Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Implementácia metodiky pre zabezpečenie IT systémov verejnej správy

Bakalárska práca

2018
Matej Štubniak

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Implementácia metodiky pre zabezpečenie IT systémov verejnej správy

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	RNDr. Jaroslav Janáček, PhD.
Konzultant:	Mgr. Lukáš Hlavička

Bratislava, 2018
Matej Štubniak

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Matej Štubniak
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Implementácia metodiky pre zabezpečenie IT systémov verejnej správy
Implementation of Methodology for Securing Public Administration IT Systems

Anotácia: Práca sa venuje implementácii metodiky CSIRT.SK pre zabezpečenie IT systémov verejnej správy so zameraním na oblasť pracovných staníc.

Cieľ: - analyzovať bezpečnostné požiadavky uvedené v metodike
- navrhnúť a implementovať softvérový nástroj na vykonanie potrebných nastavení systémov tak, aby spĺňali požiadavky metodiky pre vybranú oblasť

Vedúci: RNDr. Jaroslav Janáček, PhD.
Konzultant: Mgr. Lukáš Hlavička
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Dátum zadania: 31.10.2017

Dátum schválenia: 06.11.2017 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Týmto spôsobom by som sa chcel poďakovať môjmu školiteľovi RNDr. Jaroslavovi Janáčkovi, PhD. za zorganizovanie spolupráce a bezproblémové jednanie a môjmu konzultantovi Mgr. Lukášovi Hlavičkovi za odbornú pomoc, trpezlivosť a ochotu stretnúť sa aj napriek časovej tiesni.

Abstrakt

Táto práca sa zaoberá implementáciou metodiky pre zabezpečenie IT systémov verejnej správy, ktorú vydala bezpečnostná jednotka CSIRT.SK, pričom sa zameriava na oblasť pracovných staníc obmedzenú na operačné systémy Microsoft Windows. Jej cieľom nie je len vykonať implementáciu tak, aby spĺňala požiadavky uvedené v metodike a uľahčiť tým administrátorom zabezpečovanie IT systémov, ale aj dané požiadavky zanalyzovať a poukázať na bezpečnostné hrozby spojené s ich nesplnením. Výsledná implementácia má perspektívu byť reálne použiteľná, je však ponechaný priestor aj pre vylepšenie niektorých opatrení.

Kľúčové slová: implementácia, metodika, zabezpečenie, pracovné stanice, Windows

Abstract

This thesis deals with the implementation of methodology for securing public administration IT systems published by security unit CSIRT.SK, focusing on area of workstations limited to Microsoft Windows operating systems. Its goal is not only to make implementation, that fulfills requirements of the methodology and facilitates the securing of IT systems, but also to analyze the requirements and to point out the security threats associated with failure to comply them. The resulting implementation has a perspestive to be applicable in real life, but space for improvement of some measures is also left.

Keywords: implementation, methodology, securing, workstations, Windows

Obsah

Úvod

Zabezpečenie IT systémov je v súčasnosti veľmi dôležitá oblasť. Často dnes počujeme o informatizácii alebo digitalizácii verejnej správy, kde sa nepochybne jedná aj o prácu s citlivými informáciami, preto v žiadnom prípade nemožno túto oblasť opomenúť. Stačí napríklad vziať do úvahy mesačné prehľady ohľadom závažných bezpečnostných udalostí a kritických zraniteľností IT systémov z roku 2018, ktoré publikuje bezpečnostná jednotka CSIRT.SK. Je tu jednoznačne nezanedbateľný počet stále nových a nových zraniteľností softvérov a pokusov o ich zneužitie. Je preto potrebné zavádzať opatrenia, ktorými minimalizujeme riziká vyplývajúce z týchto skutočností. Okrem toho sa ukazuje, že sú tu aj bezpečnostné hrozby, ktoré sú síce veľmi dobre známe, no napriek tomu sa objavujú incidenty, pri ktorých sa napĺňajú. Chceme zabraňovať aj týmto prípadom.

Pri zabezpečovaní IT systémov verejnej správy treba pamätať na to, že nie všetci pracovníci v tejto sfére sú dostatočne uvedomelí, pokiaľ ide o hrozby súvisiace s prácou v počítačových systémoch. Okrem školení v tejto oblasti je preto tiež dôležité vytvárať také opatrenia, ktoré aj napriek tomuto faktu bezpečnostným incidentom zabránia, ak to je možné.

Pre tento účel využijeme metodiku pre zabezpečenie IT systémov verejnej správy od už spomenutej jednotky CSIRT.SK, ktorá sa v tejto oblasti už niekoľko rokov úspešne zaoberá zvládaním bezpečnostných incidentov, odstraňovaním ich následkov, ale aj poskytovaním služieb vzdelávacieho charakteru.

Keď hovoríme o IT systémoch, máme tým na mysli okrem samotných pracovných staníc aj celú sieťovú infraštruktúru, v rámci ktorej bývajú stanice zasadené a jej jednotlivé prvky. Metodika sa dotýka všetkých týchto oblastí, v práci sa však budeme venovať len oblasti pracovných staníc. Bezpečnostné aspekty spojené s počítačovými sieťami ale samozrejme budeme mať na pamäti.

Cieľom práce je teda implementovať bezpečnostné požiadavky obsiahnuté v spomenutej metodike. Implementácia sa bude týkať iba počítačov s operačným systémom Microsoft Windows. Treba tiež poznamenať, že aj keď máme na zreteli IT systémy verejnej správy, výsledná implementácia bude vhodná tiež pre domácich používateľov, ktorí chcú mať kvalitne zabezpečené svoje zariadenia, môže byť však pre nich miestami nadmieru obmedzujúca.

Kapitola 1

Metodika

V tejto kapitole predstavíme samotnú metodiku, o ktorej sme hovorili v úvode, keďže z nej vychádza celá táto bakalárska práca. Podrobnejšie sa pozrieme na to, čím konkrétne sa zaoberá a v čom spočívajú jej ciele.

Celý názov dokumentu je Metodika pre systematické zabezpečenie organizácií verejnej správy v oblasti informačnej bezpečnosti (ďalej len metodika) [3], pričom pracujeme momentálne s najnovšou verziou 2.0. Autorom je CSIRT.SK (Computer Security Incident Response Team Slovakia).

1.1 Cieľ metodiky

Cieľom metodiky je, aby v nej uvedené požiadavky a opatrenia aplikované v IT systémoch viedli k takému zabezpečeniu, ktoré v čo najvyššej možnej miere sťažuje kompromitáciu daného IT systému a v prípade kompromitácie nejakej časti systému minimalizuje jej dôsledky a možnosti útočníka, pokiaľ ide o pokračovanie v kompromitácii (čiže ak sa napríklad útočníkovi podarí kompromitovať bežného používateľa, nemalo by mu to umožniť kompromitovať ďalších či dokonca celú sieť). Jedná sa teda o viacúrovňovú hĺbkovú ochranu, ktorá by mala byť odolná voči štandardným útokom v oblasti IT systémov.

Metodika je primárne určená na využitie v IT systémoch verejnej správy so zvýšenými požiadavkami na bezpečnosť. Požiadavky a opatrenia spomenuté v nej sú však všeobecne užitočné a môžu byť vhodné pre uplatnenie v rôznych oblastiach, kde sa využívajú IT systémy.

Cieľom metodiky však nie je objasňovať, prečo je dôležité zaviesť dané opatrenie a čomu vlastne zabraňuje. To je preto ďalší cieľ, ktorým sa chce táto práca zaoberať (okrem samotnej implementácie). Niektoré konkrétne opatrenia však už sú pomerne obšírne popísané v dokumente Quick Wins [4].

1.2 Obsah metodiky

Obsahovo sa metodika dotýka opatrení organizačných, technických a administratívnych, v tejto práci sa však budeme zaoberať len technickými. Konkrétne sa zameriame na časť Minimálne požiadavky na zabezpečenie pracovných staníc pristupujúcich k implementovanému riešeniu (kapitola 7 metodiky [3]). Metodika neobsahuje detailné pokyny pre implementáciu daných opatrení a práve preto je cieľom tejto práce vykonať ich implementáciu (niektoré konkrétne opatrenia z metodiky na základe konzultácie s predstaviteľmi CSIRT.SK nebudeme implementovať).

Čo sa týka obsahu metodiky, treba tiež poznamenať, že s tým, ako sa postupom času vyvíjajú útoky na IT systémy, je vyvíjaná aj metodika. Nie je teda statickým dokumentom, ale predpokladáme pribúdanie nových opatrení. Takéto zmeny by sa potom prirodzene mali prejaviť aj v jej implementácii.

1.3 Oblasť implementácie metodiky

Implementácia takýchto opatrení očividne vyžaduje do istej miery rozdielny prístup na jednotlivých operačných systémoch. Keďže na Slovensku sú vo sfére verejnej správy prevažne využívané operačné systémy Microsoft Windows, rozhodli sme sa, že práve pre ne budeme metodiku implementovať. Relevantné budú pre nás iba podporované verzie, to sú konkrétne Windows 7, 8.1 a najnovší 10.

Budeme sa však snažiť o to, aby základné princípy nášho riešenia implementácie metodiky mohli slúžiť ako vzor pre implementáciu aj na ostatných operačných systémoch.

Kapitola 2

Návrh riešenia

V tejto kapitole sa budeme zaoberať tým, ako pristúpiť k implementácii metodiky a ako bude vyzerať kostra riešenia.

2.1 Požiadavky na implementáciu

Implementáciu opatrení vyplývajúcich z metodiky chceme vykonať tak, aby pre administrátora daného IT systému bolo relatívne jednoduché a pohodlné ich sfunkčniť. Vhodným riešením pre túto požiadavku je napísanie skriptov, teda akýchsi programov, ktoré bude spúšťať a ktoré budú vykonávať potrebné nastavenia. Kvôli prehľadnosti bude pre každé opatrenie vytvorený samostatný skript (pre niektoré dokonca aj viacero skriptov - o tom ale ešte budeme hovoriť neskôr). Aby nemusel administrátor spúšťať postupne každý jeden skript, naprogramujeme jeden špeciálny, ktorý budeme volať Master skript. Ten bude v našom riešení hlavným programom, ktorý bude spúšťať skripty pre jednotlivé opatrenia - práve ten bude administrátor spúšťať pre zabezpečenie daného IT systému. Skripty však budú napísané tak, aby pre nastavenie nejakého konkrétneho opatrenia nebolo nutné zakaždým spúšťať Master skript, ale aby ich bolo možné spúšťať aj úplne samostatne.

Ďalšou dôležitou požiadavkou je, aby si administrátor mohol vybrať, či dané opatrenie chce využiť (môžu existovať okolnosti, v rámci ktorých bude niektoré opatrenie zabraňovať vykonávaniu funkcií, ktoré by nevyhnutne mali fungovať, možno aj napriek nejakému riziku či komplikácii). Túto požiadavku splníme jednoducho tak, že pred vykonaním každého skriptu pre jednotlivé opatrenia bude administrátor dopytovaný, či ho chce spustiť alebo nie.

2.2 Spôsob implementácie

Ako už bolo spomenuté, implementáciu metodiky budeme vykonávať iba pre operačné systémy Microsoft Windows. Keďže ideme písať skripty, ponúkajú sa nám dva príkazové riadky od spoločnosti Microsoft - Command Prompt a PowerShell, ktoré sú automaticky súčasťou všetkých distribúcií Windowsov, ktorými sa zaoberáme. Rozhodnutie je z našej strany pomerne jednoznačné pre druhý menovaný príkazový riadok a skriptovací jazyk, keďže PowerShell zvládne všetko, čo Command Prompt a ešte oveľa viac. Je tu však aj určitá komplikácia, keďže má viacero verzií. Pre verzie OS Windows, o ktorých sme hovorili v predchádzajúcej kapitole, bude potrebné písať skripty tak, aby správne fungovali od PowerShellu verzie 2.0 po verziu 5.1 (momentálne najnovšia). Práve z tohto dôvodu bude nutné skripty pre niektoré opatrenia písať viackrát, aby boli kompatibilné jednak s verziou PowerShellu (v starších verziách chýbajú niektoré užitočné príkazy) a tiež s verziou OS Windows (na rozdielnych verziách môžu byť potrebné nastavenia vykonávateľné rozdielnym spôsobom).

2.3 Štruktúra Master skriptu

Podstatnou úlohou pre voľbu štruktúry Master skriptu je rozhodnúť sa, akým spôsobom budeme rozlišovať spúšťanie skriptov pre rôzne verzie OS a PS (PowerShell) tam, kde je to potrebné. Mohlo by sa javiť, že tých kombinácií pre jednotlivé verzie môže byť dosť veľa. Pri dôkladnejšom preskúmaní sme si však uvedomili, že bude v zásade potrebné rozlišovať hlavne verzie OS, pretože už len na základe nich máme určité informácie ohľadom prítomnej verzie PS. Vieme predpokladať nasledujúce fakty:

- na Windows 7 bude PowerShell 2.0 alebo vyššie;
- na Windows 8.1 bude PowerShell 4.0 alebo vyššie;
- na Windows 10 bude PowerShell 5.0 alebo vyššie. [22]

Ako jeden z prvých spôsobov nás možno napadne ten, že budeme vytvárať vetvy typu `if-elseif-else`, v ktorých sa budeme na verzie priamo pýtať a na základe toho skripty spúšťať. Tomuto spôsobu by sme sa chceli vyhnúť, keďže je potom nutné písať niektoré skripty pre tie isté opatrenia do každej z vetiev, čo sa nám veľmi nepáči.

Ako rozumné riešenie sa nám javí byť to, že v samotnom názve skriptu zakomponujeme číslo verzie OS, ak ju bude potrebné rozlišovať (ukázalo sa, že vo väčšine prípadov to nebude potrebné). Výhodou tohto spôsobu je, že v zdrojovom kóde Master skriptu bude pre každé opatrenie uvedený len jeden skript.

Teraz sa pozrieme na to, ako presne to budeme programovať. Prvá vec, ktorú zistíme, bude verzia operačného systému, v ktorom je skript spustený. To sa dá zistiť

pomocou príkazu `[System.Environment]::OSVersion.Version`. Ten vracia objekt, ktorého atribúty Major a Minor verziu jednoznačne určujú. Tú si uložíme do premennej *$os* (v PS majú názvy premenných predponu *$*).

Názvy skriptov budeme vyberať takto: Ak pre dané opatrenie stačí napísať jeden skript, ktorý bude fungovať vo všetkých verziách OS, potom bude názov skriptu tvaru "*názov_opatrenia.ps1*" (.ps1 je prípona, ktorú majú skripty v PS). Ak je ale pre dané opatrenie nutné rozlišovať verzie OS, potom bude názov skriptu tvaru "*názov_opatrenia_$os.ps1*". Tu by sme ešte dodali, že reálne sme vždy rozlišovali maximálne dve verzie. V takých prípadoch sme potom pre dvojicu verzií OS, ktoré nebolo treba rozlišovať, naprogramovali jeden skript, ktorý je spúšťaný na oboch verziách (teda napríklad skript pre Windows 8.1 spúšťa skript pre Windows 7).

Master skript bude teda vyzerať tak, že po získaní údajov o verzii OS bude nasledovať "vymenovanie" skriptov pre jednotlivé opatrenia, teda ich postupné spúšťanie - ľubovoľný skript v Master skripte spustíme príkazom `& ".\názov_skriptu.ps1"`.

2.4 Štruktúra skriptov pre jednotlivé opatrenia

Po spustení každého skriptu sa v príkazovom riadku zobrazí dopyt na administrátora, či chce dané opatrenie využiť. Implementované to bude štandardným spôsobom, ktorý sa používa v prostredí PowerShellu aj v dopytoch od samotného výrobcu. Výsledok je podobný ako napríklad v Linuxovom príkazovom riadku - znakom '`Y`' zvolíme možnosť áno, znakom '`N`' možnosť nie (je to samozrejme case-insensitive, teda nie sú nerozlišované veľké a malé písmená) a okrem toho máme na výber aj znak '`?`', ktorým môžeme zvoliť možnosť Pomoc.

2.5 Testovanie

Kvôli správnemu fungovaniu skriptov bude potrebné testovanie, a to na všetkých troch verziách Windowsu, pre ktoré implementujeme. Na to nám poslúžia virtuálne mašiny od Microsoftu. Tie sú k dispozícii na "*https://developer.microsoft.com/en-us/microsoft-edge/tools/vms*". Sú to všetko edície Enterprise.

Kapitola 3

Opatrenia

V tejto kapitole budeme rozoberať konkrétne opatrenia a požiadavky, o ktorých pojednáva metodika, teda to, prečo sú potrebné a ako budú implementované.

Na začiatku je potrebné zdôrazniť, že pre vykonanie skriptov, ktorých funkcionalitu ideme popisovať, je nutné spustiť PowerShell ako administrátor, a to kvôli tomu, aby skripty nezlyhali na prístupových oprávneniach. To neznamená byť prihlásený ako administrátor a z tej pozície obyčajným spôsobom spustiť program (kliknutie na ikonu ľavým tlačidlom myši alebo Enter). Je skutočne potrebné ho explicitne spustiť ako administrátor, to znamená pravým tlačidlom myši kliknúť na ikonu programu a zvoliť možnosť Spustiť ako Administrátor. Druhá možnosť je držať stlačené Ctrl a Shift a počas toho kliknúť ľavým tlačidlom myši na ikonu programu.

Ďalšie upresnenie sa týka už spomenutých dopytov na dotyčného spúšťajúceho skripty. Tieto dopyty majú spravidla tvar "Chcete vykonať toto opatrenie?". Tým nechceme navodiť dojem, že bežne očakávame odmietanie týchto opatrení. Všetky z nich sú vysoko odporúčané, ale samozrejme môžu existovať opodstatnené dôvody, pre ktoré je vhodné niektoré opatrenie odmietnuť.

V nasledujúcom budeme okrem iného používať z angličtiny výraz default, pričom ho budeme skloňovať podľa slovenského jazyka. Významom tohto slova je prednastavená hodnota a používame ho kvôli jeho výstižnosti. Podobným spôsobom budeme používať aj výraz cmdlet (číta sa ako commandlet). Tak sa nazýva skupina príkazov v PowerShelli tvaru `sloveso-podstatné_meno`, napríklad `Get-Help`.

K jednotlivým opatreniam nebudeme písať, ktoré skripty ich zavádzajú. Z názvov skriptov bude zrejmé, ktorého opatrenia sa týkajú.

V mnohých podkapitolách budeme implementáciu vykonávať pomocou zmien vo Windows Registry (budeme hovoriť, že v registroch). Registre sú v skratke niečo podobné súborovému systému, pretože tam tiež existujú cesty (directories). Sú tam uchovávané dôležité prvky konfigurácie celého systému. Jednotlivé hodnoty v registroch budeme meniť pomocou cmdletu `Set-ItemProperty`.

3.1 Anti-malware riešenie

Veľmi dôležitým prvkom zabezpečenia pracovnej stanice je aj tzv. anti-malware riešenie (tým nie je myslené nič iné, ako možno trochu bežnejší pojem - antivírusová ochrana). Jeho hlavnou funkciou je ochrana v reálnom čase pred vírusmi, malvérom a ďalšími ohrozeniami bezpečnosti. Okrem toho zvykne napríklad umožňovať kontrolu konkrétnych programov, o ktorých sa chce používateľ uistiť, že sú bezpečné, kontrolu prijatých e-mailov, pravidelné skenovanie počítača na prítomnosť vírusov alebo aj presunutie podozrivých súborov do karantény a podobne. Dôležitou súčasťou tohto nástroja bývajú pravidelné aktualizácie, ktoré sú nevyhnutné na to, aby bolo zariadenie chránené pred najnovšími hrozbami.

Metodika prirodzene požaduje tento typ ochrany pre pracovné stanice. Na Windowsoch sa pre tento účel núka Windows Defender, ktorý je tu automaticky k dispozícii. Pri vhodnej konfigurácii by mal spĺňať požiadavky metodiky týkajúce sa anti-malware riešenia.

Chceme podotknúť, že vzhľadom na existenciu množstva iných takýchto programov by mal správca systému posúdiť potrebu nasadenia Windows Defendera. Ak má totiž organizácia nasadený niektorý iný, možno platený antivírusový program a jeho konfigurácia spĺňa požiadavky metodiky, môže byť rozumnejšie nemeniť daný stav. Každopádne, ak nie je prítomný žiadny taký program, využitie Windows Defendera je vysoko odporúčané.

3.1.1 Implementácia

V tejto časti využijeme v PowerShelli modul Defender, ktorý umožní relatívne pohodlnú prácu s nastavením Windows Defendera. Postačia nám príkazy `Update-MpSignature` a `Set-MpPreference` s niekoľkými prepínačmi. V skripte budeme postupovať takto:

- vykonáme aktualizáciu definícií - súbory, ktoré Windows Defender Antivirus používa na ochranu zariadenia pred najnovšími hrozbami [7];
- aktivujeme program zapnutím ochrany v reálnom čase;
- nastavíme pravidelné úplné skenovanie pracovnej stanice raz za týždeň tak, ako to požaduje metodika (ako čas sme zvolili pondelok o 12:00). Okrem toho nastavíme, aby bolo skenovanie spustené aj vtedy, ak je počítač práve používaný;
- pre splnenie požiadaviek metodiky taktiež zapneme pravidelnú aktualizáciu definícií každý deň o 11:00;
- pre obe tieto pravidelné činnosti zapneme aj ich náhradné vykonávanie v prípade vynechania, napríklad z dôvodu vypnutia PC;

- nakoniec nastavíme uchovávanie bezpečnostných skenov na 6 mesiacov tak, ako to špecifikuje metodika.

Keďže na Windows 7 nemusí byť kvôli verzii PowerShellu dostupný modul Defender, vytvoríme preň skript zvlášť. Fungovať bude tak, že ak je k dispozícii modul Defender, spustí skript pre zvyšné Windows verzie, inak vypíše výzvu na manuálne nasadenie Windows Defendera.

3.2 Aktualizácie OS

Operačné systémy Microsoft Windows sú v oblasti počítačov dlhodobo najpoužívanejšie na svete. Pri takej miere používania tiež stúpa miera pravdepodobnosti objavenia nejakej zraniteľnosti tohto OS a výskytu jej zneužitia. Ak takúto zraniteľnosť objaví ako prvý útočník, môže to viesť k úspešnému útoku na cieľový IT systém a k jeho kompromitácii. Aj napriek tomu, že úroveň zabezpečenia stále rastie, týmto prípadom sa vo všeobecnosti nedá s istotou predísť. Lepší prípad je, ak takúto zraniteľnosť objaví samotný výrobca alebo bezpečnostná komunita. Vtedy môže výrobca vytvoriť a dať k dispozícii bezpečnostnú aktualizáciu, ktorá daný problém vyrieši. Túto aktualizáciu však môže zanalyzovať útočník, vďaka čomu vie zistiť, o akú zraniteľnosť sa jedná a tam, kde aktualizácia nebola včas zavedená, bude môcť pohodlne zaútočiť.

Preto je veľmi dôležité aktualizácie zavádzať čo najskôr. Z toho dôvodu je tiež zrejmé, že pre IT systémy verejnej správy nie je vhodné, aby používali také verzie OS, ktoré nie sú podporované výrobcom - pretože tie nedostávajú potrebné aktualizácie.

3.2.1 Implementácia

Metodika požaduje, aby bolo na pracovných staniciach zapnuté automatické sťahovanie a inštalácia aktualizácií. Tento stav dosiahneme niekoľkými úpravami v registroch. Pre OS Windows 7 a Windows 8.1:

- v *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update* nastavíme hodnotu ScheduledInstallDay na 0, hodnotu ScheduledInstallTime na 12 a hodnotu AUOptions na 4;
- to znamená automatické sťahovanie a inštaláciu aktualizácií každý deň v čase o dvanástej hodine. [8]

Pre OS Windows 10:

- najskôr zistíme, či existuje *HKLM:\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU* pomocou cmdletu `Test-Path`;

- ak nie, tak ho vytvoríme pomocou cmdletu `New-Item` (*HKLM:\SOFTWARE\Policies\Microsoft\Windows* určite existuje);
- hodnoty tu nastavujeme rovnaké ako tie, ktoré sú spomenuté vyššie;
- v prípade, že sme na 64-bitovom systéme, vykonáme analogické úpravy aj v *HKLM:\SOFTWARE\Wow6432Node\Policies\Microsoft\Windows\WindowsUpdate\AU* (to zistíme podľa hodnoty príkazu `[System.IntPtr]::Size` - ak 8, potom je to 64-bit, inak 32-bit).

3.3 Firewall

Firewall je vo všeobecnosti zariadenie, ktoré filtruje komunikáciu medzi prvkami sieťovej infraštruktúry, pričom sa riadi určitou politikou pre prepúšťanie paketov a definovanými pravidlami. Je jedným zo základných pilierov bezpečnosti v súvislosti s počítačovými sieťami.

Podľa metodiky má byť na pracovných staniciach implementovaný lokálny firewall. Pre tento účel máme na Windowsoch vstavaný nástroj Windows Firewall (pre Windows 10 je to Windows Defender Firewall). Ten ako defaultnú politiku má nastavené povolenie komunikácie smerom dnu (inbound) a zakázanie komunikácie smerom von (outbound). Ďalej má automaticky definovaných niekoľko pravidiel, ktoré fungujú ako výnimky pre danú politiku. Celé to teda je nastavené tak, že každá inbound komunikácia je zakázaná, ak sa na ňu nevzťahuje pravidlo, ktoré by ju povoľovalo a opačne, každá outbound komunikácia je povolená, ak sa na ňu nevzťahuje niektoré pravidlo, ktoré by ju zakazovalo.

Takáto politika však nespĺňa požiadavky metodiky, ktorá hovorí, že by pravidlá mali byť spravované a implementované na princípe "least privilege", teda povoliť iba to, čo je nevyhnutne potrebné pre činnosť, ktorú ma vykonávať daná pracovná stanica. Preto zvolíme takú politiku, ktorá bude defaultne blokovať všetku inbound aj outbound komunikáciu okrem tých, ktoré povoľuje niektoré pravidlo.

Čo sa týka pravidiel, budeme implementovať iba outbound pravidlá týkajúce sa konkrétnych portov s transportnými protokolmi. Existujú však aj ďalšie atribúty, ktoré je možné konfigurovať, napríklad kompletne povoliť komunikáciu nejakej aplikácii a podobne.

Vo Windows (Defender) Firewalle je možné politiku a pravidlá priraďovať nasledujúcim tzv. profilom:

- doménový - vzťahuje sa na siete v rámci domény;
- privátny - vzťahuje sa na súkromné a domáce siete;

- verejný - vzťahuje sa na verejné siete.

Príkazy, ktoré budeme používať v skriptoch, sa budú vždy týkať všetkých troch týchto profilov.

3.3.1 Implementácia

Implementáciu vykonáme pre Windows 7 pomocou príkazu `netsh advfirewall` a pre Windows 8.1 a 10 využijeme modul NetSecurity. [13] [14] Budeme postupovať takto:

- firewall zaktivizujeme, ak náhodou ešte nebol;
- vymažeme všetky prítomné pravidlá;
- nastavíme defaultnú politiku blokovania všetkej inbound aj outbound komunikácie;
- vytvoríme pravidlá pre komunikáciu prostredníctvom niekoľkých základných aplikačných protokolov tým, že povolíme tok dát smerom von pre konkrétne porty s transportnými protokolmi. Chceme ujasniť, že aj keď hovoríme o outbound komunikácii, zahŕňa aj odpovede na dopyty danej stanice;
- nastavíme veľkosť logovacieho súboru na maximum - 32767 kilobajtov.

Uvádzame ešte protokoly aplikačnej vrstvy, pre ktoré implementujeme spomenuté pravidlá a im prislúchajúce porty s transportnými protokolmi:

- DNS - port 53 s UDP aj TCP;
- HTTP - port 80 s TCP;
- HTTPS - port 443 s TCP;
- SMB - porty 139 a 445 s TCP.

Na záver tejto podkapitoly dodávame, že bude rozumné v budúcnosti vytvoriť ďalší skript konfiguračného charakteru alebo doplniť tento tak, aby uľahčoval pridávanie nových pravidiel pre Windows (Defender) Firewall, keďže nie je možné obsiahnuť všetky výnimky vopred a je potrebné reagovať na špecifické podmienky v jednotlivých organizáciách.

3.4 Logovanie

Aj napriek všetkým snahám o zabezpečenie pracovných staníc sa môže stať, že dôjde ku kompromitácii a v takom prípade je cieľom čo najrýchlejšie problém vyriešiť a zamedziť tomu, aby sa niečo podobného charakteru opakovalo v budúcnosti. Pre tento účel využijeme logovanie, čo je vlastne akési zaznamenávanie udalostí - umožňuje spätné posudzovanie útoku, a to bez ohľadu na to, či viedol ku kompromitácii alebo nie (touto oblasťou sa zaoberá forenzná analýza). Vo Windowsoch máme na to k dispozícii nástroj Windows Event Log.

3.4.1 Implementácia

Na implementáciu využijeme príkaz `auditpol` s prepínačom `/set`, pomocou ktorého je možné pracovať s politikami týkajúcimi sa logovania. Pomocou prepínačov `/success:enable` a `/failure:enable` (čím dosiahneme zaznamenávanie úspešných aj neúspešných pokusov o danú činnosť) nastavíme podľa požiadaviek z metodiky logovanie týchto kategórii respektíve podkategórií:

- Logon a Logoff - prihlásenie a odhlásenie;
- Process Creation - spustenie procesu alebo služby. Býva zaznamenané meno programu a meno používateľa, ktorý ho spustil;
- Account Management - zahŕňa operácie s používateľmi a skupinami, teda ich vytváranie, modifikovanie, mazanie, pridávanie a odoberanie používateľov do a zo skupín, zmeny hesiel používateľov a podobne;
- RPC Events - vytváranie RPC (remote procedure call) spojení (počas nich sú vykonávané procedúry zadané zo vzdialených počítačov);
- Other Account Logon Events - môže zahŕňať odomykanie a zamykanie počítača, nové spojenia so vzdialenými PC a ďalšie. [16] [10]

3.5 Doména

Podľa metodiky by mala byť každá pracovná stanica pripojená do domény. To samozrejme z našej pozície nevieme zabezpečiť. Preto aspoň správcu systému upozorníme, ak to tak nie je.

3.5.1 Implementácia

Pre implementáciu využijeme WMI (Windows Management Instrumentation). Pomocou príkazu `gwmi Win32_ComputerSystem` obdržíme inštanciu triedy Win32_ComputerSystem, ktorej metóda `PartOfDomain()` vráti príslušnú boolovskú hodnotu podľa toho, či stanica je v doméne alebo nie.

3.6 Administrátorské oprávnenia

Na Windowsoch musí vždy figurovať aspoň jeden administrátorský účet (člen skupiny Administrators). Administrátorské oprávnenia je však možné prideliť aj bežným používateľom (členom skupiny Users). Metodika ale hovorí, že bežní používatelia nemajú mať tieto oprávnenia. Na jednej strane sa to môže zdať obmedzujúce, keďže oni napríklad nemôžu inštalovať softvér (to zabezpečujeme v inej podkapitole). Faktom však je, že čím väčšie práva používateľom dáme, tým väčšia je pravdepodobnosť narušenia bezpečnosti pracovnej stanice, či už vezmeme do úvahy inštaláciu škodlivého softvéru alebo nesprávnu zmenu konfigurácie systému, spôsobené pochybením používateľa.

3.6.1 Implementácia

V podstate je len potrebné prezrieť skupiny Users a Administrators (pre splnenie požiadavky musia mať prázdny prienik). Použijeme na to príkazy `net localgroup Users` a `net localgroup Administrators`. Na ich výstup sa môžeme pozerať ako na pole stringov (reťazcov). Pomocou dvoch for-cyklov prezrieme vhodné indexy týchto polí (v poliach sa totiž nenachádzajú len mená členov týchto skupín). V prípade nájdenia používateľa s administrátorskými právami skript vyzve na jeho odstránenie zo skupiny Administrators.

Pri tejto implementácii sme narazili na malý problém s verziami PowerShellu. Na spracovanie výstupu z príkazov `net localgroup` sme totiž chceli použiť metódu `Where()`. Tá je však k dispozícii až od PS verzie 4.0. Nakoniec sme využili obyčajnú prácu s poľom, čo sa ukázalo byť možno ešte jednoduchším riešením, ale hlavne vhodným pre všetky požadované verzie PS.

3.7 Lokálni administrátori

Podľa metodiky by mali byť lokálne administrátorské účty neaktívne. To je prirodzená požiadavka, pretože s ohľadom na celú doménu je lepší prípad, keď je pracovná stanica kompromitovaná cez bežného používateľa, ako keby bola kompromitovaná cez lokálneho administrátora.

Túto požiadavku splníme tým, že ak je stanica v doméne a sú na nej nejaké lokálne administrátorské účty aktívne, tak ich vypíšeme s tým, že by mali byť neaktívne.

Druhej časti tejto požiadavky ohľadom hesiel sa venovať nebudeme.

3.7.1 Implementácia

Implementáciu vykonáme v tom istom skripte, kde budeme riešiť aj ostatných privilegovaných používateľov. Spôsob, ktorým dosiahneme výpis, bude zrejmý z časti, kde sa im venujeme, preto nateraz detaily vynechávame.

3.8 Automatické prihlasovanie

Na Windowsoch je možné nastaviť systém tak, aby bol niektorý používateľ prihlasovaný automaticky bez toho, aby musel zadávať svoje prihlasovacie údaje. Z bezpečnostného hľadiska je to samozrejme neprípustné, pretože ak sa niekto so zlým úmyslom dostane k danému počítaču, nemusí existovať nič, čo by mu bránilo sa doň prihlásiť. Spoliehať sa na to, že sa k nemu nikto nedostane, asi väčšinou nie je rozumné.

Metodika teda logicky špecifikuje, aby pracovné stanice vždy vyžadovali na prihlásenie heslo.

3.8.1 Implementácia

Vykonáme jednu zmenu v registroch:

- v *HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon* nastavíme hodnotu AutoAdminLogon na 1.

3.9 Šetrič obrazovky a uzamknutie

Metodika požaduje, aby po najviac tridsiatich minútach nečinnosti bol spustený šetrič obrazovky a uzamknutá pracovná stanica. Z hľadiska bezpečnosti je dôvod zrejmý. Pre úplnosť uvedieme príklad. Mohlo by sa napríklad stať, že zamestnanec spoločnosti zabudne vypnúť počítač a ten sa ani po nejakom rozumnom čase stále neuzamkne, v takom prípade by fyzický prístup útočníka ku PC mohol predstavovať hrozbu, či už z dôvodu prítomnosti citlivých údajov v ňom alebo z dôvodu prístupu k samotnej sieťovej infraštruktúre spoločnosti, ktorá možno nie je ideálne zabezpečená, prostredníctvom tohto PC.

3.9.1 Implementácia

Vykonáme tri zmeny v registroch v *HKCU:\Control Panel\Desktop*:

- nastavíme hodnotu ScreenSaveActive na 1;
- nastavíme hodnotu ScreenSaverIsSecure tiež na 1;
- nastavíme hodnotu ScreenSaveTimeOut na 1800, čo predstavuje 1800 sekúnd, teda 30 minút.

3.10 Právo inštalácie softvéru

Z bezpečnostného hľadiska je zo zrejmých dôvodov rozumné, ak bežnému používateľovi (na Windowsoch tým myslíme člena skupiny Users) nie je umožnené inštalovať nový softvér. Vyhneme sa tak situácii, že dobrovoľne (aj keď možno omylom) nainštaluje škodlivý softvér, ktorý predstavuje bezpečnostnú hrozbu pre daný IT systém.

V oblasti verejnej správy by takéto opatrenie nemalo robiť problém z hľadiska poskytovania funkcionality danej pracovnej stanice, keďže zvyčajne ide o nejaké konkrétne účely, na ktoré sa používa. V krátkych časových intervaloch teda zvyčajne nie je potrebné inštalovať nový softvér. Výnimky vyrieši administrátor.

3.10.1 Implementácia

Právo na inštaláciu softvéru bežným používateľom odoberieme jednoduchým nastavením v registroch:

- v *HKLM:\SOFTWARE\Classes\Msi.Package\DefaultIcon* nastavíme hodnotu (Default) na 'C:\Windows\System32\msiexec.exe,1';
- podstatný je posledný znak - ak je 1, používatelia nemajú toto právo, ak je 0, majú ho.

3.11 Počet uložených hashov

Podľa metodiky by na pracovnej stanici mal byť počet uložených hashov (odtlačkov) prihlasovacích účtov nastavený na maximálne dva. Cieľom tohto opatrenia je, aby bol v prípade kompromitácie pracovnej stanice minimalizovaný počet uniknutých autentizačných údajov, aj keď sa jedná len o hashe hesiel. Vo Windowsoch je dva minimálny počet, pri ktorom je možné sa za každých okolností prihlásiť na pracovnú stanicu v doméne jednému používateľovi. [4]

3.11.1 Implementácia

Toto opatrenie implementujeme jednoduchou zmenou v registroch:

- v *HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon* nastavíme hodnotu CachedLogonsCount na 2;
- defaultne tam býva hodnota 10.

3.12 AppLocker

Jedným z najdôležitejších opatrení týkajúcich sa zabezpečenia pracovných staníc je tzv. aplikačný whitelisting. Pojmom whitelist je myslený určitý zoznam alebo množina, ktorých prvky majú nejaké oprávnenie, povolenie či privilégia. Preto ak hovoríme o aplikačnom whitelistingu, máme na mysli mechanizmus, prostredníctvom ktorého vytvárame množinu aplikácii, ktoré majú povolenie na to, aby boli spustené.

Vzhľadom na to, že väčšina útokov na pracovné stanice sa v určitom momente dostane do fázy spúšťania nejakého skriptu alebo programu tretích strán, týmto opatrením môžeme zmariť veľké množstvo útokov.

Existujú rôzne nástroje umožňujúce využívať aplikačný whitelisting, my sa budeme zaoberať AppLockerom od spoločnosti Microsoft, ktorý je súčasťou vyšších edícií OS Windows (teda netreba ho sťahovať, no na druhej strane nie je možné ho používať na ostatných edíciách). Ak je k dispozícii, môžeme ho nájsť v Local Security Policy v časti *SecuritySettings\Application Control Policies*.

V AppLockeri existujú tri možnosti, na základe ktorých možno vytvárať pravidlá pre povolenie spúšťania programov - digitálny podpis súboru, cesta k súboru alebo hash súboru. Ďalej, môžeme kontrolovať spúšťanie týchto typov súborov: spúšťateľné súbory, skripty, Windows Installer súbory, DLL (Dynamic Link Library) súbory a (len pre Windows 8.1 a Windows 10) tzv. Packaged apps a Packaged app installers. Okrem toho tiež vyberáme, na ktorú skupinu používateľov sa dané pravidlo vzťahuje.

Ako už bolo spomenuté, AppLocker je k dispozícii len pre vyššie edície OS Windows, konkrétne sú to:

- pre Windows 7 - Enterprise a Ultimate;
- pre Windows 8.1 - Enterprise;
- pre Windows 10 - Enterprise a Education.

V dokumentácii požiadaviek pre používanie tohto nástroja si môžeme všimnúť, že AppLocker nájdeme pre Windows 7 a 8.1 aj na edíciách Pro, nie je však možné vytvorené politiky uplatňovať a teda ani ich v takom prípade nebudeme vytvárať. [15]

Pre ostatné edície je možné použiť iné nástroje, tým sa však v tejto práci nebudeme venovať.

Čo sa týka pravidiel, použijeme defaultné pravidlá pre jednotlivé typy súborov. Možno ich odpozorovať v štandardnom grafickom používateľskom rozhraní (GUI) pre AppLocker, kde sa dajú automaticky vygenerovať. Sú to tieto pravidlá:

- povolenie všetkým používateľom pre všetky DLL súbory, spúšťateľné súbory a skripty nachádzajúce sa v *C:\Program Files* (pre 64-bitové systémy aj v *C:\Program Files (x86)*);
- povolenie všetkým používateľom pre všetky DLL súbory, spúšťateľné súbory a skripty nachádzajúce sa v *C:\Windows*;
- povolenie administrátorom pre všetky DLL súbory, spúšťateľné súbory, skripty a Windows Installer súbory v PC;
- povolenie všetkým používateľom pre všetky Windows Installer súbory na základe digitálneho podpisu súboru;
- (neplatí pre Windows 7) povolenie všetkým používateľom pre všetky Packaged apps a Packaged app installers na základe digitálneho podpisu;
- povolenie všetkým používateľom pre všetky Windows Installer súbory nachádzajúce sa v *C:\Windows\Installer*.

3.12.1 Implementácia

Na riadenie AppLockeru bude treba použiť špeciálne cmdlety, ktoré je možné v PowerShelli importovať príkazom `Import-Module AppLocker`. Jednotlivé body spomenuté vyššie implementujeme takto:

- pomocou príkazu `Get-AppLockerFileInformation`, ktorému zadáme príslušné parametre, získame potrebné informácie o súboroch, pre ktoré chceme vytvárať pravidlá. Sú to informácie ohľadom cesty, hashu súboru alebo poznatky prameniace z digitálneho podpisu;
- výstup z neho pomocou tzv. pipe nasmerujeme do príkazu `New-AppLockerPolicy`, ktorý zo vstupných informácii a niektorých ďalších parametrov (hlavne typ pravidla a používateľská skupina) vytvorí novú AppLocker politiku;
- daný výstup opäť nasmerujeme pomocou pipe, tentoraz do príkazu `Set-AppLockerPolicy`, ktorý vstupnú politiku nasadí. Je dôležité pri viacnásobnom použití tohto cmdletu použiť prepínač `-Merge` kvôli zlučovaniu jednotlivých politík, pretože bez neho `Set-AppLockerPolicy` prepisuje pôvodnú politiku novou. [17]

3.13 UAC

UAC (User Account Control) je technológia, ktorej cieľom je zvýšiť zabezpečenie Windowsov voči škodlivým programom. Môže kontrolovať ten typ udalostí, keď sa nejaký program pokúša robiť zmeny v počítači ako je napríklad inštalácia nového softvéru alebo keď sa používateľ pokúša robiť zmeny v nastaveniach systému, ktoré vyžadujú administrátorské oprávnenie a podobne (napr. keď chceme spustiť Windows PowerShell s administrátorským oprávnením). Ak k takej udalosti dôjde, vyskakuje dialógové okno, ktoré si pýta súhlas alebo prihlasovacie údaje administrátora (používateľské meno a heslo) - v závislosti od konfigurácie UAC a toho, či je prihlásený bežný používateľ alebo administrátor.

V nastaveniach UAC je možné si vybrať, v akej miere si prajeme byť notifikovaný o týchto udalostiach a zadávať administrátorský súhlas respektíve prihlasovacie údaje administrátora. V štandardnom GUI pre manuálne konfigurovanie UAC máme tieto štyri možnosti:

- notifikovať vždy o všetkých takýchto udalostiach - ak udalosť nastane, bude zakalená obrazovka a vyskočí spomenuté dialógové okno;
- (defaultná možnosť) notifikovať len vtedy, ak sa nejaký program pokúša robiť zmeny v PC alebo v nastaveniach, ktoré vyžadujú administrátorské oprávnenie - tiež bude zakalená obrazovka a vyskočí dialógové okno;
- rovnako ako v predchádzajúcej možnosti, ale obrazovka nebude zakalená, tým pádom ostatné programy môžu ovplyvňovať dané dialógové okno;
- žiadne notifikácie - vypnutý UAC (ak sa bežný používateľ pokúša o zmeny, ktoré vyžadujú administrátorské oprávnenie, budú automaticky zamietnuté).

Podľa metodiky by mal byť UAC zapnutý a nastavený na najvyššiu hodnotu, čo je prvá možnosť z predchádzajúcich uvedených.

3.13.1 Implementácia

Požadovaný stav dosiahneme štyrmi zmenami v registroch, konkrétne v *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System* nastavíme tieto hodnoty:

- ConsentPromptBehaviorAdmin na 2;
- ConsentPromptBehaviorUser na 1;
- EnableLUA na 1;
- PromptOnSecureDesktop na 1. [18]

3.14 EMET

EMET (Enhanced Mitigation Experience Toolkit) je nespoplatneným nástrojom od spoločnosti Microsoft, ktorého cieľom je zabrániť útočníkom získať prístup k počítačom. Je zameraný proti technikám, ktoré zneužívajú zraniteľnosti v kóde cieleného softvéru. Často to bývajú bežné techniky a známe postupy, ktoré útočníci využívajú a EMET je schopný také pokusy rozoznať - v takom prípade chráni daný PC presmerovaním, ukončením a blokovaním týchto útokov.

Obrovská výhoda EMETu je v tom, že môže zabrániť útokom typu zero-day - to sú útoky využívajúce zraniteľnosti, ktoré ešte neboli objavené výrobcom softvéru ani bezpečnostnou komunitou, takže v ohrození je každý používateľ takého softvéru. Tento nástroj teda môže v určitých prípadoch zastúpiť funkciu bezpečnostných aktualizácií alebo dokonca aj antivírusových ochrán.

Metodika hovorí, že by mal byť nainštalovaný a nakonfigurovaný so všetkými ochranami, ktoré je možné zapnúť. Je možné, že činnosť niektorých zastaraných alebo nesprávne naprogramovaných aplikácií môže byť vyhodnotená ako pokus o útok - je teda potrebné konfiguráciu vyladiť (teda možno niektoré ochrany vypnúť), ale tiež aj zvážiť, či je nutné také aplikácie používať alebo či ich prípadne nie je možné nejakým spôsobom nahradiť.

3.14.1 Implementácia

Najprv potrebujeme inštalačný súbor - ten získame z "*https://www.microsoft.com/en-us/download/details.aspx?id=50766*" (nevieme zaručiť ako dlho bude tento link aktuálny, ale každopádne je jednoduché tento súbor vyhľadať na stránke Microsoftu alebo prostredníctvom služieb ako je Google). Má názov "*EMET Setup.msi*". Ten spustíme v PowerShelli pomocou súboru "*msiexec.exe*", ktorý sa vo Windowsoch nachádza v *C:\Windows\System32*. Ako aj jeho názov napovedá, slúži na spúšťanie Windows Installer súborov. Použijeme ho v PowerShelli ako príkaz, a to s týmito prepínačmi:

- `/i` - štandardná inštalácia;
- `/passive` - počas vykonávania sa zobrazí iba okno, ktoré ukazuje mieru pokroku inštalácie, takzvaný progress bar;
- `/norestart` - po úspešnej inštalácii počítač nebude reštartovaný.

Konfigurovanie budeme vykonávať pomocou súboru s názvom "*EMET_Conf.exe*", ktorý nájdeme v Program Files v priečinku, kde je nainštalovaný EMET. Využijeme takzvané EMET Protection Profiles (ochranné profily), čo sú XML súbory obsahujúce nastavenia pre tento nástroj. Tri takéto súbory sú automaticky získané pri in-

štalácii, pričom obsahujú predkonfigurované nastavenia týkajúce sa bežných programov od Microsoftu a ďalších overených dodávateľov. Konkrétne sú to "*Recommended Software.xml*", "*Popular Software.xml*" a "*CertTrust.xml*". Nachádzajú sa v *Deployment\Protection Profiles* (v priečinku, kde je nainštalovaný EMET). Nasadíme ich použitím prepínača `--import` pri spustení "*EMET_Conf.exe*". Je možné s nimi aj ďalej narábať, teda modifikovať ich, prípadne pomocou nich vytvárať nové Protection Profiles. [12]

3.15 Privilegovaní používatelia

Dá sa očakávať, že pracovné stanice v organizáciách verejnej správy budú aj súčasťou nejakej internej domény. V takom prípade sa nestačí zaoberať len lokálnymi administrátormi, sú tu totiž prítomné aj iné typy privilegovaných používateľov - členovia Domain Admins a Enterprise Admins.

Domain Admins je skupina, ktorej členovia majú oprávnenie na spravovanie celej domény. Defaultne majú členstvo v skupine Administrators (lokálni administrátori) vo všetkých počítačoch v doméne. Enterprise Admins je v zásade celkom podobná skupina, akurát že existuje len v koreňovej doméne tzv. lesa (množina domén). [9]

Metodika hovorí, že členovia týchto skupín nemajú mať oprávnenie na prihlásenie sa na pracovné stanice. Defaultne toto oprávnenie majú. To môže predstavovať bezpečnostné riziko. Tak ako už aj bolo spomenuté, treba si uvedomiť, že aj napriek všetkej snahe existuje možnosť, že bude pracovná stanica kompromitovaná. V prípade, že by napríklad člen skupiny Domain Admins bol aj lokálnym administrátorom na pracovných staniciach, tak ak by sa útočníkovi podarilo kompromitovať celú pracovnú stanicu, mal by priestor na pokračovanie v útoku, a to smerom na celú doménu. Tomu chceme určite zabrániť.

3.15.1 Implementácia

Toto opatrenie má samozrejme zmysel zavádzať len vtedy, ak sa počítač nachádza v doméne. Ak teda áno, vykonáme nasledujúce kroky:

- pomocou príkazu `net localgroup Administrators` získame zoznam lokálnych administrátorov a keďže sme v doméne, tak tiež zoznam členov skupín Domain Admins a Enterprise Admins;
- ak sa objaví prípad, že člen skupiny Domain Admins alebo Enterprise Admins je aj v skupine Administrators, skript vyzve na jeho odstránenie z tejto skupiny.

3.16 AutoRun

Podľa metodiky by malo byť vypnuté automatické spúšťanie programov po vložení vymeniteľného média. Táto funkcia sa nazýva AutoRun. Je zjavné, že ak by sme ju nechali zapnutú, vloženie zariadenia s nebezpečným obsahom by mohlo infikovať počítač škodlivým softvérom. Je teda dôležité splniť túto požiadavku.

3.16.1 Implementácia

AutoRun vypneme jednoduchou zmenou v registroch:

- v *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer* nastavíme hodnotu NoDriveTypeAutoRun na 255 (0xff v hexadecimálnej číselnej sústave);
- hodnota 255 značí vypnutie AutoRunu na všetkých jednotkách. [6]

3.17 NetBIOS

NetBIOS (Network Basic Input/Output System) je protokol, ktorý umožňuje programom na rôznych počítačoch navzájom komunikovať v lokálnej sieti. Poskytuje služby relačnej a transportnej vrstve referenčného modelu OSI. [20]

Tento protokol predstavuje pre IT systém možnú hrozbu, pretože schopný útočník dokáže prostredníctvom neho zhromaždiť o príslušnej sieti údaje, ktoré mu môžu byť užitočné pri hľadaní zraniteľností organizácie a potom aj pri útoku.

Metodika požaduje, aby bol protokol NetBIOS nad IPv4 vypnutý.

3.17.1 Implementácia

Implementáciu vykonáme pomocou WMI:

- najskôr zavoláme príkaz `gwmi Win32_NetworkAdapterConfiguration`, ktorý vráti inštanciu triedy Win32_NetworkAdapterConfiguration;
- táto trieda reprezentuje vlastnosti sieťových adaptérov; [19]
- pre každý adaptér zavoláme jeho metódu `SetTcpipNetbios()` s hodnotou 2, čo značí vypnutie NetBIOSu.

3.18 Kontá Microsoft

Spoločnosť Microsoft poskytuje možnosť vytvárania bezplatných kont, pomocou ktorých je možné sa prihlasovať na počítače s OS Windows. Tieto kontá potom možno používať aj pre prácu so službami ako Outlook, Office, OneDrive a tak ďalej.

Podľa metodiky lokálne účty nesmú byť zlinkované s kontami Microsoft. Inými slovami, nemalo by byť možné prihlásiť sa do daného PC pomocou týchto kont. Táto požiadavka je oprávnená, pretože z hľadiska organizácie nie je bezpečné, aby prihlasovacie údaje na jej pracovné stanice boli uložené v databázach Microsoftu (aj keď sa to asi nezdá veľmi pravdepodobné, únik dát niekedy môže nastať). Používaním týchto kont tiež uľahčujeme prípadný útok úplným preberaním (brute-force) na používateľské účty, pretože útočník nutne nemusí mať fyzický prístup k pracovnej stanici.

3.18.1 Implementácia

Urobíme jedinú zmenu v registroch:

- v *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System* nastavíme hodnotu NoConnectedUser na 3;
- tá určuje, že nie je možné sa prihlasovať pomocou Microsoft kont a ani žiadne pridávať. [5]

V prípade, že je splnenie tejto požiadavky z nejakého vážneho dôvodu zamietnuté (aj keď to veľmi neočakávame, keďže metodika používa pojem "nesmú", čo znamená povinný zákaz), vypíšeme výzvu na odinštalovanie služby OneDrive, ak je prítomná v PC. Tá je totiž použiteľná len pri prihlásení sa cez Microsoft konto, teda ak zavedieme toto opatrenie, nie je potrebné ju odstraňovať (službami tohto charakteru sa zaoberáme v nasledujúcej podkapitole).

3.19 Služby pre synchronizáciu dát

Ďalšia vec, ktorú metodika zakazuje, je synchronizácia dát na pracovnej stanici prostredníctvom verejných služieb ako OneDrive, Dropbox, Google Drive, MEGAsync a podobne. Tak ako sme uviedli aj v predchádzajúcej podkapitole, únik dát je možnosť, ktorú treba mať neustále na pamäti a teda hlavne pokiaľ ide o citlivé údaje, je nevhodné ich mať synchronizované, prípadne len ukladané prostredníctvom týchto typov služieb.

Okrem toho tieto služby ukrývajú aj ďalšie nebezpečenstvo. Google Drive napríklad funguje tak, že v počítači existuje zložka, ktorá reprezentuje obsah dát v tejto službe. Ak používateľ niektoré dáta v tejto zložke vymaže, sú odstránené aj z Google Drive,

ale nie úplne, pretože ešte sú v koši. Ak je však potom kôš vyprázdnený, tieto dáta sú úplne odstránené. Niektorému používateľovi by sa mohlo stať, že takto omylom vymaže dáta, o ktorých si myslí, že ich maže len z danej zložky, nie z Google Drive - taký prípad už aj nastal. Je to síce chyba na strane používateľa, naviac nepravdepodobná a spoločnosť Google sa takým snaží predísť, napriek tomu bude rozumnejšie vyhnúť sa týmto službám v IT systémoch verejnej správy úplne.

3.19.1 Implementácia

Bude potrebné skontrolovať, či sa takéto služby nachádzajú v počítači. Opäť na to využijeme registre. Z *HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall* získame zoznam názvov nainštalovaného softvéru (v prípade 64-bitových systémov aj z *HKLM:\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall*). Potom zistíme, či sa názov niektorej z nechcených služieb nachádza v zozname. Urobíme to pomocou prepínača `-contains`, ktorý je case-insensitive. Až na veľkosť písmen je ale potrebné zadať názvy presne tak, ako sú uložené v PC (pre Google Drive je to napríklad "Backup and Sync from Google").

Náš skript kontroluje konkrétne iba Google Drive, Dropbox a MEGAsync. V prípade potreby je možné do premennej *$in* ľahko doplniť ďalšie.

Ak sa nájde nejaká zhoda, skript vyzve na okamžité odinštalovanie danej služby.

3.20 Rozšírenia webových prehliadačov

Posledný bod metodiky v časti venovanej pracovným staniciam hovorí o tom, že webové prehliadače by mali mať implementované niektoré rozšírenia. V tejto práci sa budeme venovať len tým najdôležitejším - rozšíreniam na blokovanie spúšťania skriptov.

Implementáciu vykonáme iba pre prehliadač Mozilla Firefox, ale odporúčame ďalej pokračovať aj s ostatnými prehliadačmi.

Využijeme dobre známe a overené rozšírenie NoScript, ktoré je naviac dostupné bezplatne. Toto rozšírenie poskytuje ochranu pred spúšťaním tzv. plug-inov - určitých špecifických funkcií na webových stránkach. Vie zabrániť zneužitiu niektorých známych bezpečnostných zraniteľností ako napríklad Meltdown alebo Spectre. Funguje na princípe whitelistingu. [1]

3.20.1 Implementácia

Táto implementácia bude vyžadovať aj určité manuálne úkony zo strany toho, kto spustil skript. Nebude to ale nič dlhé ani náročné.

Najprv zistíme či vôbec je Mozilla nainštalovaná na danom počítači. To urobíme pomocou príkazu `Test-Path`, pričom si teda dovolíme predpokladať inštaláciu na štandardnom mieste - v Program Files. Ak nie, tak skript skončí. Inak bude spustený klasický dopyt.

Hlavná časť skriptu funguje takto:

- vypíše inštrukciu ohľadom pridania NoScriptu, kde sa píše, že za niekoľko sekúnd sa v prehliadači otvorí stránka, kde bude treba kliknúť na pridanie NoScriptu a potvrdiť to;
- pomocou príkazu `Start-Sleep` s prepínačom `-Seconds` vynútime desaťsekundovú pauzu, čo je predpokladaný čas čítania inštrukcie;
- spustíme prehliadač pomocou `firefox.exe`.

3.21 WPAD

WPAD (Web Proxy Auto-Discovery) je protokol na automatické vyhľadávanie proxy servera. Presnejšie, je to spôsob vyhľadávania takzvaných PAC (Proxy Auto-Configuration) súborov, ktoré umožňujú automatickú konfiguráciu webových prehliadačov pre používanie proxy servera. Tieto súbory sú vyhľadávané prostredníctvom dopytov na DHCP (Dynamic Host Configuration Protocol) servery a/alebo DNS (Domain Name System) servery.

Tento protokol teda umožňuje organizáciám automaticky nakonfigurovať prehliadače na pracovných staniciach tak, aby používali nimi určené proxy servery. Vďaka tomu je možné požadovaným spôsobom ľahko nakonfigurovať množstvo prehliadačov bez akejkoľvek námahy.

Na všetkých operačných systémoch Microsoft Windows a webových prehliadačoch Internet Explorer a Microsoft Edge je WPAD defaultne zapnutý. To sa môže z určitého hľadiska javiť ako rozumné, avšak z bezpečnostného hľadiska môže aktívny WPAD predstavovať hrozbu, ktorej sa určite chceme vyhnúť.

WPAD totiž funguje tak, že spomenutý PAC súbor (má názov "*wpad.dat*") je automaticky vyhľadaný v lokálnej sieti na základe mena pracovnej stanice v nej. Dopyty pre takéto vyhľadávanie môže za normálnych okolností spravovať nejaký súkromný server v rámci lokálnej siete, ak sa však z nej tento dopyt dostane von, útočník môže poskytovať vlastný PAC súbor, vhodne upravený, ktorý prehliadač nasmeruje na proxy server útočníka. V takej situácii bude mať útočník k dispozícii všetku komunikáciu daného prehliadača. [21]

Z toho dôvodu chceme na pracovných staniciach WPAD vypnúť, tak ako o tom hovorí aj metodika.

3.21.1 Implementácia

Implementáciu vykonáme dvoma jednoduchými nastaveniami v registroch:

- v *HKCU:\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Wpad* nastavíme hodnotu WpadOverride na 1;
- v *HKLM:\SYSTEM\CurrentControlSet\Services\WinHttpAutoProxySvc* nastavíme hodnotu Start na 4.

3.22 Oprávnenie Debug procesu

Toto oprávnenie umožňuje používateľom pripojiť sa k procesu a pozorovať všetku jeho činnosť. Vie tak poskytovať prístup k dôležitým citlivým dátam. Útočník môže zneužiť túto záležitosť a dostať sa k informáciám z operačnej pamäte, heslám, dokonca modifikovať jadro alebo nasadiť škodlivý kód. [11]

Defaultne ho majú členovia skupiny Administrators, bežní používatelia nie. Podľa metodiky ho nemajú mať ani lokálni administrátori.

Cez Local Security Policy sa k tomuto právu dostaneme v časti *Security Settings\Local Policies\User Rights Assignment*.

3.22.1 Implementácia

Pôvodne sme chceli toto opatrenie implementovať zmenou v registroch, lenže nepodarilo sa nám nájsť kľúč, ktorý by tomuto oprávneniu prislúchal, a to ani použitím nástroja RegShot, ktorý slúži na monitorovanie zmien v registroch. Okrem toho sme sa dočítali, že nič také ani v registroch nie je. [2]

Rozhodli sme sa, že využijeme príkaz `ntrights.exe`. Kvôli tomu je potrebné nainštalovať Windows Resource Kit Tools. To vykonáme spustením súboru "*rktools.exe*", ktorý stiahneme z "*https://www.microsoft.com/en-us/download/details.aspx?id=17657*". Nebude to však tichá inštalácia (bez zásahov používateľa). Napriek tomu by to nemalo byť nič náročné. Naviac skript pre istotu vypíše malú nápovedu k inštalácii. Po ukončení inštalácie bude vypísaná výzva pre ukončenie a následné opätovné spustenie PowerShellu (ako administrátor), kde by mal byť spustený skript "*debug.ps1*".

Pomocou nainštalovaného `ntrights.exe`, ktorému zadáme názov oprávnenia - SeDebugPrivilege, prepínač `-u` s názvom danej používateľskej skupiny a prepínač `-r` pre odobratie oprávnenia, odoberieme členom Users a Administrators Debug privilégium.

Skript pre toto opatrenie dáme kvôli obštrukciám s inštaláciou až na úplný koniec Master skriptu.

Netreba zabudnúť na reštart počítača, ktorý je nevyhnutný pre dokončenie zavádzania doterajších opatrení. Preto vytvoríme posledný skript, "*restart.ps1*", ktorý

spúšťame pri odmietnutí odstránenia Debug privilégií a taktiež na konci skriptu "*debug.ps1*". Samotný reštart je vykonávaný cmdletom `Restart-Computer`.

Záver

V tejto práci sme vykonali implementáciu opatrení pre zabezpečenie IT systémov verejnej správy pre oblasť pracovných staníc a to tak, aby spĺňala požiadavky nachádzajúce sa v metodike a tým sa stala vhodnou pre použitie v systémoch so zvýšenými bezpečnostnými požiadavkami. Predchádzalo tomu zoznámenie sa so samotnou metodikou a špeciálne s tou kapitolou, ktorá sa venuje pracovným staniciam. Následne sme sa zaoberali návrhom riešenia. Rozhodli sme sa, že pre operačné systémy Microsoft Windows budeme využívať PowerShell skripty, ktoré sú praktickým nástrojom pre vykonávanie potrebných nastavení a zároveň nie je zložité pre správcu daného systému s nimi pracovať (predpokladáme správcu s určitou úrovňou bezpečnostného povedomia a schopnosti práce v PowerShelli). Ďalej sme sa pustili do analýzy jednotlivých požiadaviek, skúmali sme dôvody, pre ktoré sú v metodike uvedené a v neposlednom rade sme pátrali po tom, ako jednotlivé nastavenia systému naprogramovať.

Venovali sme sa všetkým základným prvkom počítačovej bezpečnosti od automatických aktualizácií a aplikačného whitelistingu až po firewall, antivírusovú ochranu či používateľské práva. Ukázalo sa, že pri zabezpečení je nutné pamätať aj na niektoré na prvý pohľad triviálne veci ako je napríklad uzamykanie obrazovky či automatické spúšťanie programov po vložení vymeniteľného média.

Ako sme si mohli všimnúť, úroveň zabezpečenia je vysoká aj vďaka tomu, že miestami majú viaceré implementované opatrenia dopad na tú istú oblasť, čo vedie k tomu, že aj napriek zlyhaniu jedného opatrenia môže druhé zabrániť nechcenej udalosti. Je teda len správne, ak existuje viacero vrstiev ochrany voči potenciálnej hrozbe.

Dôležitou stránkou našej snahy nebola len prevencia voči spomínaným nebezpečenstvám, ale aj čo najvyššia možná miera zmiernenia dopadov úspešných útokov a následné rýchle riešenie problému a zamedzenie podobným útokom. Z toho dôvodu sme implementovali opatrenia v súvislosti s prístupom na pracovné stanice a uchovávanie bezpečnostných logov.

Dotkli sme sa aj používania niektorých bežných funkcií, ktoré poskytujú dnešné počítače, ako je používanie Microsoft kont alebo synchronizácia a ukladanie dát prostredníctvom verejných služieb ako Google Drive, Dropbox a podobne. Vysvetlili sme, prečo nie je prípustné tieto funkcie využívať na systémoch so zvýšenými bezpečnostnými požiadavkami.

V tejto časti by sme ešte chceli spomenúť opatrenia, u ktorých vidíme priestor na určité vylepšenia, prípadne body z metodiky, ktorým sme sa nevenovali a teda odporúčame na to pamätať pri použití nášho riešenia:

- pre firewall bude užitočné vytvoriť nový skript, prípadne modifikovať už hotový tak, aby bolo pohodlné pridávať nové pravidlá;
- pre AppLocker bude užitočné vytvoriť takú funkcionalitu, ktorá bude pridávať nové aplikácie do množiny povolených pre spúšťanie;
- čo sa týka webových prehliadačov, odporúčame doplniť skript pre zavedenie rozšírenia na blokovanie spúšťania skriptov aj pre ďalšie prehliadače;
- bod číslo 7.9 i. a 7.10;
- bod číslo 7.8 d. - na jeho splnenie nemáme dostatočný vplyv;
- body 7.2 a 7.8 c. - je pomerne zložité dosiahnuť splnenie takýchto typov požiadaviek.

V tejto práci sme sa celkom dosť zaoberali aj aktualizáciami, je preto vhodné teraz spomenúť, že aj toto riešenie implementácie metodiky bude skôr či neskôr potrebovať revíziu, či už kvôli zmenám v metodike alebo kvôli vývoju Windowsov. V krátkom čase sa napríklad blíži koniec technickej podpory nástroja EMET, ktorému sme sa venovali v tejto práci. Okrem toho sa tiež dá v určitom časovom horizonte očakávať strata podpory pre staršie verzie Windows. Pri využívaní práce preto treba mať túto skutočnosť na pamäti.

Čo sa týka napredovania v oblasti zabezpečenia IT systémov verejnej správy, okrem pracovných staníc, ktorým sme sa venovali v tejto bakalárskej práci, je potrebné sa venovať aj zabezpečeniu celej infraštruktúry vrátane serverov a tzv. domain controllerov. Pre tento účel odporúčame pustiť sa do implementácie ďalšej časti metodiky a to konkrétne do kapitoly Minimálne požiadavky na zabezpečenie internej infraštruktúry (kapitola 6 metodiky [3]). Vidíme tu však priestor aj na pokračovanie v oblasti pracovných staníc. My sme sa totiž venovali iba Windowsom, môže byť ale užitočné sa pozrieť aj na ostatné operačné systémy, predovšetkým máme na mysli Linux a macOS.

Literatúra

[1] NoScript. https://noscript.net.

[2] Which registry key is modified by Security Policy SeDebugPrivilege? https://superuser.com/questions/1016430/which-registry-key-is-modified-by-security-policy-sedebugprivilege?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa.

[3] CSIRT.SK. Metodika pre systematické zabezpečenie organizácií verejnej správy v oblasti informačnej bezpečnosti, 2017. Elektronická verzia na stiahnutie: https://www.csirt.gov.sk/informacna-bezpecnost/osvedcene-postupy/metodika-zabezpecenia-ikt-8a6.html.

[4] CSIRT.SK. Quick Wins (opatrenia, ktoré za nízke náklady významne zvýšia bezpečnosť infraštruktúry), 2017. https://www.csirt.gov.sk/informacna-bezpecnost/osvedcene-postupy/quick-wins-pre-zabezpecenie-organizacie-8a4.html.

[5] Alfred Geeks. 3 Ways to Block or Disable Microsoft Account in Windows 10 / 8. https://www.top-password.com/blog/block-or-disable-microsoft-account-in-windows-10-8.

[6] Mary Landesman. Disable AutoRun/AutoPlay. https://www.lifewire.com/disable-autorun-on-a-pc-153344.

[7] Microsoft. https://support.microsoft.com/sk-sk/help/4013263/windows-10-protect-my-device-with-windows-defender-antivirus.

[8] Microsoft. https://support.microsoft.com/sk-sk/help/328010/how-to-configure-automatic-updates-by-using-group-policy-or-registry-s.

[9] Microsoft. *Active Directory Security Groups*. https://docs.microsoft.com/en-us/windows/security/identity-protection/access-control/active-directory-security-groups.

[10] Microsoft. *Advanced security audit policy settings.* https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/advanced-security-audit-policy-settings.

[11] Microsoft. *Debug programs.* https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/debug-programs.

[12] Microsoft. Enhanced Mitigation Experience Toolkit 5.5 User Guide. Dostupné pre stiahnutie na https://www.csirt.gov.sk/informacna-bezpecnost/osvedcene-postupy/metodika-zabezpecenia-ikt-8a6.html.

[13] Microsoft. *NetSecurity.* https://docs.microsoft.com/en-us/powershell/module/netsecurity/?view=win10-ps.

[14] Microsoft. *Netsh Commands for Windows Firewall with Advanced Security.* https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771920(v=ws.10).

[15] Microsoft. *Requirements to use AppLocker.* https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/requirements-to-use-applocker.

[16] Microsoft. *Security Audit Policy Reference.* https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/dd772623(v%3dws.10).

[17] Microsoft. *Use the AppLocker Windows PowerShell cmdlets.* https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/use-the-applocker-windows-powershell-cmdlets.

[18] Microsoft. *User Account Control.* https://msdn.microsoft.com/en-us/library/cc232771.aspx.

[19] Microsoft. *Win32_NetworkAdapterConfiguration class.* https://msdn.microsoft.com/en-us/library/aa394217(v=vs.85).aspx.

[20] Margaret Rouse. NetBIOS (Network Basic Input/Output System). https://searchnetworking.techtarget.com/definition/NetBIOS.

[21] Mark Stockley. When domain names attack: the WPAD name collision vulnerability. https://nakedsecurity.sophos.com/2016/05/25/when-domain-names-attack-the-wpad-name-collision-vulnerability.

[22] Wikipedia. *PowerShell*. https://en.wikipedia.org/wiki/PowerShell#PowerShell_1.0.

Dodatok A

Príloha obsahuje PowerShell skripty, ktoré tvoria výslednú implementáciu metodiky, dva inštalačné súbory, s ktorými môžu skripty potenciálne pracovať a dokument Metodika pre systematické zabezpečenie organizácií verejnej správy v oblasti informačnej bezpečnosti, verzia 2.0. Tieto súbory sú k dispozícii na priloženom CD.