Poznatky z penetračných testov

Ing. Zuzana Vargová
Špecializovaný útvar CSIRT.SK
DataCentrum
Cintorínska 5, 81488 Bratislava
e-mail:zuzana.vargova@csirt.sk

Abstrakt

CSIRT.SK (*Computer Security Incident Response Team*) provides services associated with security incidents handling and impact elimination followed by the recovery of affected information and communication technologies. These services include performing penetration tests of information systems (IS) for public administration institutions. Penetration test is a simulation of a real attack against IS, focusing on finding security vulnerabilities which an attacker may misuse to gain control over the affected system. While performing penetration tests of organization´s internal infrastructure, as well while conducting external testing of publicly available services, we often find similar imperfections at many institutions. In this article, we will describe most frequently repeating vulnerabilities, potentially resulting in compromise of IS and/or infrastructure. We also include recommendations for elimination of these vulnerabilities.

1. Penetračné testovanie

Penetračný test, skrátene *pentest*, je procesom, pri ktorom sa tester snaží simulovať útok na informačný systém. Používa pritom podobné až rovnaké postupy, akými by sa o pokus o prienik do IS snažil reálny útočník. IT odborníkov, vykonávajúcich (okrem iného) penetračné testy, označujeme aj pojmom etický hacker (angl. *ethical hacker*). Pentester nevykonáva útok na systém s úmyslom spôsobiť škody, ale naopak: cieľom je poukázať na bezpečnostné slabiny systému, ktoré môže reálny útočník zneužiť, a v neposlednom rade odporučiť opatrenia na elimináciu rizika. Penetračné testovanie umožňuje ohodnotiť aktuálny stav IS alebo organizácie z hľadiska bezpečnosti, odhaľuje riziká a poskytuje podklady pre nasledujúce kroky, ktorými sa zistené nedostatky odstránia.

CSIRT.SK vykonáva penetračné testovanie informačných systémov inštitúcií verejnej správy od roku 2013. Odvtedy bolo ukončených viac ako 100 individuálnych testov a desiatky retestov. Z pohľadu predmetu testovania možno rozlišovať 2 základné typy pentestov:

- *Interný*, kde ide o test vnútornej infraštruktúry objednávateľa,
- *Externý*, pri ktorom sú testované externé, verejne dostupné služby, najčastejšie webové aplikácie.

2. Externý penetračný test

Častejšie požadované a časovo menej náročné sú *externé penetračné testy*. CSIRT.SK vykonal veľké množstvo testov webových aplikácii inštitúcii verejnej správy od roku 2014. Išlo najmä o bezpečnostné kontroly projektov vypracovaných v rámci *Operačného programu Informatizácia spoločnosti*, resp. pokračujúceho *Operačného programu Integrovaná infraštruktúra*. Vzhľadom na fakt, že požiadavky

na bezpečnosť výsledného produktu nebývali súčasťou zadaní projektov, pri kontrole bývalo spravidla zistených množstvo nedostatkov. Viaceré z nich mohli viesť k úniku dát dotknutej inštitúcie, ohrozeniu bezpečnosti používateľov či dokonca ku kompromitácii servera, na ktorom bola služba prevádzkovaná. Ovládnutý server by následne mohol byť útočníkom použitý ako pivot a poslúžiť na kompromitáciu (časti) internej infraštruktúry inštitúcie.

Zraniteľnosti webových aplikácií (ktoré sú predmetom externých testov najčastejšie), možno rámcovo rozdeliť do nasledujúcich oblastí:

- Šifrovanie
- Konfigurácia
- Aplikačné chyby
- Údaje
- Správa relácii

Rozdelenie do jednotlivých kategórií nemusí byť jednoznačné, nakoľko niektoré opatrenia môžu byť riešené tak na aplikačnej úrovni, ako aj na úrovni webového servera.

2.1. Vybrané zraniteľnosti webových aplikácií

Prvou kategóriou zraniteľností, ktoré pri externom testovaní pozorujeme, sú nedostatky v použití šifrovania na zabezpečenie komunikácie servera s používateľom. Rozumieme tým spravidla medzery v konfigurácii *SSL/TLS*[1].

V zmysle platného *Výnosu o štandardoch pre ISVS [č. 55/2014 Z. z.]* [1] a jeho novelizácií je o. i. pre transport dát, prístup k elektronickým schránkam i pre aplikačné protokoly elektronických služieb štandardom použitie *HTTPS* a *TLS*. Aj v súčasnosti však existujú prípady, kedy je prístup na stránku nezabezpečený, a to dokonca aj pri odosielaní prihlasovacích údajov používateľa. V takom prípade je jednoducho možné sieťovú komunikáciu odpočúvať a meno a heslo používateľa zachytiť.

V posledných rokoch bolo zistených viacero závažných zraniteľností protokolu *SSL/TLS*, ktoré mohli viesť ku kompromitácii zabezpečeného spojenia. Napriek tomu, že bezpečnostné záplaty boli vydané, pri testoch sa stále stretávame so servermi zraniteľnými voči *POODLE* alebo dokonca *DROWN* útoku.

Nedostatky v zabezpečení šifrovanej komunikácie možno eliminovať aktualizáciou používaných komunikačných protokolov a používaním dostatočne silných šifrovacích a hashovacích algoritmov. Vzhľadom na rýchly vývoj v oblasti IKT je potrebné sledovať aktuálne požiadavky na silu použitých kryptografických primitív. V neposlednom rade je nutnosťou reagovať na novoobjavené bezpečnostné nedostatky protokolov a ich implementácií na konkrétnych zariadeniach. Vydané bezpečnostné záplaty a konfiguračné opatrenia je potrebné nasadiť čo najrýchlejšie.

Ďalšia skupina nedostatkov sa týka nastavenia webového servera, teda softvéru spracúvajúceho požiadavky klienta a odpovede servera. Medzi chyby v konfigurácií webového servera patrí okrem mnohých iných napríklad:

- *directory listing*, teda možnosť zobrazenia a prechádzania obsahu adresára na serveri,

[1] SSL/TLS *Secure Socket Layer* resp. *Transport Layer Security* sú komunikačné protokoly zabezpečujúce dôvernosť a integritu obsahu komunikácie. Používajú ich webové prehliadače, aplikácie na prenos súborov, zabezpečenie VPN spojenia alebo VoIP komunikácie.

- ponechanie defaultných chybových hlásení servera či rôznych ladiacich funkcionalít, ktoré môžu útočníkovi poskytnúť dôležité informácie o správaní systému a použitej technológii,
- nedodržiavanie bezpečnostných odporúčaní pre použitie špecifických hlavičiek odpovedí servera (nastavenie napr. *X-XSS-Protection*, *Strict-Transport-Security* atď.)
- *local* či *remote file inclusion*.

Posledne spomenuté umožňujú útočníkovi nahrať na server súbor, obsahujúci napríklad *webshell*[2], prostredníctvom ktorého získa útočník kontrolu nad webovým serverom. Zabezpečenie a izolácia verejne dostupného obsahu od nižších systémových úrovní servera (operačný systém, databáza, ...), obmedzenie akcií, ktoré možno ako bežný internetový používateľ vykonať (napríklad nahrávanie súborov na server – ide o fotografiu na používateľov profil, alebo o spustiteľný kód?) a znalosť interakcií nutných medzi webovým serverom a podkladovými systémami sú preto významnými súčasťami opatrení zvyšujúcich bezpečnosť webových služieb a ich podkladovej infraštruktúry.

Zraniteľnosti vznikajú aj na úrovni samotnej aplikácie, v programovom kóde či logike aplikácie. Medzi nedostatky na aplikačnej úrovni možno zaradiť viaceré z najzávažnejších hrozieb pre webové služby: *SQL injekcie*, reflexný a perzistentný *Cross-Site Scripting* a iné injekcie kódu. Zraniteľnosť spočíva v tom, že vstup používateľa je bez ošetrenia postúpený na spracovanie (v prípade SQL injekcie databázou). V prípade, že obsahuje riadiace reťazce databázového či iného programovacieho jazyka, budú vykonané.

Úspešné *SQL Injection* môže viesť od získania údajov o systéme a uložených informáciách cez odcudzenie kompletného obsahu databázy až po získanie prístupu k príkazovému riadku servera, na ktorom databáza beží. V prípade nedostatočnej izolácie servera, čo, žiaľ, nie je výnimočným problémom, môže kompromitácia databázového servera viesť k ovládnutiu nezanedbateľnej časti infraštruktúry organizácie. Reflexívny a perzistentný *Cross-Site Scripting* patrí medzi útoky ohrozujúce viac používateľa ako samotný webový server či inštitúciu. Ide pri nich najčastejšie o možnosť nahrať na zraniteľnú stránku skript, ktorý sa pri návšteve stránky stiahne a vykoná na strane klienta. Prostredníctvom *XSS* možno napríklad získať používateľov identifikátor relácie uložený v *cookie*[3] a následne (po splnení ďalších podmienok) vystupovať pod jeho identitou.

Ochrana voči útokom ako *XSS* a *SQL Injection* spočíva v dôslednom filtrovaní používateľských vstupov a ich sanitáciou pred ďalším spracovaním. Napríklad, je možné obmedziť sadu znakov, ktoré je možné zadať. Ak od používateľa aplikácia vyžaduje zadanie mena a adresy, znaky ako <, >, :, ×, ÷, $ a pod. pravdepodobne nebudú potrebné. V prípade, že aplikácia má akceptovať aj nealfanumerické znaky, je potrebné ich pred použitím prekódovať tak, aby neboli interpretované programovacími alebo skriptovacími jazykmi technológií, ktoré so vstupom budú

---

[2] *Webshell* je malý program alebo skript, ktorý možno nahrať na zraniteľný server a následne otvoriť v prehliadači. Poskytuje tak webové rozhranie umožňujúce spúšťanie systémových príkazov a vykonávanie akcii ako prezeranie adresára, zmeny prístupových oprávnení, zmeny obsahu adresárov webového servera a podobne.

[3] *Cookie* je krátka správa, ktorú odosiela webový server webovému prehliadaču, ktorý ju ukladá v textovom súbore. Správa je odosielaná v každej nasledujúcej požiadavke smerujúcej na stránku príslušného servera. Cookies sa používajú na prispôsobenie vzhľadu a obsahu stránky používateľovi, a častokrát ako tzv. *Session cookies*, relačné cookies, na uloženie identifikátora konkrétneho spojenia (autentifikovaného) používateľa.

pracovať (napríklad jazyk SQL a znaky “, -- , '). Vloženie netlačiteľných, prázdnych a nulových znakov by malo byť vylúčené.

Ďalšími opatreniami je nastavenie *Secure* a *HTTPOnly* príznakov pre identifikátory spojenia. Prvý deklaruje, že cookie môže byť prenášaná iba zabezpečeným (HTTPS) kanálom, druhý obmedzuje možnosť manipulácie cookie na server, ktorý ju vydal. Server by mal vo svojich odpovediach nastavovať hlavičky *X-XSS-Protection*, aktivujúcu vstavané ochrany prehliadačov pred *XSS* útokmi, a *Strict-Transport-Security*, vyžadujúcu aby klient k stránke pristupoval výlučne zabezpečeným kanálom. *Cross-Origin Resource Sharing (CORS)* mechanizmus je ďalšou ochranou konfigurovateľnou na úrovni webového servera. Moderné prehliadače spravidla uvedené typy hlavičiek podporujú. Pre podrobnejšie informácie o zraniteľnostiach i o spôsobe ich eliminácie odporúčame [2] a [3].

Pri externých službách sa spravidla možno stretnúť s možnosťou autentifikácie. Používateľ následne vystupuje s inými privilégiami ako neprihlásený návštevník, aplikácia často rozlišuje rôzne roly používateľov s rozličnými oprávneniami a možnosťami práce so zdrojmi. Kľúčovou požiadavkou v takom prípade je, aby relácia používateľa bola korektne spravovaná a bola dodržaná bussiness logika aplikácie. Na správu relácií je z bezpečnostného hľadiska kladených viacero podmienok, ktorých vymenovanie nie je v kapacitných možnostiach príspevku. Ako príklady pravidiel, ktoré by správa spojení mala dodržať, môžeme spomenúť nasledovné:

- Po odhlásení je relácia používateľa zneplatnená a jedinečný identifikátor relácie je zrušený. Nie je možné použiť tento identifikátor v následnej požiadavke na webový server.
- Server neakceptuje identifikátor poskytnutý klientom, ak nebol generovaný týmto serverom a odoslaný v bezprostredne predchádzajúcej komunikácii s konkrétnym klientom.
- Identifikátory relácie majú dostatočnú dĺžku, komplexnosť a sú generované náhodne[4]: na základe znalosti viacerých ID nemožno predpovedať nasledujúce generované identifikátory.
- Pre daného používateľa s pridelenou rolou nie sú dostupné zdroje prislúchajúce iným používateľom s inou rolou ani odlišnému používateľovi s rovnakou rolou (pokiaľ to systém nepožaduje).
- Počas trvania relácie nie je možné, aby si používateľ zmenil jemu pridelený identifikátor na identifikátor prislúchajúci inej relácii iného používateľa a pokračoval v používaní aplikácie s jeho oprávneniami (tzv. únos spojenia, angl. *session hijacking*).
- Pre každú individuálnu požiadavku na webový server by mal byť vygenerovaný jedinečný token (nonce), aby sa zabránilo opakovanému odoslaniu totožnej požiadavky z totožného spojenia (viazaného na používateľa prostredníctvom identifikátora relácie). Ide o opatrenie limitujúce ďalší typ útoku zameraného na používateľa - *Cross-Site Request Forgery.*

Na záver kapitoly, venovanej zraniteľnostiam pozorovaným pri externých pentestoch, dávame do pozornosti kontrolný zoznam na overenie bezpečnosti webových aplikácií [4]. Bol vypracovaný pracovníkmi CSIRT.SK a stručne sumarizuje najdôležitejšie bezpečnostné aspekty pri vývoji a prevádzke webových stránok. Je možné ho využiť pri vykonávaní interného auditu bezpečnosti webových aplikácií a webových stránok.

[4] Identifikátory sú vytvorené dostatočne robustným pseudonáhodným generátorom.

3. Interný penetračný test

Interné penetračné testy sa zameriavajú na hľadanie bezpečnostných medzier v internej infraštruktúre testovanej organizácie. V prípade inštitúcií verejnej správy ide o rozsiahle siete tak po stránke počtu používateľov, ako aj z pohľadu množstva použitých technológií a rozličných IS či z hľadiska fyzického alebo logického rozloženia. Z toho dôvodu sú vo všeobecnosti interné testy oveľa komplexnejšie ako testy externé. Dotýkajú sa množstva technologických oblastí, od nastavenia Windows domény cez správu sieťových prvkov či architektúru siete až po politiku hesiel, ktorá sa v organizácii používa. Zovšeobecniť priebeh testu je zložité. Výsledok každého jednotlivého kroku určuje smer, ktorým sa špecialista, vykonávajúci pentest, ďalej vydá. Cieľom je kompromitácia čo najväčšej časti infraštruktúry.

V nasledujúcich podkapitolách detailnejšie popíšeme niektoré z častých vektorov útoku, ktoré boli pri interných testoch úspešné. Najčastejšie boli použité viaceré postupy, ktorých kombinácia viedla k ovládnutiu siete. Postupy boli použité v infraštruktúrach s nasadenými adresárovými službami Windows domény, v prostredí *MS Windows Active Directory*.

Vzhľadom na komplexnosť a citlivú povahu problematiky nie je cieľom príspevku podať vyčerpávajúci popis exploitácie. Ďalšie informácie ohľadne zraniteľností a bezpečnostných opatrení, ako aj odkazy na zdroje k danej problematike možno nájsť na stránke https://www.csirt.gov.sk. Množstvo odporúčaní, ktoré eliminujú riziká spomenuté v predošlých i nasledujúcich kapitolách, je obsiahnutých v *Metodike pre zabezpečenie organizácií v oblasti informačnej bezpečnosti* [5]. Tento materiál bol vypracovaný v spolupráci CSIRT.SK a Úradu podpredsedu vlády pre investície a informatizáciu. Metodika obsahuje požiadavky na organizáciu rozdelené do viacerých oblastí: popri zabezpečení webových služieb či pracovných staníc sa venuje aj zabezpečeniu internej infraštruktúry, bezpečnostným aspektom vývoja a nasadenia IS či administratívnej a organizačnej bezpečnosti.

2.1. Exploitácia známych zraniteľností

Jedným z najčastejších spôsobov, ktorým pri testoch CSIRT.SK získava prvotný prístup k IS, je kompromitácia zariadenia so zraniteľnými verziami OS alebo aplikácií. Na mnohé zraniteľnosti sú verejne dostupné exploity, ktorých použitím možno získať príkazový riadok zasiahnutého zariadenia.

Exploit často vedie k získaniu oprávnení na úrovni systému. V takom prípade nasleduje vytvorenie (lokálneho) používateľského účtu a jeho zaradenie do skupiny lokálnych administrátorov.

V prípade, že získané privilégia sú príliš nízke, možno sa pokúsiť o ich eskaláciu a pokračovať ako v predchádzajúcom prípade.

Scenáre konania po získaní prístupu k prvému zariadeniu sa líšia od prípadu k prípadu. No i keby sa nepodarila eskalácia privilégií a na systéme máme práva iba na čítanie, môžeme vykonať prieskum prostredia, zobraziť sieťovú konfiguráciu zariadenia, prezerať adresárovú štruktúru, či použiť inštalované nástroje na získanie ďalších informácií.

2.2. Získanie hesiel z pamäte a z MS Cache zariadenia

Po získaní prvotného prístupu k zariadeniu je za istých okolností (ktoré podľa našej skúsenosti bývajú splnené) možné získať heslá používateľov, ktorí sa v minulosti na zariadenie prihlasovali.

MS Windows totiž ukladá prihlasovacie údaje používateľov vo viacerých formách a na viacerých lokalitách, v závislosti od typu prihlásenia, požiadaviek na dostupnosť služieb a kompatibilitu s inými systémami a pod.

Prístupové dáta používateľov, ktorí sa autentifikovali na určité zariadenie, sa ukladá v jeho registroch (hoci nie vo forme meno:heslo) - v tzv. *MSCache*. Táto funkcionalita zabezpečuje, že offline konzolová autentifikácia týchto používateľov je možné aj v situácii, keď server (radič domény) nie je k dispozícii. *Cache* môže obsahovať 0 až 50 prihlasovacích údajov. Ak je parameter, určujúci počet zapamätúvaných prihlásení, nastavený na vysokú hodnotu, môže útočník v neskoršej fáze získať poverenia používateľov ich skopírovaním z registra a ich obnovou[5]. Prihlasovacie údaje pretrvávajú zapísané v registroch aj po reštarte zariadenia.

Ďalším miestom, kde sú na bežiacom systéme uložené prihlasovacie dáta účtov, je pamäť RAM. Konkrétne, do pamäte procesu *lsass*[6] sú ukladané údaje najmä z prihlásení vykonaných prostredníctvom *Vzdialenej pracovnej plochy* (angl. *Remote Desktop*). Prihlasovacie údaje používateľov (v otvorenom tvare) je možné získať vytvorením výpisu (tzv. *dump*) vyrovnávacej pamäte procesu *lsass* a jej analýzou. K dispozícii je na tento účel viacero voľne dostupných nástrojov, umožňujúcich lokálne aj vzdialené získanie autentifikačných dát.

Týmto spôsobom je nezriedka možné získať prihlasovacie údaje administrátora. Ak exploitovaným systémom je server, správcovo heslo bude pravdepodobne v pamäti kvôli vykonávaniu správy. Ak je kompromitovaná stanica bežného používateľa, je pravdepodobné, že prihlásenie administrátora bude v pamäti od času pridania stanice do domény. Prihlasovacie údaje totiž zotrvávajú v pamäti *lsass* procesu až do najbližšieho rebootu zariadenia, čo v prípade serverov môžu byť mesiace až roky.

Už s jediným kontom bežného doménového používateľa je možné získať zoznam ďalších doménových používateľov, doménových politík atď. Následne možno pokračovať slovníkovým útokom na heslá používateľov a pod.

Predpokladom na úspech popísaného postupu je, že používateľ, pod akým k systému pristupujeme, má aktívne *DEBUG*[7] privilégium, následkom čoho môže výpis pamäte procesu *lsass* získať. Zakázanie *DEBUG* práva preto túto hrozbu eliminuje. Výpis pamäte procesu však môže vytvoriť každý proces, bežiaci pod kontom *System*. V prípade, že útočník získa prístup k zariadeniu pod vysoko privilegovaným *System* kontom, bude schopný heslá z pamäte získať a samotné zakázanie *DEBUG* privilégia nepostačuje.

Ďalším odporúčaním je obmedzenie počtu hesiel, ktoré sa do vyrovnávacej pamäte ukladajú, na 2: prvým bude konto fyzickej stanice, ktorá sa voči doméne tiež autentifikuje, a druhým bude konto používateľa zariadenia, aby sa mohol prihlásiť aj v prípade nedostupnosti radiča domény.

---

[5] Heslo je pozmenené tzv. soľou (angl. *salt*), odvodenou z používateľského mena a hashované. Hash je zašifrovaný heslom, ukladaným v ďalšom registri. Obnovenie hesla z dešifrovaného hashu môže byť časovo náročné, ale nie je nemožné.

[6] Lsass.exe, teda *Local Security Authority Subsystem Service* je proces OS Windows, zodpovedajúci za vynucovanie bezpečnostných politík. Overuje prihlasovanie používateľov k Windows systému, riadi zmeny hesiel, vydávanie prístupových tokenov. Záznamy o uvedených akciách ukladá do Windows Security logu.

[7] DEBUG privilégium umožňuje používateľovi pripojiť k procesu alebo jadru OS debugger. Vývojár programu potrebuje mať toto privilégium, pretože mu umožňuje ladiť program. V produkčnom prostredí by DEBUG malo byť v maximálnej možnej miere obmedzené.

2.3. Získanie hesla lokálneho administrátora

Jednou z metód, používaných kedysi na správu staníc, je zápis doménových politík do súboru dostupného doménovým používateľom. Štandardne ide o súbor *Groups.xml*. Obsahovať môže účet lokálneho administrátora pracovných staníc spolu s heslom, hashovaným v reverzibilnom formáte. Kompromitácia jedinej stanice a prečítanie tohto hesla vedie ku kompromitácii všetkých staníc, ktoré daný administrátor spravuje. Z tohto dôvodu Microsoft tento spôsob správy staníc už viac ako 10 rokov neodporúča. Žiaľ, skúsenosti z praxe ukázali, že sa s ním možno stretnúť aj dnes.

Získanie administrátorského prístupu otvára možnosť získania výpisu pamäte procesu z viacerých zariadení, čím sa zvyšuje pravdepodobnosť, že na niektorej nájdeme aj heslo doménového administrátora. Na staniciach možno nájsť citlivé údaje o systémoch používaných v organizácii, interné dokumenty, emaily, návody na prístup a použitie k ďalším nástrojom či aplikáciám, ktoré inštitúcia používa, atď.

2.4. Nedostatočná segmentácia siete a oddelenie rolí

Jedným z nedostatkov, ktorý významne uľahčuje potenciálnym útočníkom prístup k interným systémom, je nedostatočná segmentácia siete a separácia rolí jednotlivých častí IS. Nie výnimočne sa stretávame so servermi umiestnenými v spoločnom segmente s používateľskými stanicami či s webovým serverom lokalizovaným priamo v internej sieti, nie v demilitarizovanej zóne. Bežní používatelia nebývajú dostatočne separovaní od používateľov s privilegovaným či dokonca administrátorským prístupom. Oprávnenia bežných používateľov môžu byť navyše nastavené až príliš voľne, umožňujúc prístup (hoci iba na sieťovej, nie správcovskej úrovni) k prostriedkom, ku ktorým by prístup mal byť kontrolovaný. Obmedzenie prístupových oprávnení používateľských účtov ako aj účtov služieb by malo byť implementované na princípe *Least privilege*: kontu prideliť minimálnu množinu nutných oprávnení.

Je potrebné oddeľovať používateľov od administrátorov a aj v rámci skupín zamestnancov separovať roly a na základe nich prideľovať nutné oprávnenia. Osobitnou skupinou sú administrátorské účty, používané na správu serverov či pracovných staníc. Mali by byť nastavené tak, aby kompromitácia jedného administrátorského konta neviedla ku kompromitácii všetkých ostatných prvkov. Napríklad, účet na správu domény by mal byť iný ako účet pre správu mailového servera alebo ako účet na administráciu bežných pracovných staníc.

2.5. Kompromitácia používateľov a domény

Po získaní prihlasovacích údajov prvého používateľa (nie nutne administrátora) postupujeme pri penteste ďalej do siete: pokúšame sa o autentifikáciu na ďalšie zariadenia, snažíme sa o získanie ďalších prístupových informácií a enumerujeme existujúcich používateľov. Ako sme už okrajovo spomenuli, na získanie zoznamu používateľov domény[8] stačí jediný účet. Ba čo viac, za určitých okolností je možné získať hashe doménových používateľov alebo dokonca heslá v otvorenom tvare [5].

Active Directory podporuje rôzne spôsoby autentifikácie. V prípade, že na sieti nie je používaná autentifikácia protokolom *Kerberos*, pravdepodobne je použitá *NTLM* autentifikácia. Aby bol použitý protokol *Kerberos* (na príslušne

[8] V závislosti od typu konta získaný zoznam nemusí byť kompletný

konfigurovaných zariadeniach, ktoré tento typ autentifikácie podporujú), musí byť v autentifikačnej požiadavke použité doménové meno požadovaného prostriedku. V prípade, že volanie obsahuje IP adresu, bude použitá *NTLM* autentifikácia. *NTLM* hashe, či už zachytené zo sieťovej komunikácie, alebo získané z databázy *SAM (Security Account Manager)* je možné pri v súčasnosti dostupnom výpočtovom výkone zlomiť, najmä pri nedostatočnej komplexnosti hesla.

Tu sa dostávame k neustále platnému výroku: *Reťaz je len taká silná, ako jej najslabšie ohnivko*. Povestným najslabším článkom zostáva človek-používateľ alebo správca systému. Ak si používateľ nastaví slovníkové heslo, teda heslo pozostávajúce zo slov bežného jazyka, s vysokou pravdepodobnosťou ho bude možné prelomiť pomocou vopred vypočítaných hashových tabuliek. Platí to i vtedy, ak je výraz viac alebo menej pozmenený použitím veľkých a malých písmen alebo štandardných substitúcií za čísla alebo špeciálne znaky (1 či ! namiesto l, 3 namiesto E a pod.) Administrátor významne ovplyvňuje bezpečnosť systému nastavením politiky hesiel: požiadavky na dĺžku hesla, použitie alfanumerických znakov, kontrola na prekrývanie hesla s používateľským menom a podobne.

Bezpečnostné politiky riadia aj spôsob ukladania hesiel v databáze domény: štandardne sú ukladané *NTLMv2* hashe hesiel. (*LM* formát, ktorého rozbitie je triviálne, nie je od Windows Vista používaný na ukladanie hesiel.) Je však možné vynútiť ukladanie hesiel v reverzibilnom formáte. Vyžadovať to môžu napríklad niektoré aplikácie, používajúce autentifikačný protokol *CHAP (Challenge-Handshake Authentication Protocol)*. Vo výsledku to spôsobí, že heslá na serveri sú ukladané v reverzibilnom formáte a je možné ich získať s minimálnou námahou. Takáto politika by preto nikdy nemala byť nastavená plošne pre všetkých doménových používateľov.

Z vyššie uvedených niekoľkých príkladov je zrejmé, že často je možné ovládnuť IS vďaka nevhodne zvolenému heslu a s prispením konfiguračných nedostatkov doménových bezpečnostných politík. Vstavané bezpečnostné mechanizmy a východiskové nastavenia najnovších verzii operačných systémov obsahujú mnohé ochrany. Tie však nesmú byť vypnuté alebo znefunkčnené v záujme jednoduchosti správy systému alebo pohodlia používateľov. Osobitnou kapitolou je konfigurácia diverzifikovaných prostredí, kedy je potrebné na nezanedbateľne dlhý čas zabezpečiť spätnú kompatibilitu starších a novších informačných systémov. V podobných, nezriedkavých, prípadoch je potrebná osobitná pozornosť, aby zachovanie funkčnosti neotvorilo dvere útočníkovi. V prípade extrémne starých systémov s ukončenou podporou výrobcu (čo znamená, že na ne nie sú vydávané bezpečnostné záplaty a aktualizácie, teda sú z princípu zraniteľné a navždy budú) je vhodná ich izolácia do samostatných systémov za sieťovým firewallom a zavedenie dôsledného riadenia a monitorovania prístupu.

2.6. Nedostatočné zabezpečenie prístupu na úrovni sieťovej vrstvy

Všetky vyššie spomínané scenáre predpokladajú, že sa ako útočník dokážeme pripojiť do siete subjektu, získať IP adresu a začať so získavaním informácií o sieti.

Pri vykonávaní penetračných testov sa stretávame s absenciou opatrení, ktoré by zabránili odpočúvaniu komunikácie na sieti. V kombinácii s nedostatočnou segmentáciou siete to vedie k možnosti zachytávať prevádzku nielen zo segmentu používateľov, ku ktorému sa štandardne pripájame, ale aj z iných podsietí. Je možné vykonať útok *Man-In-The-Middle*, kedy útočník presmeruje prevádzku cez svoje zariadenie a funguje ako prostredník medzi komunikujúcimi stranami. Prechádzajúcu prevádzku môže nielen čítať, ale aj meniť. V praxi to vedie k možnosti zachytiť

autentifikačné údaje, ktoré sú nezriedka odosielané v nešifrovanej forme (*HTTP* autentifikácia k intranetu, autentifikácia *Pripojenia vzdialenej pracovnej plochy* a odchytenie hesla prihlasujúceho sa administrátora, *LDAP* či *SMTP*), riadiace reťazce *SNMP* (tzv. *community strings*) či dokonca NTLM autentifikáciu (spomeňme si na fakt, že *NTLM* autentifikáciu je možné zlomiť, najmä v prípade použitia nedostatočne komplexného hesla), prípadne heslo na pripojenie k *RADIUS* serveru.

Autentifikácia pomocou protokolu *Kerberos* je odolná voči podobnému útoku, ani pri zachytení nie je jej obsah možné použiť na získanie prístupu. Ako sme už však spomenuli, na to, aby bol použitý protokol *Kerberos*, musí byť vo volaní doménové meno požadovaného prostriedku. V prípade, že prostriedok, voči ktorému sa autentifikujeme, voláme jeho IP adresou, bude použitá *NTLM* autentifikácia

Informácie, ktoré možno získať odpočúvaním siete, sa rôznia podľa nasadených technológií a stupňa použitého zabezpečenia. V ideálnom prípade je všetka komunikácia šifrovaná a pri jej zachytení je útočníkovi neužitočná. Prax však ukazuje, že nastavenie šifrovania aj v prípade, že ho zariadenia/služby podporujú, nie je pravidlom.

Obrana voči prieniku do siete a následnému odpočúvaniu komunikácie pozostáva z viacerých úrovní: Na obmedzenie fyzického prístupu do siete je vhodné implementovať štandard 802.1X. Ide o IEEE štandard sieťového protokolu pre *Port-based Network Access Control (PNAC),* ktorá poskytuje možnosť autentifikácie zariadení snažiacich sa o pripojenie k lokálnej sieti. Cieľom je zabrániť v pripojení ľubovoľného zariadenia (napríklad súkromného notebooku) k internej sieti. Hoci existujú mechanizmy na obídenie 802.1X, minimálne docielime spomalenie útočníka. Pri penetračných testoch sme sa stretli s malým počtom inštitúcií, ktoré takúto kontrolu prístupu uplatňujú. Znamená to, že ľubovoľné zariadenie je možné sieťovým káblom, „požičaným“ napríklad z multifunkčnej tlačiarne na chodbe, pripojiť k internej sieti a začať so skenovaním, zberom dát, ...

Dostupnosť jednotlivých častí siete by mala byť riadená aj pre legitímnych používateľov. Nie je dôvod, aby bol z adresného rozsahu prideleného bežným používateľom dosah na podsieť s bezpečnostnými alebo sieťovými prvkami (firewally, IDS/IPS, monitoring, prepínače, smerovače, ...).

Oddelenie na úrovni linkovej vrstvy možno implementovať pomocou *VLAN* resp. *PVLAN* (*Private Virtual LAN*). *VLAN* umožňuje rozdelenie používateľov do manažovateľných skupín, aj keď nie sú pripojení k jednému prepínaču. V jednej kancelárii tak môže byť administrátorský počítač aj počítače používateľov pripojené k jednému switchu, avšak zariadenie administrátora bude patriť k inej virtuálnej sieti ako používatelia, na sieťovej úrovni budú oddelené. Na základe *VLAN* je možné definovať pravidlá prístupu k prostriedkom, či už sú to servery, manažmentové rozhrania sieťových prvkov či tlačiarne. *PVLAN* zachádza v separácii ďalej: koncept spočíva v definovaní portov prepínača, ktoré môžu komunikovať výlučne s daným uplinkom. Prevádzka medzi segregovanými portami je transparentná: zariadenie pripojené k jednému portu nevidí komunikáciu na inom izolovanom porte, hoci zariadenia môžu patriť do rovnakej *VLAN* a teda do rovnakej IP podsiete. Opatrenie zabraňuje v odpočúvaní komunikácie v rámci jednej *LAN*.

3. Záver

V príspevku sme zhrnuli výsledky pozorovaní z vykonávania interných a externých penetračných testov, uskutočnených útvarom CSIRT.SK v posledných 3 rokoch. Zdôraznili sme, že scenáre pri interných testoch sa výrazne líšia od prípadu k prípadu, nebolo preto cieľom poskytnúť presný popis prieniku do systému. Tak

v prípade nedostatkov webových aplikácii, ako aj pri popise zraniteľností a konfiguračných medzier vnútornej infraštruktúry sme uviedli opatrenia, ktorých nasadenie môže hrozbu kompromitácie zmierniť.

Dúfame, že komplexný pohľad na bezpečnosť a systematické riešenie všetkých jej úrovní sa postupne stane bežnou praxou. Rovnako veríme, že bezpečnosť vyvíjaných aplikácií bude štandardnou požiadavkou objednávaných IS. Dodávané IS by, najmä v prípade štátnych projektov, nemali byť prebraté, pokiaľ nebude zabezpečená a podľa možností dokázaná primeraná úroveň ich bezpečnosti. Kladenie väčšieho dôrazu na bezpečnostné aspekty od návrhu, vývoja až po nasadenie informačných systémov by do značnej miery prispelo k tomu, aby opísané zraniteľnosti neboli v budúcnosti zneužiteľné ani pri penteste, ani – a to predovšetkým – v prípade reálneho útoku.

Literatúra

[1] Výnos o štandardoch pre ISVS [č. 55/2014 Z. z.].
Dostupné na Internete, http://www.informatizacia.sk/standardy-is-vs/596s

[2] Domovská stránka projektu OWASP: https://www.owasp.org/index.php/Main_Page

[3] D. Studdart, M.Pinto: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition. Wiley, 2 edition, 2011.

[4] Kolektív CSIRT.SK: Kontrolný zoznam pre bezpečnosť webových aplikácií. 2015.
Dostupné na Internete, https://www.csirt.gov.sk/doc/Checklist.pdf

[5] Kolektív CSIRT.SK: Metodika pre zabezpečenie organizácií v oblasti informačnej bezpečnosti. 2016.
Dostupné na Internete, http://www.csirt.gov.sk/doc/Metodika_OPII_vRC1.0.pdf

[6] S. Metcalf: Dump Clear-Text Passwords for All Admins in the Domain Using Mimikatz DCSync. 22.11.2015.
Dostupné na Internete, https://adsecurity.org/?p=2053