Grafy útokov v kybernetickej bezpečnosti

Analýza a návrh riešenia

Bc. Vladimír Homola

2Im, 2022 – 2023

Abstrakt. V práci sa venujeme spracovaniu bezpečnostných údajov, analýze kybernetických útokov, technikám modelovania útokov, konkrétne generovaniu ich grafov. Bližšie sa venujeme nástroju pre generovanie grafov útokov – MulVAL. Tento nástroj je už pre dnešné potreby zastaralý a navyše nie je jednoduché ho na dnešných zariadeniach spustiť. Po preštudovaní aktuálneho stavu poznania pre tento nástroj sme sa rozhodli, že pôvodné jadro nástroja, ktoré používa logické programovanie, necháme nezmenené a prerobíme len vstup a výstup nástroja. Na vstupe bude hlavnou zmenou, že zraniteľnosti sa budú sťahovať z externej databázy a budú stále aktuálne. Na výstupe chceme, aby vygenerovaný graf bol lepšie čitateľný.

Kľúčové slová: kybernetická bezpečnosť, graf útoku, MITRE rámec, MulVAL

Obsah

Zoznam ilustrácií

Úvod

Informačná a kybernetická bezpečnosť zasahujú do mnohých aspektov spoločenského, politického a obchodného života. Majú obrovské dôsledky pre sieťovú a osobnú bezpečnosť jednotlivcov a rodín. V roku 2021 boli priemerné náklady súvisiace s porušením ochrany údajov celosvetovo 4,24 milióna dolárov. Hoci mnohé porušenia ochrany údajov vedú k ohrozeniu osobných údajov, množstvo dobre zverejnených útokov proti dopravným, lekárskym a priemyselným kontrolným systémom preukázalo, že narušenia kybernetickej bezpečnosti môžu mať vážny dopad na osobnú bezpečnosť.

Mitigačné stratégie sa pomerne často zameriavajú na ochranu systémov pred útočníkmi, ktorí majú zámer spôsobiť úmyselné poškodenie systému a/alebo údajov. Avšak k mnohým zlyhaniam kybernetickej bezpečnosti dochádza jednoducho ako dôsledok správania používateľa. Často v dôsledku neúmyselných chýb spôsobených neúplným chápaním bezpečnostných mechanizmov.

Obrana rozsiahlych podnikových sietí pred útokmi je náročná úloha, ktorej čelia dnešní správcovia sietí. Obranné prístupy proti takýmto útokom boli tradične väčšinou zamerané na hostiteľa, pričom pozornosť sa venovala identifikácii slabých miest jednotlivých hostiteľov a prijímaniu opatrení na ich mitigáciu. Nástroje na skenovanie zraniteľností poskytujú informácie o zraniteľnosti jednotlivých hostiteľov a pomáhajú pri dosahovaní týchto cieľov. Jedným z hlavných problémov tohto prístupu je však to, že kladie väčší dôraz na lokálne špecifické informácie o hostiteľovi a nezohľadňuje ich vo svetle globálneho bezpečnostného kontextu siete.

Vnímanie kybernetických útokov je dôležitým výskumným problémom, ktorý si vyžaduje lepšie techniky a metódy napomáhajúce vnímaniu a posudzovaniu kybernetických útokov. Pomerne často je pre pozorovateľov ťažké si predstaviť analýzu a pochopenie zložitých vzorcov. Dobre navrhnuté diagramy a grafické systémy môžu tomuto procesu pomôcť.

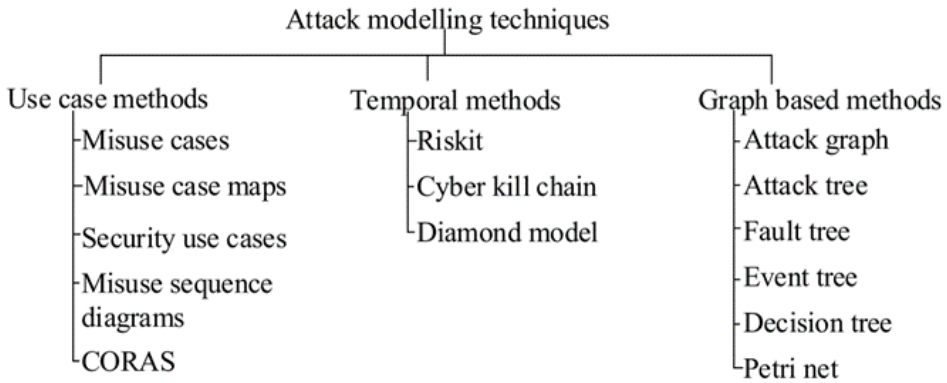
1 Techniky modelovania útokov

Techniky modelovania útokov (AMT) sa používajú na modelovanie a vizualizáciu sekvencie a/alebo kombinácie udalostí, ktoré umožňujú úspešný kybernetický útok na počítač alebo sieť. AMT možno rozdeliť do troch kategórií:

- metódy, ktoré sú založené na rámci prípadov použitia,
- metódy, ktoré predstavujú kybernetický útok z časovej perspektívy a
- metódy založené na grafoch.

Tieto metódy sú zvýraznené na Obr. 1. Z metód načrtnutých na tomto obrázku sú grafy útokov a stromy útokov najobľúbenejšou metódou znázornenia kybernetických útokov.

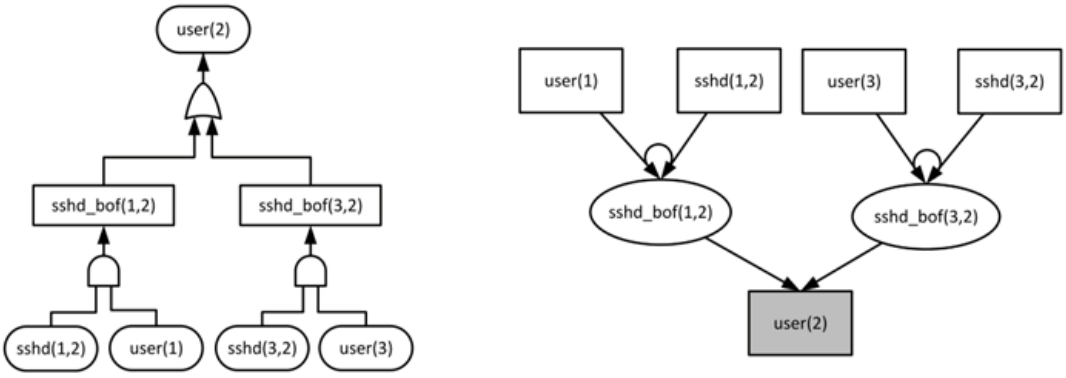
AMT umožňujú pozorovateľom vyhodnotiť najdôležitejšie informácie v diagrame a pomáhajú odstraňovať intelektuálnu záťaž z bezpečnostných expertov – ktorí musia vyhodnocovať scenáre kybernetických útokov a vyhodnocovať potenciálne opatrenia (mitigácie). V dôsledku toho môžu byť bezpečnostné problémy prezentované spôsobom, ktorý umožňuje rozhodovateľovi – či už odborníkovi alebo neodborníkovi, rýchlejšie pochopiť problém, lepšie vnímať rizikové aspekty a ľahšie vnímať zložité koncepty. Za takýchto okolností AMT poskytujú efektívne nástroje a robia tento proces jasnejším a jednoduchším, a tým uľahčujú diskusiu a môžu pomôcť vnímať kybernetické útoky.



Obr. 1 Techniky modelovania útoku [1]

Príklad dvoch AMT – strom chýb a graf útoku je uvedený na Obr. 2. Príklad na Obr. 2 ukazuje, ako je útočník schopný vykonať sériu nástrojov využívajúcich zraniteľnosti (exploitov) (sshd_bof) na sekvencii hostiteľských výpočtových zariadení

(označených v zátvorkách), a tým získať užívateľské privilégiá (user) na každom z nich. Príklad tiež ukazuje jeden z predpokladov (sshd), ktoré sú nevyhnutné na to, aby bol útok úspešný. Tento príklad ukazuje, ako je možné vizualizovať sled zneužití (exploitov), aby sa pomohlo vnímaniu kybernetických útokov.



Obr. 2 Modely útoku (naľavo strom porúch a napravo graf útoku) [1]

V nasledujúcich podkapitolách si bližšie priblížime viac diamantový model, kill-chain (reťaz zabíjania) model a graf útoku.

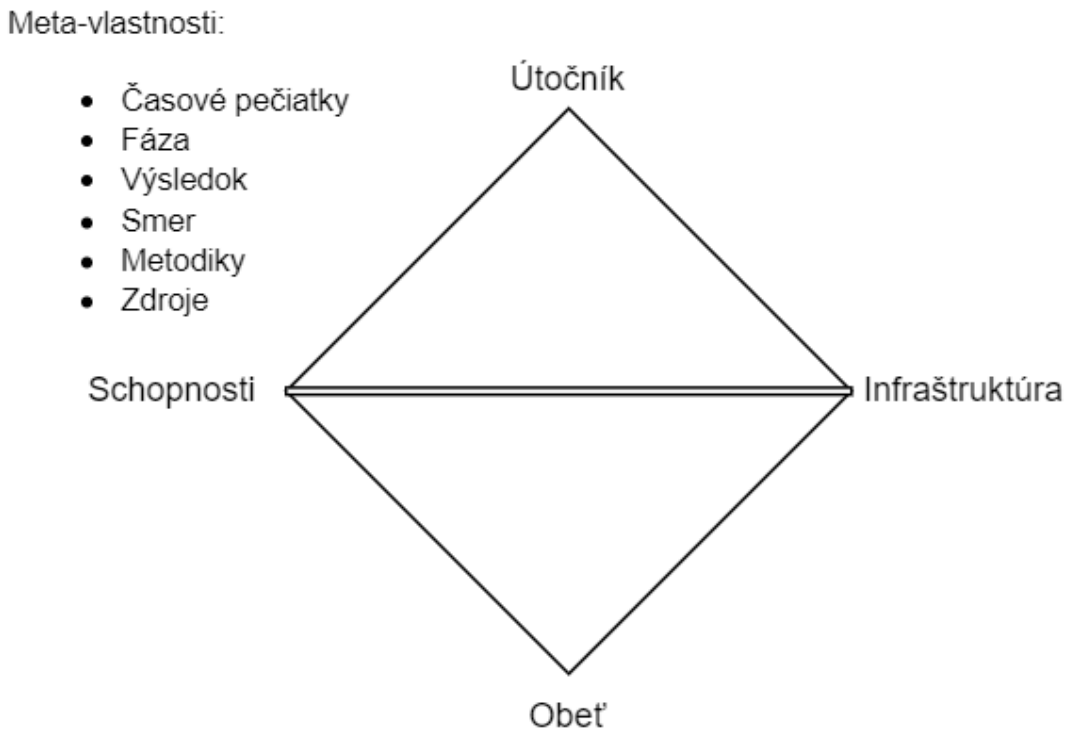
1.1 Diamantový model

Diamantový model môžeme označiť ako jednoduchý, pretože pozostáva len zo štyroch hlavných komponentov. Je jedným z nových modelov analýzy kybernetického útoku, kde útočník útočí na obeť v závislosti od dvoch kľúčových motivácií namiesto použitia série krokov, ako je kill-chain alebo graf útoku [15]. Tento model pozostáva zo štyroch základných prvkov, akými sú útočník, infraštruktúra, schopnosti a obeť. Útočníkom je aktér (alebo skupina aktérov), ktorý zaútočí na obeť po analýze svojich schopností voči obeti. Na začiatku útočník začína bez znalosti schopností obete. Po analýze schopností obete môže útočník zistiť, že má väčšie schopnosti ako obeť a zaútočiť. Tento model je dôležitý pri zaoberaní sa s pokročilejšími útočníkmi, to sú napr. takí, ktorí už získali určitú kontrolu nad sieťou. Útočník tiež analyzuje svoje technické a logické schopnosti ovládať akúkoľvek časť siete obete [15]. Podobu diamantového modelu možno vidieť na Obr. 3. Diamantový model tiež obsahuje niektoré meta-vlastnosti:

- časové pečiatky
- fáza

- výsledok
- smer
- metodiky
- zdroje

V prípade kybernetického útoku diamantový model identifikuje fázy na základe časovej pečiatky. Komponenty diamantového modelu možno vidieť na Obr. 3, ktorý znázorňuje, že útočník hľadá príležitosť zaútočiť na obeť v závislosti od schopností alebo infraštruktúry.



Obr. 3 Diamantový model [15]

1.2 Kill-chain (reťaz zabíjania)

Reťaz zabíjania je jednou z techník modelovania útoku, ktorá definuje útok ako reťaz akcií (udalostí). Konkrétne sa zameriava na modelovanie tzv. štruktúrovaných útokov, keďže postup útočníka sa dá vyjadriť v usporiadanej reťazi útočníkových krokov. Technika reťaze zabíjania je opísaná Ministerstvom obrany USA ako útok na cieľ [15], kde definovali niektoré fázy, ako napríklad nájsť (find), zamerať (fix), sledovať (track), zacieliť (target), zasiahnuť (engage) a odhadnúť (assess). Reťaz zabitia sa používa aj

v iných oblastiach ako je kybernetická bezpečnosť. V tejto oblasti sa používa hlavne na opis jednotlivých krokov útoku. Postupný výskum rozdelil reťaz zabíjania na sedem krokov útoku [15]:

Krok 1 Prieskum (Reconnaissance): Útočník zhromažďuje informácie pred útokom. Informácie je možné zbierať z internetu, ktorý je verejne dostupný.

Krok 2 Zbrojenie (Weaponization): Útočník vytvorí správu so škodlivým obsahom (náložou), ktorý odošle obeti. Škodlivou náložou môže byť vírus, trójsky kôň alebo spustiteľný súbor, ktorý môže vykonať nejakú akciu na počítači obete alebo v sieti.

Krok 3 Doručenie (Delivery): Útočník odošle škodlivý obsah obeti pomocou nejakého komunikačného prostriedku. Útočník môže poslať škodlivú nálož e-mailom ako prílohu alebo odkaz, ktorý stiahne škodlivú nálož.

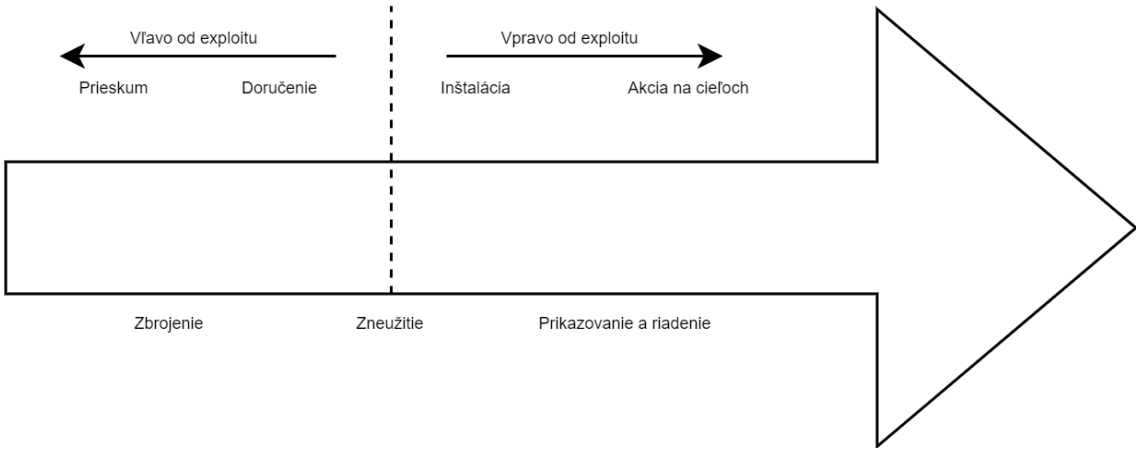
Krok 4 Zneužitie/Exploitácia (Exploitation): V tejto fáze dochádza k skutočnému zneužitiu. Ak si obeť stiahne škodlivý obsah do svojho počítača, začne sa hlavné zneužívanie. Toto je fáza, v ktorej útočník potrebuje pomoc obete. Toto je tiež jedna z fáz, kedy môže byť reťaz prerušená nestiahnutím škodlivého obsahu, ktorý útočník posiela.

Krok 5 Inštalácia (Installation): Nainštaluje sa malvér na infikovaný počítač alebo počítač obete. Na infikovanie počítača obete môže byť potrebné, aby samotná obeť spustila škodlivý obsah, alebo sa môže spustiť automaticky. Toto je tiež fáza, v ktorej môže byť reťaz prerušená nespustením škodlivého obsahu.

Krok 6 Prikazovanie a riadenie (Command and control): Prostredníctvom nainštalovaného malvéru útočník vytvorí príkazový a riadiaci kanál (command and control channel) na prístup k interným aktívam obete. V tejto fáze útočník úspešne získal kontrolu nad strojom obetí.

Krok 7 Akcia na cieľoch/obetiach (Action on objectives): Útočníci dosiahnu svoj cieľ na infikovanom počítači alebo sieti obete. Toto môže byť vstupná brána útoku. Útočník môže napr. postupovať smerom k cenným údajom z databázy cez webový server.

Reťaz zabíjania je rozdelená na dve hlavné fázy/časti, ktoré sa nazývajú vľavo od exploitu alebo prieniku a vpravo od exploitu alebo prieniku. Na Obr. 4 môžeme vidieť kroky reťaze zabíjania, kde môžeme pozorovať, že ľavá časť reťaze zabíjania dáva obeti príležitosť/možnosť reťaz prerušiť. Ak sa útočníkovi podarilo dostať do pravej časti, pre obeť to bude náročné útok zastaviť alebo znížiť straty.



Obr. 4 Model reťaze zabíjania (kill-chain) [15]

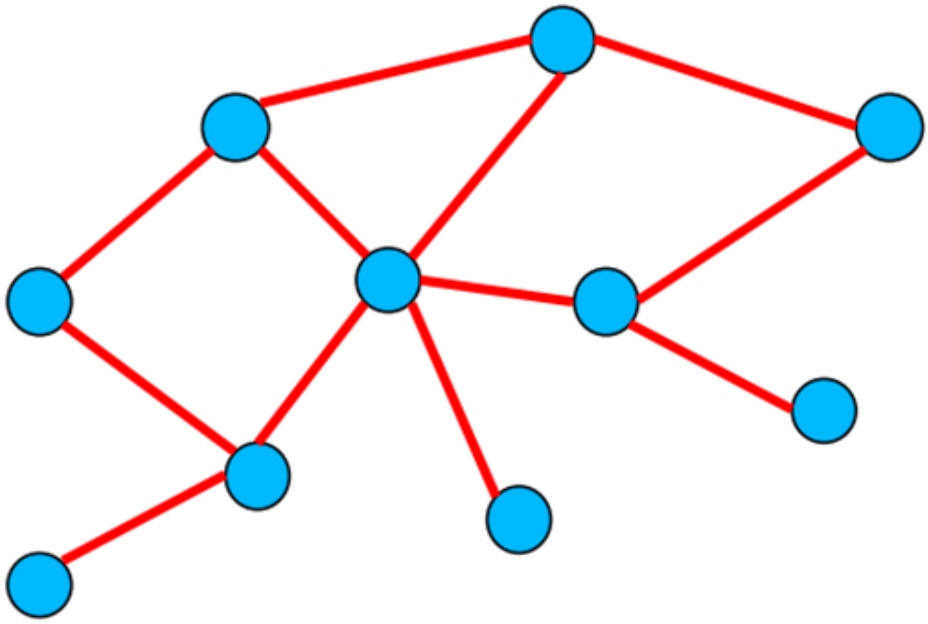
Ľavá časť je počiatočná fáza, v ktorej sa útočník pokúsi získať kontrolu nad systémom. Napríklad útočník chce získať prístup do systému, začne prieskum, to znamená, že zhromažďuje informácie pred útokom, skúma sieť obetí, profilov a iných profilov dostupných na internete. Predpokladajme, že sa útočníkovi podarilo získať e-mailovú adresu obete. E-mail sa môže stať bránou na vstup do systému alebo siete obetí. Bežné zbrane môžu byť trojanizované PDF, PPT, DOC, BNP atď., súbory rôznych typov odosielané obeti ako príloha. Odosielateľ (útočník) sa snaží byť legitímny, ako len môže. Môže do mailu tiež umiestniť odkaz na kliknutie. Predmet e-mailu, ID a názov domény by mali vyzerať ako skutočné. Po doručení e-mailu útočník čaká na odpoveď obete. V takejto situácii si obeť môže stiahnuť prílohu alebo kliknúť na odkaz, alebo len vymazať e-mail, ktorý sa jej zdá podozrivý. Exploitácia hrá v tejto situácii zásadnú úlohu, pretože ak obeť klikne alebo stiahne prílohu mailu, reťaz zabíjania pokročí (posunie sa na ďalší krok), t. j. na počítači obete sa spustí malvér [15].

V tejto súvislosti je jasné, že pre každú organizáciu alebo jednotlivca je pre kybernetickú obranu najdôležitejšia fáza vľavo od exploitu. Z analýzy fázy vľavo od exploitu môžeme nájsť množstvo informácií o takýchto útokoch. Napríklad to môže

indikovať spôsob útoku útočníkov. Odosielanie IP, e-mailovej domény, miesta a ďalších relevantných informácií možno získať na obranu obete. Pochopenie spôsobu útoku útočníka môže tiež pomôcť organizáciám poučiť zamestnancov o tom, ako sa s takýmto útokom vysporiadať. Vo väčšine prípadov môže byť reťaz zabíjania v tejto situácii prerušená v počiatočnom štádiu. Pochopenie reťaze zabíjania môže organizáciám pomôcť chrániť sa pred potenciálnymi kybernetickými útokmi [15].

1.3 Grafy kybernetických útokov

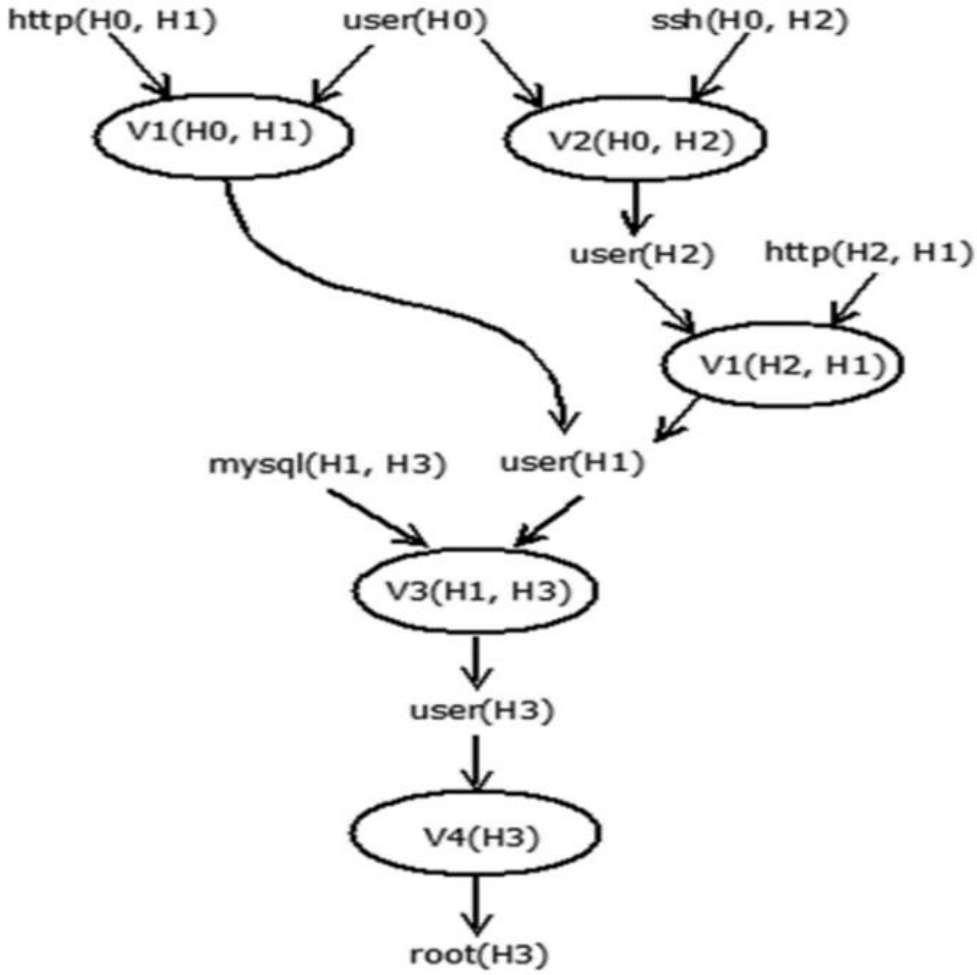
V našej práci sme si na znázornenie sledu udalostí, ktoré môžu viesť k úspešnému kybernetickému útoku, vybrali grafy útokov. Na pochopenie toho, čo je graf útoku si najprv definujeme, čo je to graf. Grafom G nazývame dvojicu (V, E), kde V je množina vrcholov grafu (kruhy) a E ($E \subseteq V \times V$) je množina hrán grafu (čiary, šípky). Príklad znázornenia grafu môžeme vidieť na Obr. 5.



Obr. 5 Znázornenie grafu

Graf útoku potom definujeme ako znázornenie všetkých ciest cez systém (reprezentovaný grafom), ktoré končia v stave, kde útočník úspešne dosiahol svoj cieľ. Grafy útokov sú koncepčné diagramy používané na analýzu toho, ako môže byť cieľ napadnutý. Je to dôležité pri analýze kybernetických hrozieb na počítačovom systéme alebo sieti. Graf útoku je stromovo štruktúrovaný graf, ktorý má viacúrovňových

potomkov s jedným koreňom. Graf útoku môžeme vidieť na Obr. 6. V tomto konkrétnom prípade sa útočníkovi podarilo získať administrátorské (root) oprávnenia na hostiteľovi H3. Môžeme si všimnúť, že tento graf sa trochu líši od grafu útoku z Obr. 2. Existuje totiž viac typov grafov útoku a prístupov na ich generovanie. V ďalšej časti práce si ich viac rozoberieme a naučíme sa ich čítať.



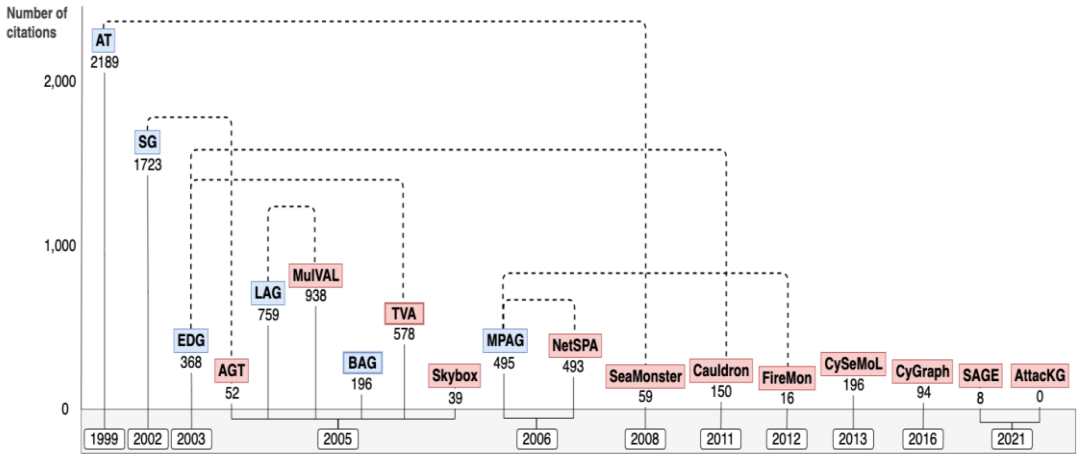
Obr. 6 Graf kybernetického útoku [3]

Graf útoku v podstate pozostáva z uzlov (vrcholov) a pri zobrazení konkrétneho útoku môže byť zložitý, to znamená, že môže obsahovať tisíce uzlov s množstvom rôznych ciest. Generovanie grafov útokov je výpočtovo zložitý problém najmä v prípade veľkých sietí [16]. Hlavnou myšlienkou grafu útoku je cesta od útočníka k sieti obete. Techniky grafov útoku pomáhajú odhaliť prieniky do systému a jeho zraniteľnosti. Grafy útokov môžu byť užitočné v mnohých oblastiach bezpečnosti počítačových sietí vrátane detekcie narušenia, forenznej analýzy, analýzy rizík a kybernetickej obrany. Správca siete používa graf útoku na identifikáciu [15]:

- Zraniteľnosti systému
- Ako môže/mohlo k útoku dôjsť
- Množiny krokov, ktoré zabránia útočníkovi dosiahnuť svoj cieľ

Hlavnou výhodou grafu útokov je, že pomáha identifikovať potenciálne útoky na sieť. Analýza pomáha identifikovať potrebné kroky, ak existuje nejaké slabé miesto v sieti. Pomocou tejto techniky je možné vypočítať návratnosť investícií (ROI – Return on Investment) pre bezpečnosť. Organizácie sa predovšetkým vyhýbajú kontrole bezpečnosti alebo zraniteľnosti, pretože je to drahé. Na druhej strane, ak sú náklady príliš vysoké na to, aby si ich spoločnosť mohla dovoliť, je nepravdepodobné, že sa spoločnosť rozhodne pre drahú možnosť. Spoločnosti teda potrebujú jasnú víziu investícií do kybernetickej bezpečnosti [15].

Generovanie grafu útoku je náročná úloha, pretože graf môže obsahovať stovky uzlov, čo sťažuje identifikáciu platnej hrozby útoku. Technika grafu útoku zahŕňa aj množstvo neistôt. Na riešenie týchto neistôt niektorí výskumníci používajú algoritmus Monte Carlo [16], pretože sa dokáže vysporiadať s neistotou a má štatistické závislosti. Na vytvorenie grafu sa používajú aj iné algoritmy, ako napríklad prehľadávanie do hĺbky a šírky. Existuje množstvo teoretických prác, ktoré boli vykonané v oblasti generovania grafov útokov. Na Obr. 7 môžeme vidieť vývoj metód generovania grafov útokov (x-ová os) a počet citácií jednotlivých reprezentácií (typov) grafov útokov (GÚ, modrá farba) a príslušných nástrojov generovania grafov útokov (červená farba).



Obr. 7 Vývoj metód generovania grafov útokov: rok vydania a počet citácií reprezentácií GÚ (modrá) a príslušných nástrojov generovania GÚ (červená) [17]

Na Obr. 8 sa nachádza tabuľka, ktorá obsahuje porovnanie rôznych nástrojov na generovanie a vizualizáciu grafov útokov. Jednotlivé stĺpce od prvého po posledný obsahujú nasledujúce parametre:

- Meno (Name)
- Vývojári (Developers)
- Dostupnosť (Accessible)
- Typ grafu útoku (AG Type)
- Škálovateľnosť (Scalability)
- Intuitívna úroveň (Intuitive Level)
- Rok (Year)
- Počet citácií (No. of References)
- Práca na vyhľadanie (Paper search)
- Nástroj na vyhľadanie (Tool search)

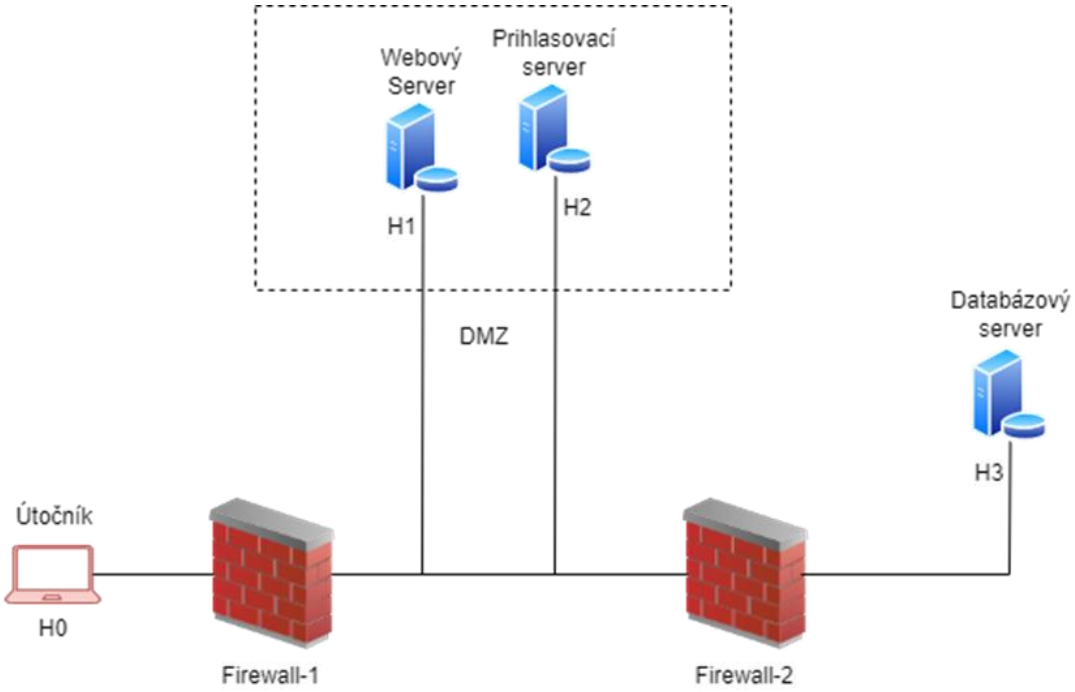
Name	Developers	Accessible	AG Type	Scalability	Intuitive Level	Year	No. of References	Paper Search	Tool Search
Attack Graph Toolkit	Carnegie Mellon University	Open source	SG	Poor, Exponential	Fair	2005	52	"Scenario graphs applied to security": 16	["Attack Graph Toolkit"]: 52
MulVAL	Kansas State University	Open source	LAG	$O(N^2)$ - $O(N^3)$	Good	2005	938	"A scalable approach to attack graph generation": 757	["MulVAL"]: 938
TVA	George Mason University	Not open source, difficult to obtain	EDG	$O(N^3)$	Good	2005	578	"Topological analysis of network attack vulnerability": 578	["Topological Vulnerability Analysis"]: 547
Skybox View	Skybox Security, Inc.	Commercial Software	Unknown	$O(N^3)$	Good	2005	39	"Proactive Security for a Mega-Merger": 39	["skybox view" "attack graph"]: 15
NetSPA	Massachusetts Institute of Technology	Not open source, difficult to obtain	MPAG	O(NlgN)	Fair	2006	493	"Practical attack graph generation for network defense": 493	["NetSPA"]: 357
SeaMonster	Norwegian Univ. of Science and Technology and SINTEF research foundation	Open source	AT	Polynomial	Fair	2008	59	"SeaMonster: Providing tool support for security modeling": 37	["seamonster" "attack tree"]: 59
Cauldron	PROINFO Company, George Mason University	Commercial Software	EDG	$O(N^3)$	Good	2011	150	"Cauldron mission-centric cyber situational awareness with defense in depth": 142	[Cauldron "attack graph"]: 150
FireMon	FireMon, Massachusetts Institute of Technology	Commercial Software	MPAG	O(NlgN)	Good	2012?	16	No paper	["firemon" "attack graph"]: 16
CySeMoL	Royal Institute of Technology, Stockholm, Sweden	Not open source, difficult to obtain	Unknown	Polynomial?	Not Provided	2013	196	"The Cyber Security modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures": 168	["cysemol"]: 196
CyGraph	MITRE	Not open source, difficult to obtain	Unknown	Scales well[a]	Very Good[b]	2016	94	"CyGraph: graph-based analytics and visualization for cyber security": 94	["cygraph" "attack graph"]: 46
SAGE	Delft University of Technology, Netherlands, Rochester Institute of Technology, US	Open source	Alert-driven	NA[c]	Good	2021	8	"Alert-driven Attack Graph Generation using S-PDFA": 2	["SAGE" "attack graph"]: 8
AttacKG	Zhejiang University, National University of Singapore, Northwestern University	Open source	CTI-based	NA[c]	Fair	2021	0	"Attackg: Constructing technique knowledge graph from cyber threat intelligence reports": 0	["AttacKG" "attack graph"]: 0

Obr. 8 Porovnanie nástrojov na generovanie a vizualizáciu grafov útokov [17]

Ďalej môžeme vidieť, že veľa z daných nástrojov nie je voľne dostupných (open-source) a dostať sa k nim je veľmi náročné. Z voľne dostupných nástrojov a celkovo všetkých nástrojov je najpoužívanejší nástroj MulVAL (má najviac citácií). Tomuto nástroju sa bližšie venujeme v podkapitole 3.1 a súvisí s ním aj náš hlavný cieľ práce – zmodernizovať tento nástroj.

2 Typy grafov kybernetických útokov a prístupy ich generovania

V tejto kapitole sa zameriame na tri konkrétne typy grafov útokov (graf privilégií, exploit dependency graf, logický graf) a prístupy ich generovania (prístup sčítania stavov, prístup topologickej analýzy zraniteľností, prístup logického programovania). Všetky tri prístupy aplikujeme na sieť zobrazenú na Obr. 9. V tejto konfigurácii siete Firewall-1 riadi prevádzku medzi vonkajšou a vnútornou sieťou. Predpokladaná lokácia útočníka je na hostiteľovi H0 vo vonkajšej sieti. V demilitarizovanej zóne (DMZ) beží webový server na hostiteľovi H1 a prihlasovací server (cez ssh) na hostiteľovi H2. Webová služba vyžaduje prístup k back-end databázovému serveru, ktorý beží na hostiteľovi H3. Firewall-1 umožňuje http a ssh prenos na webový server a prihlasovací server a blokuje všetku ostatnú prevádzku. Firewall-2 umožňuje prístup k databázovému serveru iba z webového servera. Hostiteľ H1 používa zraniteľnú verziu webového servera Apache, ktorá má zraniteľnosť V1 (CVE-2006-3747), ktorá umožňuje vzdialenému útočníkovi zneužiť a získať používateľské oprávnenia na webovom serveri. Služba ssh na H2 má zraniteľnosť V2 (CVE-2002-0640), ktorá umožňuje vzdialeným útočníkom získať používateľské oprávnenia. Databázový server H3 je linuxový box s databázou MySQL, ktorá má vzdialene zneužiteľnú zraniteľnosť V3 (CVE-2009-2446), ktorá umožňuje útočníkovi získať používateľské oprávnenia. Linuxové jadro v hostiteľovi H3 má tiež zraniteľnosť V4 (CVE-2004-0495), ktorá umožňuje miestnemu používateľovi získať oprávnenia roota. Cieľom útočníka je získať oprávnenia roota na databázovom serveri.



Obr. 9 Jednoduchá sieť

2.1 Graf privilégií

Každý uzol v grafe privilégií predstavuje množinu privilégií vlastnených používateľom alebo množinou používateľov a každá hrana predstavuje zraniteľnosť. V strome útokov každá cesta k listovému uzlu predstavuje postupnosť útokov, pomocou ktorých môže útočník dosiahnuť cieľový stav zo svojho počiatočného stavu. Graf útoku je v podstate upevnená reprezentácia stromu útokov, kde sú niektoré alebo všetky spoločné uzly naprieč rôznymi cestami útoku zlúčené.

2.2 Prístup sčítania stavov

Prístupy založené na sčítaní stavov boli počiatočnými pokusmi o automatizované generovanie grafu útoku. Pri tomto prístupe sa používa tzv. graf sčítania stavov. Uzly v ňom predstavujú možný stav systému počas vykonávania útoku. Stav systému pozostáva z informácií o hostiteľovi (hostiteľoch), úrovniach prístupu používateľov a doterajších účinkoch útoku. Hrany predstavujú zmenu stavu systému spôsobenú jediným zásahom (akciou) útočníka a môžu byť ohodnotené na základe úsilia útočníka alebo času potrebného na úspech. Na vstupe pre samotný algoritmus generovania grafu sú: šablóny útoku (predstavujú útoky (známe aj predpokladané) vo forme podgrafu popisujúce

podmienky na úspešné vykonanie útoku a tiež nové podmienky, ktoré vzniknú po úspešnom vykonaní útoku), konfiguračný súbor (obsahuje informácie o uvažovanom sieťovom systéme, tieto informácie zahŕňajú topológiu siete, konfiguráciu sieťových prvkov, ako sú hostitelia, smerovače, firewally atď.) a profil útočníka (obsahuje informácie o schopnostiach útočníka). Samotný algoritmus generovania grafu útoku začína od počiatočného stavu. Priraďuje šablóny útokov konfigurácii sieťového systému a profilu útočníka dopredným spôsobom a generuje graf iteratívne. Tento prístup generovania však trpí problémom cyklu v grafe, eliminovaním nadbytočných uzlov a ciest, exponenciálnym priestorom stavov, čím je v praxi nevyužiteľný.

2.3 Exploit dependency graf

Tento typ grafu útoku používa dva typy uzlov: uzly exploitu a uzly bezpečnostných podmienok. Exploit uzly predstavujú útoky (zneužitie určitých zraniteľností) a uzly bezpečnostných podmienok predstavujú buď predpoklady útoku alebo následky útoku (exploit je nimi definovaný). Orientované hrany z uzlov bezpečnostných podmienok do útočných uzlov predstavujú predpoklady útoku, z ktorých všetky musia byť splnené, aby bol útok úspešný. Orientovaná hrana z útočného uzla do uzla bezpečnostnej podmienky predstavuje následky útoku. Výhodou grafov exploit dependency je, že namiesto modelovania hostiteľov sa modelujú exploity na hostiteľoch, čím sa znižuje výpočtová zložitosť. Na druhej strane tento model vyžaduje nízko-úrovňové informácie o útoku.

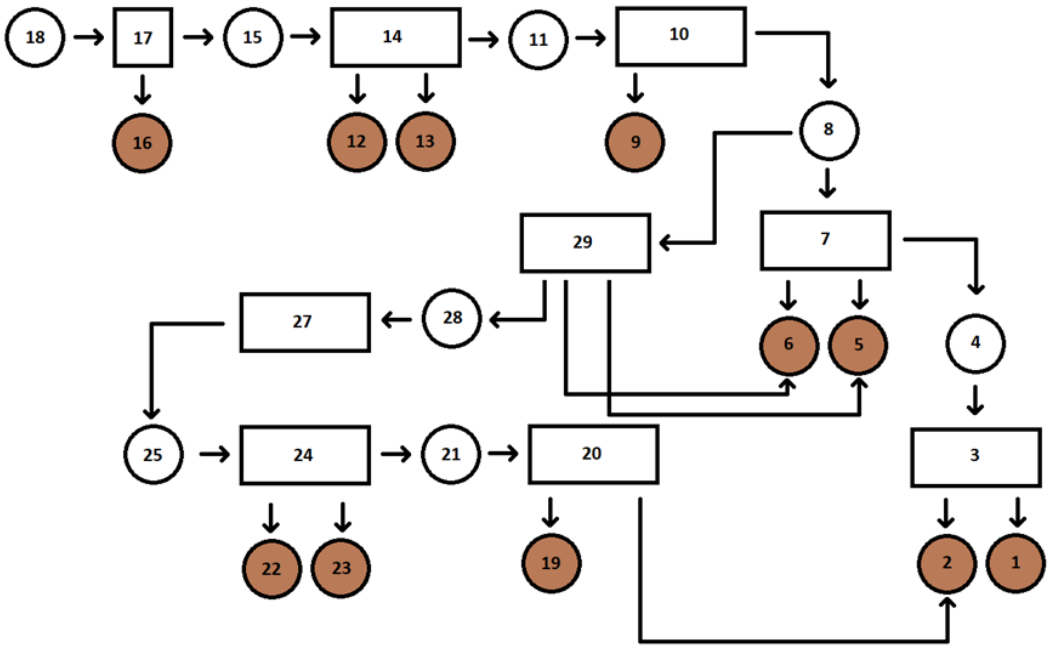
2.4 Prístup topologickej analýzy zraniteľností

Skoršie prístupy ku generovaniu grafov útokov trpeli problémami so škálovateľnosťou, pretože reprezentácia grafu útoku použitá v týchto prístupoch, t. j. graf sčítania stavov predpokladal úplný exponenciálny stavový priestor. Predpoklad monotónnosti správania útočníka bol kľúčovým faktorom pri riešení tohto problému. Tento predpoklad hovorí, že predpoklady jedného útoku sa nikdy nezrušia úspešným vykonaním ďalšieho útoku. Hoci to nemusí byť pravda vo všetkých prípadoch (t. j. útok pretečenia vyrovnávacej pamäte služby spôsobí jej ukončenie, čím sa zabráni ďalšiemu použitiu pri iných útokoch), predpoklad monotónnosti pomáha znižovať zložitosť analýzy z exponenciálnej na polynomiálnu. V najhoršom prípade má tento prístup počet uzlov kvadratický vzhľadom k počtu exploitov. V grafe útoku exploit dependency sa

každý exploit alebo závislosť objaví iba raz a medzi nezávislými exploitmi nie sú žiadne hrany. Zatiaľ čo v grafe útoku sčítania stavov môžu existovať hrany medzi exploitmi, aj keď medzi nimi neexistujú žiadne závislosti. Teraz sa vrátime k Obr. 4. V exploit dependency grafe ovály predstavujú exploity a sú označené zodpovedajúcimi zraniteľnosťami. Ostatné uzly predstavujú buď nejaký stav siete, alebo schopnosť útočníka. Napríklad stav siete http (H0, H1) znamená dostupnosť webovej služby na hostiteľovi H1 z hostiteľa H0. Schopnosť útočníka user(H0) znamená, že útočník má oprávnenia používateľa na hostiteľovi H0. Orientované hrany do a von z exploit uzlov vyjadrujú predbežné a následné podmienky útoku. Napríklad využitie zraniteľnosti MySQL CVE-2009-2446 na hostiteľovi H3 z hostiteľa H1, t.j. V3(H1, H3) vyžaduje predbežné podmienky user(H1) a mysql(H1, H3) a generuje následnú podmienku (následok) user(H3). V situácií na obrázku 4 teda existujú dve cesty útoku vedúce k tomu, že útočník získa oprávnenie root na H3. Sú to V1 (H0, H1) → V3 (H1, H3) → V4 (H3) a V2 (H0, H2) → V1 (H2, H1) → V3 (H1, H3) → V4 (H3).

2.5 Logický graf

Uzol v grafe logického útoku je logickým tvrdením, ktorý kóduje iba určitú časť stavu siete. Na rozdiel od grafu sčítania stavov nereprezentuje ani nekóduje celý stav siete. Hrany predstavujú kauzálne vzťahy medzi rôznymi konfiguráciami siete (hrany v grafe logického útoku predstavujú vzťah „závisí od“). Veľkosť logického grafu útoku je polynomiálna vzhľadom na analyzovanú sieť. Na Obr. 10 sa nachádza logický graf útoku zodpovedajúci konfigurácii siete uvedenej na Obr. 9. Obsahuje dva typy uzlov. Derivačné uzly majú tvar obdĺžnika a uzly faktov tvar kruhu. Derivačné uzly sú označené pravidlami interakcie (3.2.6.5) a uzly faktov sú označené logickými tvrdeniami vo forme predikátu aplikovaného na jeho argumenty. Hnedé kruhy sú primitívne uzly faktov, t. j. fakty, ktoré platia v počiatočnom stave. Biele kruhy predstavujú odvodené uzly faktov, t. j. nové fakty, ktoré sa generujú ako výsledok aplikácie pravidiel interakcie na existujúce fakty.



Obr. 10 Logický graf útoku

Tu je úplný list označení jednotlivých uzlov logického grafu na Obr. 6:

1. hacl(H0, H1, httpProtocol, httpPort)
2. located(Attacker, H0)
3. direct network access
4. netAccess(H0, H1, httpProtocol, httpPort)
5. networkService(H1, httpd, httpProtocol, httpPort, Apache)
6. vulExists(H1, ’CVE-2006-3747’, httpd, remoteExploit, privEscalation)
7. remote exploit of a server program
8. execCode(H1, Apache)
9. hacl(H1, H3, dbProtocol, dbPort)
10. multi-hop access
11. netAccess(H1, H3, dbProtocol, dbPort)
12. networkService(H3, mysqld, dbProtocol, dbPort, mysql)
13. vulExists(H3, ’CVE-2009-2446’, mysqld, remoteExploit, privEscalation)
14. remote exploit of a server program
15. execCode(H3, Apache)
16. vulExists(H3, ’CVE-2004-0495’, linux-kernel, localExploit, privEscalation)

17. local exploit of OS kernel
18. execCode(H3, root)
19. hacl(H0, H2, sshProtocol, sshPort)
20. direct network access
21. netAccess(H0, H2, sshProtocol, sshPort)
22. networkService(H2, sshd, sshProtocol, sshPort, SSH)
23. vulExists(H2, 'CVE-2002-0640', sshd, remoteExploit, privEscalation)
24. remote exploit of a server program
25. execCode(H2, SSH)
26. hacl(H1, H2, httpProtocol, httpPort)
27. multi-hop access
28. netAccess(H2, H1, httpProtocol, httpPort)
29. remote exploit of a server program

3 Prístup logického programovania

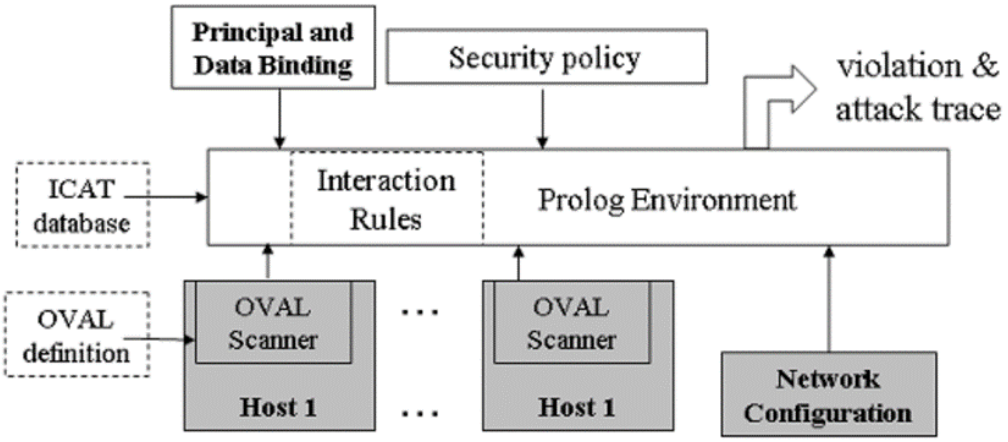
3.1 MulVAL

Viachostiteľská, viacstupňová analýza zraniteľnosti (Multi host, multistage vulnerability analysis - MulVAL) je prístup k analýze bezpečnosti siete založený na logickom programovaní. Používa znázornenie grafu útoku známeho ako graf logického útoku, ktorý zobrazuje logické závislosti medzi cieľmi útoku a informáciami o konfigurácii siete. MulVAL používa Datalog ako svoj modelovací jazyk. Informácie v databáze zraniteľností, ktoré poskytuje komunita hlásiaca chyby, informácie o konfigurácii každého počítača a siete a ďalšie relevantné informácie sú všetky zakódované ako Datalog tvrdenia (fakty). Uvažovací mechanizmus (Reasoning engine) MulVALu pozostáva zo súboru pravidiel Datalog, ktoré zachytávajú správanie operačného systému a interakciu rôznych komponentov v sieti. Integrácia informácií od komunity nahlasujúcej chyby a bežných nástrojov na skenovanie do mechanizmu uvažovania je teda jednoduchá. Uvažovací mechanizmus v MulVAL dobre škáluje s veľkosťou siete. Po zhromaždení potrebných údajov možno analýzu vykonať v priebehu niekoľkých sekúnd pre siete s tisíckami počítačov. Vstupy do MulVAL analýzy sú:

- Odporúčania (Advisories),
- Konfigurácia hostiteľa (Host configuration),
- Konfigurácia siete (Network configuration),
- Principáli (Principals),
- Interakcia (Interaction) a
- Politika (Policy).

Keďže Datalog je podmnožinou Prologu, zakódované informácie možno priamo načítať do prostredia Prologu a spustiť. MulVAL používa prostredie XSB, pretože umožňuje tabuľkové spúšťanie Prolog programov. Tabuľkové spúšťanie je forma dynamického programovania, ktorá zabraňuje prepočítavaniu už predtým vypočítaných faktov. Taktiež poskytuje logické programovanie v deklaratívnom štýle, čo znamená, že poradie pravidiel neovplyvňuje výsledok vykonania. MulVAL framework je znázornený na Obr. 11. Môžeme vidieť, že skener OVAL beží na každom stroji a vydáva správu o zraniteľnosti a príslušné konfiguračné parametre. Do prostredia XSB sa načítajú n-tice zo

skenerov, konfigurácia siete (reprezentovaná ako HACL), pravidlá a bezpečnostná politika definovaná administrátorom.



Obr. 11 MulVAL framework [4]

3.2 Podobné práce

Na určenie bezpečnostného dopadu softvérových zraniteľností na konkrétnu sieť je potrebné zvážiť interakcie medzi viacerými sieťovými prvkami. Práca [4] je pre nás „pilotnou“ prácou, pretože detailne rozoberá, ako funguje MulVAL prostredie. Hovorí o tom, že aby bol nástroj na analýzu zraniteľnosti užitočný v praxi, sú kľúčové dve funkcie. Po prvé, model použitý v analýze musí byť schopný automaticky integrovať formálne špecifikácie zraniteľnosti od komunity nahlasujúcej chyby. Po druhé, analýza musí byť schopná škálovať na siete s tisíckami strojov. Je tu opísané, čo vstupuje do MulVAL analýzy a ako prebiehajú jednotlivé štádiá generovania grafu. Toto všetko nám pomôže pri vytváraní nového nástroja.

V práci [3] autori rozoberajú rôzne typy grafov kybernetických útokov a ako prebieha ich generovanie. Konkrétne hovoria o grafe sčítania stavov, grafe exploit dependency, grafe útoku s viacerými predpokladmi a aj o logickom grafe, kde spomínajú MulVAL framework. Ďalej rozoberajú tzv. minimum cost network hardening, to znamená, že keď existuje v grafe útoku nejaká cesta zraniteľností, ktorú by vedel útočník zneužiť, ako tieto zraniteľnosti „zaplátať“ čo najefektívnejšie.

Pochopiť a predstaviť si kybernetické útoky vie byť náročná úloha. V práci [1] sa autori zaoberajú technikami modelovania útokov (AMT). Z nich sa zameriavajú hlavne

na stromy útoku a grafy útoku a ich vizuálnu syntax. Analyzovali vyše 180 grafov a stromov z hľadiska toho, ako znázorňujú útok. Dospeli k tomu, že dnes neexistuje nejaká štandardná metóda ako reprezentovať stromy a grafy útoku, čo sa týka vizualizácie a že v tejto oblasti je potrebný ešte ďalší výskum, aby sa dospelo k štandardizácií vizualizácie.

Od roku 2005 je možné nájsť viacero prác, ktoré rozšírili nástroj MulVAL. Autori v [5] doplnili do MulVAL prostredia vylepšenú reprezentáciu konštrukcií sieťových ciest a priradenie hodnoty dátovým aktívam v modeli. Článok [6] rozširuje rámec MulVAL tak, aby zahŕňal komplexnejšie bezpečnostné politiky existujúce v pokročilých operačných systémoch. Autori v článku [7] rozšírili MulVAL prostredie o niekoľko metód. V prvej metóde použili Common Vulnerability Scoring System (CVSS) na výpočet pravdepodobnosti premenných zraniteľnosti a Common Configuration Scoring System (CCSS) na výpočet pravdepodobnosti zraniteľnosti konfigurácie zabezpečenia systému. V druhej metóde uvádzajú vzájomnú závislosť premenných zraniteľnosti. Nakoniec v tretej metóde analyzujú vplyv zmeny konfigurácie zabezpečenia systému na pravdepodobnosť zraniteľnosti v kontexte Bayesovskej pravdepodobnosti.

Iným príkladom je článok [8]. Autori navrhli a implementovali v rámci prostredia MulVAL znalostnú bázu (známu aj ako „pravidlá interakcie“) na praktické generovanie grafov útokov. Štruktúrovaný návrhový postup je potrebný na vytvorenie znalostnej základne, ktorá umožňuje komplexnú analýzu, ktorá je veľmi dôležitá pre skutočné hodnotenie rizík. V rámci článku [9] bol navrhovaný nový rámec MulVAL, ktorý implementuje nový kanál. Ten zahŕňa špecializovaný lingvistický model kybernetickej bezpečnosti naučený pomocou NVD repozitátora, modelu rekurentnej neurónovej siete používaný na extrakciu útočných entít, model logistickej regresie používaný na doplnenie chýbajúcich informácií a nový model založený na strojovom učení. V článku [10] bol predstavujeme rozšírený model zabezpečenia siete pre MulVAL, ktorý zohľadňuje topológiu fyzickej siete, podporuje komunikačné protokoly krátkeho dosahu, modeluje zraniteľnosti pri navrhovaní sieťových protokolov a modeluje špecifické priemyselné komunikačné architektúry. Jedným z novších prístupov k doplneniu MulVAL rámca s cieľom začleniť kybernetické útoky na produkčné systémy je článok [11]. Autori v rámci tohto článku vyvinuli rozšírenie v podobe generovania a analýzy grafov útokov. Pomocou rozšírenia môžu odborníci v oblasti bezpečnosti aplikovať metódy analýzy grafov útokov v prostrediach, ktoré obsahujú komponenty strojového učenia.

V článku [13] použili Common Vulnerabilities and Exposures (CVE) databázu, z ktorej stiahli zraniteľnosti a extrahovali z nich relevantné informácie, ktoré následne upravili do tvaru, s ktorým dokáže MulVAL pracovať na vstupe (generovali z nich pravidlá interakcie). Dosiahli úspešnosť parsovania dát 88,15%. Použitím tohto prístupu dokážu byť grafy generované MulVALom stále aktuálne, keďže sa pravidelne sťahujú z databázy novo-objavené zraniteľnosti. V závere článku ukázali aj porovnanie grafu vytvoreného pôvodným MulVALom s novým grafom.

Grafy útokov sú dôležitými nástrojmi na analýzu bezpečnostných zraniteľností v podnikových sieťach. Počiatočné nástroje na generovanie grafov mali veľmi zlú škálovateľnosť vo veľkých sieťach, keďže mali exponenciálnu výpočtovú zložitosť. Logické grafy útokov sú veľmi dobre čitateľné pre človeka a majú polynomiálnu výpočtovú zložitosť. Toto konštatujú autori článku [14]. Vo svojom výskume okrem toho vylepšili nástroj MulVAL. Do jeho engine-u pridali tzv. derivačnú stopu, vďaka ktorej boli schopní vylepšiť výpočtovú zložitosť generovania grafu len na kvadratickú.

3.3 Odporúčania

Sú písané v jazyku OVAL, ktorý formalizuje, ako rozpoznať prítomnosť zraniteľností v počítačových systémoch. Skener OVAL berie na vstupe formalizované definície zraniteľnosti a testuje počítač na zraniteľný softvér. Výsledok testu sa konvertuje na klauzuly Datalogu, ako je nasledujúca:

```
vulExists(webServer, 'CAN-2002-0392', httpd).
```

Konkrétne skener identifikoval zraniteľnosť CAN-2002-0392 na webovom serveri počítača. Zraniteľnosť sa týkala serverového programu httpd. Účinok zraniteľnosti – ako ju možno zneužiť a aký je jej dôsledok, avšak nie je v OVAL formalizovaný. ICAT, databáza zraniteľností vyvinutá Národným inštitútom pre štandardy a technológie, poskytuje informácie o vplyve zraniteľnosti. Relevantné informácie z ICAT databázy sa prevádzajú do Datalog klauzúl ako napr.:

```
vulProperty('CAN-2002-0392', remoteExploit,
privilegeEscalation).
```

Táto zraniteľnosť umožňuje vzdialenému útočníkovi spustiť ľubovoľný kód so všetkými oprávneniami.

3.4 Konfigurácia hostiteľa

Skener OVAL vie extrahovať konfiguračné parametre na hostiteľovi. Napríklad môže dať na výstup informácie o servisnom programe (číslo portu, oprávnenia, atď.). Výstup sa konvertuje na klauzuly Datalogu ako:

```
networkService(webServer, httpd, TCP, 80, apache).
```

To znamená, že program httpd beží na stroji webServer ako user apache a počúva na porte 80 pomocou protokolu TCP.

3.5 Konfigurácia siete

MulVAL modeluje konfigurácie siete (smerovač a firewally) ako abstraktné zoznamy riadenia prístupu hostiteľa (HACL - host access-control list). Tieto informácie môže poskytnúť nástroj na správu brány firewall, ako je napríklad Smart Firewall. Tu je príklad položky HACL, ktorý umožňuje TCP spojenia z internetu na port 80 na webovom serveri:

```
hacl(internet, webServer, TCP, 80).
```

3.6 Principáli

Principála si môžeme predstaviť ako objekt, ktorý sa vie autentifikovať. V tomto kroku MulVAL mapuje principálov na ich používateľské účty na sieťových hostiteľoch. Mapovania by mali byť definované nasledovne:

```
hasAccount(user, projectPC, userAccount).
hasAccount(sysAdmin, webServer, root).
```

3.7 Interakcia

Pri viacstupňovom útoku sémantika zraniteľnosti a operačný systém určujú možnosti útočníka v každej fáze. Kódujú sa ako Hornove klauzuly (t.j. Prolog), kde prvý riadok predstavuje záver a zvyšné riadky predstavujú podmienky umožňujúce dospieť k tomuto záveru. Napríklad Pravidlo 1: Vzdialené zneužitie zraniteľnosti eskalácie privilégií v službe:

```
execCode(Host, User) :-
     networkService(Host, Program, Protocol, Port, User),
     vulExists(Host, VulID, Program, remoteExploit,
privEscalation),
     netAccess(Attacker, Host, Protocol, Port).
```

Toto je všeobecné pravidlo, ktoré špecifikuje predbežné a následné podmienky pre tento útok:

```
if
     (Program beží s oprávneniami používateľa na hostiteľovi
ako služba, ktorá používa protokol Protocol
a počúva na porte Port) AND
     (obsahuje vzdialene zneužiteľnú zraniteľnosť, ktorej
dopad je eskalácia privilégií)) AND
     (útočník má prístup k službe cez sieť)
then
     (útočník môže na stroji spustiť ľubovoľný kód ako
používateľ User)
```

3.8 Politika

V MulVALe politika popisuje, ktorý principál môže mať aký prístup k údajom. Všetko, čo nie je vyslovene povolené, je zakázané.

```
allow(Everyone, read, webPages).
```

```
allow(systemAdmin, write, webPages).
```

Keďže Everyone je písané veľkým písmenom, je to premenná Prologu, čo znamená, že sa môže zhodovať s ľubovoľným používateľom.

Záver

V tomto článku sme sa venovali teoretickej stránke generovania grafov kybernetických útokov. Uviedli sme techniky modelovania útokov (AMT) a zamerali sme sa teda na konkrétnu podmnožinu AMT, a to grafy útokov. Ukázali sme si, aké typy grafov útokov existujú a ako sa generujú. Ďalej sme sa viac zamerali na grafy logického útoku, konkrétne prostredie MulVAL.

Od posledného článku sme detailne rozobrali MulVAL prostredie, aby sme na základe získaných vedomostí navrhli riešenie na prerobenie prostredia. Od pôvodného plánu prerobiť úplne celé prostredie sme odstúpili, nakoľko by to bolo časovo veľmi náročné na implementáciu pre jednotlivca, čo nám potvrdili aj vedomosti nadobudnuté z prečítania viacerých vedeckých článkov, kde tiež konštatovali, že jadro MulVALu – logické programovanie, je to, čo robí z MulVALu silný nástroj na generovanie grafov kybernetických útokov. Aj preto sme sa rozhodli jadro ponechať a zamerať sa na prerobenie vstupu a výstupu nástroja.

Práca v tejto oblasti je momentálne vo fáze implementovania navrhnutého riešenia, ktoré bude tvoriť praktickú časť práce. Navrhnuté riešenie možno rozdeliť na dve časti: vstup a výstup. Na vstupe bude cieľom upraviť vstupné dáta (dáta budú z externej databázy) tak, aby s nimi prostredie MulVALu dokázalo pracovať. V rámci vstupu je dôležité zapracovať MITRE útočný rámec a zohľadniť informácie, ktoré nám poskytujú databázy zraniteľností, ktoré pracujú s CVSS skóre verzie 3. Dôležitou zmenou je možnosť ďalej pracovať s výstupom, ktorý nám nástroj MulVAL umožňuje. Na výstupe je cieľom vylepšiť čitateľnosť vygenerovaného grafu (prípadne umožniť používateľovi ho detailnejšie prezerať napr. po kliknutí na jednotlivé hrany a vrcholy grafu) a ukladať grafy spôsobom, aby bolo možné ďalej s výstupom pracovať ako s grafom. Po teoretickej stránke je práca takmer hotová.

Zoznam použitej literatúry

1. Lallie, H. S., Debattista, K., & Bal, J. (2020). A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*, *35*, 100219.
2. Kaynar, K. (2016). A taxonomy for attack graph generation and usage in network security. *Journal of Information Security and Applications*, *29*, 27-56.
3. Barik, M. S., Sengupta, A., & Mazumdar, C. (2016). Attack graph generation and analysis techniques. *Defence science journal*, *66*(6), 559.
4. Ou, X., Govindavajhala, S., & Appel, A. W. (2005, August). MulVAL: A Logic-based Network Security Analyzer. In USENIX security symposium (Vol. 8, pp. 113-128).
5. Bacic, E., Froh, M., & Henderson, G. (2006). Mulval extensions for dynamic asset protection. CINNABAR NETWORKS INC OTTAWA (ONTARIO).
6. Saha, D. (2008, October). Extending logical attack graphs for efficient vulnerability analysis. In Proceedings of the 15th ACM conference on Computer and communications security (pp. 63-74).
7. Sembiring, J., Ramadhan, M., Gondokaryono, Y. S., & Arman, A. A. (2015). Network security risk analysis using improved MulVAL Bayesian attack graphs. International Journal on Electrical Engineering and Informatics, 7(4), 735.
8. Inokuchi, M., Ohta, Y., Kinoshita, S., Yagyu, T., Stan, O., Bitton, R., ... & Shabtai, A. (2019, July). Design procedure of knowledge base for practical attack graph generation. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (pp. 594-601).
9. Binyamini, H., Bitton, R., Inokuchi, M., Yagyu, T., Elovici, Y., & Shabtai, A. (2020). An automated, end-to-end framework for modeling attacks from vulnerability descriptions. arXiv preprint arXiv:2008.04377.
10. Stan, O., Bitton, R., Ezrets, M., Dadon, M., Inokuchi, M., Yoshinobu, O., ... & Shabtai, A. (2020). Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks. IEEE Transactions on Dependable and Secure Computing.
11. Bitton, R., Maman, N., Singh, I., Momiyama, S., Elovici, Y., & Shabtai, A. (2021). Evaluating the Cybersecurity Risk of Real World, Machine Learning Production Systems. arXiv preprint arXiv:2107.01806.

12. Sadlek, L., Čeleda, P., & Tovarňák, D. (2022, April). Identification of Attack Paths Using Kill Chain and Attack Graphs. In NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium (pp. 1-6). IEEE.
13. JING, James Tan Wee, et al. Augmenting mulval with automated extraction of vulnerabilities descriptions. In: *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 2017. p. 476-481.
14. OU, Xinming; BOYER, Wayne F.; MCQUEEN, Miles A. A scalable approach to attack graph generation. In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006. p. 336-345.
15. AL-MOHANNADI, Hamad, et al. Cyber-attack modeling analysis techniques: An overview. In: *2016 IEEE 4th international conference on future internet of things and cloud workshops (FiCloudW)*. IEEE, 2016. p. 69-76.
16. SHANDILYA, Vivek; SIMMONS, Chris B.; SHIVA, Sajjan. Use of attack graphs in security systems. *Journal of Computer Networks and Communications*, 2014, 2014.
17. TAYOURI, David, et al. A Survey of MulVAL Extensions and Their Attack Scenarios Coverage. *arXiv preprint arXiv:2208.05750*, 2022.