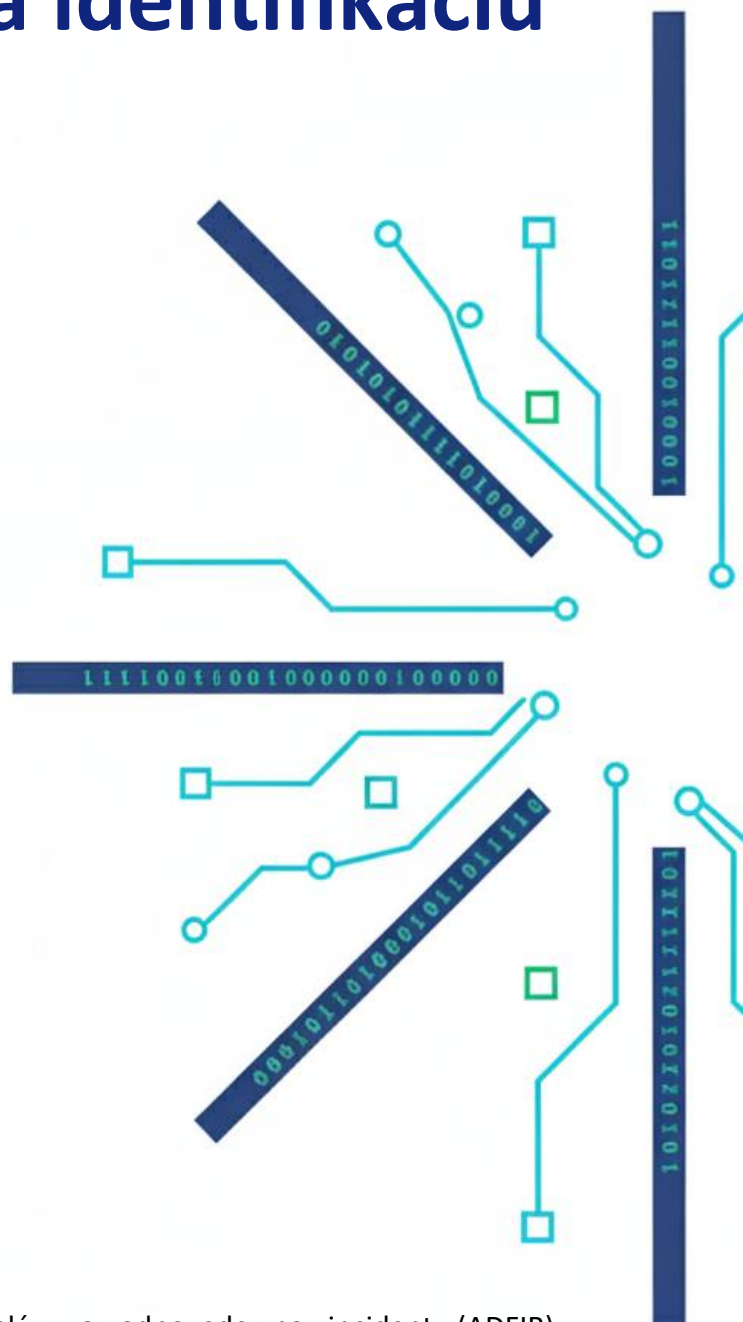


D20 – Metóda na identifikáciu vzťahov (I)



Projekt Automatizácia digitálnej forenznej analýzy a odpovede na incident (ADFIR) financovaný Európskou úniou – Next GenerationEU prostredníctvom Plánu obnovy a odolnosti Slovenskej republiky pod číslom projektu č. 09I05-03-V02-00079.

Obsah

Popis projektu	2
Úvod.....	3
1 Formálna konceptová analýza	4
2 Dekompozícia formálneho kontextu	6
2.1 <i>Faktorizácia binárnych matíc</i>	6
2.2 <i>Formálne koncepty ako faktory</i>	7
2.3 <i>Problém pokrytia množiny</i>	8
3 Algoritmus na hľadanie faktorov	10
4 Aplikácia algoritmu na hľadanie faktorov	12
4.1 <i>Všeobecné a špecifické faktory</i>	12
4.2 <i>Faktory súvisiace s útokom</i>	15
Záver	17
Bibliografia	18
Prílohy	19

Popis projektu

Projekt **Automatizácia digitálnej forenznej analýzy a odpovede na incident** (ďalej len „ADFIR“) je financovaný Európskou úniou – Next GenerationEU prostredníctvom Plánu obnovy a odolnosti Slovenskej republiky pod číslom projektu č. 09I05-03-V02-00079. Tento projekt sa zaoberá jednou z kľúčových výziev v oblasti kybernetickej bezpečnosti a informačnej bezpečnosti – ako spracovať obrovské množstvo digitálnych dôkazov, ktoré vznikajú počas incidentov kybernetickej bezpečnosti alebo forenznych vyšetrovaní. V súčasnosti je tento proces veľmi náročný z hľadiska ľudských zdrojov a času. Automatizácia pomocou metód strojového učenia môže preto výrazne **zlepšiť kvalitu digitálnej forenznej analýzy** a skrátiť čas potrebný na jej vykonanie. Celkovo to umožňuje bezpečnostným tímom efektívnejšie reagovať na kybernetické hrozby. Hlavné prínosy tohto projektu sú:

- **Rýchlejšie riešenie incidentov v oblasti kybernetickej bezpečnosti.** Projekt ADFIR zavádza automatizované prístupy k zberu, spracovaniu a analýze digitálnych stôp. Vďaka tomu môžu bezpečnostné tímy rýchlejšie identifikovať príčiny incidentov a prijať účinné opatrenia na ich riešenie.
- **Zníženie pracovnej záťaže forenznych analytikov.** Rutinné a časovo náročné úlohy spojené so spracovaním digitálnych stôp budú nahradené automatizovanými metódami. To umožní analytikom sústrediť sa na zložitejšie prípady a strategické rozhodovanie.
- **Vyššia kvalita a konzistentnosť výstupov.** Použitie jednotných metodík a nástrojov zaručuje, že spracované digitálne stopy budú presnejšie, konzistentnejšie a ľahšie overiteľné. To výrazne znižuje riziko chýb spôsobených ľudskými faktormi.
- **Možné využitie v trestnom konaní.** Výstupy projektu budú vyvinuté v súlade s právnymi požiadavkami a normami, čo umožní, aby digitálne stopy boli akceptované ako relevantné dôkazy pre vyšetrovanie a súdne konania.

Úvod

Digitálna forenzná analýza pracuje s veľkým množstvom dát pochádzajúcich z rôznych zdrojov, ako sú súborové systémy, registre operačného systému či webové artefakty. Jednotlivé udalosti a ich vlastnosti pritom často nadobúdajú význam až vo vzájomných súvislostiach.

Cieľom je navrhnúť a aplikovať metódu na identifikáciu vzťahov v dátach z digitálnej foreznej analýzy. Jedným z prístupov vhodných na identifikáciu vzťahov je formálna konceptová analýza. Umožňuje v dátach hľadať formálne koncepty tvorené množinou objektov a množinou atribútov, ktoré majú tieto objekty spoločné.

Počet formálnych konceptov však môže vzhľadom na štruktúru formálneho kontextu rásť exponenciálne. Pri rozsiahlych datasetoch je generovanie konceptového zväzu časovo náročné a následná manuálna analýza a interpretácia veľkého množstva konceptov prakticky nemožná. Navrhnutá metóda preto nevyžaduje generovanie úplného konceptového zväzu, ale vyberá takú podmnožinu formálnych konceptov, ktorá pokrýva všetky incidencie vstupného datasetu. Metóda využíva maticovú faktorizáciu založenú na formálnych konceptoch. Na výber faktorov používa algoritmus Rice-Siff Factorization, ktorý kombinuje greedy prístup k problému pokrytia množiny s postupným vytváraním kandidátnych formálnych konceptov.

1 Formálna konceptová analýza

Formálna konceptová analýza (FCA) [1] je matematický rámec založený na teórii zväzov, ktorý slúži na vyhľadávanie konceptov v binárnych dátach a ich hierarchické usporiadanie. Vstupné dáta možno reprezentovať ako binárnu tabuľku, v ktorej riadky zodpovedajú objektom, stĺpce atribútom a hodnota 1 vyjadruje, že daný objekt má príslušný atribút.

Formálne, nech B a A sú neprázdne množiny objektov a atribútov, a nech $R \subseteq B \times A$ je binárna relácia. Potom sa trojica (B, A, R) nazýva **formálny kontext** – príklad uvedený na **Obrázok 1**.

	i	ii	iii
a	×		
b	×	×	
c		×	

Obrázok 1 - Formálny kontext

Na hľadanie spoločných atribútov podmnožiny objektov a spoločných objektov podmnožiny atribútov sa používajú operátory $\uparrow: \mathcal{P}(B) \rightarrow \mathcal{P}(A)$ a $\downarrow: \mathcal{P}(A) \rightarrow \mathcal{P}(B)$, kde $\mathcal{P}(B)$ a $\mathcal{P}(A)$ označujú potenčné množiny množín B a A .

Pre podmnožinu objektov $X \subseteq B$ definujeme

$$X^\uparrow = \{a \in A \mid \forall b \in X: (b, a) \in R\}.$$

Analogicky, pre podmnožinu atribútov $Y \subseteq A$ definujeme

$$Y^\downarrow = \{b \in B \mid \forall a \in Y: (b, a) \in R\}.$$

Operátory $\uparrow$ a $\downarrow$ tvoria **Galoisovu konexiu**. Ich zložením $X^{\uparrow\downarrow}$ a $Y^{\downarrow\uparrow}$ získavame uzáverové operátory.

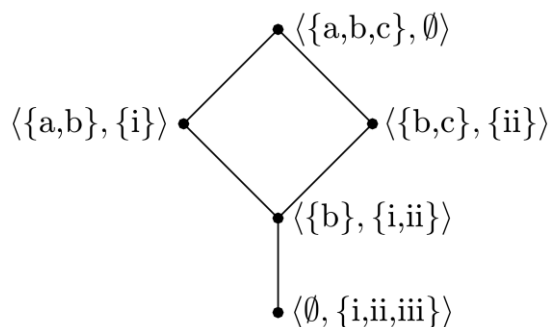
Dvojica (X, Y) , pre ktorú platí $X^\uparrow = Y$ a zároveň $Y^\downarrow = X$, sa nazýva **formálny koncept**. Objektová časť konceptu X sa nazýva extent a atribútová časť konceptu Y sa nazýva intent.

Formálny koncept teda vyjadruje úplnú dvojicu – uzavretú podmnožinu objektov spolu so všetkými atribútmi, ktoré má táto podmnožina spoločné. V tabuľkovej reprezentácii môžeme formálny koncept chápať ako maximálny obdĺžnik vyplnený jednotkami. To znamená, že ho nemôžeme rozšíriť o ďalší riadok alebo stĺpec bez toho, aby obsahoval nulu.

Množinu všetkých formálnych konceptov formálneho kontextu (B, A, R) označujeme $\mathcal{B}(B, A, R)$.

Formálne koncepty možno usporiadať pomocou vzťahu $(X_1, Y_1) \leq (X_2, Y_2)$ práve vtedy, keď $X_1 \subseteq X_2$, resp. $Y_2 \subseteq Y_1$.

Čiastočne usporiadanú množinu formálnych konceptov $(\mathcal{B}(B, A, R), \leq)$ nazývame **konceptový zváz**. Konceptový zváz môžeme vizualizovať pomocou čiarového diagramu (**Obrázok 2**).



Obrázok 2 - Konceptový zváz

Podľa základnej vety o konceptových zvázoch [1] je tento zváz úplný, teda pre každú podmnožinu formálnych konceptov v ňom existuje infimum aj supremum.

2 Dekompozícia formálneho kontextu

Reprezentujme formálny kontext (B, A, R) binárnou incidenčnou maticou. Dekompozíciu formálneho kontextu potom môžeme chápať ako dekompozíciu incidenčnej matice.

Cieľom dekompozície je identifikovať menšiu množinu latentných štruktúr, označovaných ako **faktory**, ktoré vystihujú vzťahy medzi podmnožinami objektov a podmnožinami atribútov. Hľadajú sa dve binárne matice

$$U \in \{0,1\}^{|B| \times k}$$

a

$$V \in \{0,1\}^{k \times |A|},$$

kde k predstavuje počet faktorov. Matica U opisuje vzťah medzi objektmi a faktormi, zatiaľ čo matica V opisuje vzťah medzi faktormi a atribútmi. Pôvodná incidenčná matica sa následne rekonštruje pomocou Boolean súčinu týchto matíc.

2.1 Faktorizácia binárnych matíc

Nech $R \in \{0,1\}^{|B| \times |A|}$ je incidenčná matica formálneho kontextu. **Faktorizácia binárnej matice** je rozklad

$$R = U \circ V,$$

kde

$$U \in \{0,1\}^{|B| \times k}, V \in \{0,1\}^{k \times |A|}$$

a symbol $\circ$ označuje Boolean súčin matíc. Pre každý prvok výslednej matice platí

$$(U \circ V)_{ij} = \bigvee_{\ell=1}^k (U_{i\ell} \wedge V_{\ell j}),$$

kde $\wedge$ označuje logickú konjunkciu a $\vee$ logickú disjunkciu.

Príklad rozkladu matice uvádzame na **Obrázok 3**.

$$\begin{matrix} & R & & U & & V \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} & = & \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} & \circ & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Obrázok 3 - Rozklad matice

Hodnota $(U \circ V)_{ij} = 1$ teda nastane práve vtedy, keď existuje aspoň jeden faktor ℓ , pre ktorý súčasne platí

$$U_{i\ell} = 1$$

a

$$V_{\ell j} = 1.$$

V závislosti od požadovanej kvality rekonštrukcie rozlišujeme presnú a aproximačnú faktorizáciu. Pri **presnej faktorizácii** platí

$$R = U \circ V,$$

teda rekonštruovaná matica je totožná s pôvodnou incidenčnou maticou. Pri **aproximačnej faktorizácii** sa pripúšťa

$$R \approx U \circ V,$$

pričom môžu vzniknúť dva typy chýb:

- **undercover** – nastáva, keď niektorý jednotkový prvok z pôvodnej matice nie je pri faktorizácii zrekonštruovaný,
- **overcover** – nastáva, keď zrekonštruovaná matica obsahuje jednotku na pozícii, na ktorej mala pôvodná matica hodnotu nula.

2.2 Formálne koncepty ako faktory

Pri rozklade binárnej matice môžeme jednotlivé faktory reprezentovať formálnymi konceptmi [2]. Nech

$$\mathcal{F} = \{\langle X_1, Y_1 \rangle, \dots, \langle X_k, Y_k \rangle\} \subseteq \mathfrak{B}(B, A, R)$$

je vybraná množina formálnych konceptov. Táto množina následne indukuje maticu objektov a faktorov $U_{\mathcal{F}}$ a maticu faktorov a atribútov $V_{\mathcal{F}}$.

Základnou vlastnosťou formálnych konceptov ako faktorov je ich **univerzálnosť**. Pre každú binárnu maticu R existuje množina formálnych konceptov $\mathcal{F} \subseteq \mathfrak{B}(B, A, R)$, pre ktorú platí

$$R = U_{\mathcal{F}} \circ V_{\mathcal{F}}.$$

Formálne koncepty teda vždy postačujú na presnú rekonštrukciu pôvodnej incidenčnej matice.

Formálne koncepty sú zároveň **optimálnymi faktormi**. Ak má matica R ľubovoľný rozklad $R = U \circ V$ s k faktormi, potom existuje množina formálnych konceptov $\mathcal{F}$ taká, že

$$|\mathcal{F}| \leq k$$

a

$$R = U_{\mathcal{F}} \circ V_{\mathcal{F}}.$$

Rozklad s minimálnym možným počtom faktorov teda možno vždy nájsť medzi formálnymi konceptmi. Množina formálnych konceptov preto predstavuje postačujúci a neredundantný priestor kandidátnych faktorov na hľadanie optimálneho rozkladu.

Optimalitu formálnych konceptov vieme zdôvodniť aj geometricky. Každý obdĺžnik indukovaný ľubovoľným faktorom je obsiahnutý v niektorom maximálnom obdĺžniku zodpovedajúcemu formálnemu konceptu. Nahradením pôvodných faktorov formálnymi konceptmi sa zachová presná rekonštrukcia matice a počet faktorov sa nezvýši.

Významnou vlastnosťou formálnych konceptov ako faktorov je tiež ich **priama interpretovateľnosť**. Vo faktorovej analýze sa faktory zvyčajne interpretujú ako latentné dimenzie, ktoré sa prejavujú prostredníctvom pôvodných pozorovaných atribútov. Pri použití formálnych konceptov je význam faktora explicitne určený jeho extantom a intentom. Extent X_ℓ identifikuje množinu objektov, na ktoré sa faktor vzťahuje, zatiaľ čo intent Y_ℓ určuje množinu atribútov, prostredníctvom ktorých sa faktor prejavuje.

V digitálnej forenznej analýze môže jeden faktor reprezentovať opakujúci sa vzor udalostí, napríklad kombináciu súborových artefaktov, registrov operačného systému a sieťovej komunikácie, ktorá sa vyskytuje vo viacerých objektoch. Faktory tak umožňujú nahradiť veľké množstvo jednotlivých incidencií menším počtom interpretovateľných štruktúr. Bezpečnostný analytik môže následne skúmať tieto faktory namiesto analýzy každého objektu samostatne, čo uľahčuje identifikáciu opakujúcich sa vzorov správania a potenciálnych bezpečnostných incidentov.

2.3 Problém pokrytia množiny

Hľadanie rozkladu matice s minimálnym počtom faktorov úzko súvisí s problémom pokrytia množiny (Set Cover Problem).

Keďže každý faktor je reprezentovaný formálnym konceptom (X, Y) , ktorý pokrýva obdĺžnik incidencií $X \times Y \subseteq R$, cieľom je nájsť takú čo najmenšiu množinu formálnych konceptov $F \subseteq \mathcal{B}(B, A, R)$, aby ich zodpovedajúce obdĺžniky pokryli všetky jednotkové prvky incidenčnej matice, teda

$$R = \bigcup_{(X,Y) \in F} (X \times Y).$$

Optimalizačný problém pokrytia množiny je NP-ťažký. Z tohto dôvodu sa v praxi používajú aproximačné a heuristické algoritmy. Greedy prístup v každom kroku vyberie formálny koncept, ktorý pokrýva najväčší počet doposiaľ nepokrytých incidencií, a tento postup opakuje, až kým nie je pokrytá celá matica.

Výpočtovú zložitosť navyše zvyšuje skutočnosť, že počet formálnych konceptov môže byť vo všeobecnosti exponenciálny. Pre formálny kontext s n objektmi a m atribútmi môže počet formálnych konceptov dosiahnuť až $2^{\min(n,m)}$. Pri veľkých kontextoch je náročné vygenerovať

celý konceptový zväz a následne prehľadať všetky kandidátne faktory. Preto sa používajú algoritmy, ktoré namiesto prehľadávania všetkých formálnych konceptov postupne vytvárajú a vyhodnocujú iba vhodných kandidátov.

3 Algoritmus na hľadanie faktorov

Algoritmus **Rice-Siff Factorization (RSF)** [3] je navrhnutý na presnú faktorizáciu incidenčnej matice pomocou formálnych konceptov. Vychádza z greedy prístupu k problému pokrytia množiny, ale namiesto prehľadávania celého konceptového zväzu vytvára množinu kandidátnych faktorov postupným zlučovaním formálnych konceptov.

Počas behu si algoritmus udržiava množinu $R_{\text{un}} \subseteq R$, ktorá obsahuje incidencie zatiaľ nepokryté žiadnym vybraným faktorom. Na začiatku platí $R_{\text{un}} = R$. Po výbere každého faktora sa z množiny R_{un} odstránia incidencie, ktoré daný faktor pokrýva. Proces pokračuje, až kým sa množina R_{un} nevyprázdni ($R_{\text{un}} = \emptyset$).

V každej iterácii hlavného cyklu sa vytvoria dve množiny konceptov, $\mathcal{C}_1$ a $\mathcal{C}_2$. Obe množiny sa inicializujú množinou objektových konceptov

$$\mathcal{C}_1 = \mathcal{C}_2 = \{(\{b\}^{\downarrow}, \{b\}^{\uparrow}) \mid b \in B\}.$$

Množina $\mathcal{C}_1$ predstavuje aktuálnu pracovnú množinu konceptov, ktoré sa postupne zlučujú. Množina $\mathcal{C}_2$ uchováva všetkých kandidátov vytvorených počas aglomerácie, teda pôvodné objektové koncepty aj ich následné suprema.

Na výber dvojice konceptov na zlúčenie sa používa stavovo závislá semimetrika ρ_{un} . Pre formálne koncepty (X_1, Y_1) a (X_2, Y_2) je definovaná vzťahom

$$\rho_{\text{un}}((X_1, Y_1), (X_2, Y_2)) = 1 - \frac{|(X_1 \times Y_1) \cap (X_2 \times Y_2) \cap R_{\text{un}}|}{|(X_1 \times Y_1) \cup (X_2 \times Y_2)|}.$$

Jej hodnota závisí od aktuálnej množiny nepokrytých incidencií. Dvojice konceptov s menšou hodnotou ρ_{un} majú väčšie spoločné pokrytie v reziduálnej relácii, a preto predstavujú vhodnejších kandidátov na zlúčenie.

Na výber faktora sa používa reziduálna váha pokrytia

$$w(X, Y) = |(X \times Y) \cap R_{\text{un}}|.$$

Hodnota $w(X, Y)$ udáva počet zatiaľ nepokrytých incidencií, ktoré dokáže pokryť koncept (X, Y) . Po ukončení aglomeračnej fázy sa z množiny $\mathcal{C}_2$ vyberie koncept s maximálnou reziduálnou váhou.

Zlučovanie konceptov prebieha vo vnútornom cykle, v ktorom sa najskôr zostaví množina vhodných dvojíc

$$\mathcal{P} = \{(O_i, O_j) \in \mathcal{C}_1 \times \mathcal{C}_1 \mid Y_i \cap Y_j \neq \emptyset\}.$$

Podmienka na neprázdny prienik intentov je dôležitá. Ak totiž $Y_i \cap Y_j = \emptyset$, potom supremum týchto konceptov má prázdny intent a nepokrýva žiadnu incidenciu. Takéto zlúčenie nemá význam pre riešenie problému pokrytia.

Z množiny $\mathcal{P}$ sa vyberie dvojica s najmenšou hodnotou ρ_{un} . Tieto dva koncepty sa z pracovnej množiny $\mathcal{C}_1$ odstránia a vytvorí sa ich supremum

$$O_{new} = O_i \vee O_j.$$

Pre $O_i = (X_i, Y_i)$, $O_j = (X_j, Y_j)$ je definované vzťahom

$$O_i \vee O_j = ((X_i \cup X_j)^{\uparrow\downarrow}, Y_i \cap Y_j).$$

Nový koncept sa vloží do oboch množín $\mathcal{C}_1$ a $\mathcal{C}_2$. Aglomerácia pokračuje, kým pracovná množina neobsahuje najviac jeden koncept alebo kým neexistuje dvojica s neprázdny prienikom intentov.

Po skončení vnútorného aglomeračného cyklu sa z množiny $\mathcal{C}_2$ vyberie koncept (X, Y) s maximálnou váhou $w(X, Y)$. Vybraný koncept sa pridá do množiny faktorov F a ním pokryté incidencie sa odstránia.

Potom sa množiny kandidátov opätovne inicializujú objektovými konceptmi a celý postup sa vykoná vzhľadom na aktualizovanú reziduálnu reláciu.

Výstupom algoritmu je množina faktorov F . Pseudokód algoritmu uvádzame na **Obrázok 4**.

Algorithm 1: Rice-Siff Factorization (RSF)	
Input:	A formal context (B, A, R) with $R \subseteq B \times A$
Output:	Factor set F
1	$F \leftarrow \emptyset$
2	$R_{un} \leftarrow R$
3	while $R_{un} \neq \emptyset$ do
4	$C_1 \leftarrow \{(\{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow}) \mid x \in B\}$
5	$C_2 \leftarrow C_1$
6	while $ C_1 > 1$ do
7	$P \leftarrow \{(O_i, O_j) \in C_1 \times C_1 \mid Y_i \cap Y_j \neq \emptyset\}$
8	if $P = \emptyset$ then break
9	Find $(O_i, O_j) \in P$ minimizing $\rho_{un}(O_i, O_j)$
10	Remove O_i, O_j from C_1
11	$O_{new} \leftarrow O_i \vee O_j$
12	Insert O_{new} into C_1 and C_2
13	Select $(X, Y) \in C_2$ maximizing $w(X, Y)$
14	$F \leftarrow F \cup \{(X, Y)\}$
15	$R_{un} \leftarrow R_{un} \setminus (X \times Y)$
16	return F

Obrázok 4 - Pseudokód algoritmu RSF [3]

Jedna iterácia hlavného cyklu má časovú zložitosť $O(|B|^3 |A|)$. Keďže v každej iterácii pokryje aspoň jednu dovtedy nepokrytú incidenciu, tento cyklus sa vykoná najviac $|R|$ -krát.

4 Aplikácia algoritmu na hľadanie faktorov

Na aplikáciu algoritmu RSF sme použili dataset z dátovej sady „NIST Data Leakage Case“, ktorý bol agregovaný s použitím časového okna 1 800 sekúnd. Podrobné predspracovanie datasetu je popísané vo výstupe KPB2-02. Jednotlivé riadky datasetu reprezentujú časové zhľuky udalostí vytvorené v rámci tohto intervalu.

Väčšina atribútov už má binárny charakter, pričom hodnota 1 označuje výskyt príslušnej vlastnosti v danom časovom zhľuku a hodnota 0 jej neprítomnosť. Pred aplikáciou algoritmu bolo odstránených 5 nebinárnych atribútov, ktoré obsahovali časové, identifikačné alebo pomocné informácie. Výsledkom predspracovania je binárna matica $R \in \{0,1\}^{276 \times 218}$, následne použitá ako vstup algoritmu RSF.

Výstupom algoritmu je **86 faktorov**. Každý faktor zodpovedá jednému formálnemu konceptu a reprezentuje určitú kombináciu atribútov spoločnú pre skupinu objektov. Celý výstup uvádzame v prílohe NIST_factors_1800s.xlsx, kde sa nachádzajú v riadkoch jednotlivé faktory, atribúty charakterizujúce daný faktor, počet atribútov a počet objektov vo faktore.

Jednotlivé faktory sa líšia počtom atribútov v intente aj počtom objektov v extente. V získanom rozklade obsahujú faktory od 1 do 107 atribútov a od 1 do 220 objektov.

4.1 Všeobecné a špecifické faktory

Faktory s malým počtom atribútov často zachytávajú všeobecnejšie vzory, ktoré sa vyskytujú pri väčšom počte objektov. Naopak, faktory s väčším počtom atribútov spravidla reprezentujú špecifickejšie kombinácie vlastností a obsahujú menší počet objektov. Tento vzťah vyplýva z vlastností formálnych konceptov – čím viac atribútov obsahuje intent, tým menej objektov má všetky vybrané atribúty. Napríklad faktor 83 obsahuje 1 atribút a 220 objektov, zatiaľ čo faktor 41 obsahuje 107 atribútov a iba 1 objekt. Počet atribútov ani počet objektov však sami o sebe ešte neurčujú význam faktora. Pri interpretácii je potrebné zohľadniť najmä význam atribútov a súvis faktora s analyzovaným bezpečnostným scenárom.

Za najvšeobecnejšie považujeme faktory s najväčším počtom objektov v extente. Takéto faktory zachytávajú atribúty alebo kombinácie atribútov, ktoré sa vyskytujú vo veľkej časti analyzovaných časových zhľukov. Reprezentujú často sa opakujúce vzory, a preto pravdepodobne nepoukazujú na neštandardné alebo podozrivé správanie. Poskytujú však základný obraz o dominantných atribútoch v analyzovanom datasete. V **Tabuľka 1** uvádzame 3 najvšeobecnejšie faktory.

Č. faktora	Počet atribútov	Atribúty	Počet objektov
83	1	mask_MSIE_WebCache_container_record	220
2	2	is_url_visit, mask_MSIE_WebCache_container_record	213
5	1	mask_MSIE_Cache_File_URL_record	103

Tabuľka 1 - Najvšeobecnejšie faktory analyzovaného datasetu

Za špecifické považujeme faktory s malým počtom objektov v extente. Takéto faktory zachytávajú kombinácie atribútov, ktoré sa v datasete vyskytujú iba zriedkavo. Ich význam z pohľadu digitálnej forenznej analýzy závisí od konkrétnych atribútov v intente faktora, napríklad od prítomnosti atribútov súvisiacich so spúšťaním skriptov, zmenami registra, prístupom k citlivým súborom alebo inými podozrivými aktivitami. V získanom rozklade obsahuje 15 faktorov iba 1 objekt. V **Tabuľka 2** uvádzame 3 z nich.

Č. faktora	Počet atribútov	Atribúty	Počet objektov
41	107	access_count_gt_1, contains_cab, contains_inf, contains_update_task, contains_vbs, ContainsCmdCommand, ContainsScriptExtension, event_registry, has_exe_target, has_macb_accessed, has_macb_modified, has_modified_flag, has_modified_time, has_sha256, has_transition_type, HasAandBNoMC, is_accessed, is_admin_activity, is_advertising_cookie, is_analytics_cookie, is_apis_google, is_application_experience, is_bing_domain, is_error_event, is_executable, is_failed_login, is_file, is_file_entry, is_filename_attr, is_fileshare_access, is_from_driverstore, is_from_ie, is_from_pinned_location, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_history_container, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_link_file, is_login_event, is_modified_after_access, is_network_related, is_ntfs, is_policy_change, is_privilege_change, is_repeated_record, is_root_entry, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_suspicious_task_name, is_system_path, is_terminal_services, is_third_party, is_tracking_cookie, is_unallocated, is_unpinned, is_unusual_location, is_url_visit, is_user_profile_shortcut, is_xml_file, is_youtube_domain, IsCSS, IsDynamicLibrary, IsInAppData, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_anomaly, macb_is_only_access_birth, MissingMInMACB, ModifiesStartType, task_starts_at_logon, visit_to_search_engine, visited_microsoft, visited_msn, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_AppCompatCache_Registry_Key, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_Cookies, mask_Chrome_History, mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_MSIE_WebCache_containers_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_System, mask_Setup_API_Log, mask_Task_Cache_Registry_Key	1
34	90	access_count_gt_1, contains_inf, contains_path_recent, ContainsPowershellCommand, event_registry, has_hostname_reference, has_macb_accessed, has_macb_modified, has_modified_flag, has_modified_time, has_sha256, has_transition_type, is_accessed, is_admin_activity, is_analytics_cookie, is_apis_google, is_application_experience, is_error_event, is_executable,	1

		is_failed_login, is_file, is_file_entry, is_filename_attr, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_link_file, is_login_event, is_network_share, is_ntfs, is_policy_change, is_privilege_change, is_root_entry, is_runonce_key, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_system_location, is_system_path, is_terminal_services, is_tracking_cookie, is_unallocated, is_unpinned, is_unusual_location, is_url_visit, is_user_account_event, is_user_profile_shortcut, is_xml_file, is_youtube_domain, IsCSS, IsInAppData, IsSuspiciousPath, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_is_only_access_birth, MissingMInMACB, ModifiesImagePath, ModifiesStartType, opened_control_panel, unknown_shell_item_type, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_Cookies, mask_Chrome_History, mask_MRUList_Registry_Key, mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_System, mask_User_Account_Information_Registry_Key	
43	84	access_count_gt_1, accessed_local_file, contains_cab, contains_inf, has_exe_target, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, is_accessed, is_application_experience, is_content_container, is_executable, is_file, is_file_entry, is_filename_attr, is_fileshare_access, is_frequent_execution, is_from_system32, is_from_users_dir, is_high_access_count, is_history_container, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_link_file, is_login_event, is_modified_after_access, is_multiple_volumes, is_network_path, is_ntfs, is_obfuscated_path, is_privilege_change, is_rare_app, is_repeated_record, is_root_entry, is_run_often, is_service_event, is_specific_container, is_standard_info, is_suspicious_extension, is_system_hive, is_system_location, is_system_path, is_system_process, is_temp_launch, is_unallocated, is_url_visit, is_user_profile_shortcut, is_xml_file, IsDynamicLibrary, IsInAppData, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, macb_is_only_access_birth, MissingMInMACB, ModifiesImagePath, ModifiesStartType, opened_control_panel, opened_recycle_bin, unknown_shell_item_type, UsesQuotesInPath, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_Windows_Shortcut, mask_AppCompatCache_Registry_Key, mask_BagMRU_Registry_Key, mask_MRUListEx_Registry_Key, mask_MSIE_WebCache_container_record, mask_MSIE_WebCache_containers_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Item, mask_Registry_Key, mask_Run_Run_Once_Registry_Key,	1

		mask_Setup_API_Log, mask_User_Account_Information_Registry_Key, mask_WinPrefetch	
--	--	--	--

Tabuľka 2 - Faktory s malým počtom objektov v extente

4.2 Faktory súvisiace s útokom

Významný atribút *is_attack_related* sa nachádza v 4 získaných faktoroch.

Najvšeobecnejší z nich, faktor 24, obsahuje 2 atribúty a 21 objektov. Tento faktor neposkytuje podrobný opis útoku, ale zachytáva spoločnú vlastnosť väčšej skupiny označených objektov.

Extenty ďalších 3 faktorov obsahujú vždy 1 objekt, pričom ich intenty pozostávajú z 50 až 60 atribútov. Tieto intenty zahŕňajú kombinácie atribútov odvodených zo súborových a webových artefaktov, artefaktov registra Windows a systémových udalostí. Faktor 30 obsahuje najmä atribúty súvisiace s prístupom k citlivým súborom a sieťovým zdieľaním, zatiaľ čo faktor 32 zahŕňa atribúty spojené so spúšťaním príkazov a skriptov a so zmenami mechanizmov automatického spúšťania. Uvedené faktory tak zachytávajú konkrétne kombinácie vlastností prítomné v časových zhlukoch označených ako súvisiace s útokom. V **Tabuľka 3** uvádzame vyššie popísané faktory súvisiace s útokom.

Č. faktora	Počet atribútov	Atribúty	Počet objektov
24	2	mask_MSIE_WebCache_container_record, is_attack_related	21
27	50	access_count_gt_1, accessed_local_file, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, has_transition_type, is_accessed, is_apis_google, is_application_experience, is_file, is_file_entry, is_filename_attr, is_from_system32, is_from_users_dir, is_google_domain, is_gstatic_domain, is_https, is_image_file, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_login_event, is_modified_after_access, is_ntfs, is_policy_change, is_privilege_change, is_runonce_key, is_script_file, is_service_event, is_standard_info, is_suspicious_domain, is_system_hive, is_unallocated, is_url_visit, is_xml_file, IsCSS, IsSystemHive, IsUserHive, ModifiesStartType, mask_WinEVTX, mask_NTFS_file_stat, mask_File_stat, mask_Chrome_Cache, mask_Chrome_History, mask_MSIE_WebCache_container_record, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_Setup_API_Log, is_attack_related	1
30	60	access_count_gt_1, accessed_local_file, accessed_sensitive_file, contains_secret_path, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, has_transition_type, is_accessed, is_application_experience, is_file, is_file_entry, is_filename_attr, is_from_users_dir, is_google_domain, is_gstatic_domain, is_high_access_count, is_https, is_image_file, is_in_appdata, is_in_temp_or_spool, is_modified_after_access, is_network_share, is_ntfs, is_obfuscated_path, is_root_entry, is_run_often, is_script_file, is_sensitive_file, is_service_event, is_specific_container, is_standard_info, is_suspicious_domain, is_suspicious_keyword, is_system_location, is_unpinned,	1

		is_url_visit, is_xml_file, IsInAppData, IsSystemHive, IsUserHive, key_contains_run, key_new_registry_branch, opened_control_panel, unknown_shell_item_type, mask_WinEVTX, mask_File_entry_shell_item, mask_NTFS_file_stat, mask_File_stat, mask_BagMRU_Registry_Key, mask_Chrome_Cache, mask_Chrome_History, mask_MSIE_WebCache_container_record, mask_MSIE_Cache_File_URL_record, mask_OLECF_Dest_list_entry, mask_OLECF_Item, mask_Registry_Key, mask_WinPrefetch, is_attack_related	
32	51	access_count_gt_1, accessed_local_file, contains_cab, contains_cur, contains_inf, contains_ppd, contains_vbs, ContainsCmdCommand, ContainsPowershellCommand, has_macb_accessed, has_macb_modified, has_modified_flag, has_sha256, is_accessed, is_executable, is_file, is_file_entry, is_filename_attr, is_from_driverstore, is_from_system32, is_from_users_dir, is_in_appdata, is_in_system32, is_in_temp_or_spool, is_in_winsxs, is_modified_after_access, is_network_path, is_ntfs, is_runonce_key, is_script_file, is_standard_info, is_suspicious_extension, is_unallocated, is_url_visit, is_xml_file, IsDynamicLibrary, IsSystemHive, IsUserHive, key_autorun_related, key_contains_winlogon, key_new_registry_branch, key_suspicious_value, ModifiesImagePath, ModifiesStartType, mask_NTFS_file_stat, mask_File_stat, mask_MSIE_WebCache_container_record, mask_Registry_Key, mask_Run_Run_Once_Registry_Key, mask_System, is_attack_related	1

Tabuľka 3 - Faktory súvisiace s útokom

Záver

Metóda na identifikáciu vzťahov v dátach z digitálnej forenznej analýzy využíva maticovú faktorizáciu založenú na formálnych konceptoch a algoritmus RSF, ktorý kombinuje greedy výber s postupným vytváraním kandidátnych formálnych konceptov. Výsledkom je presný rozklad vstupnej incidenčnej matice, ktorý umožňuje identifikovať skupiny objektov charakterizované spoločnými atribútmi.

Metóda identifikuje vzťahy medzi časovými zhukmi udalostí a ich vlastnosťami, a zachytáva všeobecné aj špecifické kombinácie atribútov prítomné v analyzovaných dátach. Faktory s menším počtom atribútov spravidla opisujú všeobecnejšie vzory prítomné pri väčšom počte objektov. Naopak, faktory s väčším počtom atribútov zachytávajú špecifickejšie kombinácie vlastností a ich extenty sú zvyčajne menšie. Pri analýze získaných faktorov sa môžeme zamerať na vybraný významný atribút a sledovať, s akými ďalšími atribútmi sa v jednotlivých faktoroch vyskytuje. Dôležitým prínosom navrhutej metódy je aj interpretovateľnosť nájdených faktorov.

Navrhnutá metóda poskytuje systematický a interpretovateľný spôsob skúmania vzťahov medzi foreznými udalosťami a ich vlastnosťami.

Bibliografia

- [1] Ganter, B., Wille, R. (2024). *Formal Concept Analysis: Mathematical Foundations*. Springer.
<https://doi.org/10.1007/978-3-031-63422-2>
- [2] Belohlavek, R., Vychodil, V. (2010). *Discovery of optimal factors in binary data via a novel method of matrix decomposition*. Journal of Computer and System Sciences.
<https://doi.org/10.1016/j.jcss.2009.05.002>
- [3] Antoni, Ľ., Kotlárová, D., Krídlo, O., López-Rodríguez, D., Ojeda-Aciego, M. (2026). *Effective greedy Boolean matrix factorization via the Rice-Siff algorithm*. International Journal of Approximate Reasoning.
<https://doi.org/10.1016/j.ijar.2026.109747>

Prílohy

ADFIR-D20-KPB4-01-NIST_factors_1800s.xlsx