Analýza škodlivého kódu a objavovanie zraniteľností

(*Tisícdňový chrobák)*

Mgr. Ladislav Bačo

CSIRT.SK
Computer Security Incident Response Team Slovakia
DataCentrum, Cintorínska 5, 814 88 Bratislava
email:ladislav.baco@csirt.sk

Abstrakt

Vulnerability CVE-2016-4116 in Adobe Flash Player has been discovered during our analysis of an older public-available malware sample. This article describes our analysis of this vulnerability, responsible disclosure, additional research and identification of other similar flaws. Exploitation of this vulnerabilities in signed programs for bypassing the User Access Control and DLL hijacking.

Úvod

V minulosti sa nové vzorky škodlivého kódu objavovali iba zriedkavo, každá nová vzorka mohla byť podrobne preskúmaná. V dnešnej dobe to už možné nie je, existuje obrovské množstvo jedinečných vzoriek škodlivého kódu a každý deň sa objavujú stotisíce nových[1]. Mnohé antivírové produkty používajú rôzne heuristiky na odhaľovanie potenciálne škodlivých programov, nezachytávajú však ani len všetky známe vzorky, na základe rôznych testov stále nezachytávajú aj jednotky až desiatky percent známych vzoriek [2]. Pri neznámych vzorkách je miera detekcie ešte výrazne nižšia, a v mnohých prípadoch je potrebný počiatočný ľudský podnet, aby vzorku začali zachytávať.

Metódy analýzy škodlivého kódu

Existuje niekoľko spôsobov analýzy škodlivého kódu a tiež niekoľko kritérii, na základe ktorých vieme jednotlivé metódy rozdeliť. Na základe toho, kto vykonáva analýzu, môžeme hovoriť o automatizovanej (strojovej) alebo manuálnej (ručnej, ľudskej) analýze škodlivého kódu. Na základe toho, čo sa so vzorkou deje počas analýzy, môžeme hovoriť o statickej (kód/program zo vzorky sa nevykonáva) alebo dynamickej/behaviorálnej analýze (kód zo vzorky sa vykonáva). Na základe pracovných postupov, časovej a vedomostnej náročnosti môžeme taktiež použiť nasledovnú kategorizáciu:

- základná statická analýza,
- behaviorálna analýza,
- dynamická analýza,
- pokročilá statická analýza – reverzné inžinierstvo.

Základná statická analýza by mala zodpovedať otázku, či môže byť vzorka nebezpečná. Zisťuje informácie o vzorke bez jej spustenia, napr. na základe metadát, sekvencií znakov v nej (textové reťazce, špecifické postupnosti inštrukcií), pribalených obrázkov a súborov v nej (resources). V prípade PE (.exe, .dll) súborov je tiež možné využiť názvy sekcií programu a zoznam importovaných a exportovaných funkcií. V neposlednom rade je vhodné pozrieť sa aj na entropiu vzorky a skontrolovať jej hash, či to nie je hash známej vzorky detekovanej antivírusovými produktmi (napr. pomocou služby VirusTotal). Základnú statickú analýzu môže vykonávať človek a taktiež ju najčastejšie používajú rôzne antivírusové produkty.

Behaviorálna analýza sa snaží zistiť, čo robí vzorka za normálnych okolností. To znamená, čo robí vzorka, keď ju používateľ spustí a v rámci možností s ňou interaguje ako bežný používateľ. Pri behaviorálnej analýze sa vzorka spúšťa v bezpečnom kontrolovanom prostredí, najčastejšie vo virtuálnom počítači, v ktorom je simulované reálne prostredie. Záleží na konkrétnej situácii a citlivosti vzorky, niekedy je potrebné simulovať aj Internetovú komunikáciu. Počas behu vzorky sa monitorujú prejavy vzorky a jej činnosť, napr. zaznamenáva sa sieťová komunikácia, prístup k súborom a registrom, vytváranie procesov a vlákien a podobne. Vhodné sú napríklad nástroje Wireshark a Process Monitor, ktorý budeme používať aj my pri)dalšom skúmaní. Behaviorálnu analýzu môže vykonávať človek, taktiež sa však čoraz častejšie využíva aj automatizovane v niektorých bezpečnostných riešeniach typu sandbox[1].

Dynamická analýza by mala priniesť odpoveď na otázku, ako funguje vzorka. Pri tejto metóde je vzorka spustená v ladiacom nástroji (debuggeri), ktorý umožňuje vidieť vykonávané inštrukcie, obsah pamäte a registrov v procesore a pod. Pri podmienených skokoch v programe môže analytik vynútiť vykonávanie odlišných vetiev kódu. Toto je jedno z riešení výrazného problému behaviorálnej analýzy – skrytá funkcionalita vzorky, ktorá bude odhalená iba za splnenia určitých podmienok. Dynamická analýza vyžaduje hlbšie pochopenie samotného programu, preto už nie je veľmi vhodná pre automatizované/strojové vykonávanie.

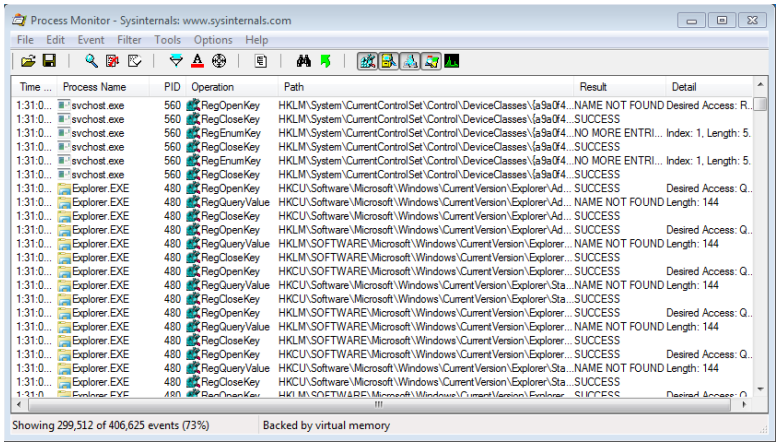
Posledným typom analýzy je **pokročilá statická analýza** v podobe **reverzného inžinierstva**. Táto metóda môže odhaliť, čo všetko dokáže vzorka robiť a akým spôsobom to robí. Uplatňuje sa častokrát v prípadoch, keď vzorka obsahuje viaceré ochrany pred analýzou a odmieta fungovať v debuggeri alebo v analytickom prostredí. Pri tejto metóde analytik pomocou disasemblera alebo dekompilátora preloží inštrukcie strojového kódu (alebo bajtkódu) vzorky do ľudsky čitateľnej podoby a snaží sa pochopiť význam a odhaliť podmienky, za ktorých sa vykonajú jednotlivé časti.

Pri hĺbkových analýzach analytik často kombinuje vyššie uvedené typy analýz (najmä dynamickú analýzu a reverzné inžinierstvo) s cieľom zefektívniť proces analýzy.

Priblížili sme si niektoré metódy analýzy škodlivého softvéru, z ktorých v nasledujúcom texte budeme využívať najmä behaviorálnu analýzu. Vzorky budeme sledovať pomocou programu Process Monitor z balíka systémových utilít SysInternals pre operačné systémy Windows [3].

Tento nástroj dokáže monitorovať jednotlivé procesy bežiace v systéme a ich prístup k súborovému systému a registrom, pokusy o sieťovú komunikáciu, vytváranie a narábanie s procesmi a vláknami a taktiež načítavanie súorov DLL (dynamicky linkované knižnice).

[1] Sandboxy vytvárajú izolované prostredie s limitovaným prístupom k iným procesom a zdrojom v systéme s cieľom obmedziť možnú škodu spôsobenú nedôveryhodným programom

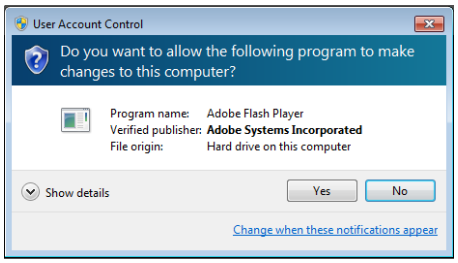


Obr. 1: Používateľské rozhranie programu Process Monitor

Identifikácia zraniteľnosti CVE-2016-4116

Inšpiráciou pre tento príspevok bola jedna staršia verejne dostupná vzorka škodlivého softvéru. Pri príprave tréningovej úlohy bol zámer použiť vzorku bankového trójskeho koňa Zeus, ktorá by už bola dostatočne známa, aby ju zachytávali takmer všetky antivírusové riešenia a aby boli verejne dostupné aj analýzy podobných vzoriek.

Na základe vyššie uvedených požiadaviek sme siahli po staršej vzorke z verejne dostupného repozitára malvéru slúžiaceho na vzdelávacie účely. Vzorka kategorizovaná ako variant bankového trojana Zeus tváriaca sa ako faktúra s hodnotou MD5 hashovacej funkcie `ea039a854d20d7734c5add48f1a51c34` pochádzala z novembra 2013, mala by byť teda dostatočne známa[2]. Po stiahnutí sme si chceli overiť jej správanie pomocou behaviorálnej analýzy. Chvíľku po spustení vzorky sa však na testovací počítač chcel nainštalovať program Adobe Flash Player, ako nás upozornilo dialógové okno kontroly UAC (User Account Control, Kontrola používateľských kont).



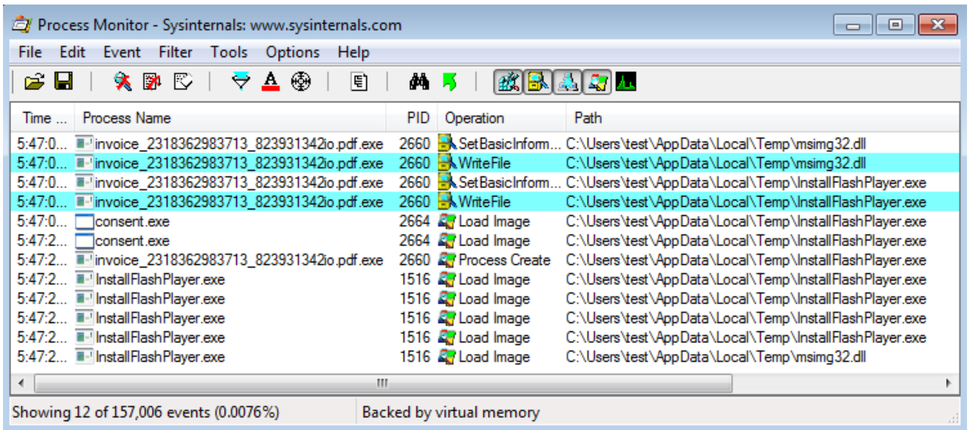
Obr. 2: Pokus o inštalovanie Adobe Flash Player s administrátorskými oprávneniami

UAC, resp. kontrola používateľkých kont je bezpečnostný mechanizmus predstaveý s operačným systémom Microsoft Windows Vista. Používateľom s administrátorskými oprávneniami umožňuje spúšťať aplikácie s oprávneniami štandardného používateľa, pokiaľ administrátor neschváli zvýšenie oprávnení prostredníctvom dialógového okna. V prípade, že takúto aplikáciu spúšťa štandardný používateľ, systém vyžaduje zadanie mena a hesla administrátora [7].

Trójske kone rodiny Zeus majú viaceré schopnosti, medzi ktoré patrí napr. odchytávanie prihlasovacích mien a hesiel na stránkach internetového bankovníctva [4]. No inštalovanie

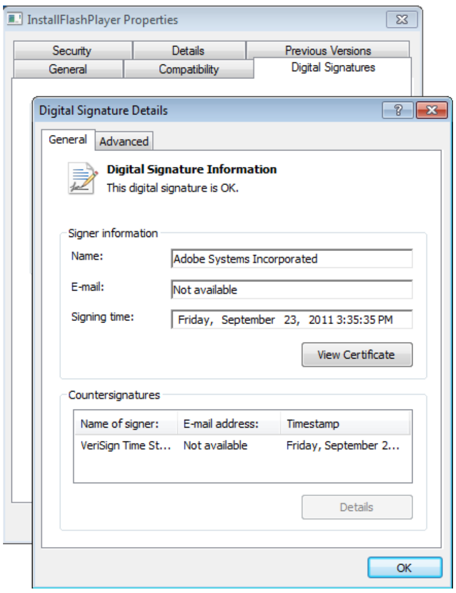
[2]V čase písania článku ju detegovalo 49 z 57 AV riešení zapojených do služby VirusTotal; prvýkrát sa vyskytla vo VirusTotale koncom novembra 2013

prehrávača Adobe Flash Player sa v prezretých analýzach nespomínalo nič [4, 5, 6]. Preto sme sa rozhodli uvedené správanie bližšie preskúmať. Z vyššie uvedených metód bola zvolená behaviorálna analýza, ktorá je dostatočne rýchla a efektívna na získanie informácií o správaní vzorky. Process Monitor zachytil množstvo udalostí, po lokalizovaní udalostí týkajúcich sa inštalátora Adobe Flash Player a vyfiltrovaní ostatných, je výstup programu Process Monitor zachytený na nasledujúcom obrázku.



Obr. 3: Udalosti objasňujúce pôvod inštalátora

Z Process Monitora je vidno, že skúmaná vzorka (vytvorila a následne) zapísala skryté súbory `msimg32.dll` a `InstallFlashPlayer.exe` do adresára `%TEMP%` a následne došlo k pokusu spustiť tento inštalátor, ktorý sa prejavil jeho načítaním do pamäte a zobrazením dialógového okna UAC (proces `consent.exe`) [8]. Program `InstallFlashPlayer.exe` je legitímny inštalátor staršej verzie Adobe Flash Player a bol podpísaný spoločnosťou Adobe 23. septembra 2011.

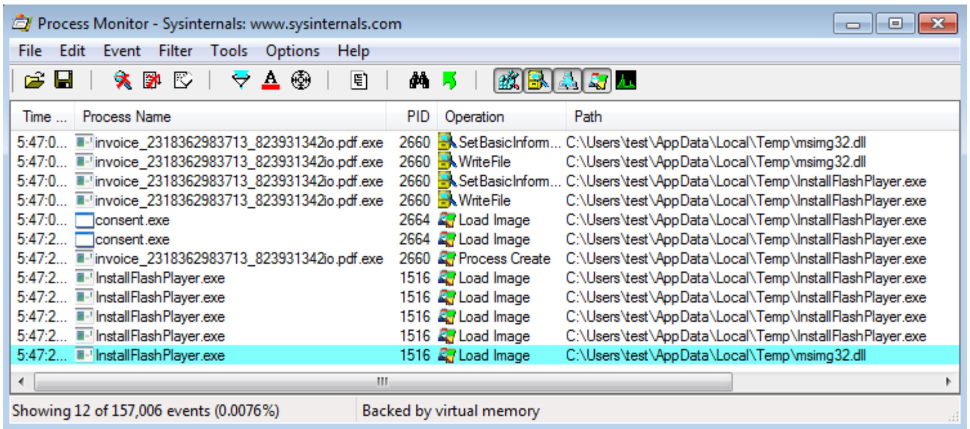


Obr. 4: Legitímny inštalátor podpísaný spoločnosťou Adobe

Po schválení administrátorských oprávnení je legitímny inštalátor spustený ako nový proces a nasleduje štandardné načítavanie dynamicky linkovaných knižníc (.dll súbory). V operačných systémoch Microsoft Windows sa takéto DLL knižnice štandardne vyhľadávajú

najprv v adresári, z ktorého je aplikácia spustená, potom v systémových adresároch[3], v adresári Windows, v aktuálnom adresári[4] a v adresároch špecifikovaných premennou prostredia `%PATH%` [9].

Spustený inštalátor vyžaduje knižnicu `msimg32.dll`. Vďaka štandardnému poradiu načítavania týchto knižníc je načítaná knižnica z rovnakého adresára, v ktorom je umiestnený inštalátor, teda z adresára `%TEMP%`. Je to ten súbor, ktorý pred spustením inštalátora vytvorila analyzovaná vzorka škodlivého softvéru. Po načítaní tejto knižnice sa teda bude vykonávať kód pochádzajúci zo škodlivej vzorky, avšak už s oprávneniami admininistrátora a maskovaný ako legitímny program spoločnosti Adobe.



Obr. 5: Načítanie škodlivej DLL knižnice

Inštalátor prehrávača Adobe Flash Player použitý v skúmanej vzorke teda obsahuje zraniteľnosť typu Dll hijacking umožňujúcu injektovať škodlivý DLL súbor do spusteného procesu. Následným testovaním sme zistili, že rovnakú zraniteľnosť obsahuje aj v tom čase[5] aktuálny inštalátor Adobe Flash Player verzie 21.0.0.213. To znamená, zraniteľnosť zneužívaná vo vzorke starej 876 dní je stále neopravená. Navyše, táto zraniteľnosť bola prítomná už v inštalátore z roku 2011, teda bola v tom čase stará minimálne 1670 dní. Takže sme vlastne identifikovali starú 0-day zraniteľnosť (chybu, „bug“), ktorá ešte nebola opravená a je aktívne zneužívaná útočníkmi. Táto skutočnosť je vlastne inšpirácia pre mierne poetický podtitul tohto článku: **Tisícdňový chrobák**.

Nahlásenie zraniteľnosti

V rámci procesu zodpovedného odhalenia zraniteľnosti bola vyššie identifikovaná zraniteľnosť nahlásená v rámci programu Zero Day Initiative (ďalej len ZDI) spoločnosti TippingPoint. V rámci tohto programu sú po akceptovaní nahlásených zraniteľností ich nálezcovia odmenení a zároveň sú zraniteľnosti nahlásené výrobcovi dotknutého produktu a spolupracuje sa s ním na opravení a následnom zverejnení informácií o zraniteľnosti [12].

Nahlásiť zraniteľnosť v rámci programu ZDI sme zvolili z viacerých dôvodov; boli sme zvedaví, ako funguje proces nahlasovania a skúmania zraniteľnosti, ako ohodnotia uvedenú zraniteľnosť a taktiež je veľkou výhodou odbremenenie nálezcu od komunikácie s výrobcom produktu. Do dvoch týždňov od nahlásenia zraniteľnosti sme obdržali vyjadrenie, podľa

[3] štandardne `C:\Windows\System32` alebo jeho ekvivalent

[4] pokiaľ je vypnutý režim `SafeDllSearchMode`, aktuálny adresár sa prehľadáva ako druhý v poradí. Tento režim je však štandardne zapnutý

[5] apríl 2016

ktorého tento typ zraniteľnosti nespadá do programu ZDI, ale zároveň sme dostali odporúčanie kontaktovať priamo spoločnosť Adobe, ktorá na základe ich skúsenosti komunikuje veľmi dobre. Takéto odporúčanie bolo pre nás príjemným prekvapením, a tak sme následne kontaktovali priamo spoločnosť Adobe.

Následne prebehla komunikácia s PSIRT tímom spoločnosti Adobe (Product Security Incident Response Team), počas ktorej PSIRT potvrdil nahlásenú zraniteľnosť. V priebehu ďalšej komunikácie bol zraniteľnosti priradený identifikátor CVE-2016-4116 a bolo potvrdené jej orpavenie pri nadchádzajúcej aktualizácii.

Paralelne s týmto procesom bolo otestovaných aj viacero ďalších softvérových produktov na podobné zraniteľnosti (viď nižšie), pričom sme zaznamenali podobnú zraniteľnosť aj pri produktoch Adobe Reader, Adobe Acrobat Reader, Adobe Acrobat Reader DC a Adobe Acrobat DC. Uvedené zraniteľnosti sme taktiež nahlásili spoločnosti Adobe. V rámci tohto procesu boli opravené spolu s ďalšou podobnou zraniteľnosťou a sú evidované pod označením CVE-2016-1090. Pre úplnosť uvádzame časovú os komunikácie so spoločnosťou Adobe:

- 28.04.2016 10:07 Zraniteľnosť vo Flash Player bola nahlásená spoločnosti Adobe,
- 28.04.2016 11:31 Potvrdenie prijatia hlásenia, priradenie označenia PSIRT-5186,
- 02.05.2016 11:18 Zraniteľnosť v Reader, Acrobat Reader, Acrobat Reader DC a Acrobat DC bola nahlásená spoločnosti Adobe,
- 02.05.2016 20:22 Potvrdenie prijatia hlásenia, priradenie označenia PSIRT-5199,
- 05.05.2016 Zverejnenie opravy Adobe Reader a spol.,
- 10.05.2016 14:30 Informovanie sa o stave riešenia,
- 10.05.2016 23:36 Priradenie CVE-2016-4116 prípadu PSIRT-5186, potvrdenie plánovanej aktualizácie,
- 12.05.2016 Zverejnené opravy Adobe Flash Playera[13],
- 13.05.2016 Zverejnené varovanie o zraniteľnosti na našej stránke[14],
- 19.05.2016 Aktualizovanie Security Bulletinu pre Adobe Reader a Acrobat na základe ďalšej komunikácie[15].

Poznámka. Neskoršou analýzou pôvodnej vzorky bolo zistené, že v skutočnosti nepredstavuje vzorku bankového trójskeho koňa Zeus, ale ide o trójskeho koňa ZeroAccess. Pre túto rodinu škodlivého kódu bolo zaznamenaná nami pozorované správanie, ale pravdepodobne nebolo oznámené spoločnosti Adobe vzhľadom na to, že až do nášho nahlásenia bola táto zraniteľnosť stále prítomná aj v novších produktoch [10, 11].

Testovanie a overovanie zraniteľností typu DLL hijacking

Vzhľadom na to, že podobné zraniteľnosti sú ľahko zneužiteľné a môžu byť prítomné aj v produktoch iných spoločností, rozhodli sme sa otestovať aj inštalátory iných softvérových produktov. Pre tento účel sme si vytvorili nástroj na automatizované testovanie, ktorý simuloval naše manuálne identifikovanie a testovanie zraniteľnosti CVE-2016-4116.

V prvej fáze spustil na krátku dobu testovanú vzorku softvéru, pričom na pozadí bežal Process Monitor. Po chvíli bola testovaná vzorka násilne ukončená a bol skontrolovaný záznam aktivity na prítomnosť pokusov o načítanie knižnice z adresára, v ktorom sa nachádzala

testovaná vzorka[6]. Následne v druhej fáze postupne pre všetky takto identifikované názvy DLL knižníc bola do adresára so vzorkou pridaná testovacia DLL knižnica s požadovaným menom a testovaná vzorka bola opätovne spustená. V prípade, že naša testovacia knižnica bola skutočne načítaná, spustila nový proces – kalkulačku. Toto spustenie bolo opäť detekované pomocou zoznamu bežiacich procesov.

Po ukončení testovania náš nástroj pre každú vzorku vygeneroval správu o tom, ktoré knižnice sú potenciálne náchylné na DLL hijacking a pre ktoré bol tento útok naozaj úspešný. Na základe tohto testovania boli objavené podobné zraniteľnosti aj v rodine produktov Adobe Reader a taktiež v inštalátore jedného webového prehliadača. V druhom prípade však ešte v priebehu testovania pred nahlásením zraniteľnosti bola vydaná nová verzia prehliadača, ktorá uvedenú zraniteľnosť už neobsahovala.

Listing 1: Zdrojový kód testovacej DLL knižnice

```
#include <windows.h>

BOOL WINAPI DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
{
        system("C:\\Windows\\System32\\calc.exe");
        return TRUE;
}
```

Ďalší výskum a (staro)nové zraniteľnosti

Vo februári 2017 sme opäť narazili na podobnú zraniteľnosť, avšak tentoraz už nemala takúto priamočiaru a jednoduchú príčinu. Pomocou tejto zraniteľnosti je možné zneužiť viacero programov od rôznych výrobcov softvéru na útoky s cieľom presvedčiť používateľa, aby programu s legitímnym podpisom výrobcu pridelil zvýšené oprávnenia a následne sa cez tento program spustí škodlivý kód. Zraniteľnosť bola nahlásená výrobcovi softvéru, avšak podľa oficiálneho vyjadrenia sa týmto prípadom nebudú zaoberať, pretože z ich pohľadu to nie je bezpečnostná zraniteľnosť a takéto správanie programov je „by design“. Výrobcovi bolo oznámené, že v tom prípade bude uvedená zraniteľnosť aj ukážka jej zneužitia prezentovaná na konferencii. Účastníci konferencie SASIB tak budú mať možnosť sami posúdiť, či prezentovaná ukážka skutočne nepredstavuje bezpečnostné riziko.

Odporúčania na záver

Mnoho používateľov sťahuje a ukladá nové programy (a ich inštalačky) do štandardného adresára pre stiahnuté súbory. V tomto adresári je častokrát veľa iných starších súborov. Môže sa stať, že niektorý zo starších súborov bude práve DLL knižnica, ktorá bude neskôr načítaná do inštalátora, ktorý bude spustený s administrátorskými právami. Navyše, útočníci môžu zneužiť zraniteľnosť/útok typu drive-by download a stiahnuť do tohto adresára pre stiahnuté súbory škodlivú DLL knižnicu, a následne pomocou prvkov sociálneho inžinierstva na webstránke prinútiť používateľa stiahnuť a nainštalovať legitímny program, v ktorom zneužijú zraniteľnosti v načítavaní DLL súborov[7][16].

Druhou možnosťou na podobné zneužitie zraniteľností v načítavaní DLL knižníc môže byť umiestnenie škodlivého DLL súboru do adresára pre dočasné súbory (%TEMP%), ktorý

[6]boli použité filtre na načítanie .dll súborov z aktuálneho adresára, pričom výsledok operácie bol NAME NOT FOUND

[7]Typickým príkladom, ktorý môže byť ľahko využitý v sociálnom inžinierstve, je práve spomínaný Adobe Flash Player, prípadne Java pre podporu rôznych appletov na webstránkach.

je častokrát využívaný pri inštalovaní nových programov na rozbalenie inštalátora a jeho následné spustenie.

Odporúčaním pre používateľov môže byť jednak pravidelné mazanie dočasných súborov a premazávanie adresára pre stiahnuté súbory. Toto opatrenie však nemusí byť dostatočné najmä v prípade spomínaných drive-by download útokoch. Univerzálnejšie odporúčanie pre používateľov by mohlo znieť nasledovne:
akonáhle si po spustení nejakého stiahnutého programu hocijaký program vypýta administrátorské oprávnenia, zrušíme ho a skontrolujeme, či je spúšťaný stiahnutý program (príslušný .exe súbor) naozaj podpísaný certifikátom[8]. Takéto overenie by v prípade prvotnej vzorky škodlivého softvéru zabránilo škodlivému kódu v získaní administrátorských oprávnení.

Zraniteľnosti typu DLL hijacking sú prítomné v mnohých programoch. Vo niektorých prípadoch však ich zneužitie nepredstavuje veľké riziko, keďže útočník, ktorý môže umiestniť škodlivú DLL knižnicu do adresára so zraniteľným programom, môže už aj priamo prepísať tento program[9]. V týchto prípadoch však môže byť zraniteľnosť využívaná na maskovanie škodlivých útočníkových súborov v podobe nenápadných zdanlivo legitímnych DLL knižníc. V iných prípadoch však môže byť táto zraniteľnosť využitá na oklamanie používateľa a získanie administrátorských oprávnení, čo zneužívala aj skúmaná vzorka malvéru a ukážky prezentované počas konferencie. Taktiež boli zaznamenané zraniteľnosti, v ktorých sa mohla škodlivá DLL knižnica načítať z ľubovoľného adresára[10]. Takýto scenár je zneužiteľný napr. pri otváraní súborov (dokumentov) zo zdieľaných diskov, kam môže umiestniť umiestniť škodlivé DLL hociktorý používateľ s platným prístupom.

Pokiaľ existuje možnosť zneužitia DLL hijacking zraniteľnosti, mali by dotknuté programy prejsť testovaním a kontrolou a zistené zraniteľnosti by mali byť opravované ešte pred publikovaním týchto programov. Ako sme ukázali, testovanie DLL hijacking zraniteľnosti je jednoduché a dá sa dobre automatizovať.

Literatúra

[1] https://usa.kaspersky.com/about-us/press-center/press-releases/2016/Kaspersky˙Lab˙Number˙of˙the˙Year˙2016˙323000˙Pieces˙of˙Malware˙Detected˙Daily, navštívené 28.2.2017

[2] https://www.av-test.org/typo3temp/˙processed˙/csm˙0915˙Linux˙Tabelle˙scanwerte˙neu2˙en˙43e6000a5c.png navštívené 28.2.2017

[3] https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx, navštívené 13.2.2017

[4] http://www.talosintelligence.com/zeus-trojan-analysis, navštívené 13.2.2017

[5] http://www.ioactive.com/pdfs/ZeusSpyEyeBankingTrojanAnalysis.pdf, navštívené 13.2.2017

[6] https://www.symantec.com/content/en/us/enterprise/media/security˙response/whitepapers/zeus˙king˙of˙bots.pdf, navštívené 13.2.2017

[7] https://msdn.microsoft.com/en-us/library/windows/desktop/dn742497(v=vs.85).aspx, navštívené 14.2.2017

[8] https://technet.microsoft.com/en-us/library/dd835561(v=ws.10).aspx, navštívené 14.2.2017

[8] pomocou pravého kliku na .exe súbor a kontrolou podpisu podľa Obrázka 4

[9] sú pre programy uložené v adresári `Program Files` sú na takýto útok vyžadované administrátorské oprávnenia

[10] typicky v situáciách, kedy bol zraniteľný program asociovaný s určitým typom súborov a pri otvorení týchto súborov hľadal DLL knižnice v adresári s otváraným súborom

[9] https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx, navštívené 14.2.2017

[10] https://www.symantec.com/content/en/us/enterprise/media/security˙response/whitepapers/trojan˙zeroaccess˙infection˙analysis.pdf, navštívené 13.2.2017

[11] https://www.symantec.com/content/en/us/enterprise/media/security˙response/whitepapers/zeroaccess˙indepth.pdf, navštívené 13.2.2017

[12] http://www.zerodayinitiative.com, navštívené 20.2.2017

[13] https://helpx.adobe.com/security/products/flash-player/apsb16-15.html, navštívené 2.3.2017

[14] https://www.csirt.gov.sk/oznamenia-a-varovania-803.html?id=147, navštívené 2.3.2017

[15] https://helpx.adobe.com/security/products/acrobat/apsb16-14.html, navštívené 2.3.2017

[16] https://en.wikipedia.org/wiki/Drive-by˙download, navštívené 28.2.2017