Penetrační testování

Úvod do penetračního testování

Penetrační testování může být zcela samostatný proces, pokud se bavíme o penetračním testování firemní infrastruktury nebo lidského faktoru. Doporučuje se však zařadit i do poslední fáze testování SW, za funkční testy a zátěžové testy.

Penetrační testování je v současné době velmi diskutované téma, zvláště v kruhu ICT manažerů a top managementu firem. Dle výsledků nedávné studie Cost of Data Breach, která se zabývala výškou finančních ztrát v případě odcizení firemních dat, je průměrná ztráta při bezpečnostním incidentu, při kterém dojde k odcizení dat, 25,2 milionu korun českých.1

Právě penetrační testování je odpovědí na otázku: Jak minimalizovat riziko tak vysoké ztráty. Typickým dilematem každého přemýšlivého (nejen) ICT manažera pak je, zda-li má zaplatit okolo 300 000 tisíc korun (což je běžná cena v ČR za penetrační testování od externí firmy, samozřejmě s výraznou odchylkou oběma směry v závislosti na hloubce testování) a tím výrazně snížit riziko odcizení dat, anebo tyto peníze ušetřit a riskovat mnohonásobně vyšší ztrátu v případě napadení.

Nicméně je třeba zdůraznit, že ani po negativním výsledku penetračních testů nejsou firemní systémy 100% bezpečné a neproniknutelné. Jak uvádí jedna ze známých internetových stránek o bezpečnosti: „Security is a process – not a destination"2, tedy volně přeloženo: Bezpečnost sítě není cíl, ale proces. Testování je totiž vždy limitováno přidělenými prostředky, ať už se jedná o čas, finance nebo odpovědný personál.

Hlavním cílem této práce je tedy popsání vybraných metod a fází penetračního testování způsobem, který bude pochopitelný i pro čtenáře bez technického vzdělání. Popsána bude i aplikace metod na reálném firemním prostředí, přičemž bude proveden i návrh vhodných opatření proti nalezeným hrozbám.

Jedním z vedlejších cílů práce pak bude objasnění tématu sociálního inženýrství, včetně ukázky jeho praktického využití.

Limitující podmínky

Práce se zabývá pouze nejčastěji používanými metodami penetračního testování a jejím smyslem není úplný výčet všech metod. Vzhledem k nezměrnému množství metod to ani není možné. Zaměření práce bude na externí a interní testy infrastruktury firmy. Nebudeme se zabývat testem firemních webových stránek, aplikací, mobilních aplikací ani Wi-Fi sítí.

V práci bude použito mnoho termínu a zkratek z oblasti síťové komunikace. Tyto výrazy budou vždy stručně vysvětleny v poznámce pod čarou, aby měl čtenář představu, co daný výraz znamená. Pro hlubší pochopení síťové problematiky však doporučuji nastudovat knihu Cisco Certified Network Associate (CCNA) od autora Todda Lammlea. Pro plné pochopení práce se tedy předpokládá znalost síťové infrastruktury a jejich protokolů.

Specifické výrazy, které nesouvisí se sítí, budou vysvětleny v následující kapitole.

Slovník

DMZ – (DeMilitarized Zone) podsíť oddělená z bezpečnostních důvodů od vnitřní sítě, ve které jsou umístěné služby, které jsou k dispozici z celého internetu

ERP – (Enterpriste Resource Planning) IT systém, ve kterém firma řídí svoje hlavní procesy a oblasti,jako např.: nákup, prodej, plánování, marketing, finance, řízení zásob

CRM – (Customer Relationship Management) IT systém pro řízení vztahu se zákazníky. Obsahuje všechny relevantní informace k jednomu zákazníkovi (osobní informace o zákazníkovi, objednané služby atd.)

Domain Controller – Server, který odpovídá za autentizaci (ověření přihlašování, oprávnění na soubory) uživatelů v doméně MS Windows

Access point – vysílač Wi-Fi signálu

Postup a metody používané pro penetrační testování

V rámci správy ICT provádíme penetrační testování pro ověření současného stavu bezpečnosti. Má tedy logicky smysl provádět testy pouze na konceptuálně zavedeném bezpečnostním systému firemního IT. Firma musí mít vyřešenou otázku zabezpečení infrastruktury, tedy správně nastavené a nakonfigurované jednotlivé síťové prvky, jako jsou routery, switche nebo firewall. Dále musí mít firma zabezpečené koncové stanice, typicky pomocí antiviru a doménových bezpečnostních politik. Neposlední oblastí, kterou by měla firma mít z hlediska bezpečnosti pokrytou, je například oblast webových serverů firmy a uživatelských práv.

Penetrační testování nejčastěji využívá právě chyb v zabezpečení, které jsou způsobeny špatným návrhem sítě nebo systému, špatnou konfigurací systému nebo síťových prvků, chybami v programování nebo selháním lidského faktoru.

K tomuto účelu máme obecný postup, který se může v detailech lišit, nicméně velmi základně nám ho znázorňuje následující graf.

Obrázek 1 - Fáze penetračního testování

Zdroj: Vlastní zpracování

V první přípravné fázi probíhají schůzky s klientem (typicky firmou), kde se dohaduje rozsah, cena, časování a další důležité náležitosti budoucího testování. Po podpisu smlouvy můžeme přistoupit k testování samotnému.

V druhé fázi dochází k otestování všech dohodnutých částí systému, sítě, nebo jiných zranitelných částí firmy. Můžeme si představit, že nám bude v pozici testované firmy v první fázi nabídnuto testovací menu, kde místo jídel budeme vybírat moduly, ze kterých si složíme svoje několikachodové testování.

Právě tak, jako každá restaurace má trochu jiné menu, tak i jednotlivé firmy, zabývající se penetračním testováním, nabízí jiné moduly.

Podle Seleckého3 jsou hlavní moduly **externí testy firemních sítí, interní testy firemních sítí, testy bezdrátových sítí a testy webových aplikací**. Každý z těchto modulů je pak složen právě ze 4 fází:

a) Definování cíle a rozsahu testu
b) Sběr dat
c) Skenování a exploitace
d) Report

Externí testování simuluje útok z internetu, stejně jako by to udělali hackeři. Cílem je získat neoprávněný přístup k datům anebo firmu jinak poškodit.
Interní testy simulují útok z vnitřní sítě, tedy typicky z řad pomstychtivých zaměstnanců, nebo to bývají útoky za účelem průmyslové špionáže. Tento druh testu může také simulovat útočníka, kterému se podařilo do interní sítě dostat.

Co se týče rozdělení každého modulu na fáze, tak osobně preferuji spíše rozdělení, ve kterém je fází 5 a skenování je samostatná kategorie, stejně jako preferuji osamostatnění exploitace. Mnou upravené rozdělení fází ilustruje následující graf:

Obrázek 2 - Detailní fáze testování

Zdroj: Vlastní zpracování

Takto upravené rozdělení skenování a exploitace totiž velmi úzce souvisí i s dvěma pojmy, které se často pletou. Jedná se o pojmy penetrační testování a vulnerability assesment, neboli česky ohodnocení zranitelností. Tyto dva pojmy jsou částečně propojené, ale penetrační testování klade důraz na získání co největšího přístupu, zatímco ohodnocení zranitelností se zabývá spíše objevováním zranitelných oblastí. Tester provádějící ohodnocení zranitelností se zastaví právě ve chvíli, kdy končí objevování zranitelností (skenování) a začíná jejich exploitace, neboli využívání. Tester provádějící penetrační test jde tak daleko, jak může v rámci dohodnuté smlouvy. Nalezené zranitelnosti se tedy snaží využít a kompromitovat tak síť nebo systém.DCIT a.s., jedna z největších Českých firem zabývajících se penetračními testy, má oproti Seleckého přístupu navíc moduly pro **testování mobilních aplikací** a
vyčleněnou kategorii pro **zátěžové testy**.
Často se také můžeme setkat s moduly pro **otestování fyzické bezpečnosti,** kde řešíme možnost náhodného či úmyslného poškození nebo zneužití jednotlivých zařízení nezbytných pro chod sítě a společnosti a také s modulem pro testování lidského elementu – **sociálním inženýrstvím**.
Přehled testovacích modulů znázorňuje následující graf:

Obrázek 3- Testování

Zdroj: Vlastní zpracování

Poté, co jsme si v přípravné fázi vybrali moduly, které chceme otestovat a v testovací fázi jsme je otestovali, přichází fáze reportovací. Přesto, že v každém testu je reportovací fáze, je třeba mít na paměti, že tyto dílčí reporty slouží hlavně k vytvoření jednolitého závěrečného reportu, který bude předán zákazníkovi. V této zprávě nesmí chybět shrnutí rozsahu a průběhu testování, celkové výsledky testů a všechny nalezené zranitelnosti a návrh na jejich odstranění.
Některé firmy nabízejí po reportovací fázi ještě další volitelnou fázi nazývanou re-test. Tato fáze probíhá cca měsíc po reportu výsledků testů a doporučení na opravu. Tento čas má firma na záplatu nalezených zranitelností a když je s opravami hotova, provede se testování ještě jednou, čímž se zaručí, že opravy byly aplikovány správně a zranitelností již nelze využít.
Musíme mít na paměti také to, že pokud je testování špatně provedeno, může způsobit zahlcení sítě nebo dokonce pád běžících systémů. V nejhorším případě může být výsledkem testování právě to, čemu se snaží firma zabránit – kompromitace systému neověřenými útočníky. Může k tomu dojít například v důsledku neodstranění administrátorských účtů, které byly při testování vytvořeny, nebo neodinstalováním nebezpečného software (například trojského koně). Nicméně v praxi málokterá firma nechá testování dojít tak daleko, aby nechala testery nainstalovat trojského koně na některé stanice. Pokud tento druh testování probíhá, tak je to řešeno kontrolovanou formou a to sice klonováním uživatelské stanice a testováním v izolovaném prostředí.
Konzultant této práce také přiznává, že i když je testování provedeno dobře, může způsobit pád systému. Důvodem bývá chybná a nečekaná konfigurace některých zařízení. Patří to zkrátka mezi drobná rizika spojená s testováním.

Přípravná fáze

Dříve, než se pustíme do samotného testování, je třeba provést důkladné přípravy. Stejně jako má svůj životní cyklus proces vývoje SW, tak i při penetračním testování se můžeme řídit pomocí podobné šablony.
Přípravnou fázi lze trochu přirovnat k tzv. inception fázi procesu vývoje SW.
Dochází v ní ke schůzkám s klientem, kde se snažíme zjistit:
a) Proč klient požaduje testování – Byl někdy nedávno napaden? Jaká má citlivá data, která potřebuje chránit? chránit? Jde firmě jen o potvrzení na papíře, aby vypadala dobře?
b) Co chce klient testovat – www stránky, síť, webovou aplikaci, VPN připojení atd.
c) Jak vypadá topologie sítě – např.: je síť segmentovaná? Jak síť komunikuje s venkovní sítí? Je zavedená DMZ?
d) Jaké jsou požadavky na zabezpečení sítě – Jaký dopad by měla na firmu případná kompromitace dat, emailů, neoprávněné užívání firemních systémů atd..
e) Jakým způsobem klient síť využívá - Kolik osob se sítí pracuje najednou? Jak se určuje oprávnění? Co může síť zpomalit / zahltit?
f) Budou IT zaměstnanci vědět o tom, že testování probíhá?
g) Byla firma už někdy testována?

h) Jakou úroveň zabezpečení firma má?

Odpovědi na tyto otázky by nám měli pomoci vytvořit prioritizovaný seznam prvků, na které se musíme při testování zaměřit. Zároveň bychom si na základě tohoto seznamu měli být schopni určit tzv. "scope" testování.

Scope

Scope nám určuje, do jaké hloubky a v jakém rozsahu můžeme testovat. Identifikuje nám zařízení, systémy, sítě a zahrnuté zaměstnance.

Příkladem může být dohoda, kde budeme testovat pouze WWW stránky firmy a budeme se snažit dostat na firemní server k datům zaměstnanců z vnější sítě bez jakékoliv znalosti o síti a bez přidělení práv. Mimo scope pak bude vše ostatní, tedy například testování z vnitřní sítě nebo testování firemní databáze.

Stanovení hloubky testování je velmi klíčový proces, který však může stanovit téměř výhradně na základně tacitních znalostí, tj. Znalost, která se nedá získat jinak, než na základě životních zkušeností.

Časování

Penetrační testování může mít vážné dopady pro síť, na které probíhá. Během kick-off meetingů tedy musíme stanovit časovou posloupnost a trvání penetračního testování. Pro management testované firmy bude často jednou z hlavních snah, aby během průběhu testů nebyl nijak ovlivněn business a každodenní aktivity. Z toho vyplývá, že testy musí probíhat pouze v některých částech dne a může se vyskytnout konflikt mezi potřebou všechno otestovat včas a zákazem testovat síť během velké business zátěže nebo během kritické doby dne, kdy probíhá například zálohování.

Některé druhy testů mohou generovat nadměrné množství síťové aktivity a shodit tak některé běžící systémy. Příkladem může být DoS (Denial of Service) útok, který spočívá v tom, že se na cílový WWW server posílá větší množství požadavků, než na které je server dimenzován a tím se přehltí a odmítne odpovídat na standardní požadavky uživatelů. Dočasným vedlejším výsledkem testování tedy může být např. nefunkční web firmy nebo nepřidělení IP adresy nově připojenému PC do sítě. Pokud takový risk firma není ochotná podstoupit, můžou být v důsledku toho některé systémy nebo sítě vyřazeny z testování.

Příkladem katastrofálního načasování testování může být již zmíněný Denial of Service test na škole v den, kdy mají probíhat online zkoušky.

Je tedy zřejmé, že je vitální mít schválené načasování od managementu firmy předtím, než s testováním začneme.

Non-disclosure agreement

Dalším krokem, bez kterého není v žádném případě možné testování provádět, je podpis tzv. NDA (non-disclosure agreement), nebo-li Dohody o mlčenlivosti.

Jedná se o dokument, ve kterém vám zákazník dává svůj souhlas k nabourání se do jeho systému v dohodnutém rozsahu a zároveň se poskytovatel zavazuje k tomu, že nevyzradí ani jinak nezneužije nebo nepublikuje nalezená data a informace o síti. Velmi důležitá je část věty: „v dohodnutém rozsahu". Během testování se nám totiž může stát, že se necháme příliš unést, nebo některý z členu testovacího týmu projeví iniciativu a otestuje (a často i prolomí) některou část sítě nebo systému, která není v dohodnutém "scope".

Pokud by poskytovatele v takovém případě zákazník zažaloval nebo pokud bychom vůbec nepodepsali NDA, octneme se před soudem, kde budeme čelit obvinění z porušení zákona č.40/2009 Sb., trestního zákoníku, část druhá, zvláštní část, § 230 Neoprávněný přístup k počítačovému systému a nosiči informací, který dále říká:

„Kdo překoná bezpečnostní opatření, a tím neoprávněně získá přístup k počítačovému systému nebo k jeho části, bude potrestán odnětím svobody na jeden rok až pět let, zákazem činnosti, propadnutím věci, jiné majetkové hodnoty nebo peněžitou pokutou v závislosti na závažnosti přečinu.

Odnětím svobody na tři až osm let bude pachatel potrestán,

a) způsobí-li činem uvedeným v odstavci 1 nebo 2 škodu velkého rozsahu, nebo

b) získá-li takovým činem pro sebe nebo pro jiného prospěch velkého rozsahu."4

Tato dohoda by měla být sepsána i v případě, že testování provádějí zaměstnanci testované firmy.

Posledním detailem, který je třeba si ujasnit, je forma výstupního reportu.

Osobně mi přijde nejrozumnější finální report pro management požadovat v pdf formě, která je uživatelsky nejpřívětivější. Naopak technický report, který je určen pro správce firemního IT, preferuji ve formě HTML souboru. Výhodou je, že v html jsou již většinou obsaženy odkazy, kde se můžeme dozvědět více o nalezené chybě a často jsou zde i odkazy na záplatu chyby. Tento druh reportu je však možné dostat pouze z automaticky generovaných reportu, například z programu Nessus5.

Před samotným testováním je ještě dobrou praxí, vyžádat si od testované firmy analýzu rizik. Pokud ji firma má zpracovanou, měli bychom se z ní dozvědět výši ztrát, v případě výpadku jednotlivých IT systémů.

Pokud jsme si odpověděli na výše uvedené otázky a máme tedy představu o tom, proč chce zákazník ověřit svoje zabezpečení a jak vypadá síť, sestavili prioritizovaný seznam prvků, které budeme testovat, správně vypracovali a podepsali Dohodu o mlčenlivosti, ke máme přesně definovaný rozsah testování, zákazník schválil cenu za naše služby a stanovili jsme časový harmonogram testování, můžeme konečně přistoupit k testování samotnému.

Externí testy

Jak již bylo definováno, externí testy jsou, jak již název napovídá, vedeny z vnější sítě – internetu. Často se však plete s pojmem a black-box test. Je to částečně pochopitelné, protože nejčastější metodou externího testování je black-box metoda, neboli testování bez přidělení práv nebo informací o síti. Nicméně teoreticky je možné provést i externí white-box nebo grey-box testování, tedy testy, kdy máme úplné nebo částečné informace o struktuře a vstupních hodnotách.

Nicméně vzhledem k tomu, že typicky chceme simulovat reálné hrozby, hodí se nám opravdu pro externí testy nejvíce black-box metoda a pro interní testy zase white a grey-box testy. Je méně pravděpodobné, že by chtěl někdo s detailními informacemi o struktuře sítě a uživatelskými jmény napadnout firmu z venku, než že by se o to pokusil někdo bez těchto znalostí.

Dále se tedy v rámci externích testů budeme zabývat výhradně black-box metodou.

Sběr dat

Prvním krokem externího testování je pokusit se opatřit informace o síti, kterou budeme testovat z volně dostupných zdrojů. V praxi to probíhá tak, že se snažíme vyhledat cílovou korporaci ve všech internetových databázích s tím, že hledáme hlavně informace o všech serverech, které firma má, geografické lokace serverů, operačních systémech na kterých servery běží, IP adresách serveru a DNS, kontaktních emailech nebo telefonů.

Dalším zdrojem informací pro nás můžou být sociální sítě, jako například Twitter nebo Facebook. Cílem pro nás bude vyhledat si na těchto sítích zaměstnance a detailně prolistovat jejich profily a příspěvky s tím, že hledáme cokoliv, co by nám mohlo být užitečné pro další fáze. Například pokud bude IT administrátor sdílet odkaz na stránku, kde se řeší záplatování určitého software, je dobrá šance, že tento software reálně běží ve firmě. Nebo pokud CEO firmy často píše o nějakém sportovním klubu, může jméno klubu figurovat v jeho heslu. Stejně tak to však může být úplně irelevantní.

Občas se vyplatí podívat se, jestli firma nehledá nové zaměstnance. V požadavcích na kvalifikaci nového pracovníka může být například: schopnost pracovat s Mac OS, z čehož lze vytvořit domněnku, že zaměstnanci nepracují s operačním systémem Windows, ale Mac OS.

WHOIS

Příkladem postupu při získávání informací může být zadání firemních stránek do vyhledávání jednoho z tzv. WHOIS serverů (např.: www.who.is) a stránku vyhledáme i v databázi www.netcraft.com. Pokud vyzkoušíme tyto 2 dotazy použít na stránky www.unicorncollege.cz, tak se z whois dotazu dozvíme, že doména byla registrovaná přes stránky https://www.active24.cz/ a adresu školy a další informace o datu založení stránek a podobně. Zajímavější je dotaz na serveru www.netcraft.com, který nám vrátí tento výsledek:

Obrázek 4 - WHOIS dotaz

Zdroj: Vlastní zpracování

Nyní máme k dispozici 2 IP adresy na testování, nameserver pro DNS dotazování, a informaci o běžícím operačním systému a web serveru.

DNS dotazování

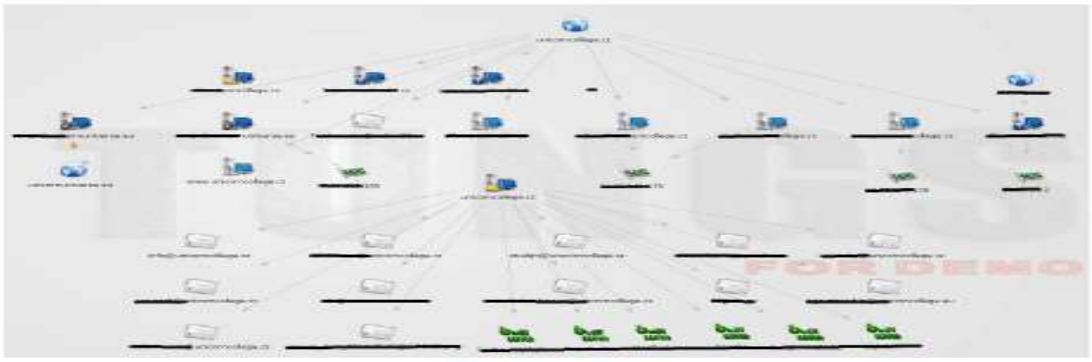
Dalším zdrojem informací může být DNS dotazování. Nejjednodušším způsobem je použití příkazu *nslookup* nebo modernějšího *dig* v příkazovém řádku. Přepnutím na typ záznamů MX, bychom měli být schopní zjistit adresu mailového server případně serverů včetně jejich priorit.

Pokud máme štěstí a DNS server je skutečně velmi špatně nakonfigurován, můžeme použít tzv. Zone transfers. „Informace o zóně jsou přenášeny z primárního jmenného serveru (primary nameserver) na sekundární (secondary nameserver) kvůli zajištění redundance dat. Redundance je nutná v případě výpadku primárního jmenného serveru, kdy jeho funkce přebírá sekundární server (zjednodušeně řečeno). Obecně lze říci, že přenosy zóny by měli být povoleny pouze na sekundární servery. Mnoho DNS serverů však umožňuje tyto přenosy na libovolný počítač. Tohle je kritické, zvláště v případě, kdy primární DNS server obsahuje také informace o interní síti organizace. Útočník pak získá pouhým přečtením informací o zóně kompletní přehled serverů ve vnitřní síti. V mnoha případech může podle jejich jmen odhadovat, k čemu konkrétní server slouží, resp. jaké aplikace jsou na nich provozovány.“6

Jedná se o možnost stáhnout si všechny záznamy z DNS na naší stanici. Tato možnost standardně slouží k synchronizaci s interní záložní DNS, nicméně pokud je špatně nakonfigurovaná, můžeme tuto funkci využívat z externí sítě. Skvělým nástrojem pro získávání informací je program Maltego7 dostupný například v Linuxové distribuci určené pro penetrační testování s názvem Kali Linux.

Umožňuje většinu z výše uvedených typů získávání informací a vše převede do graficky přívětivého výstupu. Výsledek sběru informací pak může vypadat například takto:

Obrázek 5 - Sběr informací v programu Maltego



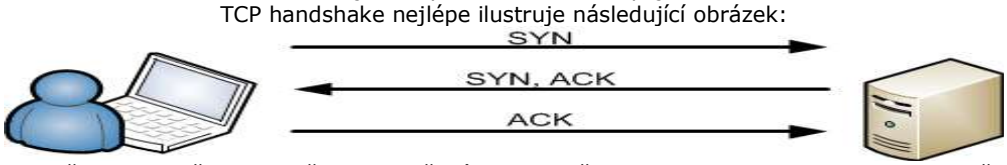
Zdroj: Vlastní zpracování

Port scanning

Máme zde vše potřebné k tomu, abychom se posunuli k další fázi sběru informací a tou je skenování portů a běžících služeb. Nejčastěji používaným nástrojem je nejspíš program NMap8, který nám umožňuje pomocí SYN scanu ověřit, zda-li je port aktivní.

Princip SYN scanu spočívá v zaslání neúplného TCP handshaku, čímž se tato technika skenování téměř nedá zjistit, protože nedojde k úplnému navázání spojení.

TCP handshake nejlépe ilustruje následující obrázek:



SYN scan funguje tak, že NMap pošle SYN požadavek a čeká na odpověď SYN, ACK. Pokud SYN, ACK odpověď nepřijde, port je buď neaktivní anebo blokován firewallem.

Pokud nám odpověď přijde, port je aktivní a může být použitý pro další testování.

Tímto způsobem NMap zjišťuje, jestli je port otevřený, aniž by se plně připojil k cílové stanici [přeloženo volně z knihy Penetration testing.

```
Starting Nmap 6.40 ( http://nmap.org ) at 2015-12-18 08:29 EST
Nmap scan report for 192.168.20.10
Host is up (0.00046s latency).
Not shown: 991 closed ports
PORT     STATE SERVICE      VERSION
21/tcp   open  ftp          FileZilla ftpd 0.9.32 beta
25/tcp   open  smtp         SLmail smtpd 5.5.0.4433
79/tcp   open  finger       SLMail fingerd
80/tcp   open  http         Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
                              mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
106/tcp  open  pop3pw       SLMail pop3pw
110/tcp  open  pop3         BVRP Software SLMAIL pop3d
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn
443/tcp  open  ssl/http     Apache httpd 2.2.12 ((Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k
                              mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0)
445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
```

Information Gathering

Nyní když máme představu, jaký druh služeb na síti běží, můžeme se pustit do III. fáze – Skenování.

Skenování

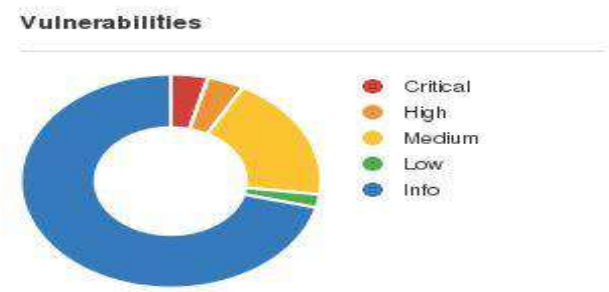
Ve skenovací fázi se budeme zabývat mapováním zranitelných míst na základě informacích získaných v předchozích fázích. Zvláště se budeme opírat o seznam IP adres, nalezených služeb a nainstalovaného SW.

Existuje velké množství nástrojů, které dokážou kvalitně splnit náš účel. Obecně je můžeme rozdělit na automatické a manuální. Výhodou manuálních testů je kompletní kontrola nad testováním a tedy větší šance vyhnout se detekci, než u automatického skenování. Selecký doporučuje12 například nástroj Pytbull nebo Metasploit, které jistě poslouží, nicméně oproti automatickým programům jsou výrazně složitější na obsluhu, náročnější na zkušenosti testera, často i časově náročnější a postrádají GUI, které práci jistě také výrazně zpříjemní. Budeme se tedy věnovat automatickým nástrojům, konkrétně si ukážeme skenování na programu Nessus.

Program v základu umožňuje několik druhů testů. Osobně začínám testem Host Discovery, který ještě jednou zkusí ověřit "živé" stanice a otevřené porty podobně jako NMap v předchozím kroku.

Dále pokračuji testem Basic Network Scan, jehož vstupními argumenty jsou IP adresy nebo rozsah (můžeme použít i tvar 192.168.1.1/24 k otestování celé podsítě). Můžeme si nastavit hloubku testu a také můžeme poskytnout administrátorská práva, pokud to povaha testu žádá, ale vzhledem k tomu, že jsme si definovali, že provádíme testování pouze black-box metodou, tuto funkci nyní nevyužijeme. Výsledkem je seznam zranitelností přehledně seřazených podle zařízení v síti, na kterém byla chyba zjištěna a také náležitě oprioritizované. Stupnice závažnosti je vzestupně podle severity Info -> Low -> Medium -> High -> Critical.

Výstupy z Basic testu můžou vypadat například takto:



Obrázek 9- Nalezené zranitelnosti

Severity ▲	Plugin Name	Plugin Family	Count
CRITICAL	MS14-066: Vulnerability in Schannel Could Allow Remote Code Execution (2992611) (uncredentialed check)	Windows	1
CRITICAL	MS15-034: Vulnerability in HTTP.sys Could Allow Remote Code Execution (3042553) (uncredentialed check)	Windows	1
HIGH	SNMP Agent Default Community Name (public)	SNMP	1
HIGH	SNMP Agent Default Community Names	SNMP	1
MEDIUM	DNS Server Cache Snooping Remote Information Disclosure	DNS	1
MEDIUM	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness	Windows	1
MEDIUM	SMB Signing Required	Misc.	1
MEDIUM	SNMP 'GETBULK' Reflection DDoS	SNMP	1
MEDIUM	SSL Certificate Cannot Be Trusted	General	1
MEDIUM	SSL RC4 Cipher Suites Supported	General	1
MEDIUM	SSL Self-Signed Certificate	General	1
MEDIUM	Terminal Services Doesn't Use Network Level Authentication (NLA) Only	Misc.	1

Pokud rozklikneme jednotlivé nalezené zranitelnosti, otevře se nám detailní přehled o chybě, obsahující název chyby, popis, relevantní odkazy, ohodnocení závažnosti a návrh na odstranění chyby.

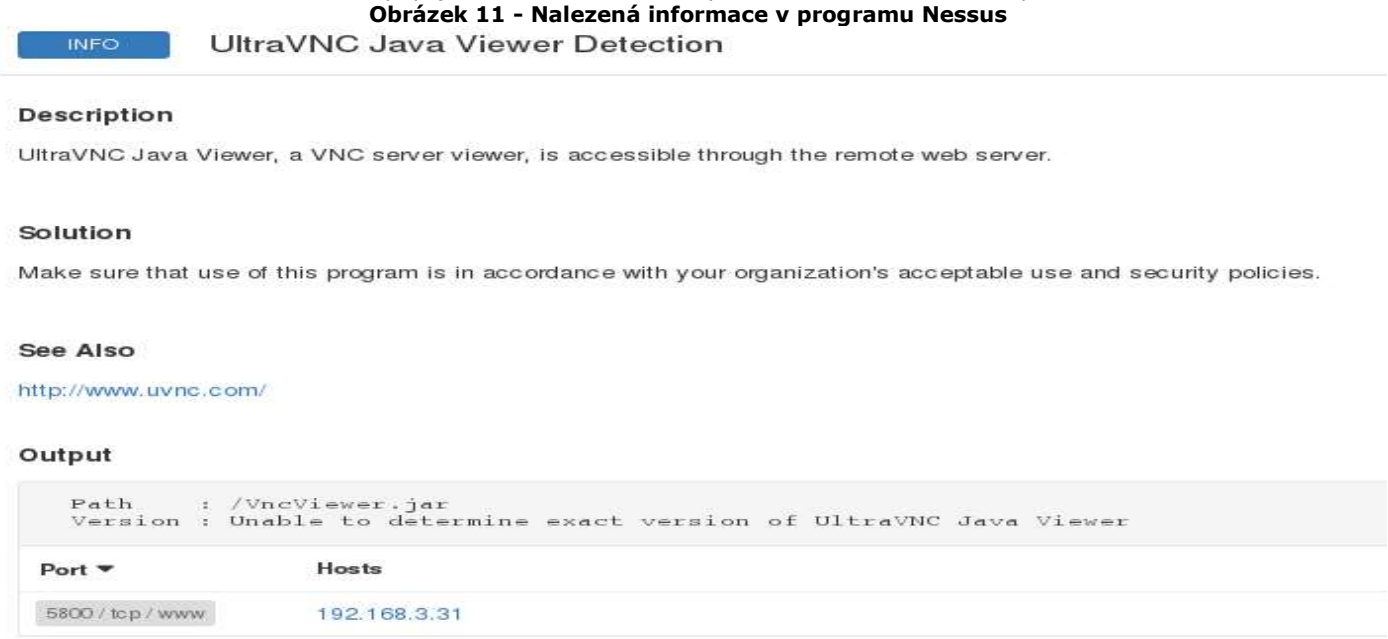
Obrázek 10 - Detail nalezené zranitelnosti

Krom toho, že Nessus najde zranitelnosti s detailním popisem a dokonce i s odkazy na stránky, kde je popsán i postup, jak zranitelnost exploitovat, vyhledá tzv. Info položky.

Nejedná se o chyby v pravém slova smyslu, ale spíše o možnosti, které můžeme využít k získání dalších cílů k útoku. V následujícím příkladu Nessus detekoval, že zařízení (v tomto případě se jedná o PC s operačním systémem Windows) podporuje vzdálené připojení přes UltraVNC13 a naslouchá na portu 5800.

Samotný fakt, že zařízení tuto funkci podporuje, není zranitelnost, nicméně můžeme si prověřit, jestli pro zabezpečení vzdáleného připojení uživatel nenastavil výchozí nebo základní heslo a podobně.

Obrázek 11 - Nalezená informace v programu Nessus



Zdroj: Vlastní zpracování

Exploitace

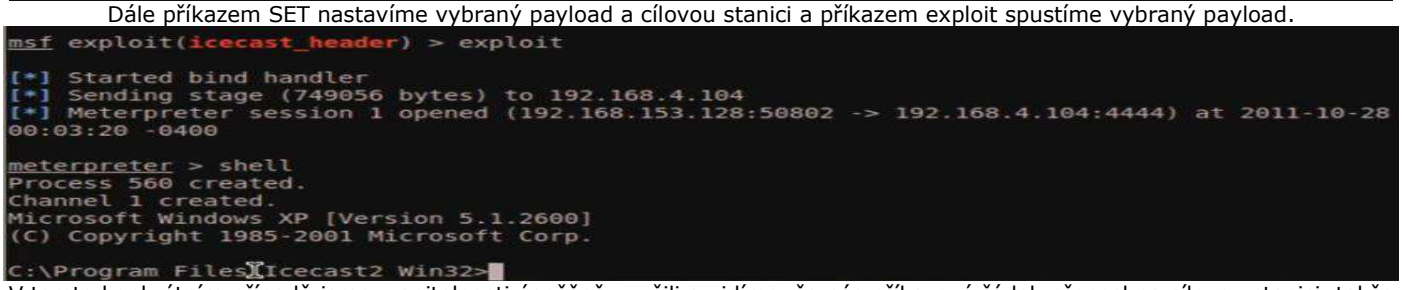
Fáze exploitace se zásadně opírá o výsledky z programů pro vulnerability assesment jako námi použitý Nessus v předchozí kapitole. Všechny nalezené zranitelnosti je před exploitací třeba důkladně prostudovat na základě poskytnutých odkazů z programu Nessus anebo, pokud nám tyto informace nestačí, nebo link neobsahuje dostatečný popis pro exploitaci, můžeme použít našeho nejlepšího přítele při hledání – vyhledávací server Google. Další užitečné odkazy, o které se můžeme opřít, jsou například servery http://www.exploit-db.org nebo http://www.cve.mitre.org . Má preference je používat poslední uvedené stránky CVE (Common Vulnerabilities and Exposures system) v kombinaci s dotazem na Google ve formátu "označení zranitelnosti: site:securityfocus.com" pro hledání pouze na uvedené stránce.

Výsledkem hledání jsou často návody na exploitaci, nicméně je třeba si dát velký pozor, protože některé návody nemusí dělat přesně to, co tvrdí, že dělat budou. Musíme tedy přesně vědět co děláme a ne jen otrocky přepisovat cizí online návod, který může v nejhorším případě vést až k poškození testovaného systému nebo zařízení. Další variantou, jak se takovým nepříjemnostem vyhnout, je používat důvěryhodné a ověřené informace a produkty, jako je například program Metaspoloit14, na jehož použití se nyní v krátkosti podíváme.

Program Metasploit je rozhodně jedním z nejpopulárnějších nástrojů na exploitaci a to hlavně díky vysoké modularizaci. Metasploit obsahuje přes 900 exploitů pro všechny nejpoužívanější operační systémy a každý exploit obsahuje několik různých payloadů (konkrétní kód, který bude spuštěn na cílovém zařízení).
Typický příklad postupu pro použití je pak vybrání a nakonfigurování exploitu a payloadu, nastavení cílového zařízení, volitelně můžeme nastavit i šifrovací techniku, která bude použitá (kvůli ztížení odhalení IDS – Intrusion Detection System15) a spuštění kódu. Navíc umí Metasploit importovat data z programu Nessus, což nám výrazně zrychlí a usnadní práci. Jednotlivé kroky použití můžeme vidět na následujících obrázcích, kde si ukážeme využití známé zranitelnosti Icecast serveru (server na streaming hudby po internetu).
Příkazem msfconsole spustíme Metasploit framework, vyhledáme klíčová slova icecast, příkazem use použijeme nalezený exploit a použitím příkazu **show payloads** si necháme zobrazit dostupné payloady.

Obrázek 12 - Metasploit příkazy 1

Dále příkazem SET nastavíme vybraný payload a cílovou stanici a příkazem exploit spustíme vybraný payload.



V tomto konkrétním případě jsme zranitelnosti úspěšně využili a vidíme, že nás příkazový řádek přenesl na cílovou stanici, takže nyní máme plnou kontrolu nad cílovou stanicí skrz příkazový řádek. Můžeme například volně procházet, stahovat nebo mazat některá data, spouštět jiné služby atd. u některých souborů, ale můžeme narazit na to, že nemáme dostatečné oprávnění pro operaci, kterou chceme provést.

Report

Poslední fází externího testování je vytvoření reportu. Tento report bude sloužit jako podklad pro vytvoření finálního Executive summary reportu pro management a finálního Technického reportu pro IT správce. Přirovnal bych tedy toto tvoření podkladů ke komentování kódu v programování. Je důležité průběžně zapisovat poznatky a výsledky, protože bez nich se stejně jako finální dokumentace k SW nedají finální reporty vytvořit. Nicméně vzhledem k tomu, že tyto dílčí reporty slouží v zásadě k naší interní potřebě, nemusí mít nutně pečlivě upravenou formu a vysvětlivky pro nezasvěcené do problematiky.
Měla by nám stačit kombinace exportů použitých programů jako je Maltego, Nessus obohacená o naše poznámky o úspěšných a neúspěšných testech a nálezech, které jsme dosáhli pomocí programů, které export reportů nepodporují, tedy například poslední použitý Metasploit.

Interní testy

V předchozí kapitole jsme se zabývali testy z vnější sítě a simulovali tak převážně útok hackerů nebo jiných cizích osob. Tato kapitola se však bude zabývat typem útoků, který je většinou firem podceňován – útok zevnitř firemní sítě. Může se tedy jednat jak o útok ze strany zaměstnanců, tak i o útočníka, kterému se podařilo nabourat se do vnitřní sítě. Míra podcenění interních testů vyplývá například z průzkumu společnosti GFI16, která zjistila, že IT manažeři a bezpečnostní experti podniků se zaměřují hlavně na sledování stavu serverů, síťových prvků a detekci útoků z vnější strany sítě.
Typickým příkladem, kdy firmě prudce vzroste zájem o interní testování je až poté, co se firma stane obětí tohoto druhu útoku. Vhodným způsobem, jak firmě ukázat důležitost interního testování ještě předtím, než se stanou obětí, je vyčíslit škody, které nastanou v případě, že by se obětí stali a došlo by k odcizení interních dat.
„Pro vytvoření prvotní představy o míře rizika ztráty interních dat ve vlastní firmě je možné vypočítat hodnotu přibližně podle kalkulátoru míry rizika úniku firemních dat, který byl vytvořen společností Symantec. Dotazníkový kalkulátor je dostupný na webových stránkách www.databreachcalculator.com/Default.aspx . Dotazník se ve třinácti krocích dotazuje například na otázky týkající se oblasti působení firmy, počtu zaměstnanců, typy záznamů, které firma uchovává, firemní politiky nebo používání šifrování. "17
Výstupem tohoto dotazníku je vypočtení míry rizika, že dojde k odcizení dat, průměrná cena za odcizení jednoho záznamu a průměrná cena za únik dat.
Manažeru firmy tedy lze předložit odhad, že pro firmy ve stejném oboru, jako je ta jeho, je riziko odcizení dat například 8,6%, průměrná cena za jeden odcizený záznam je 3472 korun a pokud by došlo k úniku dat, stálo by to firmu průměrně 2 606 324 korun.
Na základě přípravné fáze bychom měli mít jasno v tom, co chceme v rámci interních testů testovat. Podle firmy EC Council, která se zabývá vzděláváním a certifikací bezpečnostních expertů, by na „menu" interního testování neměly chybět následující kroky18:

I. Mapování interní sítě
II. Sken sítě na „živá" zařízení
III. Port scan nalezených zařízení
IV. Pokus o získání přístupu zneužitím známých zranitelností
V. Null session útok - útok fungující pouze na Windows Server 2000, který umožňuje příkazem net use získat přístup ke kritickým částem serveru
VI. Odposlouchávat síť pomocí programu Wireshark a snažit se odposlechnout hesla k POP3, FTP a celé emailové zprávy

VII. Replay útok – „útok kdy útočník zkopíruje proud zpráv mezi dvěma stranami a pošle je další straně. Pokud není odhalen (například pomocí timestampingu), napadené počítače zprocesují tento proud jako validní zprávy. Výsledkem může být soustu neblahých následků, jako například redundantní objednávky položek při používání e-shopu."19[z anglického originálu přeložil autor práce]
VIII. ARP poisoning útok
IX. MAC flooding útok
X. Man-In-The-Middle útok
XI. DNS poisoning
XII. Nabootovat PC do jiného OS a uloupit SAM soubor
XIII. Pokusit se nainstalovat na stanici keylogger, spyware a trojského koně

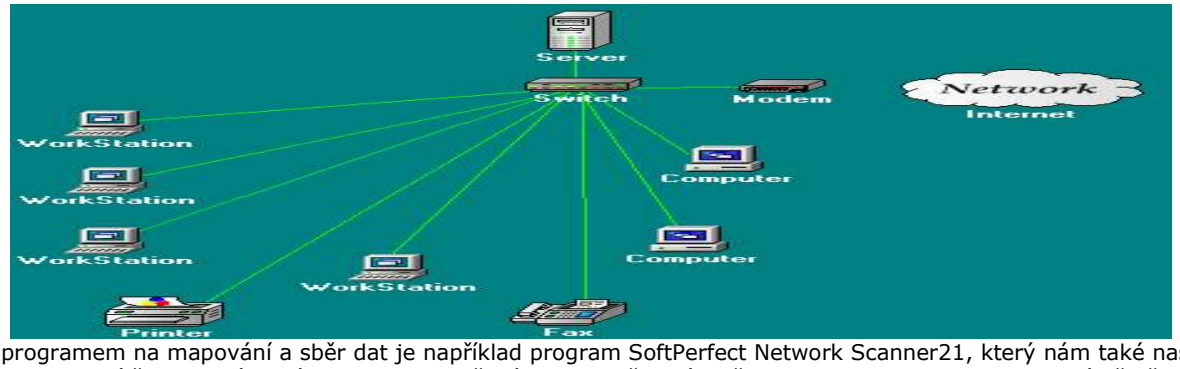
Jak je vidno, položek na „menu" je skutečně mnoho a to stále existuje celá řada dalších testů, které je možné aplikovat. Vzhledem k omezenému rozsahu této práce není možné detailně projít všechny druhy testů a proto se budeme zabývat pouze výběrem z výše uvedeného „menu" a přidáme navíc i automatický test na kontrolu zabezpečení klientských stanic a jednoduchý DoS útok.

Důležité je zmínit, že všechny doposud použité testovací nástroje a metody pro externí testy jsou aplikované i při interním testování. V daných fázích se tedy budeme zabývat pouze metodami a nástroji, které se od externího testování odlišují. Rozdíl mezi interním a externím testováním z hlediska možnosti je, že můžeme testovat L2 vrstvu a máme přímý přístup k DNS serveru, DHCP serveru a můžeme i přímo testovat všechna zařízení které jsou na síti.

Sběr dat

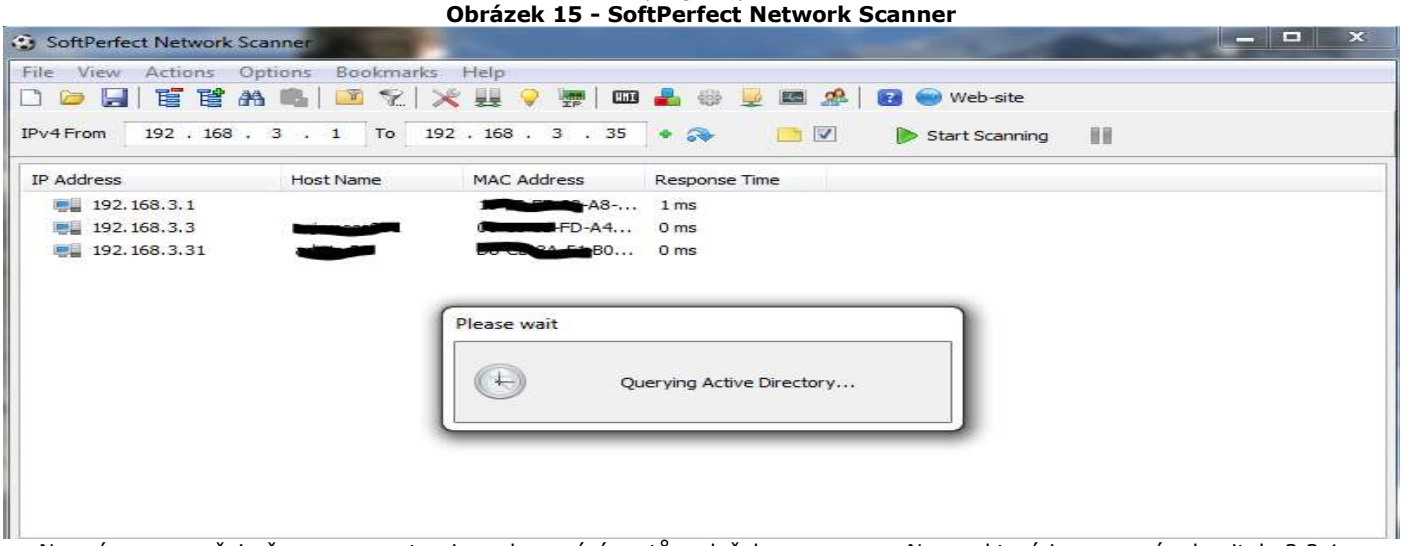
Stejně jako pro externí testování platí, že abychom mohli co nejlépe vyhodnotit slabá místa, musíme nejdřív znát její topologii a architekturu. Nad rámec již použitého programu Nmap tedy ve fázi sběru dat použijeme ještě program Friendly pinger20. Friendly pinger dokáže použitím kombinace ping a traceroute příkazů vytvořit mapu sítě a dokonce i částečně inventarizovat nainstalované aplikace a používat příkazy jako telnet nebo shutdown. Mapa malé firemní sítě pak může vypadat třeba takto:

Obrázek 14 - Friendly pinger, mapa sítě



Dalším programem na mapování a sběr dat je například program SoftPerfect Network Scanner21, který nám také naskenuje určitý segment sítě, ale navíc umí i skenovat otevřené porty a některé služby a dokonce identifikovat aktuálně přihlášené uživatele a umí i několik dalších užitečných funkcí. Jak lze vidět níže, postrádá však grafické zobrazení topologie a proto je vhodné oba programy kombinovat.

Obrázek 15 - SoftPerfect Network Scanner



Nesmíme samozřejmě zapomenout ani na skenování portů a služeb programem Nmap, které je popsané v kapitole 2.2.1. .

Skenování

Interní skenování je vhodné začít programem Nessus, jehož použití už by pro nás nemělo být nic nového. Dalším programem, který se hodí na interní skenování zranitelností, je program Microsoft Baseline Security Analyzer22

Microsoft Baseline Security Analyzer

Jedná se o nástroj, kterým je možné kontrolovat zabezpečení klientských zařízení a serverů. Program umožňuje testovat individuální zařízení stejně tak jako celé IP rozsahy. U klientských stanic je možné zkontrolovat sílu hesla, Windows zranitelnosti a také bezpečnostní aktualizace. U serverů pak to samé a navíc ještě zranitelnosti spojené s databází SQL a IIS web serverem, pokud je přítomen. Program po skenování vygeneruje report s nalezenými zranitelnostmi. Použití i vizuální stránka je prakticky totožná jako u programu Nessus.

Exploitace

Další z testů by mělo být puštění programu Wireshark23 a pokusit se odchytit hesla nebo emaily. Nicméně pokud nejsme na síti, která používá místo switchů obyčejné huby, tak se nám žádnou cizí komunikaci nejspíš odchytit nepodaří. Je to proto, že switch posílá provoz pouze určeným stranám (pokud tedy switch ví, na kterém portu se adresa nachází). Naproti tomu pokud hub přijme paket, přeposílá ho na všechny porty a přitom se všech zeptá: Jsi např.: 192.168.1.22? Pokud zařízení má adresu 192.168.1.22, tak si paket přijme a pokud ne, tak si ho prostě nevšímá. Nicméně pokud zapneme na své síťové kartě tzv. promiskuitní mód, můžeme přijímat i pakety, které nejsou určené jen pro naši IP adresu. Je tedy zřejmé, že pokud síť používá huby, je odposlouchání celé sítě velmi jednoduchou záležitostí.

To je ovšem také jedním z důvodů (vedle nevyhnutelného zahlcení větší sítě), proč se v dnešní době huby téměř nepoužívají a místo nich všude figurují switche a routery, přes které jde veškerá komunikace. Budeme tedy muset vymyslet způsob, jakým předstírat, že jsme switch nebo router, abychom se dostali k veškeré komunikaci. Jedním z možných způsobů jak toho docílit je tzv. ARP cache poisoning.

ARP Cache Poisoning

K pochopení útoku typu ARP cache poisoning (česky poněkud kostrbatě otrávení mezipaměti ARP, proto radši používám originální název) je třeba si trochu detailněji popsat co vlastně dělá protokol ARP. ARP představuje cestu, jak spojit adresy v L2 síti (HW adresy síťových karet) s adresami v L3 síti (IP adresy, které přiděluje administrátor).

Za zkratkou ARP se v angličtině skrývají slova Address Resolution Protocol – tedy protokol pro rozpoznání adres. Pokaždé, když se chceme připojit k jinému zařízení v síti obvykle používáme k identifikaci zařízení buď hostname, plnohodnotný doménový název nebo IP adresu. Předtím, než může být odeslán paket z jednoho zařízení (například notebook) na druhé (například druhý notebook), první notebook musí namapovat IP adresu druhého notebooku na Media Access Control (MAC) adresu síťové karty. Aby toho dosáhl, použije ARP k broadcastu zprávy: „Kdo má IP adresu 192.168.1.22 ? " na celou lokální síť. Zařízení které danou adresu má odpoví: „Já mám adresu 192.168.1.22 a moje MAC adresa je 00:0c:32:a4:ba:69". První notebook si následně tuto informaci uloží do své ARP mezipaměti.

Když chce první notebook někdy později poslat další zprávu, nejdříve se podívá do mezipaměti ARP, jestli pro 192.168.1.22 nemá záznam o MAC adrese, zjistí, že má a tak ji použije, místo aby používal další ARP broadcast. ARP mezipamět se však pravidelně maže a znovu obnovuje, protože prvky v síti se můžou také kdykoliv změnit. Při každém obnovení se samozřejmě posílá i ARP broadcast. Tohoto faktu právě využívá ARP Cache poisoning.

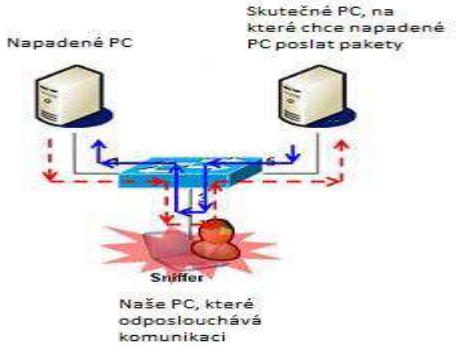
Dalším faktem, kterého ARP cache poisoning využívá je to, že není žádná garance, že stroj, který pozitivně odpoví na broadcast „Kdo je 192.168.1.22? " skutečně nemá adresu třeba 192.168.1.35. Stroj, který dotaz vyslal, prostě odpověď přijme jako validní a zapíše MAC adresu do ARP mezipaměti.

A to je vlastně celé o čem tento útok je, vysíláme sérii ARP odpovědí, které náš cíl přesvědčí, že jsme jiný stroj na síti. Když se to podaří, cílový stroj pak posílá pakety místo na zamýšlené zařízení na náš stroj, kde můžeme komunikaci odchytit. Tento druh útoku je jednou z variant provedení tzv. Man-In-The-Middle útoku.

Nesmíme pak samozřejmě zapomenout nastavit přeposílání paketů z cílového notebooku na skutečnou destinační adresu, jinak bychom způsobili výpadek spojení a náš útok by byl ihned odhalen.

Následující obrázek ukazuje, jak může úspěšný ARP cache poisoning útok vypadat.

Obrázek 16 - ARP cache poisoning



Samotný útok se pak provádí poměrně jednoduchým příkazem:

arpspoof –i eth0 –t 192.168.1.20 192.168.1.35

Přepínač –i nám specifikuje interface, který budeme pro útok používat, přepínač –t nám určí cíl (target), kde první IP adresa určí PC, které chceme napadnout a druhá určuje, za které zařízení se chceme vydávat. V tu chvíli začneme vysílat na cílovou stanici ARP odpovědi a cca po minutě bychom měli být zapsaní v cílové ARP mezipaměti.

Ke správné funkčnosti musíme spustit ještě jednu instanci arpspoofu a spustit stejný příkaz s prohozenými IP adresami, aby mohlo zařízení, za které jsme se vydávali, správně odpovídat. Příkaz pak vypadá takto:

arpspoof –i eth0 –t 192.168.1.35 192.168.1.20

Nyní máme vše připravené pro to, abychom mohli odchytit a přečíst celou komunikaci mezi těmito dvěma stanicemi pomocí programu Wireshark.

Na závěr této podkapitoly bych ještě rád dodal, že stejným způsobem, jako jsme předstírali, že jsme jiné PC v síti, můžeme předstírat, že jsme výchozí brána do internetu, tedy typicky router. Tímto způsobem pak můžeme odchytávat veškerou komunikaci, která směřuje ze sítě ven nebo odchytávat čitelná hesla, která jsou posílána přes HTTP. Je jen třeba si dát pozor, abychom se nesnažili podvrhnout příliš mnoho stanic. Kdyby totiž veškerá komunikace ve větší síti šla místo přes router, přes jeden notebook nebo hůř přes jeden virtuální stroj, mohli bychom způsobit úplný výpadek připojení v celé síti, protože notebook by byl zkrátka přetížen množstvím komunikace, kterou by musel zpracovat a přeposlat dál.

MAC flooding útok

Ještě než se dostaneme k použití Wiresharku se však podíváme na jeden z dalších prostředků jak docílit toho, abychom mohli odchytávat komunikaci z celé sítě. Na začátku kapitoly exploitace jsme si řekli, že je snadné odchytávat veškerou komunikaci v

síti, ve které místo switchů figurují huby. Teď když už víme, jak funguje ARP protokol, mělo by nám být jasné, že pravý rozdíl mezi hubem a switchem je ten, že switch si uchovává svojí tabulku MAC adres, kdežto hub o ničem takovém neví a přeposílá pakety pomocí broadcastu do celé sítě.

MAC flooding útok spočívá v tom, že se pokusíme proměnit switch v hub tím, že zahltíme jeho MAC tabulku. Switch si v tabulce mapuje MAC adresy na jednotlivé porty. Když na však na switch pošleme velké množství paketů, kde každý bude obsahovat jinou zdrojovou MAC adresu, může se nám podařit vyčerpat přidělenou paměť switche pro uchovávání těchto informací. Tím se nám v zásadě podaří vytlačit legitimní MAC adresy ze switche a přeplníme paměť switche, čímž způsobíme to, že se switch začne chovat jako hub a začne posílat pakety na všechny zařízení v síti.

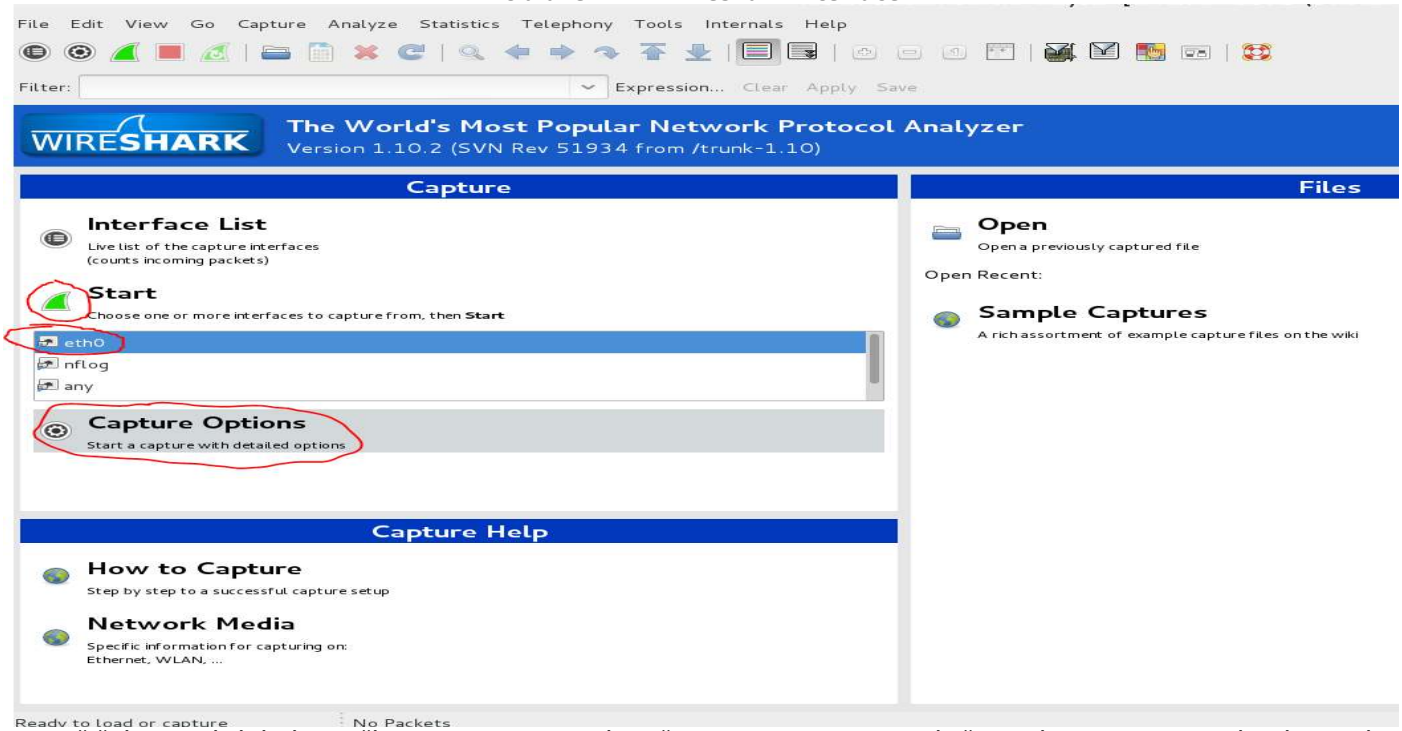
Nám už jen stačí mít připravené odposlouchávání a stejně jako při ARP cache poisoning útoku můžeme odchytávat komunikaci z celé vnitřní sítě.

Wireshark

Pokud se nám úspěšně podařilo uskutečnit ARP cache poisoning nebo MAC flooding útok, potřebujeme nástroj, kterým bychom mohli odchytávat, filtrovat a prohlížet přesměrovanou komunikaci.

Právě k tomu nám poslouží program Wireshark. Disponuje grafickým rozhraním, takže jeho použití není příliš složité. Základní rozhraní vypadá takto:

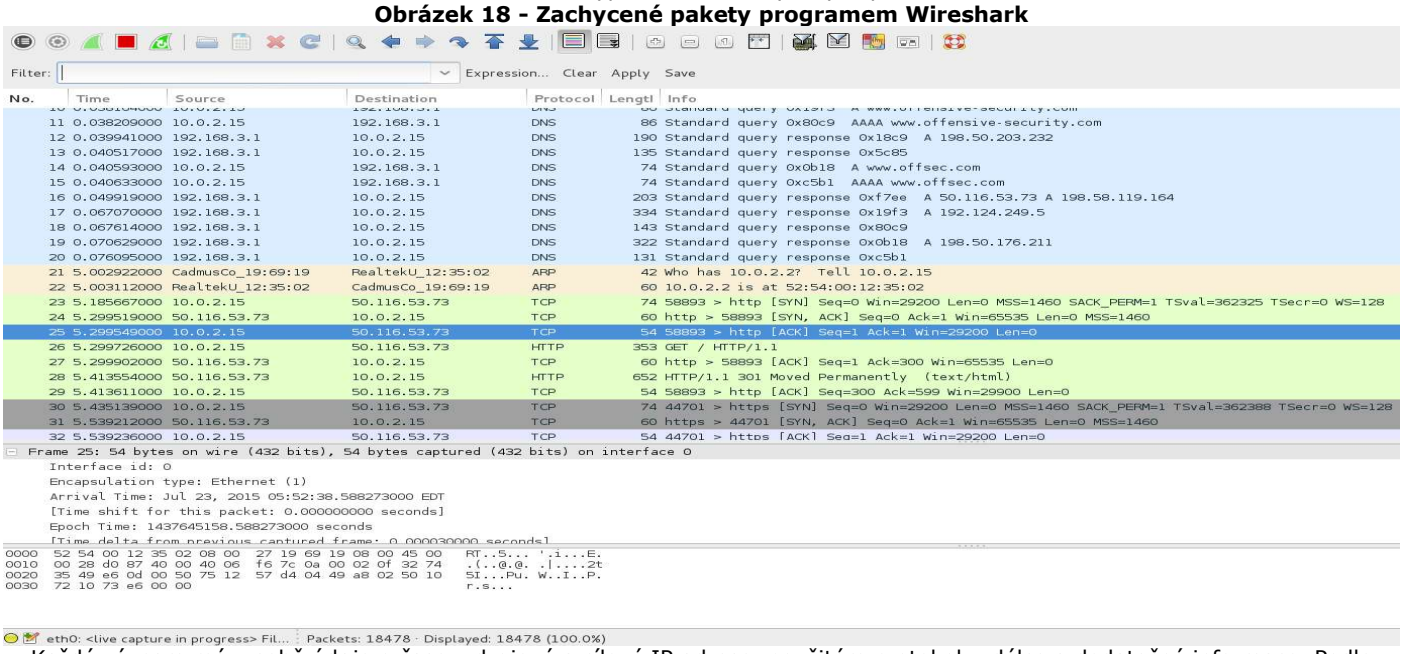
Obrázek 17 - Wireshark interface



Ke spuštění odchytávání nám stačí zvolit interface, který použijeme (eth0 pro kabelové připojení, wlan0 pro bezdrátový adaptér), případně rovnou aplikovat nějaký filtr pod záložkou Capure options a kliknutím na Start se nám zvolený interface automaticky přepne do promiskuitního módu a začne zachytávat všechny pakety určené pro naši IP adresu a také všechny broadcasty.

Takto může vypadat část zachycených paketů:

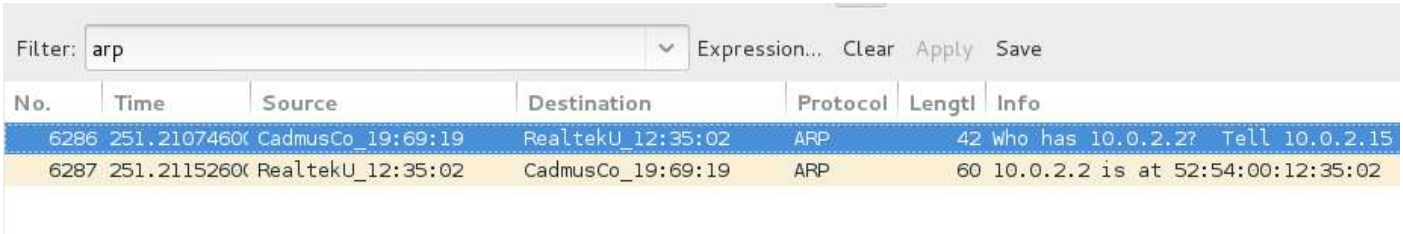
Obrázek 18 - Zachycené pakety programem Wireshark



Každý záznam má v sobě údaje o čase, zdrojové a cílové IP adrese, použitém protokolu, délce a dodatečné informace. Podle všech těchto prvků můžeme nechat zachycené pakety setřídit pro lepší orientaci.

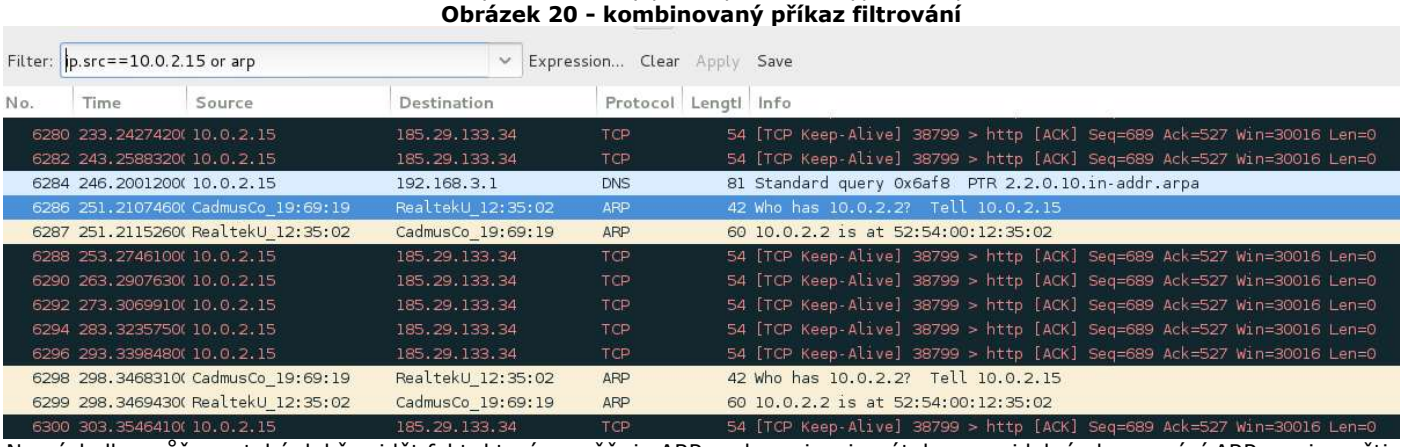
Pokud bychom chtěli pakety filtrovat, máme k dispozici okénko Filter. Pokud se budeme chtít podívat třeba na to, jak vypadá ARP komunikace, kterou jsme zmiňovali o pár odstavců výše, uděláme to následovně. Vyhledáme pakety, které používají ARP protokol a to tak, že do vyhledávácího pole Filter napíšeme jednoduše ARP.

Obrázek 19 - filtrování ARP komunikace



Tímto způsobem můžeme filtrovat všechny druhy protokolů. Pokročilejším způsobem filtrace může být příkaz ip.src==10.0.0.2 který nám vrátí všechny pakety, které byly poslány z adresy 10.0.0.2. Příkazy jde také kombinovat pomocí klíčových slov *or* nebo *and.* Takový kombinovaný příkaz pak může vypadat například takto:

Obrázek 20 - kombinovaný příkaz filtrování



Na výsledku můžeme také dobře vidět fakt, který umožňuje ARP cache poisoning útok – pravidelné obnovování ARP mezipaměti. Tolik asi k základnímu použití programu Wireshark a nyní se podíváme na další typ útoku proti L2 vrstvě.

DNS Cache poisoning

Stejně tak jako můžeme otrávit ARP mezipaměť, můžeme otrávit i záznamy Domain Name Service (DNS) .

DNS mapuje doménové jména na IP adresy. Máme-li tedy název počítače například "czpocitac123" a chceme se na něj vzdáleně připojit přes službu RDP, nemusíme znát jeho IP adresu a stačí nám do pole adresy vyplnit czpocitac123 a náš lokální DNS server nám hostname přeloží na IP adresu. Ještě jasnější příklad jak DNS funguje možná ukáže následující scénář připojení k serveru www.seznam.cz.

Běžný notebook si chce prohlédnout stránky a zadá do URL prohlížeče www.seznam.cz. V tu chvíli se notebook spojí s lokálním DNS serverem s dotazem: „Chci se připojit na www.seznam.cz, jaká je jeho IP adresa?" Předpokládejme, že lokální DNS tuto informaci nemá a tak se zeptá jiného DNS serveru, a ten třeba ještě jiného DNS serveru, dokud tuto informaci některý DNS server mít nebude a nepředá IP adresu zpátky, až do lokálního DNS serveru. Ten si informaci uloží a pošle jí do notebooku a ten se nyní přímo připojí na 77.75.76.3 a zobrazí si stránky.

DNS cache poisoning funguje na stejném principu jako ARP cache poisoning: Pošleme několik DNS odpovědí, které ukazují na falešnou IP adresu. Provedení útoku také není nijak složité. Použijeme k němu program dnsspoof, který je součásti operačního systému pro penetrační testování s názvem Kali Linux24 a který ale akceptuje jako parametr pouze předpřipravený textový soubor s dvěma položky na řádek v pořadí falešná IP adresa, cílové doménové jméno. Tento soubor si můžeme pojmenovat jak chceme, v následujícím příkladu se jmenuje files.txt a obsahuje jeden řádek ve tvaru 192.168.1.22 www.seznam.cz

Obrázek 21 - DNSSpoof

Jestli je útok úspěšný zjistíme tak, že napíšeme příkaz nslookup www.seznam.cz . Pokud nám tento příkaz vrátí IP adresu 192.168.1.22 místo 77.75.76.3, útok se zdařil.

Úspěšnost můžeme také ověřit zadáním adresy www.seznam.cz do URL prohlížeče a stránka která se zobrazí by měla být ta, kterou jsme nastavili na webserveru falešné adresy.

Zneužití této zranitelnosti nejlépe demonstruje příklad, kdy na falešnou IP adresu zkopírujeme originální stránky seznamu a upravíme je aby zadané přihlašovací údaje do emailu uložili na náš falešný server a následně přesměrovali na skutečné stránky. Uživatel si tak pravděpodobně všimne, že se přihlášení nezdařilo, zkusí to znovu, a když se mu to podaří, zapomene na celou věc, aniž by tušil, že jeho údaje již nejsou jen jeho.

Boot test a extrakce SAM souboru

K dalšímu z testů budeme potřebovat fyzický přístup k jednomu ze zařízení. Může se jednat například o volně přístupný notebook v zasedací místnosti. Do notebooku vložíme bootovatelný flash disk s nainstalovaným operačním systémem Kali Linux a restartujeme nebo zapneme, pokud je notebook úplně vypnutý. Při startu se zkusíme dostat do BIOSu a změnit bootovací prioritu na USB zařízení. Pokud vstup do BIOSu chrání heslo, zkusíme ještě přerušit standartní nabíhání OS a zvolit USB flash disk ručně. Velmi pravděpodobně se nám nakonec tak nebo onak podaří nabootovat do Kali Linuxu, kde můžeme kromě jiného extrahovat SAM soubor a v dalším testu se ho pokusit prolomit a získat tak některá uložená uživatelská jména a hesla.

SAM soubor je součástí systému Windows a uchovává sobě zahashované Windows hesla. Najdeme ho v adresáři Windows\System32\config. Když se ho pokusíme otevřít neuvidíme téměř žádný čitelný text, protože zahashované hesla stejně jako většina ostatního textu jsou ještě zašifrované 128-bitovou RC4 šifrou. Naštěstí pro nás se soubor s klíčem k rozšifrování nachází ve stejné složce jako SAM soubor a jedná se o soubor s názvem SYSTÉM. Příkazem **bkhive system key.txt** extrahujeme klíč do souboru key.txt který vzápětí použijeme k rozšifrování SAM souboru.

Příkaz k použití extrahovaného klíče na zašifrovaný SAM soubor má syntaxi

samdump2 sam key.txt > hashes.txt

Používáme program samdump2, který je taktéž obsažený v Kali Linuxu a jako argument mu předáváme SAM soubor, soubor s klíčem a výstup ve formě uživatelské jméno: zahashované heslo si necháme vypsat do souboru hashes.txt, abychom ho měli připravený pro další crackování.

Útoky na heslo

Ve firemním prostředí jsou hesla často klíčovým opěrným bodem zabezpečení. Můžeme mít bezchybně zazáplatovaný veškerý běžící software a nainstalované všechny Windows aktualizace, ale uživatele záplatovat nelze. Pokud si některý uživatel zahesluje účet heslem ve stylu „heslo123", veškerá ostatní bezpečnostní opatření jsou nám k ničemu. Ne nadarmo se říká, že bezpečnost firmy je tak silná jako její nejslabší článek a často je tímto článkem právě uživatel a jeho volba hesla.

Problematika politiky hesel ve firmě je velmi aktuální. Obecně jsem ve firmách zažil 2 zcela jiné přístupy.

První přístup se vyznačuje z mého pohledu extrémně krátkou dobou, po kterou heslo platí, než si ho uživatel musí změnit, přičemž musí heslo zároveň splňovat komplexitu (velké písmeno, číslo a speciální znak). Tento přístup typicky vedl k tomu, že většina hesel byla ve stylu: „Unor2015." Nebo „Sparta02" a vzhledem k tomu, že některá hesla se sdílela mezi více uživatelů, tak se heslo dalo se zhruba 50% úspěšností najít napsané někde poblíž pracovního místa na papírku nebo kalendáři.

Druhý přístup vyžadoval extrémně komplexní heslo s délkou alespoň 10 znaků, ale doba po kterou heslo platilo byla 6 měsíců.

Je zřejmé, že oba přístupy mají své vady. V druhém případě je to jistě příliš dlouhá doba, po kterou heslo platí, což dává například v případě úniku zahashovaného hesla útočníkovi příliš mnoho času na crackování hesla. Nicméně snaha aby si uživatelé vybrali kvalitní a odolné heslo je chvályhodná.

Osobně si myslím, že heslo by se mělo měnit tak často, jak velké riziko je, že heslo někdo odhalí. Pokud máme heslo k routeru, které je zavřené někde v trezoru, není nutné ho měnit každý rok, protože administrativní úkony se změnou spojené zdaleka nedosahují přidané hodnoty bezpečnosti, kterou změna přinese. Ideální politikou se mi pak zdá první zmíněná varianta, ale pouze v kombinaci s bezpečnostním školením zaměstnanců, kde se doporučí používat místo slov celé fráze nebo například hlášky z filmů.

Tyto fráze nebo hlášky jsou snadno zapamatovatelné, takže odpadne potřeba psát si hesla na papírky a zároveň je takové heslo „3JeJaSiZlomilSavli!" prakticky neprůstřelné proti všem druhům crackování hashů.

Abychom lépe pochopili, jaké heslo je bezpečné a proč tomu tak je, ukážeme si na příkladu, jaké útoky se proti heslům dají použít.

Útoky můžeme rozdělit na online a offline. Online útokem může být například posílání kombinací uživatelského jména a hesla na webovou aplikaci Microsoft Exchange – OWA (Outlook Web Application). Problémem při online útocích bývá fakt, že po několika neúspěšných pokus je účet často zablokován anebo je zablokovaná naše IP adresa. Existují možnosti, jak se i s těmito druhy ochran vyrovnat (například časté měnění IP adresy během crackování), ale my se spíše zaměříme na oblast offline útoků. Při offline útocích si využitím některé zranitelnosti stáhneme soubor s hesly, který typicky obsahuje pouze hashe hesel.

Hash je výsledek některé jednosměrné hashovací funkce, která slouží k přeměně čitelného, často i krátkého textu na dlouho sérii čísel a písmen, ze které nelze získat zpět původní hodnotu. Jak tedy ale poznáme, že je se zadané heslo shoduje s tím správným, když nelze z hashe dekryptovat zpět na čitelný text? Nejlépe to ilustruje příklad přihlašování do Windows. Když vytváříme uživatelský účet a zadáme heslo třeba „MáchaleTebeČekáŠibenice2x", systém Windows tento řetězec vezme a hashovací funkcí z něj udělá řetězec podobný tomuto: „b4b9a02e6f19a9bd760f388b67351e2b ", který si uloží do SAM souboru. Když potom další den PC zapneme a Windows se nás zeptá na heslo, my opět napíšeme že Máchala čeká šibenice, Windows na toto heslo opět pustí hashovací funkci a výsledný hash porovná s tím co si uložil do SAM souboru a pokud se hashe shodují, přihlásí nás.

Stejným způsobem pak probíhají útoky, máme SAM soubor s uloženým hashem hesla a pomocí crackovacího programu, jakým je třeba populární John The Ripper25, zkoušíme kombinace nebo běžná hesla, které John zahashuje a porovná s hashem ze SAM souboru.

Máme v zásadě 2 možnosti, jaký druh útoku zvolit – tzv. Bruteforce attack (útok hrubou silou), při kterém se v nejkomplikovanějším případě snažíme uhodnout heslo kombinací všech malých a velkých písmen, číslic a speciálních znaků v určitém rozsahu. Zvolíme-li tedy například malá písmena a čísla o délce hesla 6 znaků, bude první porovnání vypadat zhruba takto: „aaaaaa == skutecneheslo". Tato metoda funguje hlavně na jednoduchá hesla, kdy má heslo do 6 znaků a obsahuje pouze číslice a písmena. V takovém případě trvá teoreticky bruteforce útok proti NTLM hashi 4 hodiny a 24 minut s 57 731 386 986 možných kombinací26. Prolomit 7 místné heslo pak trvá přes 11 dní a 8 místné heslo necelé 2 roky.

Druhým způsobem útoku na heslo je slovníkový útok. Útok spočívá v tom, že crackovací program nezkouší každou možnou kombinaci, ale iteruje předem připraveným souborem, který obsahuje běžná hesla.

Příkaz, kterým bychom mohli pustit program John The Ripper na připravený textový soubor z minulé kapitoly, může vypadat takto:

john --wordlist=mywordlist2.lst --rules hashes.txt

Zajímavým přepínačem je –rules, který umožňuje aplikaci vlastních pravidel v programu John the ripper. Tyto pravidla můžeme editovat v souboru john.conf, kde můžeme pod kategorií List.Rules.Wordlist přidat například řetězec $[0-9] $[0-9]. Pokud následně v příkazu použijeme přepínač –rules, pro aplikaci pravidel, přidají se nám za každý řetězec z wordlistu 2 číselné znaky.

Na závěr této kapitoly bych ještě rád zmínil možnost použití tzv. Rainbow tables, což jsou předhashované slovníky nebo všechny možnosti v dané komplexitě a délce. Jejich použitím ušetříme čas tím, že eliminujeme fázi, kdy John the Ripper musí čitelný text převést na hash a teprve ten porovnat a program pouze porovnává hashe, což je řádově jednodušší a rychlejší proces. Jedinou malou nevýhodou je, velikost rainbow tabulek, která například pro všechny MD5 hashe pro malá písmenka a číslice v rozsahu 1-9 znaků zabírá zhruba 80GB.

Sociální inženýrství

"Firma si může pořídit ty nejlepší a nejdražší bezpečnostní technologie, vyškolit personál tak, aby byla každá důvěrná informace před odchodem domů pod zámkem, najmout si tu nejlepší firmu na noční ostrahu objektů, a přece bude ta organizace ještě zranitelná. Soukromé osoby se mohou držet všech nejlepších zásad doporučovaných odborníky, mohou otrocky nainstalovat všechny nejnovější
produkty vylepšující zabezpečení a odpovídajícím způsobem pozorně zkonfigurovat systém, mohou použít všechna jeho vylepšení či opravy, a přece jsou tyto osoby stále nechráněné. "27

Tímto odstavcem nám rozšiřuje Kevin Mitnick, proslulý mistr v oboru sociálního inženýrství pohled na nejen firemní bezpečnost. Lidský faktor dále nazývá Achillovou patou každé firmy. V rámci penetračního testování je tedy žádoucí, abychom i tuto zranitelnost otestovali.

Ve své podstatě je sociální inženýrství metoda hodnotící bezpečnost, využívající konceptu vzájemné důvěry více lidí. Hackeři, ale i běžní lidé špatných úmyslů, často používají tuto techniku pro získání neoprávněného přístupu k informacím bez potřeby obcházení zabezpečení technického rázu, jako jsou třeba korporátní firewall, IDS nebo antivir.

Často se předpokládá, že internetové zločiny jsou možné díky identifikování mezery v ochraně IT systému útočníkem. V reálné situaci sociální inženýrství hraje obvykle velkou roli v provedení zločinu a pomáhá útočníkovi dostat se přes korporátní zabezpečení. Důvěřivost uživatelů či nedostatečnost bdělosti jsou často důvodem nedovoleného přístupu k informacím, nebo-li prolomení zabezpečení.

Podle osobních zkušeností konzultanta této práce z firmy ESET software s.r.o. podlehne dobře připravenému útoku tohoto typu 7 z 8 lidí, což je číslo skutečně alarmující. Dle jeho slov se v rámci přípravné fáze domlouvá i promyšlenost útoku na zaměstnance, aby měli zaměstnanci vůbec možnost se ubránit.

Metod, kterými lze použít sociální inženýrství je skutečně nepřeberné množství, ale nejpoužívanějšími jsou:

- Phising
- Zjištění informací po telefonu
- Pokus o fyzický vstup do kanceláře / serverovny

My se nyní podíváme na tyto 3 metody blíže. Mnoho dalších metod najdeme například v knize Umění Klamu nebo The Art of Intrusion, obě od autora Kevina Mitnicka.

Phishing

Pro naše účely použijeme definici Georgie Weidman: „Pokus o oklamání uživatele za účelem získání citlivých informací pomocí emailu nebo jiných elektronických prostředků (například telefonu), vydávaje se za důvěryhodnou osobu, říkáme phising".28 Tato definice nebere v úvahu tzv. SPAMy což jsou „nevyžádané reklamní sdělení masově šířené internetem. Původně se používalo především pro nevyžádané reklamní e-maily, postupem času tento fenomén postihl i ostatní druhy internetové komunikace – např. diskuzní fóra, komentáře nebo instant messaging".29 Důvod proč se SPAMy nebudeme zabývat je, protože jsou masové a necílené a z těchto důvodu se jich v penetračním testování nepoužívá.

Phising se naopak používá velmi často, protože k němu penetrační tester nemusí mít žádné herecké vlohy jakou u útoků, které budou zmíněné dále.

Princip útoku spočívá v zaslání falešného emailu, který má však věrohodnou emailovou adresu odesílatele (například vedoucího IT oddělení nebo CEO firmy), ve kterém se snažíme přimět uživatele, aby nám zaslal své heslo nebo jiné důležité informace, které chceme. Klíčovými prvky k přesvědčení uživatele, aby nám poslal informace které chceme jsou podle mého názoru hlavně:

- Dobře provedená fáze získávání informací a použití věrohodného emailu
- znalost místního firemního žargonu
- dobře promyšlená, často nestandardní situace, která poskytnutí informace vyžaduje

Email odešleme na adresy získané například na základě fáze sběru informací, nebo získané zneužitím některé ze zranitelností při externím testování.

Naším záměrem může také být přimět uživatele, aby si prohlédl nějakou námi připravenou WWW stránku nebo aby otevřel infikovanou přílohu. Možnosti jsou omezeny pouze fantazií penetračního testera.

Není také od věci mít připravený nouzový scénář, abychom mohli v případě, že by se plán v kteroukoliv chvíli zvrtl, z celé situace naprosto nepozorovaně vycouvat.

Telefonické hovory

Na rozdíl od předchozího útoku je při útoku telefonickým hovorem jednou z klíčových vlastností dobře sehrát roli, kterou představíme, stejně tak jako nepropadnout panice, když se nám něco nedaří podle plánu, nebo nám oběť položí nečekanou otázku a podobně. Další dobrou praktikou je žádat pouze o informaci, kterou oběť mylně považuje za nevinnou nebo mít opět velmi dobře vymyšlený důvod či situaci, která nás k dostání informace "opravňuje".

Velmi dobrý příklad dobře vymyšlené situace uvádí Kevin Mittnick v příkladu s názvem "Prosil bych číslo..".

"Útočník zavolal do mechanizovaného centra přiřazování linek (MLAC) jedné telekomunikační firmy a řekl ženě, které zvedla telefon: „Dobrý den, tady je Paul Anthony. Jsem montážní technik. Poslyšte, mám tu spálenou rozvodnou skříňku. Policie si myslí, že se nějaký chytrák pokoušel podpálit svůj dům, aby získal z pojišťovny prachy. Poslali mne sem, abych tu zapojil novou skříňku s dvěma sty koncovkami. Potřeboval bych vaši pomoc. Jaká zařízení by měla fungovat na South Main pod číslem 6723?"

V jiných odděleních telekomunikačních firmy, kam zavolal, věděli, že neveřejná telefonní čísla nebo jakékoliv informace přiřazující k jménu lze sdělovat pouze oprávněným zaměstnancům. Ale o existenci MLAC vědí spíš jen pracovníci firmy. Tyto informace jsou sice chráněné, ale kdo by odmítl pomoci kolegovi, který má vykonat těžkou a důležitou práci? Dotazovaná s ním soucítila — vždyť jí se také občas přihodilo, měla velmi náročné pracovní dny — takže trochu pominula zásady a pomohla kolegovi ze stejné firmy, který měl problém. Sdělila mu označení kabelů, svorek a všechna čísla přiřazená této adrese."30

Tento příklad velmi dobře demonstruje sílu tohoto druhu útoku. I velmi jednoduchý vymyšlený příběh bez použití firemního žargonu nebo nátlaku stačil k tomu, aby uživatel udělal, co si sociotechnik přál.

Pokus o fyzický vstup do kanceláře / serverovny

Při tomto druhu testování se snažíme otestovat ochranu proti volnému pohybu cizích osob po kanceláři nebo serverovně. Pro úspěšné testování fyzického přístupu je kladen ještě větší nárok na vlastnosti sociotechnika než při telefonickém hovoru. Je to logické, protože při nečekané otázce oběti nemůžete prostě říct „Promiňte, ověřím to a zavolám hned zpátky" a mezitím si odpověď na otázku opatřit. Všechny situace musíte řešit na místě a v případě nutnosti improvizovat.

Velký rozdíl je také jestli máme domluvené testování tzv. Red týmem, které spočívá v tom, že o testu ví pouze nejvyšší IT manažer nebo security manažer a nikoliv zbytek IT teamu a ani nikdo z ostatních zaměstnanců nebo Blue týmem, kdy o testu ví celé IT avšak nikoliv zaměstnanci.

Red team testování má tu výhodu, že lépe simuluje reálný útok a IT manažer může sledovat reakci IT teamu, případně ostatních odpovědných zaměstnanců, na vzniklou situaci. Nevýhodou dobře demonstruje příběh, který zažil konzultant této práce. Několika členům red týmu se úspěšně podařilo proniknout do firemních prostor a všichni se sešli v serverovně, bez doprovodu. K tomu, aby se jim to povedlo, museli oklamat každý jednoho nebo více zaměstnanců firmy. Několik desítek minut ,co byli již všichni členové red týmu pryč, se oklamaní zaměstnanci začali potkávat a sdílet mezi sebou informace, ze kterých si dali dohromady, že byli oklamáni. Dříve než stačil IT manažer zasáhnout, zavolali zaměstnanci na policii, aby průnik ohlásili. Naštěstí jim objednavatel záhy vysvětlil, že se jednalo pouze o penetrační test, ale i tak si později vyslechli pár nemilých slov, když policii oznamovali zrušení poplachu.

Aplikace metod na firemní síť

Popis následujících testů se opírá o reálné penetrační testování provedené na nejmenované firmě. Veškeré výsledky jsou anonymizované podle požadavků testované firmy a veškeré uvedené zranitelnosti jsou již odstraněny. Většina zranitelností, na které během testování narazíme, patří mezi časté a typické zranitelnosti, které v různých variacích najdeme téměř v každé průměrné firmě. Jedinou výjimkou byl výsledek testu fyzického přístupu do firmy, kdy se sociotechnik mohl bez omezení

pohybovat po firmě, aniž by byl kýmkoliv zastaven a to ani na obou z recepcí, které byly v době oběda, kdy testování záměrně probíhalo, opuštěné. S tímto druhem svobody pohybu cizích osob po takto velké firmě se ani konzultant této práce příliš často nesetkal.

Popis testovaného prostředí

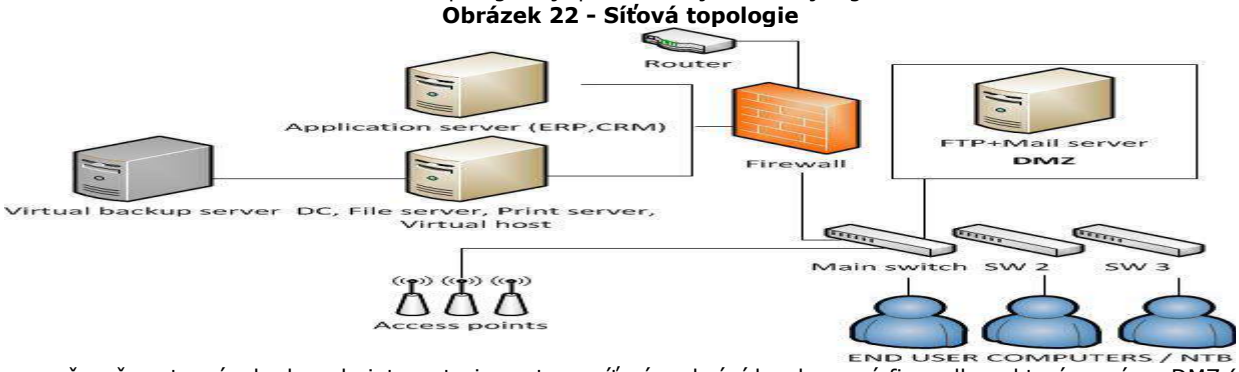
Firma, ve které jsme v praktické části prováděli penetrační testování, je typická, středně velká společnosti s cca 150 zaměstnanci. Její zaměření není pro tuto práci relevantní, takže by nám měla postačit informace, že figuruje v potravinářském průmyslu. Co nás však zajímá (pro účely sociálního inženýrství) je, že celá sídlí v jednom patře klasické kancelářské budovy, která má jednu recepci dole a ještě jednu již přímo při vstupu do kanceláří.

Co se týče kontroly vstupu do objektu kanceláří, tak je standardně třeba vyplnit na spodní obecné recepci jméno a příjmení, na základě kterého dostane návštěvník kartičku, která mu umožní vstup do vybrané kanceláře. Jméno a příjmení nicméně není ověřováno ani oproti občanskému průkazu, takže si návštěvník může jméno prakticky vymyslet, čímž je vstup v zásadě nekontrolovaný.

Síťová topologie

Síťovou topologii nejlépe naznačuje následující graf:

Obrázek 22 - Síťová topologie



Vidíme na něm, že vstupním bodem do internetu je router a síť nám chrání hardwarový firewall, za kterým máme DMZ (podsíť oddělená z bezpečnostních důvodů od vnitřní sítě, ve které jsou umístěné služby, které jsou k dispozici z celého internetu). V naší DMZ se nachází FTP server (pro sdílení a přenos souborů) a virtuální emailový server.

Velmi často se v DMZ vyskytuje i webový server, tedy server, na kterém jsou umístěné firemní WWW stránky, ale tato firma si z bezpečnostních důvodů vybrala možnost externího hostingu, kvůli snížení a diversifikaci potenciálního rizika útoku. Za firewallem se dále nachází vnitřní síť složená z několika switchů, tří access pointů, cca 150 uživatelských stanic (notebooky i desktopy), domain controlleru, file serveru a print serveru v podobě jednoho serveru, application serveru (na kterém běží ERP a CRM systémy) a navíc máme i virtuální server, který se nám stará o zálohování. Síť běží na UTP kabelech typu Cat5e a všechny switche jsou gigabytové.

Externí testy

V dohodnutém rozsahu externích testů byli pouze fáze získání informací o síti a skenování zranitelností, tedy pouze tzv. Vulnerability assesment. Management testované firmy nepotřeboval dokazovat, že daná zranitelnost skutečně jde zneužít i s přihlédnutím na výrazné zvýšení rizika narušení běžných denních business povinností při provádění exploitační fáze. Tento přístup je typický pro většinu firem s tím, že se zřejmě management firem řídí příslovím: „Aby se vyzkoušeli hasiči, není třeba zapalovat les.“

Získané informací o síti

Kombinací dotazů na stránkách http://www.google.com, http://who.is/ a http://searchdns.netcraft.com jsme se dostali k těmto informacím:

Obrázek 23 - WHOIS dotazování 2

Site	http://www.[redacted].com	**Netblock Owner**	Euroweb Internet Service Provider
Domain	[redacted].com	**Nameserver**	nsa1.invitel.net
IP address	81.0.67.[redacted]	**DNS admin**	nic@invitel.net
IPv6 address	*Not Present*	**Reverse DNS**	lnx-web-02.aspcenter.hu
Domain registrar	*unknown*	**Nameserver organisation**	whois.secura-gmbh.de
Organisation	*unknown*	**Hosting company**	Invitel
Top Level Domain	Commercial entities (.com)	**DNS Security Extensions**	*unknown*
Hosting country	HU		

Hosting History

Netblock owner	IP address	OS	Web server
Euroweb Internet Service Provider [redacted], Hungary	81.0.67.[redacted]	Linux	Apache

Name Servers

nsa1.invitel.net 62.77.20[redacted]

nsa2.invitel.net

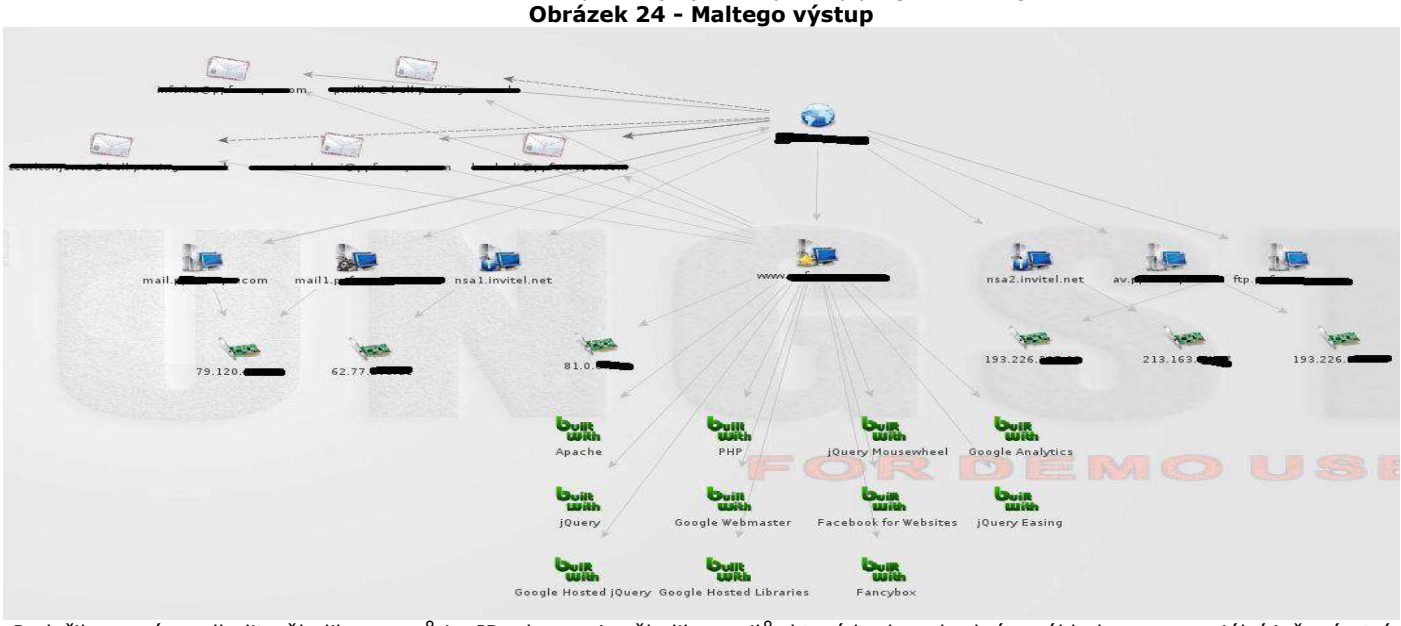
Raw Registrar Data

```
Domain Name: [redacted].com
Registry Domain ID: 1665491475_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.joker.com
Registrar URL: http://joker.com/
Updated Date: 2014-06-29T00:36:50Z
Creation Date: 2011-07-06T08:14:04Z
Registrar Registration Expiration Date: 2015-07-06T08:14:04Z
Registrar: CSL Computer Service Langenbach GmbH d/b/a joker.com
Registrar IANA ID: 113
Registrar Abuse Contact Email: abuse@joker.com
Registrar Abuse Contact Phone: +49.21186767447
Domain Status: clientTransferProhibited
Registry Registrant ID:
Registrant Name: [redacted]
Registrant Organization: [redacted]
Registrant Street: [redacted]
Registrant City: [redacted]
Registrant State/Province: --
Registrant Postal Code: H-2040
```

```
Registrant Country: HU
Registrant Phone: +36.180[redacted]
Registrant Email: [redacted].com
Registry Admin ID:
Admin Name: [redacted]
Admin Organization: [redacted]
Admin Street: [redacted]
Admin City: Budaors
Admin State/Province: --
Admin Postal Code: H-2040
Admin Country: HU
Admin Phone: +36.1801[redacted]
Admin Email: [redacted].com
Registry Tech ID:
Tech Name: Invitel Hostmaster
Tech Organization: Invitel Zrt.
Tech Street: [redacted]
Tech City: [redacted]
Tech State/Province: --
Tech Postal Code: 2040
Tech Country: HU
Tech Phone: +36.885[redacted]
Tech Email: domain@invitel.co.hu
Name Server: nsa1.invitel.net
Name Server: nsa2.invitel.net
```

Další informace nám přinesl již jednou použitý program Maltego.

Obrázek 24 - Maltego výstup



Podařilo se nám odhalit několik serverů i s IP adresami, několik emailů, které budou vhodným základem pro sociální inženýrství a také technologie, které běží na webových stránkách firmy.

Nalezené zranitelnosti

V této praktické části práce se podíváme na seznam nalezených zranitelností, ohodnocení rizika na škále A-C, kdy A je kritické riziko a C nízké riziko, popis možné exploitace a doporučení k nápravě. Jedná se o výtah z celkového reportu penetračního testování, kterého se autor práce aktivně zúčastnil.

V anonymní firmě bylo nalezeno celkem 1083 zranitelností, přičemž nejčastěji se dané zranitelnosti řešili nainstalováním nové verze SW. Často se pak také přenastavovala konfigurace jednotlivých částí systému. Rozložení nalezených zranitelností podle jejich závažnosti ukazuje následující tabulka.

Tabulka 1 - Nalezené zranitelnosti

Závažnost	Počet
Critical	**110**
High	**294**
Medium	**511**
Low	**168**
Grand Total	**1083**

Fakt o nejčastějším druhu opravy nalezených zranitelností potvrzuje i jedna z anket stránek www.darkreading.com, která se ptala několika desítek penetračních testerů, jaké jsou podle nich nejběžnější nalezené zranitelnosti31. Z té se můžeme dozvědět, že mezi nejčastěji nalezené zranitelnosti patří zastaralé verze operačních systému a software. Dále se často můžeme setkat s ponechaným výchozím nastavením na zařízeních nebo software a nebo administračním rozhraním aplikačních serverů vystavených do internetu.

Konzultant této práce přidává ze své zkušenosti ještě špatné nastavení Windows politik. Příkladem může být nevyžadování kombinace CTRL-Alt-Delete pro přihlášení, nevyžadování komplexity hesla nebo zobrazení naposledy použitého uživatelského jména. Politiky ke kontrole těchto chyb je snadné aplikovat přes Active Directory Group Policy konzoli, přesto to firmy často nedělají.

Nyní se již pojďme podívat na příklad vybraných nalezených zranitelností.

- **Microsoft Exchange Client Access Server Information Disclosure**

Riziko: B

Synopse: Je možné zjistit citlivé informace o vzdáleném mail serveru.

Popis: Microsoft Exchange Client Access Server obsahuje chybu, která vede k neoprávněnému získání informací o mail serveru. Neověřený útočník může této zranitelnosti zneužít k tomu, aby získal interní adresu mail serveru.

Doporučení: V současné době neexistuje žádná oprava, jedná se tedy o tzv. Zero day vulnerability32.

- **Switch používá výchozí jméno a heslo pro vzdálenou správu**

Riziko: C

Synopse: Je možné uhodnout přihlašovací údaje ke vzdálené konfiguraci switche.

Popis: Jeden ze switchů používá výchozí hodnoty pro přihlašování do webového rozhraní. Následně je možné měnit nastavení s plnými administrátorskými právy.

Doporučení: Změnit přihlašovací údaje.

- **SSLv3 padding vulnerability**

Riziko: B

Synopse: Je možné zjistit citlivé informace ze vzdáleného zařízení, které má povolené SSL služby.

Popis: Vzdálené zařízení je vystavené útoku typu Man-in-the-middle. Příčinou je zranitelnost protokolu SSL 3.0, konkrétně způsob, kterým protokol zpracovává tzv. Padding byty.

Doporučení: Zakázat SSL3 služby a místo nich používat TLS

Více

Více detailů můžeme nelézt například na adrese:

http://www.oracle.com/technetwork/topics/security/poodlecve-2014-3566-2339408.html

Nejvíce překvapivým zjištěním bylo nalezení výchozího hesla pro vzdálenou konfiguraci switche. Jinak se nám podařilo shromáždit dostatek informací o firemní topologii, ale samotné externí skenování zranitelnosti nalezlo méně zranitelností než testování interní. Důvodem je, že hlavní zaměření firemní IT bezpečnosti je právě na oblast, kterou externí testování testuje, společně s absencí penetračních testů firemního webu, který je častým zdrojem zranitelností.

Interní testy

- **Detekce nepodporované verze Windows 2000 a Windows XP**

Riziko: A

Synopse: Detekované operační systémy už nejsou podporované.

Popis: Na tyto operační systémy už není poskytovaná žádná podpora, nevychází na ně tedy ani žádné bezpečnostní záplaty.

Doporučení: Upgrade na novou verzi Windows

- **MS05-027: Zranitelnost v SMB umožňuje spuštění vzdáleného kódu**

Riziko: A

Synopse: Může být spuštěn škodlivý kód v důsledku chyby v SMB33 implementaci.

Popis: Windows verze 2000, 2003 a XP obsahují chybu v SMB, která umožní útočníkovi spustit škodlivý kód na vzdáleném počítači.

Doporučení: Nainstalovat záplatu kterou Microsoft vydal na tuto zranitelnost nebo nainstalovat novější operační systém.

- **SNMP agent používá výchozí název komunity (public)**

„Protokol SNMP (Simple Network Management Protocol) shromažďuje klíčové informace o síti a manipuluje s nimi. Data získává dotazováním zařízení ze stanice pro správu v pevných, či náhodných intervalech." 34

Riziko: B

Synopse: Může být uhodnut název komunity vzdáleného SNMP serveru.

Popis: Pokud si útočník opatří výchozí název komunity vzdáleného SNMP serveru, může měnit nastavení vzdáleného systému (pokud to systém umožňuje) a také může zjišťovat další informace o vzdálené stanici.

Doporučení: Pokud není používána, zakázat službu SNMP na vzdáleném zařízení nebo změnit výchozí název komunity.

- **Firemní notebooky a PC umožňují nabootování jiného OS z flash disku**

Riziko: B

Synopse: Může být spuštěn jiný operační systém, než Windows pod správou firemního IT.

Popis: Notebooky ani PC nemají zaheslovaný BIOS a zakázané bootování z jiného média, než primární hard disk. To může vést ke spuštění jiného operačního systému a kompromitaci celého souborového systému Windows, včetně extrakce zahashovaných přihlašovacích údajů k Windows.

Doporučení: Nainstalovat na všechna firemní zařízení heslo do BIOSu a upravit nastavení bootování pouze na primární hard disk.

Sociální inženýrství

V této fázi projektu byl kladen důraz na lidské chování ve smyslu odolnosti proti útokům tohoto typu. S pomocí této metody zjišťujeme rizika a slabiny lidského faktoru, struktury interní politiky a praktického využití standardů bezpečnosti.

Aplikovány byly 2 typy útoků:

- Test fyzického přístupu včetně zhodnocení „Clear desk" politiky

- Pokus o průnik do kanceláře v přestrojení za technika a zhodnocení možností napadení a informací, které může návštěvník získat

- Útok pomocí Phising emailů

V současnosti jedna z nejběžnějších hrozeb. Během tohoto útoku je poslán podvržený email z internetu převlečený za interní poštu. Ptáme se recipientů na soukromé informace, jako jsou například hesla, údaje z kalendáře či přístupy k datům. Uživatelé jsou také pozváni na neznámou webovou adresu monitorovanou našimi experty. Množství návštěv a akcí je použito na posouzení možnosti hrozby ze strany lidského faktoru.

- **Test fyzického přístupu do kanceláří**

Riziko: A

Popis : Kancelář je dostatečně vybavena pro sledování příchodu a odchodu osob a majetku. Nicméně i tak bylo možné se v kancelářských prostorech firmy pohybovat bez doprovodu, o samotě. V tomto případě pak útočník může provést cokoliv. Možnosti jsou nekonečné, od ukradení podstatného předmětu, přes focení či klonování informací až po uložení Wi-Fi routeru či vybudování permanentního přístupu. V našem případě jsme při testu penetrovali síťovou infrastrukturu instalací Wi-Fi routeru na recepci do firemní sítě.

Tímto krokem se může kdokoliv dostat do interní sítě z bezpečné vzdálenosti, minimálně z okolní ulice. S profesionálním vybavením roste tato vzdálenost na kilometry. Přístup k LAN vede k úplné kompromitaci firmy.

Během návštěvy byly stoly zaměstnanců a volně dostupné prostory prohledány za účelem nalezení utajených informací a bylo odneseno několik interních dokumentů.

Doporučení: Kvalitní výcvik bezpečnosti a ostražitosti pro zaměstnance v této oblasti. Taktéž doporučujeme zavést pravidlo, že každý návštěvník nebo technik se musí po prostorech kanceláře pohybovat pouze s doprovodem. Taktéž by bylo vhodné vybavit kancelář Wi-Fi kontrolními subsystémy jako je WIDS, tedy Wi-Fi Instrusion Detection System.

- **Phishing test**

Riziko: A

Popis : S dodaným seznamem emailů bylo možné poslat phising mail daným zaměstnancům a během několika hodin jsme získali z 50 cílů útoku 30 záznamů o uživatelském jménu a hesle. Smazáním jednoho testovacího loginu můžeme použít 29 záznamů, což dělá 58% úspěšnost. Toto číslo stále obsahuje dvojité pokusy některých uživatelů, čímž snižujeme závěrečný počet na 16, což odpovídá 32% úspěšnosti.

Získání hesel touto akcí je hlavní krok útoku proti společnosti. Otázkou je, nakolik skrytý může takový útok být. S dostatečným množstvím času může být tento přístup až individuální, což snižuje riziko prozrazení.

Doporučení : Kvalitní výcvik v bezpečnosti a ostražitosti pro zaměstnance v této oblasti. Jinými slovy je potřeba zúžit dostupnost k síti a tím dosáhnout menšího přístupu k LAN. Zároveň začlenění centrálního logu management/SIEM řešení k možnosti řízení dění na síti.

- **Test získání uživatelem uložených informace z internetového prohlížeče**

Riziko : A

Popis : Lidé mají tendenci k pohodlnosti, neměli bychom ale přijímat tyto tendence jako jsou například uložená hesla a cookies v oblastech zabezpečení informací.

Doporučení : Při ukončení prohlížeče by měli být všechny záložky a relace ukončeny, historie prohlížeče vyčištěna a cookies vymazány. Nejlepší způsob, jak se vyhnout oslabení bezpečnosti, je spouštět prohlížeč v inkognito módu.

- **Zhodnocení sociálního inženýrství**

Během tohoto průzkumu sociálním inženýrství bylo objeveno několik podstatných či dokonce kritických nálezů. Bylo možné se vydávat za interního technika, který přišel za účelem opravení tiskárny – přístup k LAN síti a instalace Wi-Fi routeru do LAN sítě umožnily přístup do LAN sítě společnosti na kilometry daleko.

Při klonování vlastního portálu společnosti a poslání jednoduchého phising emailu bylo zkompromitováno okolo 50% určených cílů, což vedlo k získání jejich hesla.

Výsledky testování

Následující zpráva popisuje nález získaný aktuálním externím a interním zhodnocením bezpečnosti a metodou sociálního inženýrství. Metody a nástroje použité během hodnocení byly výrazně omezeny závazky, pod kterými jsme se nacházeli, jmenovitě byl velký důraz kladen na zachování funkcionality všech business procesů, předpokladem tedy bylo nevyvolání systémového vypnutí či zhroucení, vzhledem k závažnosti následků způsobených těmito ději.Tato zpráva obsahuje pouze výsledky současného hodnocení a shrnutí odhalených bezpečnostních faktorů a slabin. Funkční části IT systému zákazníka nejsou zmíněny. Cíl této zprávy je nabídnout komplexní popis zabezpečení hodnoceného IT systému v čase hotovení zprávy pro management zákazníka.

- **Odhalené faktory ohrožení**

Bohužel, nehledě na kvalitně implementované elementy jsme odhalili malé množství kritických faktorů ohrožení v různých oblastech, které by měly být prozkoumány a vyřešeny IT pracovníky. Tyto faktory mohou mít negativní dopad a následně i mohou ovlivnit průběh práce zákazníka.

Ačkoliv jsme nenašli možnost přímého ovlivnění interního IT systému zákazníka, některé servery a zařízení mohou být viděny přímo z Internetu.

IT systém firmy může být také ohrožen z interní sítě, další kritické faktory ohrožení byly tedy objeveny i z tohoto směru. Tyto chyby mohou zapříčinit získání citlivých informací, uložených uvnitř systému, neautorizovanými uživateli. Škodliví uživatelé mohou zneužít tyto slabiny pro způsobení těžkostí v každodenním provozu zákazníka. Významná slabina během průzkumu zabezpečení, kterou se nám podařilo odhalit, bylo získání administrativní úrovně přístupu na dva Microsoft Windows servery. Důvody pro tyto slabiny jsou primárně v nedokonalosti v zacházení s heslem (například použití základně nastavených hesel). Slabé nastavení zabezpečení serveru (například nadbytečné služby způsobující možné ztráty informací) taktéž představuje závažné riziko.

Během hodnocení síťového provozu nastaly určité problémy s oprávněním a byly objeveny nezabezpečené proudy dat (například HTTP, telnet). Jako důsledek nedostatku šifrování, odchytávání interního síťového proudu dat je pouze částečně omezeno, díky čemuž mohou být získána hesla a hashe.

Navíc jsme objevili, že jsou používány zastaralé operační systémy a aplikace, což v důsledku umožňuje získání přístupu do interní sítě neautorizovanými uživateli.

- **Následky**

V případě, že bezpečností úroveň nebude opravena, může dojít k závažnému ohrožení na následujících úrovních IT systému :

- Úroveň obchodní
- Úroveň ochrany dat a informací
- Úroveň operační

- **Shrnutí**

Musíme podotknout, že námi zkoumaná infrastruktura je dobře navržená a zkonstruovaná.

Během průběhu tohoto projektu bylo objeveno několik podstatných či dokonce kritických informací. Bylo možné se vydávat za interního technika, který přišel za účelem opravení tiskárny – přístup k LAN síti a instalace Wi-Fi routeru do LAN sítě umožnily přístup do LAN sítě společnosti na kilometry daleko.

Při klonování vlastního portálu společnosti a poslání jednoduchého phising emailu bylo zkompromitováno okolo 50% určených cílů a byla získána jejich hesla.

Při porovnání výkonnostní úrovně testované firmy proti útoku sociálního inženýrství můžeme zkonstatovat, že úroveň je oproti jiným firmám průměrná. Vzhledem k velikosti a mezinárodnímu věhlasu firmy musíme upozornit, že by lidé měli být připraveni i na dlouhodobý skrytý útok využívající sociální inženýrství.

Návrh opatření pro snížení bezpečnostního rizika

Je třeba zdůraznit, že pro udržení očekávané úrovně zabezpečení je třeba průběžné údržby dle fráze „zabezpečení není produkt, ale spíše proces". Je běžně známo, že nejslabší článek obrany určuje zabezpečení celého systému. Doporučujeme tedy roční hodnocení IT systému zákazníka.

Po ukončení zhodnocení bezpečnosti doporučujeme zlepšit zabezpečení zákazníka na mezinárodně přijatelné standardy těmito způsoby :

1. Eliminace zastaralých verzí software
2. Spuštění projektu zlepšujícího systém ve smyslu eliminování bezpečnostních problému, například slabé či defaultní konfigurace s interní kontrolou kvality.
3. Používání adekvátních hesel a managementu přístupů
4. Interní zlepšení IT a bezpečnostního personálu pomocí znalostí a nástrojů potřebných na získání kontroly zabezpečení skrz bezpečnostní monitoring a analýzu řešení
5. Provádění ročního hodnocení bezpečnosti IT systému – nebo minimálně po výrazném IT pokroku – vzhledem ke změnám v systému a pro uzavření uveřejněných mezer v bezpečnosti

Po skončení projektu, hodnotícího zabezpečení systému, jsme rozšířili doporučení na eliminování všech identifikovaných rizik a seřadili tato rizika vzhledem k jejich důležitosti. Paralelně k tomuto řazení pak byl vytvořen checklist. Cílem tohoto checklistu pak bylo zdůraznění a shrnutí, jak pro IT manažery, tak pro celé vedení, úkolů výrazně doporučených pro zlepšení zabezpečení IT systému a také pro možnost sledování přijatých opatření.