rolul de a destabiliza sisteme bancare. El era capabil să culeagă date de conectare, precum parole, cookie-uri și alte informații de această natură de la: Citibank, MasterCard, American Express, Visa, PayPal, eBay, Gmail, Hotmail, Yahoo, Facebook, Amazon și alte câteva bănci din Orientul Mijlociu. Și existența acestuia este în strânsă legătură cu Stuxnet (ComputerWorld, 2012).

Flame, diferit de toate celelalte *malware*-uri în esența algoritmilor de programare și a arhitecturii, folosește totuși aceleași mecanisme pentru a-și atinge scopul (Dalziel, 2013). Cu toate astea, nu există nici un fel de evidență că aceste aplicații ar fi fost scrise de aceeași echipă. Ba mai mult, totul pare să ducă la ideea că au fost scrise în paralel, fie cu scopul ca o aplicație să asigure spatele celeilalte în caz de eșec – un fel de *backup* al operației, fie pentru a culege informații sau a sprijini operațiunea în ansamblul ei.

Un alt element care trebuie menționat este că Flame a avut un spectru de acțiune mai larg decât DuQu – acesta, ca efect al libertății de acțiune restrânse, a infestat un număr mult mai mic de computere (ComputerWorld, 2011) – toate, însă, aparținând unor utilizatori cu un profil foarte înalt (Kaspersky Lab ZAO [RU], 2012a). Aplicația asupra căreia îmi focalizez atenția, în acest capitol, a reușit să infesteze, cu succes, un număr de aproximativ o mie de calculatoare (Kaspersky Lab ZAO [RU], 2012a). Chiar dacă numărul victimelor poate părea mic, în dezvoltarea unei aplicații de asemenea calibru, utilizatorii casnici nu sunt importanți. Aici ținta a fost mult mai înaltă. Atât DuQu, cât și Flame au avut același scop: să fure informații de care dezvoltatorii ar fi avut nevoie pentru a crea alte arme cibernetice mai sofisticate și mai bine țintite.

Cu toate că este foarte puțin probabil ca un utilizator obișnuit de computer să se infesteze cu Flame în mod direct, ce trebuie înțeles este că simpla existență a *malware*-ului deschide noi oportunități de dezvoltare, de creare a unor versiuni modificate – unele poate chiar pe structura Flame (este ușor de înțeles că nu există drepturi de *copyright* pentru această aplicație – ca, de fapt, pentru niciuna de această natură). Exemple mai vechi, așa cum am menționat în volumul de față, ar fi virusul „Melissa", care a fost urmat imediat de „I LOVE YOU". DuQu 2.0, spre exemplu, a fost descoperit tot de Kaspersky Lab, în iunie 2015, iar declarația oficială a acestora a fost: *The philosophy and way of thinking of the 'Duqu 2.0' is a generation ahead of anything seen in the APT (Advanced Persistent Threat) world* (Kaspersky, 2015a). Dacă mai adăugăm faptul că DuQu 2.0 a fost lansat cu ocazia evenimentelor P5+1[1] și în zona în care acestea au avut loc (Kaspersky, 2015b), putem să înțelegem mai bine natura acestor riscuri.

***

Mai departe, aș dori să descriu *malware*-ul în toate dimensiunile sale. (1) Vom vedea ce face Flame și care este comportamentul lui într-un sistem infectat. (2) De asemenea, vom desluși cum acesta pătrunde într-un computer și cum se propagă spre altele în căutarea țintei și (3) vom înțelege care sunt caracteristicile unice ale lui Flame. Cu alte cuvinte, vom afla cum reușește să fie atât de flexibil și eficient.

(1) Ce face?

Fără îndoială, Flame fură informații (BBC, 2012; Symantec, 2012, Kaspersky Lab ZAO [RU], 2012a).

[1] P5+1 reprezintă un grup de țări format din China, Franța, Rusia, Marea Britanie și Statele Unite plus Germania, care își unesc eforturile diplomatice pentru a discuta planurile Iranului de dezvoltare nucleară. Întâlnirea din 2015 a avut loc la Lausanne, Elveția, în data de 2 aprilie.

- Captează sunetul, dacă sistemul are microfon încorporat (trebuie menţionat că toate laptopurile au aşa ceva), şi are senzori *software* care se declanşează automat atunci când anumite aplicaţii, precum Skype, sunt lansate în execuţie;
- Execută *screenshots* la anumite intervale de timp sau când utilizatorul întreprinde anumite sarcini, precum atunci când discută pe canale de *chat*. Ceea ce este interesant la aceste acţiuni este că utilizatorul poate fi logat pe astfel de canale prin mijlocirea unui VPN şi, astfel, conversaţiile pot fi criptate, dar având în vedere că se execută *screenshot*-uri, criptarea, în contextul dat, devine inutilă;
- Reţine parolele sau alte date de identificare ale utilizatorilor;
- Înregistrează imputurile de la nivelul tastaturii. În cazul sesiunilor de *chat*, mesajele transmise de pe sistemul infestat sunt uşor de capturat prin această metodă – cele primite sunt capturate prin executarea *screenshot*-urilor;
- Scanează, via Bluetooth, dispozitivele din apropiere care au opţiunea „discoverable mode" setată pe ON, îşi anunţă prezenţa pe acestea şi, implicit, îşi acordă accesul fără a-l mai anunţa pe utilizator acest lucru. În felul acesta, mapează zonal reţeaua de dispozitive Bluetooth;
- Citeşte şi fură date din documentele de interes existente în sistemul infestat.

Totodată, Flame acţionează ca un *botnet* (Gostev, 2012) – o reţea de computere interconectate şi care comunicând cu altele (în cazul de faţă, este vorba de preluarea de informaţii de la sistemele infectate), realizează sarcini dificile care nu pot fi executate local datorită gradului ridicat de complexitate, lucru care ar atrage atenţia utilizatorului – de exemplu, datorită unei funcţionări defectuoase (SANS, 2003). Flame a infestat câteva mii de noduri de reţea şi aproximativ 80 de domenii Internet cărora le-au fost asociate un număr de aproximativ 25 de adrese IP din toată lumea: Germania, Olanda, Marea Britanie, Elveţia, Hong Kong, Turcia, Polonia şi Malaezia, transformându-le în servere C&C (Command-and-Control) (Gostev, 2012; CNET, 2012).

Toate informaţiile furate de aplicaţie de pe computerul victimei erau întâi comprimate şi apoi transmise periodic, prin protocoale de reţea bine securizate şi criptate – precum SSL sau SSH, către un server C&C (Sotirov, 2012). Este suficient să ne gândim numai la faptul că sunt folosite canale criptate de transmisie a datelor furate şi ne putem da seama de importanţa acestora.

Niciuna din acţiunile prezentate mai sus nu relevă intenţiile dezvoltatorilor *malware*-ului. Nu se poate vorbi nici de intenţii clare de monitorizare, ci mai curând pare că dezvoltatorii au avut în vedere acumularea de mari cantităţi de informaţii (Kamluk, 2012), care să-i ajute, mai departe, în a dezvolta alte aplicaţii de cyberspionaj.

(2) Cum se propagă?

Flame ar fi cel mai uşor de asemănat cu aplicaţiile de tip „Cal Troian" (*Trojan*), foarte des întâlnite astăzi (Symantec, 2012). În contrast, însă, cu acestea, el nu parazitează alte fişiere pentru a se infiltra, folosind pentru asta fişiere şi metode proprii – este o aplicaţie de tipul *standalone* (InformationWeek, 2012; Udi, 2012).

De asemenea, mai are şi caracteristici de *backdoor*: deschide o cale de acces către sistemul infectat prin care un utilizator extern îşi poate extinde atribuţiile de administrare şi asupra acestuia (Kaspersky Lab ZAO [RU], 2012a).

Ultima caracteristică, dar poate cea mai importantă dintre toate, este că metoda de infectare copiază, totodată, comportamentul unui vierme informatic (*Worm*), uitându-se după gazde locale pe care le poate infecta (Symantec, 2012; Kaspersky Lab ZAO [RU], 2012a).

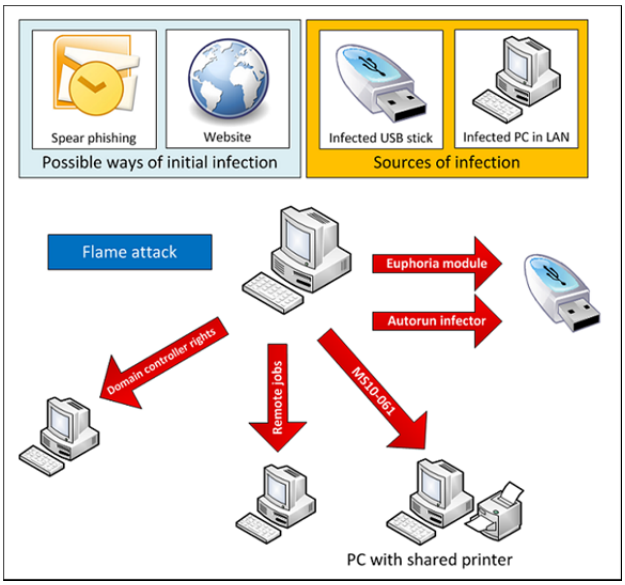

Figura 2. Propagarea *malware*-ului Flame.
Sursa: https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/

Mai înainte de toate, trebuie spus că Flame s-a propagat foarte uşor în reţeaua locală (Figura 2). Maniera în care făcea acest lucru era prin a exploata vulnerabilităţile acesteia, şi anume: MS10-061 – o vulnerabilitate a protocoalelor de printare, Print Spooler Service, şi MS10-046 (CrySyS, 2012) – o vulnerabilitate a sistemului de protecţie a Windows-ului – *Windows shell*, care s-a răspândit prin Windows Update şi care a afectat toate sistemele de operare produse de Microsoft, începând cu Windows XP şi până la Windows Server 2008 (mai enumăr aici pentru exemplificare: Windows Server 2003, Windows 7, atât versiunea pe 32 de biţi, cât şi cea pe 64 etc.). Exploatată corespunzător, prin această ultimă vulnerabilitate, Windows-ul permitea saltul incorect al *link*-urilor interne (de unde şi denumirea ei de „lnk" în jargonul specific specialiştilor în securitate informatică), realizând, totodată, un serviciu de Web propriu – *WebDAV*, care transforma fiecare calculator infestat într-un server Web de unde plecau, spre celelalte calculatoare din reţea, presupuse *update*-uri, bineînţeles certificate, în mod fals, de Microsoft – tehnică denumită *Man in the Middle* (Udi, 2012) (Figura 3). Certificatele digitale folosite, în număr de trei, au fost fabricate de dezvoltatorii *malware*-ului în urma unei greşeli a Microsoft. Reacţia companiei a fost următoarea:

When we initially identified that an older cryptography algorithm could be exploited and then be used to sign code as if it originated from Microsoft, we immediately began investigating Microsoft's signing infrastructure to understand how this might be possible. What we found is that certificates issued by our Terminal Services licensing certification authority, which are intended to only be used for license server verification, could also be used to sign code as Microsoft. Specifically, when an enterprise customer requests a Terminal Services activation license, the certificate issued by Microsoft in response to the request allows code signing without accessing Microsoft's internal PKI infrastructure (Microsoft, 2012).

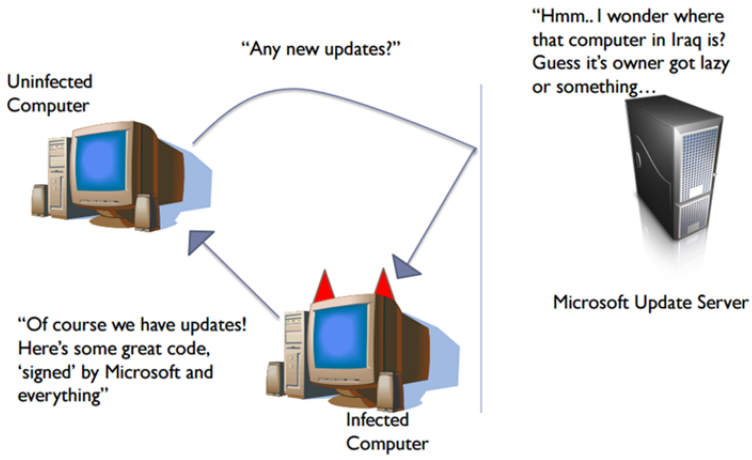

Figura 3. Tehnica Man in the Middle (MITM attack).
Sursa: Adam Udi, How did the 'Flame' family of malware spread?

Ceea ce este interesant la cele spuse mai sus, şi anume folosirea vulnerabilităţilor MS10-061 şi MS10-046, este că ambele au fost folosite şi de Stuxnet în atacul asupra centralei nucleare Natanz din Iran, ceea ce arată că aceste două *malware*-uri, deşi diferite din multe puncte de vedere, au avut totuşi caracteristici comune – probabil, echipele de dezvoltatori, chiar dacă au lucrat în paralel şi poate chiar independent una de alta, au avut acces la aceleaşi informaţii.

O altă metodă de răspândire a lui Flame este printr-un Domeniu – denumit de Microsoft „Active Directory" şi care reprezintă un server ce gestionează procedurile de autentificare a utilizatorilor într-o reţea de distribuţie a informaţiei, cum ar fi o reţea locală de calculatoare. Acesta, odată infestat de *malware*-ul în discuţie, creează căi de acces pe sistemele pe care le controlează pentru a facilita realizarea de copii locale ale acestuia – şi în acest caz, tehnica folosită este *Man in the Middle*.

Pe lângă metodele de propagare în reţelele locale, Flame se mai poate răspândi prin intermediul mediilor de stocare mobile – *memory stick*-uri sau orice alte tipuri de *drive*-uri USB (Figura 2) (Gostev, 2012). Pentru această sarcină, Flame a fost îmbogăţit cu capacitatea de a lansa procedura *autorun* în modul *stealth* (ascuns) – posibilitatea de a se lansa în execuţie imediat ce se conectează la un computer, fără ca utilizatorul să afle şi, astfel, să poată interveni în procesul de instalare (voi reveni, în cursul acestui capitol, la asta).

În ciuda tuturor acestor caracteristici de propagare şi infestare, Flame pare că-şi controlează răspândirea. Unul din mecanismele pe care le foloseşte, în acest scop, este „Mutex" (*mutual exclusion*) (CrySyS, 2012) – cunoscut, în programare, ca fiind o procedură de identificare a instanţelor proprii de lucru a unei aplicaţii în vederea evitării suprapunerii sau rescrierii acesteia (Taubenfeld, 2004). În cazul Flame, prin *mutex* se verifica dacă sistemul vizat este deja infestat – dacă răspunsul era „da", acel sistem era ocolit. Motivul pentru care opera în această manieră rezidă în faptul că mai multe instanţe operaţionale pe acelaşi computer ar fi făcut *malware*-ul mai „zgomotos" – adică ar fi putut atrage atenţia utilizatorului asupra sa, făcând, de exemplu, computerul să ruleze mai greu anumite aplicaţii. Rulând, însă, o singură instanţă, Flame putea lucra „în linişte". Un atacator obişnuit nu este interesat să-şi limiteze accesul în faza de infestare, el dorind o răspândire cât mai rapidă a *malware*-ului (Marshall & Wesley, 2013). În schimb, dezvoltatorii lui Flame au avut în vedere, încă de la bun început, aşa cum am mai spus, un număr foarte mic de computere, limitând, astfel, infestarea doar la scopul ei. Acest lucru demonstrează încă o dată că Flame a fost o armă cibernetică şi nu un instrument de lucru al grupărilor de *hackeri*.

(3) Care sunt caracteristicile malware-ului?

Cea mai notabilă dintre toate caracteristicile *malware*-ului Flame este dimensiunea – 20 MB (Kaspersky Lab ZAO [RU], 2012a). Această valoare nu pare a fi foarte mare, luând în considerare cerinţele actuale de spaţiu de stocare, însă pentru o aplicaţie de această natură, este incredibil de mare! Prin comparaţie, Stuxnet (cu care adesea este asemănat Flame, datorită rolului de armă cibernetică pe care le-au avut ambele) a avut aproximativ 500 KB (Symantec, 2011b) – o jumătate de MB, ceea ce face *malware*-ul nostru să fie de 40 de ori mai mare – trebuie menţionat aici că Stuxnet era deja considerat ca fiind de mari dimensiuni (PCWorld). 20 MB este considerat exagerat de mare pentru un *malware* şi asta în special pentru că dezvoltatorii acestor tipuri de aplicaţii nu agreează volumul din simplul motiv că durează prea mult pentru a fi *downloadat* în calculatorul victimei, limitându-se astfel răspândirea lui.

O a doua caracteristică importantă a lui Flame este arhitectura bazată pe *plug-in*-uri (Kaspersky Lab ZAO [RU], 2012a). Cu alte cuvinte, are o structură modulară inteligentă – întâi se încarcă instanţa de configurare – denumită *Boot_dll_loader*, care verifică sistemul proaspăt infestat şi determină acţiunile care trebuie executate mai departe. Dacă, spre exemplu, metoda de infestare ar fi fost prin USB, s-ar fi încărcat modulul *Infectmedia* care, la rândul său, determina următoarele acţiuni, încărcând fie modulul *Autorun_infector*, fie *Euphoria*, în funcţie de configuraţia sistemului; dacă nu, acestea ar fi rămas cuminţi în *kit*-ul *malware*-ului. De asemenea, dacă un computer nu era dotat cu microfon, *plug-in*-ul *Microbe* – cel care controla acest device – nu s-ar fi instalat. Au fost descoperite douăzeci de *plug-in*-uri care puteau fi lansate într-o instanţă Flame, ceea ce duce la o flexibilitate mare a *malware*-ului, acesta variind foarte mult de la un sistem infectat la un altul, în funcţie de nevoile şi cerinţele acestuia.

Un alt motiv pentru care Flame este atât de voluminos este că acesta conţine foarte multe componente, dintre care cele mai voluminoase ar fi librăriile de date – linii de cod, proceduri şi funcţii de programare, organizate după funcţionalitate. Astfel, avem (Kaspersky Lab ZAO [RU], 2012a):

- Un număr de trei mecanisme de arhivare, precum: zlib, libbz2 şi PPMD, care au fost implementate nativ pentru ca aplicaţia să poată rula independent de sistemul de operare şi aplicaţiile instalate pe acesta;
- SQLlite3 – o librărie de lucru cu bazele de date, dezvoltată pentru a fi rulată de pe computerele client a unor baze de date mai mari, instalate, de regulă, pe servere dedicate;
- O maşină virtuală (o emulare a unui mediu de lucru pentru executarea unor sarcini într-un alt limbaj de programare decât cel care este comun sistemului sau aplicaţiei instalate) pentru limbajul Lua – un limbaj extensibil care se bucură de interfaţă de lucru în C++, acesta fiind cel în care a fost scris, în mare parte, *malware*-ul în discuţie. În particular în cazul Flame, folosirea Lua este foarte interesantă, el nemaifiind niciodată folosit pentru atacuri cibernetice – de regulă, atacatorii folosesc limbaje de programare compacte, prin care s-ar dezvolta *malware*-uri de mici dimensiuni şi, astfel, uşor de ascuns. În plus, numărul de linii de programare scrise în acest limbaj, şi regăsite aici, nu este mai mare de câteva mii (CrySyS, 2012), ceea ce sporeşte curiozitatea pentru toţi cei care au încercat să înţeleagă Flame;
- Cinci algoritmi diferiţi de criptare. Trebuie menţionat că aceştia nu au fost foarte complicaţi astfel încât sistemului infestat să nu i se aglomereze procesorul, dar cu toate astea, nici foarte simpli, pentru a descuraja, în cazul în care ar fi fost captat un flux de date, intenţiile de decriptare (InfosecInstitute, 2012).

După cum deja se înţelege din descrierea caracteristicilor, Flame nu a fost un singur fişier executabil cu sarcina de a se lansa în momentul în care infesta un sistem. El s-a constituit dintr-o mulţime de fişiere de tip.dll (*dynamic-link library*) încărcate împreună şi, de asemenea, din foarte multe alte fişiere de formate diferite, care erau plasate în sistemul de operare (CrySyS, 2012). Acest lucru l-a făcut foarte greu de depistat, izolat şi analizat (Kaspersky Lab ZAO [RU], 2012b) – intenţie care a stat, în mod cert, la baza procesului de dezvoltare a lui Flame şi care, de altfel, este intenţia oricărui dezvoltator de astfel de aplicaţii.

Din perspectiva celor care au analizat Flame, mai sunt câteva caracteristici importante care trebuie menţionate aici, şi anume:

- Flame are modificată data de creare a fişierelor proprii – o tehnică foarte întâlnită în cazul aplicaţiilor cu caracter maliţios, care face foarte dificil de investigat momentul în care acestea au fost dezvoltate. În particular, pentru aplicaţia în discuţie, fişierele sunt datate ca fiind din 1992, 1994, 1995 şi aşa mai departe – este cât se poate de clar că aceste date sunt false (Kaspersky Lab ZAO [RU], 2012a). Această tehnică nu este nouă, fiind întâlnită prima dată la sfârşitul anilor '90, dar este în continuare foarte eficientă;
- Aplicaţia analizată în acest capitol foloseşte ceea ce, în limbaj informatic, se numeşte „process injection" (CrySyS, 2012). Cu alte cuvinte, Flame avea

capacitatea de a insera secvenţe de lucru proprii în fişiere şi procese deja existente şi în execuţie în sistemul de operare infestat – se înţelege că acestea erau curate până atunci. Acest lucru duce la influenţarea sau alterarea comportamentului unor structuri software deja existente şi acceptate de sistemele de protecţie ca fiind în afara riscului. Un exemplu, în acest sens, ar fi modificarea regiştrilor Windows-ului, cu scopul de a-şi păstra persistenţa în sistemul compromis (FireEye, 2012);
- Posibilitatea de a depista aplicaţii care pot prezenta risc pentru Flame, precum antiviruşi sau *firewall*-uri. În cazul în care acestea erau instalate, *malware*-ul se autoînscria în lista de excepţii la scanare, rămânând astfel nedepistat. Ironic, poate, dezvoltatorii au denumit acest modul „Security" (Kaspersky Lab ZAO [RU], 2012b).

***

În acest capitol, am vorbit de o armă cibernetică deosebit de discretă şi eficientă. Văzând toate caracteristicile ei şi felul în care acestea se regăsesc şi în celelalte trei *malware*-uri de care am vorbit (Stuxnet, DuQu şi Gauss), ne face să ne gândim automat la faptul că ele au fost finanţate de una sau mai multe organizaţii guvernamentale şi dezvoltate, în paralel, de echipe independente.

Flame, cu atât de multe module şi cu o mărime de 20 MB, în mod cert nu a fost dezvoltat să infesteze un număr mare de computere, ci mai degrabă pe cele care sunt cu adevărat importante pentru dezvoltatori şi din care aceştia au încercat să culeagă cât mai multe informaţii posibil, într-o manieră cât mai discretă. Dacă este să ducem argumentul mai departe, ne putem uşor da seama că, în cazul în care aplicaţie discutată în acest capitol s-ar fi răspândit mai agresiv (aşa cum poate suntem obişnuiţi cu viruşii sau orice alt tip de *malware* existent azi), ar fi crescut riscurile ca ea să fie depistată. Ţintind, însă, doar anumite sisteme şi infestându-le doar pe acelea, riscul statistic se diminuează şi el corespunzător. Dacă vreţi o comparaţie cu o situaţie de război, nu se mai trage foc de acoperire cu mitraliera de asalt, timp în care trupele proprii s-ar porni să cucerească linia întâi a inamicului, ci sunt folosiţi lunetişti, care doboară, rând pe rând, soldaţii plasaţi în prima linie a frontului.

Kit – set de instrumente şi unelte *software* necesare pentru îndeplinirea unei anumite operaţii.

Plug-in – o secvenţă de program sau o aplicaţie de mici dimensiuni care se poate insera într-una mamă pentru a îndeplini funcţii specifice.

Server C&C (Command-and-Control Servers) – un computer într-o reţea *botnet* (fantomă) care transmite comenzi şi primeşte rapoarte de la sistemele asupra cărora deţine controlul.

SSH (Secure Shell) – un protocol de criptare folosit pentru a facilita conectarea, de la distanţă, a utilizatorilor în vederea executării locale, în condiţii de siguranţă, a unor sarcini specifice. De regulă, acest protocol se foloseşte când se operează prin mijlocirea unei reţele mai puţin sigure, precum Internetul.

SSL (Secure Sockets Layer) – un protocol de criptare utilizat pentru transportul informaţiilor în condiţii de siguranţă într-o reţea de calculatoare.

Standalone – o aplicaţie care nu foloseşte module sau fişiere proprii sistemului de operare pe care e instalată şi care este programată să pornească odată cu acesta tocmai pentru a-şi asigura independenţa.

VPN (Virtual Private Network) – o extensie a unei reţele private către utilizatori specifici, prin mijlocirea tehnologiilor Internet. Ea permite utilizatorilor să folosească resursele acesteia, bucurându-se de toată securitatea necesară atât în mediul de lucru, cât şi în cadrul componentelor de transmisie a datelor.

WebDAV – o extensie a protocolului HTTP (Hypertext Transfer Protocol) care permite utilizatorilor să execute, de la distanţă, operaţiuni autorizate de conţinut Web.

Regiştrii (sistemului de operare Windows) – o bază de date organizată după funcţionalitatea setărilor fine (*low-level settings*) ale Windows-ului.

Bibliografie

BBC (2012), *Flame: Massive cyber-attack discovered, researchers say*, http://www.bbc.com/news/technology-18238326

Clark Jeff (2012), *Flame malware: big story or old news?* în The DataCenterJournal, http://www.datacenterjournal.com/flame-malware-big-story-or-old-news/

CNET (2012), *Flame malware network based on shadowy domains, fake names*, http://www.cnet.com/news/flame-malware-network-based-on-shadowy-domains-fake-names/

ComputerWorld (2011), *FAQ: What's the big deal about Duqu?* http://www.computerworld.com/article/2498889/security0/faq--what-s-the-big-deal-about-duqu-.html

ComputerWorld (2012), *Gauss malware: Nation-state cyber-espionage banking Trojan related to Flame, Stuxnet*, http://www.computerworld.com/article/2597456/security0/gauss-malware--nation-state-cyber-espionage-banking-trojan-related-to-flame--stuxnet.html

CrySyS Lab (2012), *sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks*, http://www.crysys.hu/skywiper/skywiper.pdf

Dalziel Henry (2013), *The four amigos: Stuxnet, Flame, Gauss and DuQu*, https://www.concise-courses.com/security/stuxnet-flame-gauss-duqu/

FireEye (2012), *Flamer/sKyWIper malware: Analysis*, https://www.fireeye.com/blog/threat-research/2012/05/flamerskywiper-analysis.html

Gostev Alexander (2012), *The Flame: Questions and Answers*, în Kaspersky Lab SecureList, https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/

Hanlon James (2012), *Security Think Tank: Flame a good reason to keep up with emerging threat analysis,* în ComputerWeekly.com,

http://www.computerweekly.com/opinion/Security-Think-Tank-Flame-a-good-reason-to-keep-up-with-emerging-threat-analysis
IEEE Spectrum (20121), *Sons of Stuxnet* – Dialog între Steven Cherry, reprezentant al IEEE Spectrum și Larry Constantine, profesor de matematică al Universității Madeira din Portugalia. Online la: http://spectrum.ieee.org/podcast/telecom/security/sons-of-stuxnet
InformationWeek (2012), *Meet Flame Espionage Malware Cousin: MiniFlame*, http://www.darkreading.com/vulnerabilities-and-threats/meet-flame-espionage-malware-cousin-miniflame/d/d-id/1106871
InfosecInstitute (2012), *Flame: The Never Ending Story*, http://resources.infosecinstitute.com/flame-the-never-ending-story/
Kamluk Vitaly (2012), în BBC (2012), *Flame: Massive cyber-attack discovered, researchers say*, http://www.bbc.com/news/technology-18238326
Kaspersky (2015a), *DuQu 2.0: Frequently Asked Questions*, http://media.kaspersky.com/en/Duqu-2-0-Frequently-Asked-Questions.pdf
Kaspersky (2015b), *The DuQu 2.0. Technical Details*, https://cdn.securelist.com/files/2015/06/The_Mystery_of_Duqu_2_0_a_sophisticated_cyberespionage_actor_returns.pdf
Kaspersky Eugene (2012), declarație făcută pe profilul personal de Twitter (e_kaspersky), pe 28 mai 2012, https://twitter.com/e_kaspersky/status/207114162701213696
Kaspersky Lab ZAO [RU] (2012a), *The Flame: Questions and Answers*, https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/
Kaspersky Lab ZAO [RU] (2012b), *Flame: Bunny, Frog, Munch and BeetleJuice*…, https://securelist.com/blog/incidents/32855/flame-bunny-frog-munch-and-beetlejuice-2/
Marshall Brian & Wesley Fenlon (2013), *How Computer Viruses Work*, în HowStuffWorks.com, http://computer.howstuffworks.com/virus.htm
Microsoft (2012), *Microsoft certification authority signing certificates added to the Untrusted Certificate Store*, Security Research & Defense Blog, http://blogs.technet.com/b/srd/archive/2012/06/03/microsoft-certification-authority-signing-certificates-added-to-the-untrusted-certificate-store.aspx
PCWorld, *Researchers Identify Stuxnet-like Cyberespionage Malware Called 'Flame'*, http://www.pcworld.com/article/256370/researchers_identify_stuxnetlike_cyberespionage_malware_called_flame.html
SANS Institute, InfoSec Reading Room (2003), *Bots & Botnet: An Overview*, http://www.sans.org/reading-room/whitepapers/malicious/bots-botnet-overview-1299
Sotirov Alex (2012), *Analyzing the MD5 collision in Flame*, în Trail of Bits Research, https://www.trailofbits.com/resources/flame-md5.pdf
Symantec (2011a), *W32.Duqu. The precursor to the next Stuxnet*, *http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_duqu_the_precursor_to_the_next_stuxnet.pdf*

Symantec (2011b), *W32.Stuxnet Dossier*, în Symantec Security Response, https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf
Symantec (2012), *W32.Flamer!gen*, https://www.symantec.com/security_response/writeup.jsp?docid=2012-053007-0702-99&tabid=2
Taubenfeld Gadi (2004), *The Black-White Bakery Algorithm*, în Proc. Distributed Computing, 18th international conference, DISC 2004, Vol. 18, 56-70, http://www.cs.tau.ac.il/~afek/gadi.pdf
Udi Adam (2012), *How did the 'Flame' family of malware spread?* https://www.cs.bu.edu/~goldbe/teaching/HW55813/flame.pdf
Washington Post (2012), *U.S., Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say*, https://www.washingtonpost.com/world/national-security/us-israel-developed-computer-virus-to-slow-iranian-nuclear-efforts-officials-say/2012/06/19/gJQA6xBPoV_story.html
Zetter Kim (2012), *Meet 'Flame, The Massive Spy Malware Infiltrating Iranian Computers,* în Wierd.com, http://www.wired.com/2012/05/flame/