Systém na identifikáciu súborov

Meno: Matej Fedor
Školiteľ: RNDr. Jaroslav Janáček, PhD.

Motivácia

Existujúce riešenia

Intrusion Detection Systems:

- Open Source Tripwire
- AFICK
- SAMHAIN

Nástroje pre vykonávanie bezpečnostných auditov

- AIDE
- md5deep/hashdeep

Existujúce riešenia

NIST: National Institute of Standards and Technology

- National Software Reference Library (NSRL)
- Reference Data Set (RDS)
- metadáta o súboroch známych softvérových balíkov
- výrazný nárast v septembri 2018
- udržiavané na štvrťročnej báze

RDS

- > 56 000 000 unikátnych hashov
- > 10 GB metadát
- MD5/SHA1 hash-set
- konverzný hash-set SHA1 → SHA2

Problémy uvedených riešení

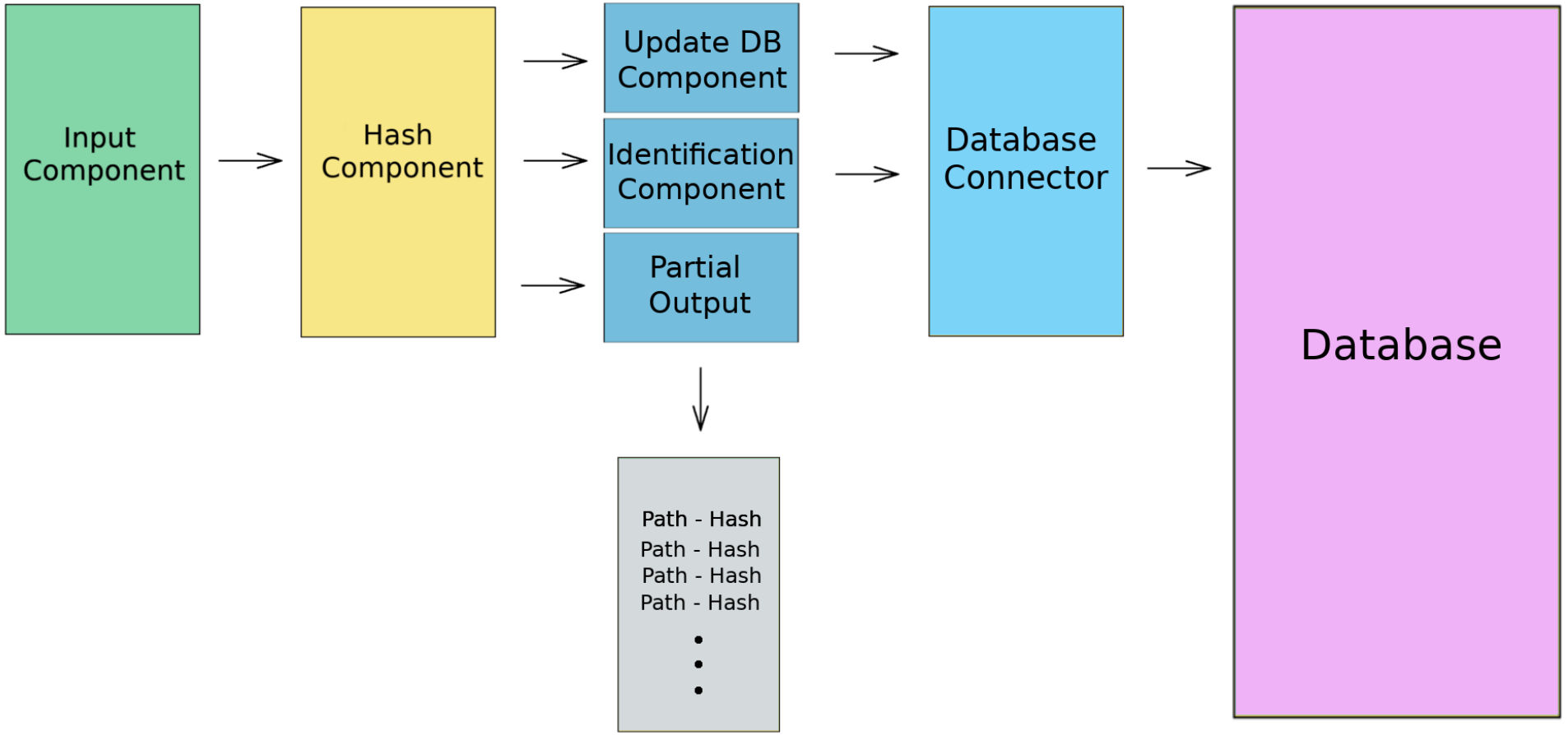
- ✗ závislosť na konkrétnej platforme(AIDE)
- ✗ bezpečnostné nedostatky (md5deep/hashdeep)
- ✗ horšia škálovateľnosť (md5deep/hashdeep)
- ✗ automatizovaná údržba referenčnej databázy
- ✗ nevhodný návrh pre použitie v danej oblasti

Riešenie

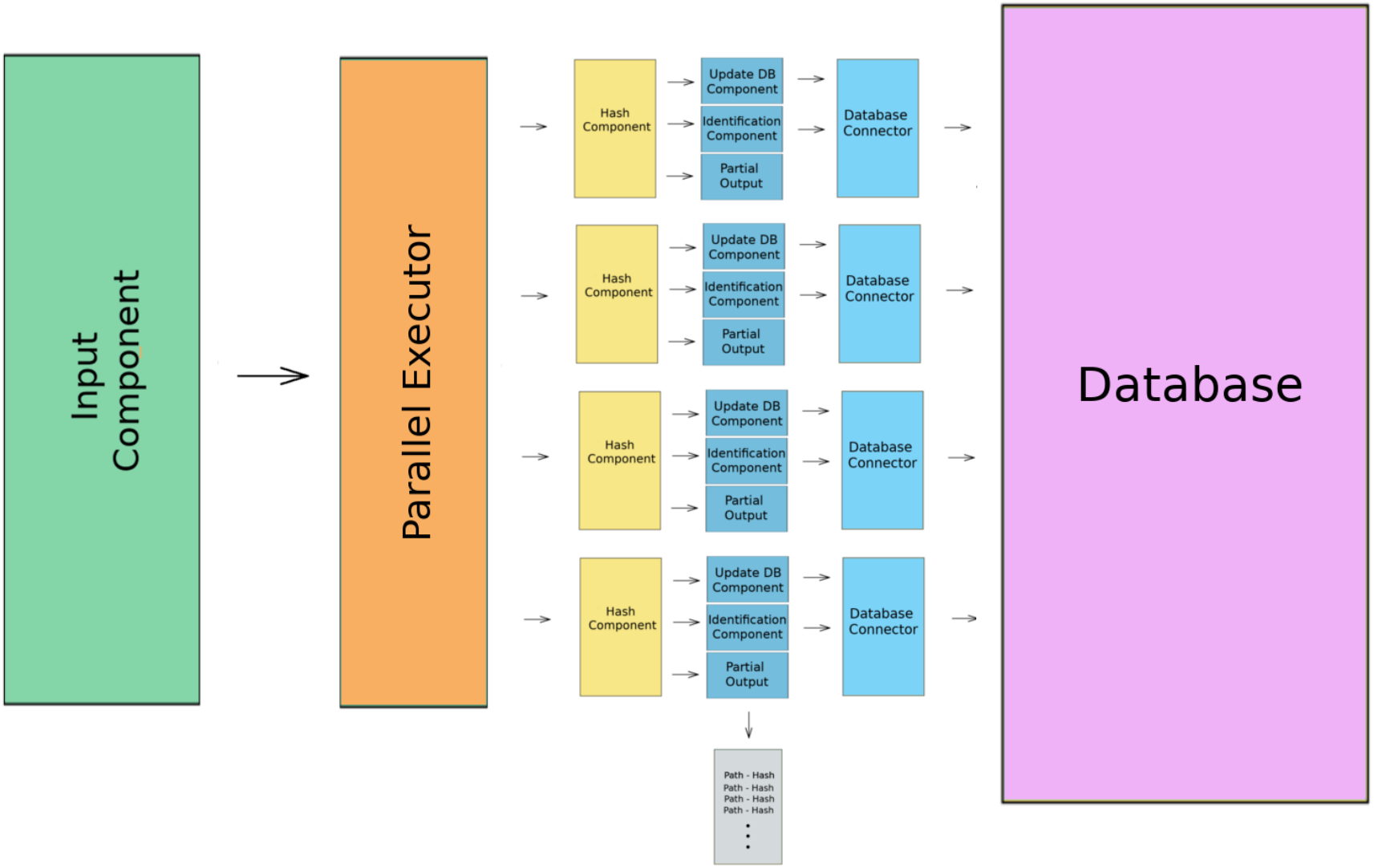
Softvérové dielo s dôrazom na:

- ✔ platformovú nezávislosť Windows/Linux
- ✔ jednoduchú distribuovateľnosť
- ✔ efektivitu a škálovateľnosť
- ✔ flexibilný prístup
- ✔ podpora pre budovanie a rozširovanie referenčnej databázy
- ✔ intuitívne spracovanie výsledkov

File-Identification-System



File-Identification-System



File-Identification-System

- 45 000 referenčných súborov
- 200 000 skúmaných súborov
- veľkosť ~kB - ~MB
- 56% redukcia potrebného času

Čas (s) (1 vlákno)	Čas (s) (4 vlákna)	Čas (s) (8 vlákien)	Čas (s) (12 vlákien)
404	199	193	173
398	201	198	179

Referenčná databáza

- dôraz na množinu zaznamenávaných atribútov
- minimalizácia redundancie
- optimalizácia vyhľadávania

Referenčná databáza

- 45 000 referenčných záznamov
- 4 500 overovaných odtlačkov
- 98% redukcia potrebného času

1 vlákno (s)	8 vlákien (s)	1 vlákno, index (s)	8 vlákien, index (s)
562	286	6	3
573	301	6	3

Budovanie databázy

- farma virtuálnych strojov s rôznymi operačnými systémami
- priebežná aktualizácia OS a zaznamenávanie stavu súborového systému do databázy

- pomocná aplikácia SysUpdate.exe

Spracovanie výsledkov

- webová aplikácia určená pre vizualizáciu výsledkov
- analýza súboru/zoznamu súborov

FILE IDENTIFICATION SYSTEM

Access your reference database comfortably. Examine a specific file or a list of files. Visualize and filter output information.
Find the nature of system files and identify interesting pieces for further analysis.

EXAMINE

... a specific file of your interest.

C:\File\Path\...\

SUBMIT

EVALUATE

... a list of FIS pre-hashed files.

SUBMIT

VISUALIZE

... a reference database information.

SUBMIT

EXAMINE

... a specific file of your interest.

C:\File\Path\...\

SUBMIT

EVALUATE

... a list of FIS pre-hashed files.

SUBMIT

VISUALIZE

... a reference database information.

SUBMIT

All files have been checked successfully. 2 valid files, 1 suspisions, 2 warnings, 0 errors and 0 unknown files were found.

EXPAND COLUMNS | SHOW SUSPICIOUS | SHOW WARNINGS | SHOW ERRORS | SHOW VALID | SHOW UNKNOWN | REMOVE FILTERS

File name	Original path	Original digest	File path	File digest	Created	Modifie
DebuggerProxy.dll	C:\Users\Matej\.vscode\ex...	09D80B9925BFEECFBD21...	C:\Users\Matej\.vscode\ex...	09D80B9925BFEECFBD21...	13.4.2020 2:50:12	13.4.202
DebuggerProxy.dll	C:\Users\Matej\.vscode\ex...	09D80B9925BFEECFBD21...	C:\Users\Matej\.vscode\ex...	09D80B9925BFEECFBD21...	10.5.2020 5:11:17	10.5.202
msvcp140.dll	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	13.4.2020 2:50:10	13.4.202
msvcp140.dll	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	10.5.2020 5:11:16	10.5.202
msvcp140.dll	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	C:\Users\Matej\.vscode\ex...	6E7896923BD527975C6B...	10.5.2020 5:11:16	10.5.202

Ďalší vývoj

- podpora pre existujúce referenčné databázy
- tvorba špecifických referenčných databáz
  - databáza zraniteľných verzií softvéru
  - databáza škodlivého obsahu

Ďakujem za pozornosť

“Súborový systém potenciálne napadnutého zariadenia analyzovaný nad distribúciou Linuxu sa však s veľkou pravdepodobnosťou bude prejavovať omnoho stabilnejšie, keďže škodlivý softvér na viacerých platformách často nebude fungovať správne.”

Pokiaľ sa bude súborový systém napadnutého zariadenia analyzovať „offline“, použitím čistého operačného systému, tak sa domnievam, že vôbec nezáleží na tom, či je to Windows alebo Linux a či je škodlivý softvér prenositeľný na viacero platforiem alebo nie.

Pri výbere programovacieho jazyka požadujete platformovú nezávislosť, kompiláciu do rýchleho natívneho kódu a iné. V diskusii spomínate programovacie jazyky C, Java a C++ (pre ktorý ste sa nakoniec rozhodli). Zvažovali ste aj iné? Napríklad jazyk Rust je niektorými považovaný za rýchlejší a bezpečnejší ako C++.

“Druhou požiadavkou je, aby bolo prakticky výpočtovo nemožné odvodiť z tohto digitálneho odtlačku akúkoľvek informáciu o pôvodnom digitálnom vstupe funkcie.”

Prečo požadujete túto vlastnosť?

*“Prvá z nich využíva referenčnú databázu pre identifikáciu, prípadne kontrolu integrity objektov súborového systému na základe ich **absolútnej cesty** …”*

Čo ak je súčasťou absolútnej cesty napríklad meno používateľa? Čo ak si používateľ zmení umiestnenie svojho domovského adresára? Nebolo by niekedy lepšie používať napríklad relatívne cesty alebo cesty obsahujúce symbolické názvy (ako napríklad %AppData%)?

“Významná bola pre nás taktiež podpora aplikácie pre čo najširšiu množinu databázových serverov a ich verzií.”

Reálne uvažujete iba MySQL (prípadne jeho fork MariaDB). Prečo neuvažujete aj iné databázy, napríklad PostgreSQL?

“Úplné statické linkovanie aplikácie so závislosťami sa nám však pre komplikácie a obmedzenosť času nepodarilo.”

Môžete objasniť o aké komplikácie išlo a či sa Vám ich už podarilo vyriešiť?

“Toto vylepšenie považujeme za vynikajúci výsledok, ktorý je brilantnou odpoveďou na obavy z praktickej použiteľnosti našej aplikácie pre veľké dáta.”

Dvojnásobné zrýchlenie programu je rozhodne zaujímavé, ale pre veľké dáta to môže byť stále málo. Superlatívy ako „brilantný“ by som nechal skôr pre rádové urýchlenie. Uvažovali ste napríklad nad použitím GPU pre výpočet odtlačku?