Grafy útokov v kybernetickej bezpečnosti

Analýza a návrh riešenia

Bc. Vladimír Homola

1Im, 2021 – 2022

Abstrakt. V práci sa venujeme spracovaniu bezpečnostných údajov, analýze kybernetických útokov, technikám modelovania útokov, konkrétne generovaniu ich grafov. Zaujíma nás už existujúci nástroj pre generovanie grafov útokov – MulVAL. Tento nástroj je už pre dnešné potreby zastaralý a navyše nie je jednoduché ho na dnešných zariadeniach spustiť. Chceme teda na základe už existujúceho nástroja MulVAL vytvoriť nový, moderný nástroj na generovanie grafov útokov.

Kľúčové slová: kybernetická bezpečnosť, graf útoku, MulVAL

1 Úvod

Kybernetická bezpečnosť zasahuje do mnohých aspektov spoločenského, politického a obchodného života. Má 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.

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.

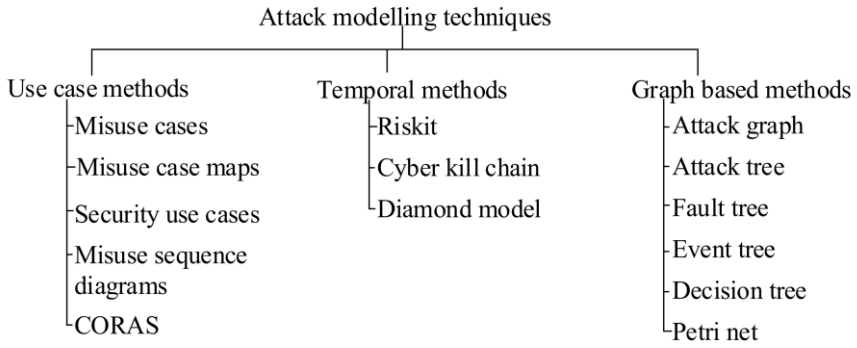
3 Techniky modelovania útokov (AMT)

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.

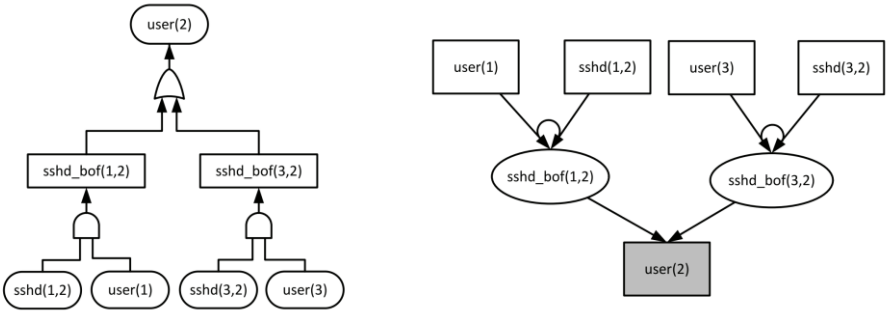
Tieto metódy sú zvýraznené na Obr. 1. Z metód načrtnutých na Obr. 1 sú grafy útokov a stromy útokov najobľúbenejšou metódou znázornenia kybernetických útokov.

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 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]

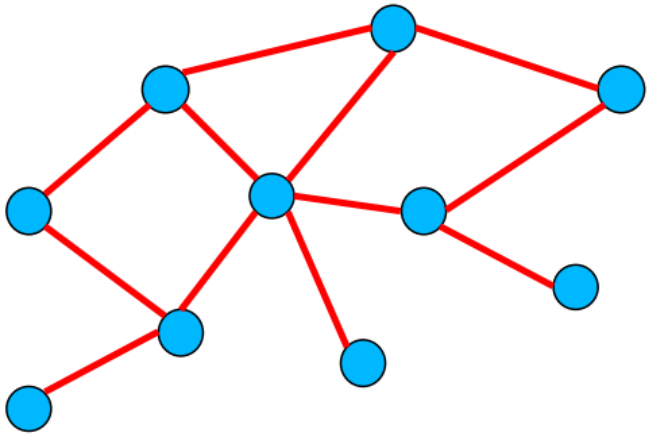
Príklad dvoch AMT – strom chýb a graf útoku je uvedený na Obr. 1. Príklad na Obr. 1 ukazuje, ako je útočník schopný vykonať sériu exploitov (sshd_bof) na sekvencii hostiteľských výpočtových zariadení (označených v zátvorkách), a tým získať užívateľské privilégiá (user) na každom z nich. Príklad tiež 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 2 Modely útoku (naľavo strom porúch a napravo graf útoku) [1]

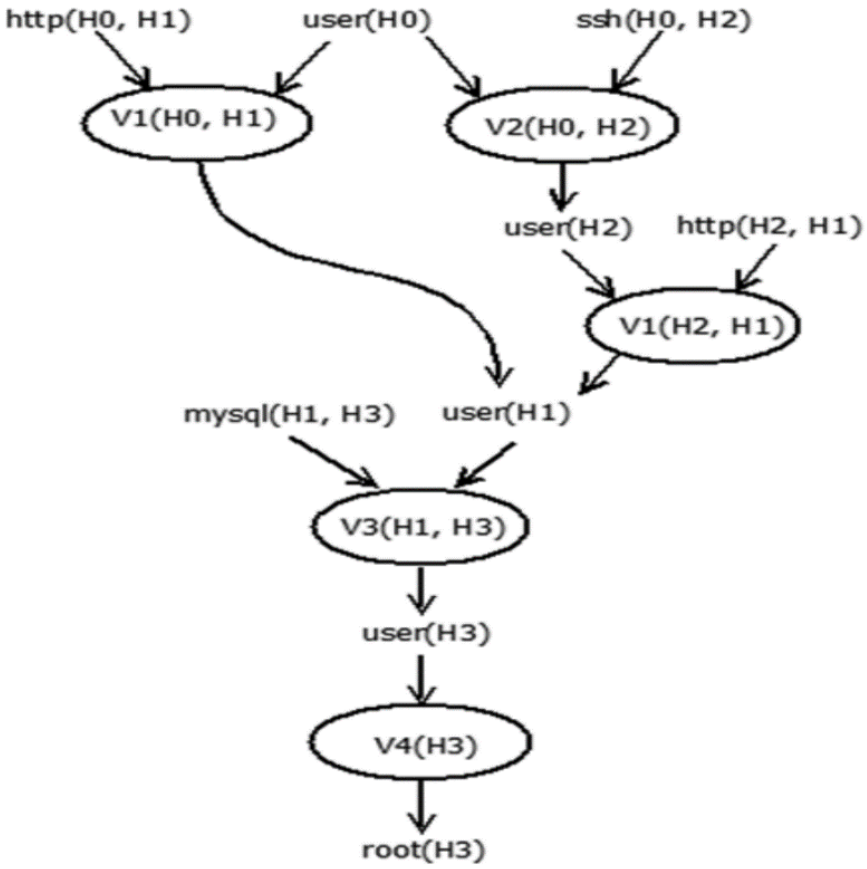
3.1 Grafy kybernetických útokov

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. 3.



Obr. 3 Znázornenie grafu

Graf útoku potom definujeme ako znázornenie všetkých ciest cez systém (reprezentovaný grafom), ktoré končia v stave, kde útočník úspešne dosiahol svoj cieľ. Graf útoku môžeme vidieť na Obr. 4. 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. 3. Existuje totiž viac typov grafov útoku a prístupov na ich generovanie. V ďalšej časti práce si ich viac rozoberieme a naučíme sa ich čítať.

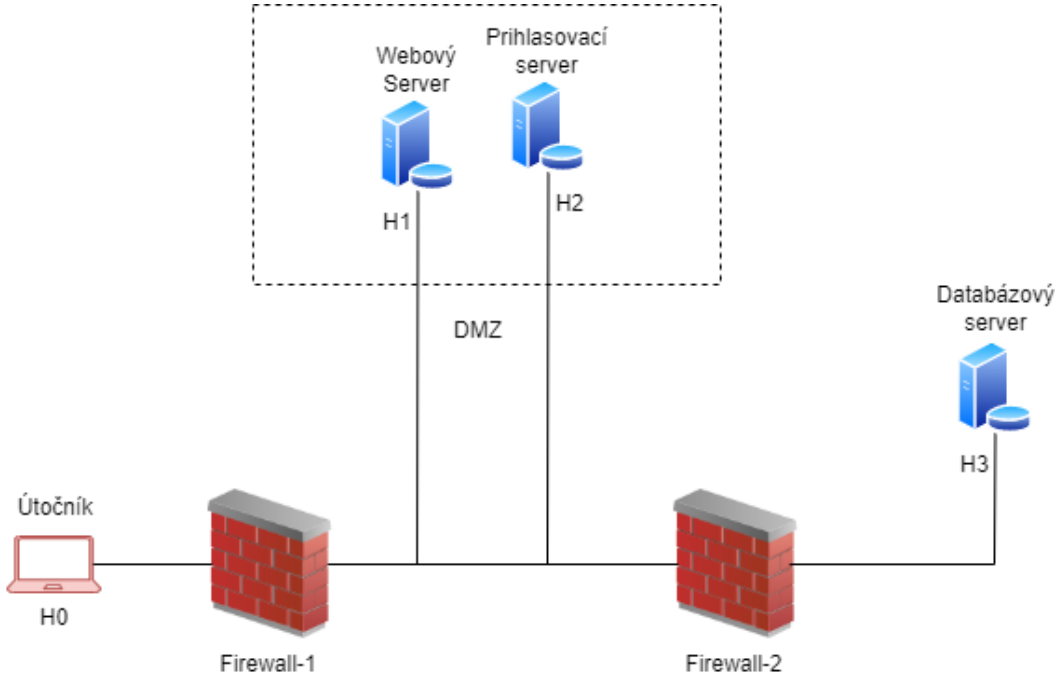


Obr. 4 Graf kybernetického útoku [3]

3.2 Typy grafov kybernetických útokov a prístupy ich generovania

V tejto podkapitole 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. 5. 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. 5 Jednoduchá sieť

3.2.1 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.

3.2.2 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.

3.2.3 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.

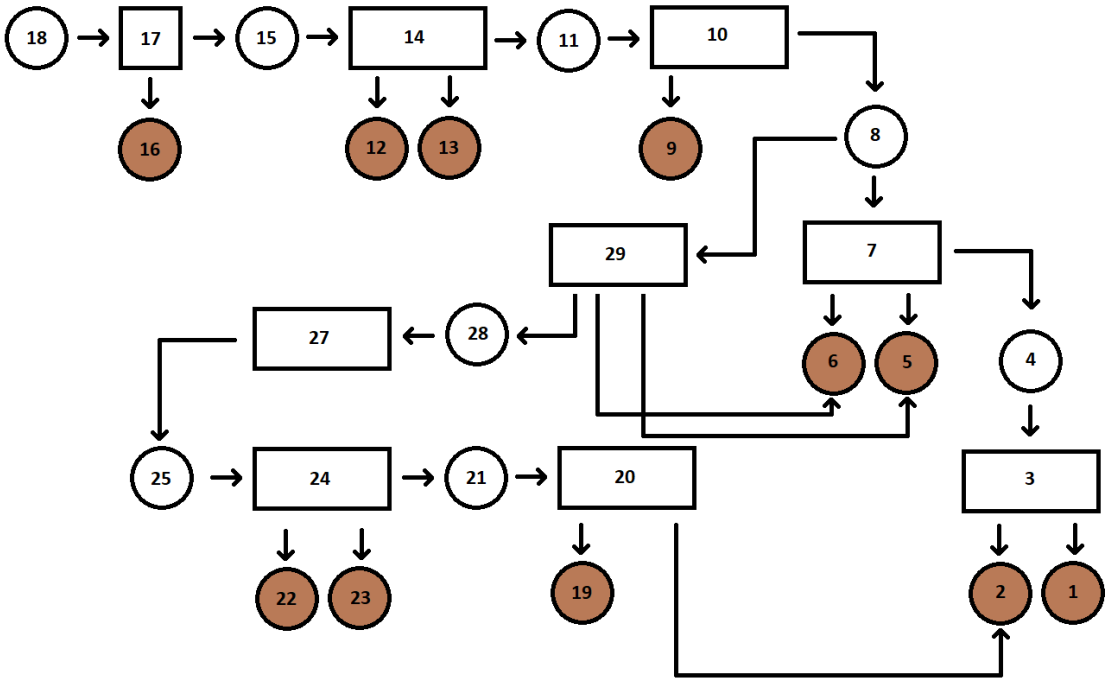
3.2.4 Prístup topologickej analýzy zraniteľností

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).

3.2.5 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. 6 sa nachádza logický graf útoku zodpovedajúci konfigurácii siete uvedenej na Obr. 5. 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. 6 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

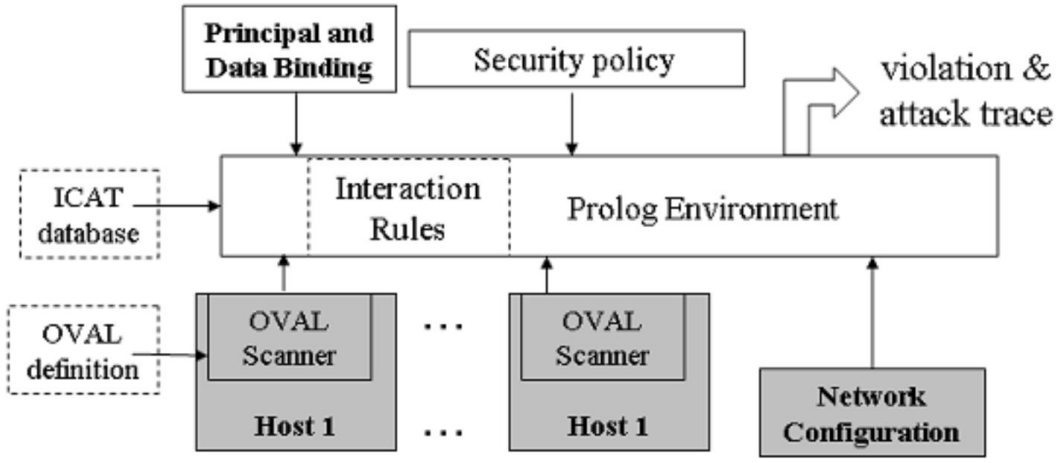
3.2.6 Prístup logického programovania (MulVAL)

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. 7. 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. 6 MulVAL framework [4]

3.2.6.1 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.

3.2.6.2 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.

3.2.6.3 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).
```

3.2.6.4 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).
```

3.2.6.5 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)
```

3.2.6.6 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.

4 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.

Práca v tejto oblasti je momentálne ešte len na začiatku. V najbližšom čase máme v pláne detailne rozobrať MulVAL prostredie, aby sme potom na základe získaných vedomostí vedeli implementovať nový nástroj na generovanie grafov kybernetických útokov. Keď už budeme v stave, že náš nástroj bude približne kopírovať prácu MulVALu, potom prídu na rad otázky o implementovaní ďalšej funkcionality do tohto nástroja. Predbežne sa črtá cesta generovania tzv. kill-chain grafov útoku [12].

Literatúra

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). 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). 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.