Tomasz Pawlicki
Arkadiusz Bolewski
Paweł Kubicz
Andrzej Romanowski

Oprogramowanie do analizy śledczej tworzone we współpracy z Katedrą Informatyki Stosowanej Politechniki Łódzkiej – – SQLite, Torrent Analyzer, Konwerter Dźwięku

Wprowadzenie i cel pracy

Niniejsza publikacja jest kolejnym artykułem poświęconym zagadnieniom szeroko pojętej informatyki śledczej. Informatyka śledcza (*Computer Forensics*) wkracza tam, gdzie kryminalistyka styka się z komputerami zarówno w sensie sprzętu, jak i oprogramowania. Od kilku lat obszary jej zastosowania, poza komputerami, obejmują także szereg innych urządzeń technologii służących do przesyłania, przetwarzania i przechowywania informacji. W poniższym artykule poruszone zostały trzy wybrane, praktyczne aspekty związane z analizą dowodową informacji w postaci elektronicznej. Pierwszy z nich to wykorzystanie w praktyce kryminalistycznej właściwości i obsługi systemu bazodanowego SQLite na przykładzie archiwum komunikatora internetowego Gadu-Gadu. Drugi prezentuje możliwości opracowanego oprogramowania do analizy plików konfiguracyjnych programów klienckich sieci do wymiany plików (*Peer-to-Peer* – P2P). Trzecim przedstawionym zagadnieniem jest projekt programu do konwersji plików dźwiękowych ułatwiający wykorzystanie i odtworzenie materiału dowodowego pochodzącego głównie z urządzeń przenośnych. Prezentowana problematyka jest przedmiotem badań dotyczących tworzenia nowoczesnych narzędzi informatycznych wspierających pracę ekspertów kryminalistyki. Nad badaniami wspólnie pracują Laboratorium Kryminalistyczne Komendy Wojewódzkiej Policji w Łodzi i Katedra Informatyki Stosowanej Politechniki Łódzkiej przy wsparciu Oddziału Łódzkiego Polskiego Towarzystwa Informatycznego. Część wyników zamieszczona w niniejszym artykule przedstawia elementy prac dyplomowych Arkadiusza Bolewskiego oraz Pawła Kubicza.

Odczyt danych z aplikacji opartych na silniku bazodanowym SQLite na przykładzie nowej wersji popularnego komunikatora internetowego Gadu-Gadu

Nowe Gadu-Gadu może być dodatkowym źródłem informacji w różnych sprawach, w których dochodzi do komunikacji między użytkownikami. Często rozmowy zawierają wiadomości na temat miejsc spotkań, transakcji. Historia Gadu-Gadu, w skrócie GG, rozpoczęła się w roku 2000, obecna wersja ma numer 10. Po roku 2008, wdrożeniu biblioteki Qt i zastosowaniu kodowania Unicode UTF-8 wydano wersję komunikatora pod nazwą Nowe Gadu-Gadu.

Niniejsza część artykułu przedstawia metody przeglądania i analizy danych SQLite na przykładzie właśnie Gadu-Gadu. Dokładniej, jest to analiza pliku *Archive.db* używanego przez komunikator. Pliki te są kontenerami dla danych bazy SQLite w wersji v3. Poniżej zostały zaprezentowane podstawowe informacje umożliwiające poznanie struktury danych zastosowanych w Nowym GG, a także metody pozwalające na tzw. ręczną analizę informacji w niej zawartych. W tym miejscu należy wspomnieć, że istnieje dostępne oprogramowanie, np. adbT, za pomocą którego można uzyskać dane pochodzące z archiwów komunikatora, jednak w trakcie przeprowadzonych przez ekspertów kryminalistyki Pracowni Badań Komputerowych LK KWP w Łodzi badań stwierdzono, iż generowane przez nie raporty zawierają błędy, co całkowicie dyskwalifikuje to oprogramowanie jako narzędzie do analiz typu *Computer Forensis*.

SQLite – informacje podstawowe

SQLite to zarazem silnik i system zarządzania relacyjną bazą danych obsługujący język zapytań SQL1[1]. Jedną z podstawowych zalet tego rozwiązania jest możliwość implementacji bazy danych jako pojedynczego pliku (do 2 TB). Dzięki takiej architekturze w praktycznych implementacjach nie jest konieczne łączenie się z serwerem bazodanowym, a tym samym uruchamianie odrębnego procesu RDBMS[2] tak, jak to ma miejsce przykładowo w rozwiązaniach klasycznych, np. MySQL. SQLite jest obecnie najpopularniejszym rozwiązaniem bazodanowym wybieranym dla systemów wbudowanych. Świadczy o tym choćby jego obecność w dwóch najważniejszych platformach Smartphone: iPhone OS oraz Android, co z kolei pozwala przewidywać dalszy jego rozwój i wzrost znaczenia. Początki projektu datuje się na rok 2000, w którym to D. Richard Hipp stworzył i udostępnił SQLite na zasadach licencji Public Domain[3].

Podstawowe cechy SQLite to: obsługa transakcji zgodnych ze standardem ACID[4], implementacja zgodna ze standardem SQL92, przechowywanie w pojedynczym pliku kompletnej bazy danych, możliwość migracji pomiędzy różnymi platformami Win32, Win64, WinCE, Linux, MacOSX. Ograniczeniem rozwiązania jest blokowanie całego pliku podczas dopisywania nowych danych oraz brak praw dostępu do bazy danych. Jednak w przypadku analizy plików będących materiałem dowodowym jest to znaczne ułatwienie.

SQLite – zastosowanie

Baza danych SQLite jest wbudowana między innymi w następujące aplikacje: Mozilla Firefox – przeglądarka internetowa na licencji *open source* (w bazach danych przechowywane są profile użytkowników), Android – środowisko programistyczne dla urządzeń przenośnych, zaprojektowane przez Google (do przechowywania danych użytkownika), Apple iPhone – multimedialne urządzenie przenośne, Nowe GG – komunikator internetowy (archiwum połączeń).

Sqlite3Explorer – menadżer bazodanowy

W celu analizy pliku można posłużyć się dostępnymi rozwiązaniami typu Menager SQLite. W omawianym przypadku użyto programu Sqlite3Explorer, dostępnego na stronach *http://www.singular.gr/sqlite/*. Poniżej przedstawiono jego opis, zważywszy jednak na nadrzędny cel artykułu ograniczono się jedynie do opisu funkcji umożliwiających uzyskanie danych z kontenera SQLite.

Okno programu (ryc. 1) dzieli się na trzy podstawowe elementy: widok schematyczny tabeli (podgląd tabeli przypomina strukturę folderów znanych z MS Windows), pole zapytań (pole umożliwiające ręczne wpisywanie zapytań SQL), pole wyniku (pole, w którym przedstawiony jest wynik zapytania). Dodatkowym elementem wspomagającym konstrukcje zapytań SQL jest *Query Bulider* dostępny w górnym pasku narzędziowym (ryc. 2). Za pomocą kreatora można stworzyć oraz przetestować zapytanie SQL. Dostępny widok przypomina opcję tworzenia kwerend znaną z MS Access.

Dodatkowe opcje sortowania dostępne w dolnej części programu pozwalają na doprecyzowanie zapytania. Tradycyjny podgląd można uzyskać przez naciśnięcie pola *Sql* umieszczonego w górnym pasku narzędziowym (ryc. 3).

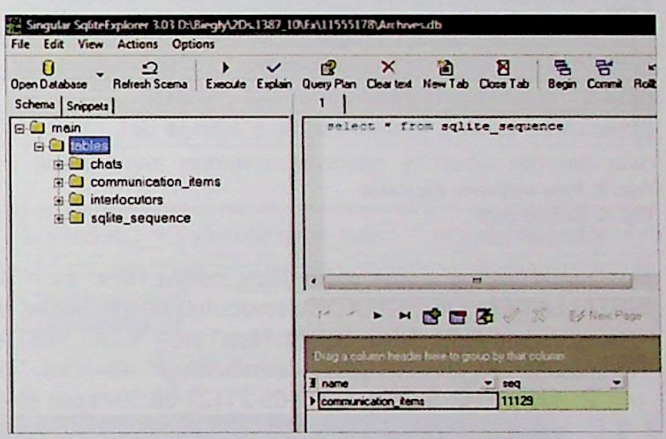
Natomiast pole *Sample* pokazuje przykładowe działanie zapytania.

Tak utworzone zapytanie można skopiować (Ctrl + C) i przenieść do *Pola zapytań* dostępnego w głównym panelu programu. W celu utworzenia nowego zapytania należy wyświetlić dostępne tabele, a następnie połączyć je w logiczną całość, przeciągając poszczególne komórki.

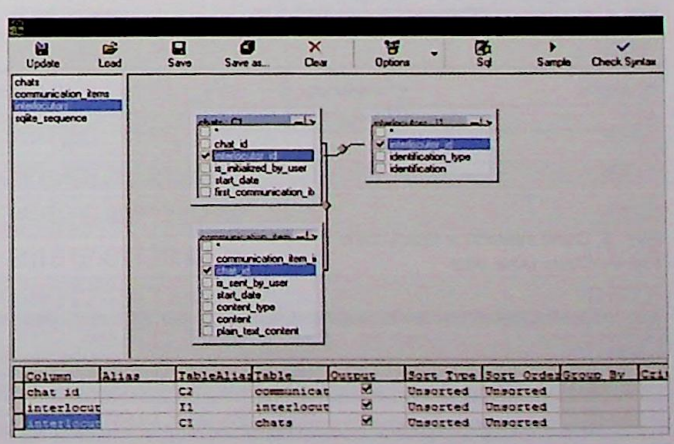
Studium przypadku

W celu przedstawienia zastosowania opisywanego narzędzia zaproponowano następujący przykład procedury analizy danych (rozmów) pochodzących z komunikatora internetowego Nowe GG z użytkownikiem o numerze 11127[5].

Ważnym elementem pracy z plikiem bazodanowym jest dokładne poznanie jego struktury oraz poszczególnych komórek determinujących prawidłowe rozpoznanie użytkownika, charakteru jego rozmowy, treści itp. Baza danych komunikatora zawarta jest w pliku: *Archive.db* w katalogu: *\Users\Nazwa Użytkownika\ AppData\ Roaming\ Gadu-Gadu 10\Numer Profilu*. W przypadku niezaszyfrowanego archiwum[6] można bezpośrednio uzyskać dostęp do wiadomości, m.in. przez podgląd pliku, np. z użyciem aplikacji wbudowanej w MS Windows – Notatnik. Jednak prowadzone w ten sposób badanie jest bardzo żmudne, a wygenerowanie raportu, w sposób umożliwiający jego płynną analizę, jest bardzo czasochłonne. Standardowa treść wiadomości zawarta jest w znacznikach: *Data/czas <span style="color:#000000; font-family:'MS Shell Dlg 2'; font-size:9pt; >{treść wiadomości}.*



Ryc. 1. Główne okno programu Singular SQLiteExplorer
Fig. 1. *Main window of SQLiteExplorer*
Źródło (ryc. 1–18): autorzy



Ryc. 2. Główne okno kreatora zapytań SQL
Fig. 2. *Main window of SQL query builder*

Po uruchomieniu programu: Sqlite3Explorer i załadowaniu pliku *Archive.db* wyświetlana jest jego struktura. Baza danych składa się z trzech podstawowych tabel zawierających odpowiednio: *chats* – rozmowy, *communication_items* – treści wiadomości, *interlocutors* – użytkownicy, z którymi przeprowadzone zostały rozmowy.

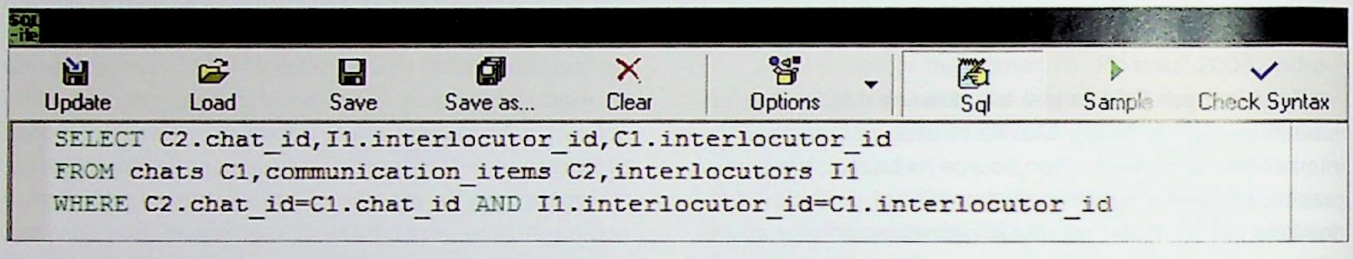
Tabela *chats*

Struktura tabeli *chats* została przedstawiona na rycinie 6.

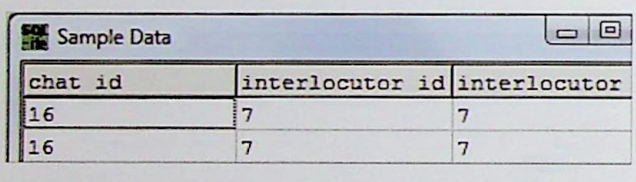
Tabela składa się z pięciu podstawowych elementów: *chat_id* – numer identyfikacyjny przeprowadzonej rozmowy, *interlocutor_id* – numer identyfikacyjny użytkownika, z którym przeprowadzona była rozmowa (przeprowadzone rozmowy zapisane są w tabeli *interlocutors*), *is_initialized_by_user* – znacznik, czy rozmowa została rozpoczęta przez „mającego archiwum" (1 = tak, 0 = nie), *start_date* – data i czas rozpoczęcia rozmowy, *first_communication_item_id* – numer identyfikacyjny pierwszej wiadomości należącej do tej rozmowy (z tabeli *communication_items*).

Tabela *communication_items*

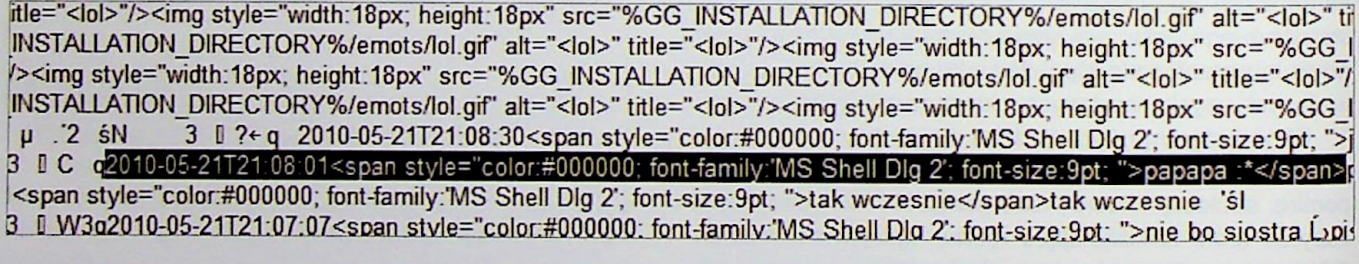
Struktura tabeli *communication_items* przedstawiona została na rycinie 7.



Ryc. 3. Zapytania SQL w formacie nieprzetworzonym
Fig. 3. *SQL statements in basic form*



Ryc. 4. Pola wynikowe zapytania
Fig. 4. *Sample data*



Ryc. 5. Fragment pliku tekstowego archiwum GG
Fig. 5. *Fragment of GG archive text file*

chat_id	interlocutor_id	is_initialized_by_user	start_date	first_communication_item_id
16	7	1	2010-02-15T09:12:49	44
18	7	0	2010-02-15T10:15:42	45
51	11	1	2010-04-26T15:23:28	109

Ryc. 6. Dane zawarte w tabeli chats
Fig. 6. *Chats table data*

commun	chat_id	is_sent_by_user	start_dat	content_type	content	plain_text_content
44	16	1	2010-02-15	0	czeŁ›Ąć ciocia co u waz</img	czeŁ›Ąć ciocia co u waz:)
458	16	1	2010-02-15	0	czeŁ›Ąć co uwaz	
459	18	0	2010-02-15	0	no czeŁ›Ąć	no czeŁ›Ąć
460	18	1	2010-02-15	0	czeŁ›Ąć	
464	18	1	2010-02-15	0	co sł uchać	

Ryc. 7. Dane zawarte w tabeli *communication_items*
Fig. 7. *Communication_items table data*

Tabela składa się z sześciu podstawowych elementów: *communication_item_id* – identyfikator wiadomości, *chat_id* – numer identyfikacyjny przeprowadzonej rozmowy w tabeli *chats*, *is_sent_by_user* – wiadomość wysłana przez „użytkownika archiwum" (1 = tak, 0 = nie), *start_date* – data i czas rozpoczęcia rozmowy, *content_type* – typ wiadomości (0 = tekstowa (ewentualnie z obrazkiem), 1 = SMS, 2 = plik), *content* – wiadmość wraz z formatowaniem tekstu.

Tabela *interlocutors*

Struktura tabeli *interlocutors* przedstawiona została na rycinie 8.

Tabela składa się z trzech podstawowych elementów: *cinterlocutor_id* – identyfikator rozmówcy w bazie, *identification_type* – rodzaj danych (0 = numer GG, 1 = numer telefonu), *identification* – numer GG lub telefonu (zgodnie z polem *identification_type*), w polu *identification* znajdują się również użytkownicy widniejący w komunikatorze GG jako „spoza listy".

Z akt sprawy wynikało, że interesujące, z punktu widzenia śledztwa, wiadomości pochodzą od użytkownika o numerze GG: 11127. W tabeli *interlocutors* sprawdzono kontakty w *Polu Zapytań*, wpisano: *select * from interlocutors*, otrzymując dane na temat zdefiniowanych w programie numerów kontaktów (ryc. 9).

interlocu	identification_type	identification
4	1	+4850283
12	1	+4850902
16	0	10364
18	0	1235
17	0	12536
21	0	13
27	0	14557
8	0	16

Ryc. 8. Dane zawarte w tabeli interlocutors
Fig. 8. *Interlocutors table data*

Na podstawie uzyskanego wyniku stwierdzono, że poszukiwanemu numerowi w bazie danych odpowiada identyfikator rozmówcy (*interlocutor_id*) 45. Następnie w tabeli *chats* sprawdzono numery identyfikacyjne przeprowadzonych rozmów (*chat_id*) dla użytkownika (*interlocutor_id*) 45. W *Polu Zapytań* wpisano więc *select * from chats WHERE interlocutor_id=45*, otrzymując dane przedstawione na rycinie 10.

Kolejnym krokiem jest sprawdzenie w tabeli *communication* poszczególnych rozmowów (*chat_id*) użytkownika o numerze 45. W celu sprawdzenia rozmowy nr 252, w *Polu Zapytań* wpisano *select * from communication_items WHERE chat_id=252* (kolejne rozmowy można przeglądać analogicznie, zmieniając jedynie ich numer porządkowy z kolumny *chat_id*). Następnie, aby dokonać eksportu wyników należy wybrać z menu *File* opcję *Save Result set as XML* i zapisać plik w wybranej lokalizacji na dysku. Tak wyeksportowany plik można później otworzyć, np. programem MS Excel, jako tabelę XML (ryc. 11).

Przeglądanie wynikowego zbioru XML z użyciem MS Excel daje dodatkową możliwość sortowania wyników, co zostało przedstawione na rycinie 12.

Mając na uwadze nadrzędny cel przeprowadzenia badań, czyli sporządzenie raportu rozmów prowadzonych przez komunikator Nowe GG z użytkownikiem o numerze 11127, należało skonstruować odpowiednie zapytanie SQL. Jak wynika z wcześniejszej analizy użytkownik o wskazanym numerze widnieje w bazie danych jako

interlocu	identification_type	identificatior
7	0	21551
45	0	11127
47	0	225820
16	0	74643

Ryc. 9. Dane o numerach GG poszukiwanych rozmówców
Fig. 9. *Output data – GG id numbers of requested interlocutors*

chat_id	interlocutor_id	is_initialized_by_user	start_date	first_communication_item_id
252	45	0	2010-06-28T21:23:51	9704
253	45	0	2010-06-29T14:10:41	9834
254	45	0	2010-06-29T19:22:32	9842
257	45	0	2010-06-29T21:47:15	9860
260	45	0	2010-06-30T17:02:58	9897
266	45	0	2010-07-01T20:43:25	10022
280	45	0	2010-07-10T13:48:52	10527
288	45	0	2010-07-14T11:57:24	10707
299	45	0	2010-07-28T16:29:28	10865
304	45	0	2010-08-09T12:46:45	10903

Ryc. 10. Dane dotyczące czatów prowadzonych przez użytkownika o numerze identyfikacyjnym 45
Fig. 10. *Data related to chats conducted by user id #45*

numer 45, w związku z tym w *Polu Zapytań* wpisano następujące zapytanie:

```
SELECT *
FROM communication_items C1,interlocutors I1,chats C2
WHERE C1.chat_id=C2.chat_id AND I1.interlocutor_id=45
```

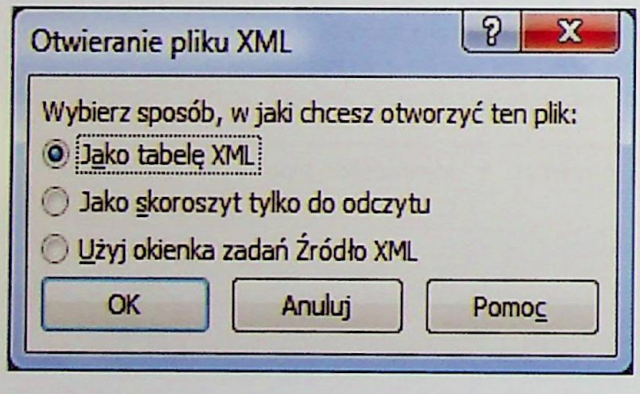
W tym miejscu należy zaznaczyć, że składnia SQL daje ogromne możliwości sortowania oraz odnajdowania konkretnych, poszukiwanych danych. Trzeba również podkreślić, że w celu budowy zapytań SQL-owych bardzo pomocne jest wspomniane wcześniej narzędzie *Query Builder*. Przykładowe zapytania mogą wyglądać następująco:

- Podgląd całego archiwum:

```
SELECT *
FROM communication_items C1,interlocutors I1,chats C2
WHEREC1.chat_id=C2.chat_idANDI1.interlocutor_id=C2.-
interlocutor_id
```

- Podgląd rozmów jednego użytkownika, w rozpatrywanym przypadku użytkownik 45:

```
SELECT *
FROM communication_items C1,interlocutors I1,chats C2
WHERE C1.chat_id=C2.chat_id AND I1.interlocutor_
id=45
```



Ryc. 11. Otwieranie pliku XML z użyciem programu MS Excel
Fig. 11. *XML file open dialog window in MS Excel application*

- Podgląd rozmów użytkowników z uwzględnieniem pól: rozmowy (*content*), data (*start_date*), numer GG rozmówcy (*identification*):

```
SELECT C1.content,C2.start_date,I1.identification
FROM communication_items C1,interlocutors I1,chats C2
WHERE C1.chat_id=C2.chat_id AND I1.interlocutor_id=C2.
interlocutor_id
```

Podsumowanie

Autorzy wykazali, że możliwy jest dostęp do pliku *Archive* komunikatora Nowe GG. Za pomocą narzędzia Sqlite3Explorer dokonano tego i tego, co pozwala na to i na to. Dzięki możliwości wykorzystania języka SQL możliwe jest praktycznie nieograniczone uzyskiwanie dowolnych informacji z archiwum rozmów dotyczących np. wysyłanych plików (czy zostały odebrane).

Z praktyki – program Torrent Analyzer

Kolejna część artykułu zawiera opis możliwości wykorzystania opracowanego narzędzia Torrent Analyzer w procedurach analizy plików konfiguracyjnych klientów sieci wymiany plików w Internecie. W prowadzonych postępowaniach często zachodzi konieczność ustalenia, czy dane pliki udostępniono szerszemu gronu użytkowników. Autorski program pozwala na praktyczną i wygodną analizę plików konfiguracyjnych wybranych programów klienckich sieci P2P posługujących się protokołem wymiany BitTorrent, a efektem końcowym działania programu może być wygenerowanie przejrzystego raportu.

Sieci typu *peer-to-peer networking* cieszą się niemalejącą popularnością od czasów Napstera[7]. BitTorrent jest protokołem wymiany plików w sieciach P2P dedykowanym do rozprowadzania dużych ilości danych. Protokół został opracowany przez Brama Cohena i opublikowany w lipcu 2009 roku. Szacuje się, że obecnie wykorzystuje go od 25% do 55% transmisji (w zależności od lokalizacji

	L	M	N	O	P	
1	chat_id	is_sent_by_user	start_date	content_type	content	plain_text_content
2						
3						
4						
5						
6						
7						
8						
9	252	0	2010-06-28 21:23	0	<span style='	masz nowe gg ;P
10	252	1	2010-06-28 22:36	0	<span style='	nie cały czas to samo
11	252	0	2010-06-28 22:36	0	<span style='	??
12	252	0	2010-06-28 22:36	0	<span style='	nie rozumiem
13	252	1	2010-06-28 22:36	0	<span style='	pytałaś czy mam nowe gg

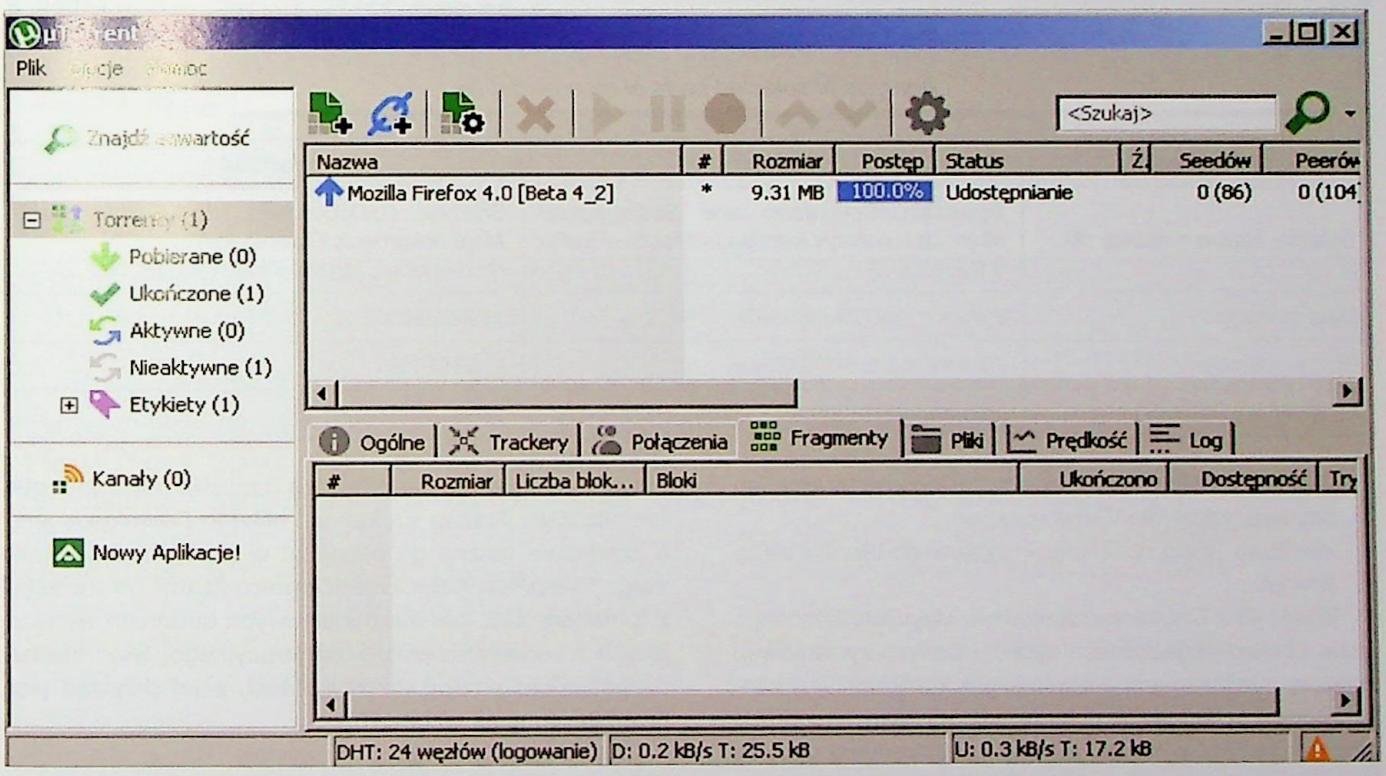
Ryc. 12. Arkusz z danymi XML otwarty w programie MS Excel
Fig. 12. *XML data opened as MS Excel spreadsheet*

geograficznej)[8] w Internecie. Nie są znane dokładne dane dotyczące tego, ile z tych dystrybuowanych treści jest nielegalnych (łamanie autorskich praw majątkowych, praw własności intelektualnej, pornografia z udziałem osób poniżej piętnastego roku życia itp.).

W praktyce kryminalistycznej kluczowa dla sprawy jest przede wszystkim analiza plików konfiguracyjnych programu klienckiego obsługującego rzeczony protokół. W związku z tym zostało opracowane odpowiednie oprogramowanie przeznaczone do analizy programów klienckich obsługujących protokół BitTorrent. Wybrane programy klienckie, które obsługuje opracowana aplikacja, np. Torrent Analyzer, opisana w niniejszym artykule to µTorrent, BitComet oraz Vuze (dawniej Azureus). Program skupia się na analizie najistotniejszych plików konfiguracyjnych, jakimi są pliki zawierające informacje na temat ilości pobranych danych, ilości wysłanych danych, nazwie dokumentu oraz pierwotnej ścieżce zapisu.

Program µTorrent – podstawowe informacje

W przypadku programu µTorrent informacje dotyczące aktualnie pobieranych i wysyłanych dokumentów znajdują się w pliku *resume.dat* bez względu na to, czy aplikacja była instalowana w systemie czy uruchomiona jako wersja *portable* (czyli nieinstalowalna). Dla ułatwienia lokalizacji poszukiwanego pliku przyjęto oznaczenie ścieżki do folderu, w którym aplikacje umieszczają najczęściej swoje pliki jako *%AppPath%*. Przykładowo w systemach:



Ryc. 13. Główne okno programu µTorrent v2.2
Fig. 13. *µTorrent v2.2 main window*

Tabela 1

Sposób kodowania poszczególnych typów danych używany w kodowaniu Bencode

Bencode encoding for various types of data

Typ danych	Wzór	Przykłady
Tekstowy	*<długość tekstu>:<wartość>*	*7:policja* -> policja
Liczbowy całkowity	**i***<wartość>***e**	*i42e* -> 42
Lista	**l***<zakodowane wartości>***e**	*l7:policja3:p2pe* -> {policja, p2p}
Słownik	**d***<zakodowana wartość><zakodowany klucz>***e**	*d2:ja4:moje2:ty5:twojee* -> {ja-moje; ty-twoje}

d10:.fileguard40:29E41A7DFED14DD39D1D370FCC5A082721BB9D4B38:Mozilla Firefox4.0[Beta4_2].torrentd8:added_oni1292039997e9:antivirusIli0ei0eee9:app_owner0:6:blocksle9:blocksizei16384e6:cachedi0e7:caption30:Mozilla Firefox 4.0 [Beta 4_2]5:codeci0e12:completed_oni1292040043e7:corrupti0e3:dhti13e12:download_url0:10:downloadedi9764624e9:downspeedi0e7:episodei0e10:episode_toi0e8:feed_url0:9:hashfailsi0e4:have19:····4:info20:FF□¨E±÷CŤń^ái˘bnż,Ŕ18:lastseencompletei1292042514e11:last_activei2270e3:lsdi8e5:movedi1e5:orderi1e21:overrid e_seedsettingsi0e4:path86:C:\Documents and Settings\arek\Moje dokumenty\Downloads\Mozilla Firefox 4.0 [Beta4_2]6:peers63438:4:prio:5:prio2i1e7:qualityi0e8:rss_name0:7:runtimei2352e9:scrambledi0e6:seasoni0e8:seedtimei2307e7:startedi2e9:superseedi0e19:superseed_cur_piecei0e4:timei1292042527e11:trackermodei3e8:trackersll44:udp://tracker.openbittorrent.com:80/announce45:http://tracker.openbittorrent.com:80/announceel39:http://tracker.publicbt.com:80/announce38:udp://tracker.publicbt.com:80/announcee41:udp://denis.stalker.h3q.com:6969/announce42:http://denis.stalker.h3q.com:6969/announcee7:ulslotsi0e8:uploadedi16384e7:upspeedi0e7:use_utpi1e5:validi1e7:visiblei1e12:wanted_ratioi1500e15:wanted_seedtimei0e5:wastei237807e8:webseedsleee

Ryc. 14. Listing – przykładowa zawartość pliku *resume.dat*
Fig. 14. *Listing – example content of resume.dat file*

Tabela 2

Analiza wybranych fragmentów pliku *resume.dat*

Analysis of selected parts of resume.dat file

Własność	Fragment	Wartość
Ścieżka zapisu z nazwą pliku	*4:path86:C:\Documents and Settings\arek\ Moje dokumenty\Downloads\Mozilla Firefox 4.0 [Beta4_2]*	Ścieżka: *C:\Documents and Settings\arek\ Moje dokumenty\Downloads* Nazwa pliku: *Mozilla Firefox 4.0 [Beta4_2]*
Data dodania	*8:added_oni1292039997e*	*1292039997*
Ile danych pobrano?	*10:downloadedi9764624e*	*9764624* B
Ile danych wysłano?	*8:uploadedi16384e*	16384 B

- Windows XP folderem tym jest *C:\Documents and Settings\<uzytkownik>\Dane aplikacji*,
- Windows Vista *C:\Users\<użytkownik>\AppData\Roaming*,
- Windows 7 *C:\Users\<użytkownik>\AppData\Roaming*,

gdzie *C:* jest oznaczeniem dysku – partycji systemowej. Zgodnie z założeniami domyślnie plik konfiguracyjny z informacjami o aktualnie pobieranych/wysyłanych plikach można odnaleźć w *%AppPath%\uTorrent\resume.dat*. Plik ten jest zakodowany systemem Bencode[9]. Sposób kodowania poszczególnych typów danych w tym systemie przedstawiony jest w tabeli 1. Przykładowa zawartość pliku *resume.dat* przedstawiona jest na rycinie 14. Dokładna analiza wybranych fragmentów przedstawiona jest w tabeli 2. Wartość czasu odpowiadająca temu, kiedy plik został dodany zapisana jest w formacie uniksowym[10].

Klient BitComet – podstawowe informacje

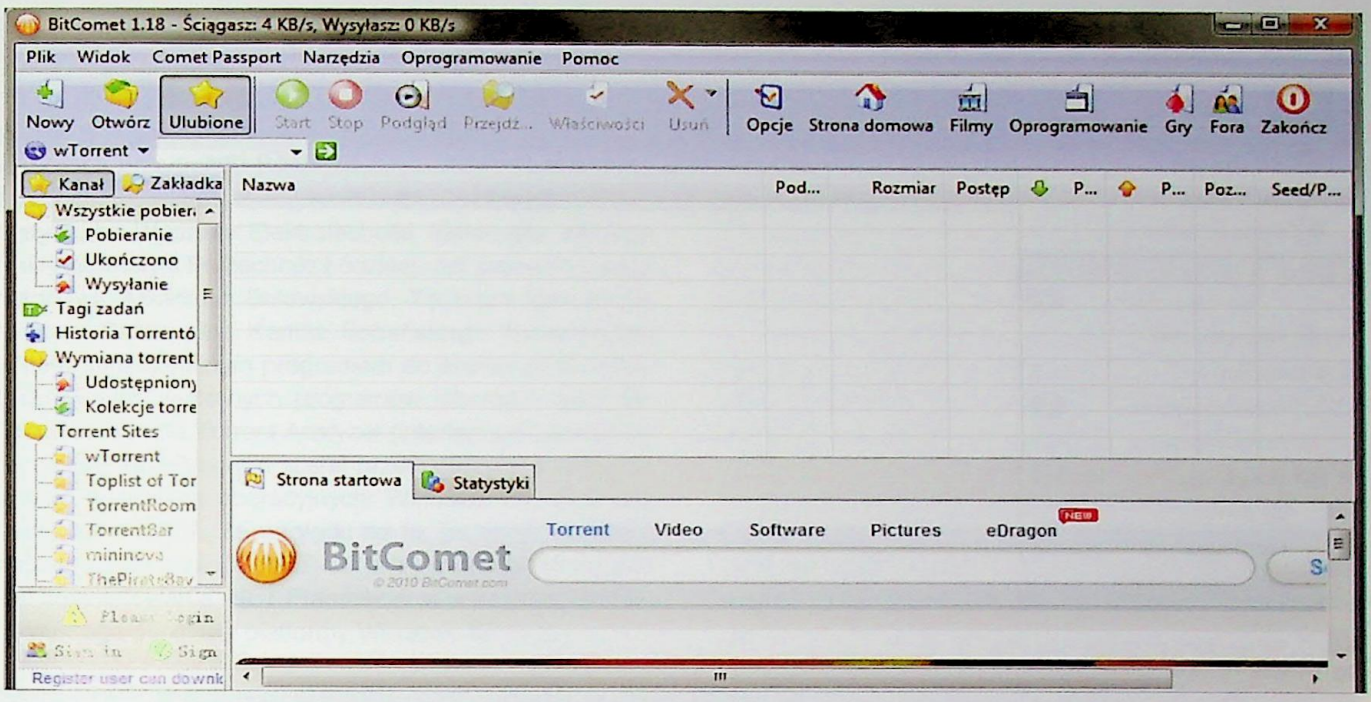
W odróżnieniu od klienta µTorrent, program BitComet (ryc. 15) zawiera interesujące, z punktu widzenia analizy kryminalistycznej, informacje na temat aktualnie pobieranych/wysyłanych plików w bardziej uniwersalnym, przejrzystym formacie (zarówno jeśli chodzi o podgląd, jak i ewentualną późniejszą obróbkę danych), jakim jest plik formatu XML. Nazwa szukanego pliku to *Dawnloads.xml*, a domyślnie można go odnaleźć w następującej lokalizacji: *%AppPath%\BitComet\Downloads.xml*. W związku z formatem XML nie ma dodatkowych utrudnień wynikających z kodowania pliku konfiguracyjnego, więc można bezpośrednio przejść do zawartości, a jej przykład jest przedstawiony na rycinie 16.

Jak można zauważyć na rycinie 16, w przypadku klienta BitComet, wszystkie informacje są przedstawione w prostej, czytelnej formie, a analizę można przeprowadzić nawet bez użycia specjalistycznego oprogramowania. Dokładną analizę wybranych fragmentów przedstawiono w tabeli 3.

Program Vuze/Azureus – podstawowe informacje

Program Vuze (ryc. 17) jest następcą klienta BitTorrent o nazwie Azureus; obecnie spotykane są jeszcze obydwie wersje programu, ale stopniowo nowsza wersja wypiera poprzednika.

W przypadku programu Vuze pliki konfiguracyjne, podobnie jak w przypadku µTorrent, zakodowane są w systemie Bencode. Poszukiwany plik nosi nazwę *downloads.*



Ryc. 15. Główne okno programu BitComet v1.18
Fig. 15. *BitComet v1.18 main window*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<BitComet Author="RnySmile" Version="0.1">
  <Torrents>
    <Torrent InfoHashHex="46467fa845b1f7438df10a5ee169a1626ebf82c0"
Torrent="Mozilla Firefox 4.0 [Beta 4_2].torrent" CreateDate="2010-12-11 06:36:37.895147"
FinishDate="2010-12-11 06:37:38.261950" SaveDirectory="C:\Downloads"
SaveName="Mozilla Firefox 4.0 [Beta 4_2]" Description="Torrent downloaded from
http://thepiratebay.org" Size="9764624" SelectedSize="9764624" Left="0" DataUpload="0"
DataDownload="10157840" DataDiscard="196608" PreviewState="0" FileOrderList="0"
TorrentFile="Torrents\Mozilla Firefox 4.0 [Beta 4_2].torrent"
SaveLocation="C:\Downloads\Mozilla Firefox 4.0 [Beta 4_2]" ShowName="Mozilla Firefox
4.0 [Beta 4_2]" AutoRun="true" SnapSent="true" Tags="Programy"/>
  </Torrents>
</BitComet>
```

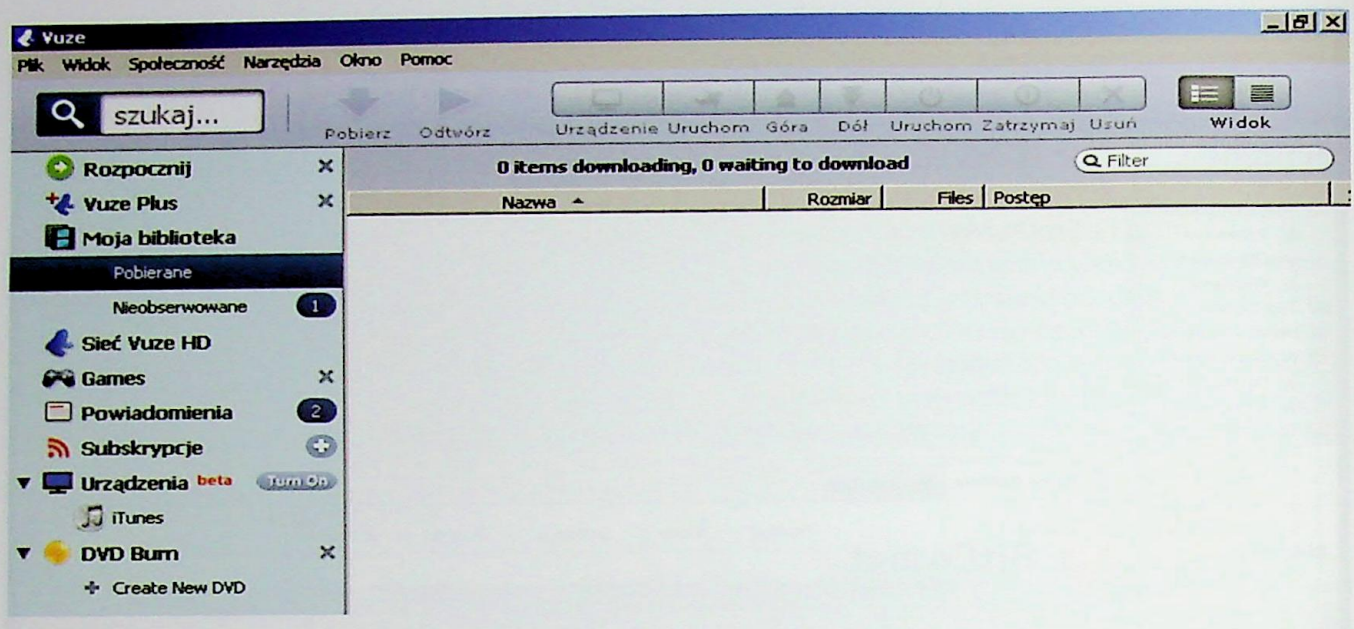
Ryc. 16. Listing – przykładowa zawartość pliku *Downloads.xml* programu BitComet
Fig. 16. *Listing – example content of Downloads.xml file*

Tabela 3

Analiza wybranych fragmentów pliku *Downloads.xml*

Analysis of selected parts of Downloads.xml file

Własność	Fragment	Wartość
Nazwa pliku	*SaveName="Mozilla Firefox 4.0 [Beta 4_2]"*	*Mozilla Firefox 4.0 [Beta 4_2]*
Data dodania	*CreateDate="2010-12-11 06:36:37.895147"*	*2010-12-11 06:36:37.895147*
Ścieżka zapisu	*SaveDirectory="C:\Downloads"*	*C:\Downloads*
Ile danych wysłano?	*DataUpload="0"*	*0* [B]
Ile danych pobrano?	*DataDownload="10157840"*	*10157840* [B]



Ryc. 17. Główne okno programu Vuze v4.5.1.0
Fig. 17. *Vuze v4.5.1.0 main window*

config. Jego lokalizacja to *%AppPath%\Azureus\downloads.config*, a jego przykładowa zawartość została przedstawiona na rycinie 18. Dokładną analizę wybranych fragmentów pliku konfiguracyjnego klienta Vuze zaprezentowano w tabeli 4. Informacja dotycząca daty dodania została zapisana w identycznym formacie jak w przypadku programu µTorrent (tzw. uniksowy format czasu).

d9:downloadsld9:allocatedi1e9:completedi1000e12:creationTimei1292049169619e9:discarded i851968e10:downloadedi10729731e15:file_prioritiesli-1ee10:forceStarti0e13:hashfailbytesi0e5:maxdli0e5:maxuli0e10:persistenti1e8:positioni1e8:save_dir60:C:\Documents and Settings\arek\Moje dokumenty\Vuze Downloads9:save_file30:Mozilla Firefox 4.0 [Beta 4_2]18:secondsDownloadingi84e18:secondsOnlySeedingi848e5:statei75e7:torrent130:C:\Documents and Settings\arek\Dane aplikacji\Azureus\torrents\Mozilla_Firefox_4_0_Beta_4_2_Faster__Safer__Smarter__Better.torrent12:torrent_hash20:FF□¨E±÷CŤń ^ái˘bnż,Ŕ8:uploadedi1784e7:uploadsi4eeee

Ryc. 18. Listing – Przykładowa zawartość pliku *downloads.config*
Fig. 18. *Listing – Example content of downloads.config file*

Tabela 4

Analiza wybranych fragmentów pliku *downloads.config*

Analysis of selected parts of downloads.config file

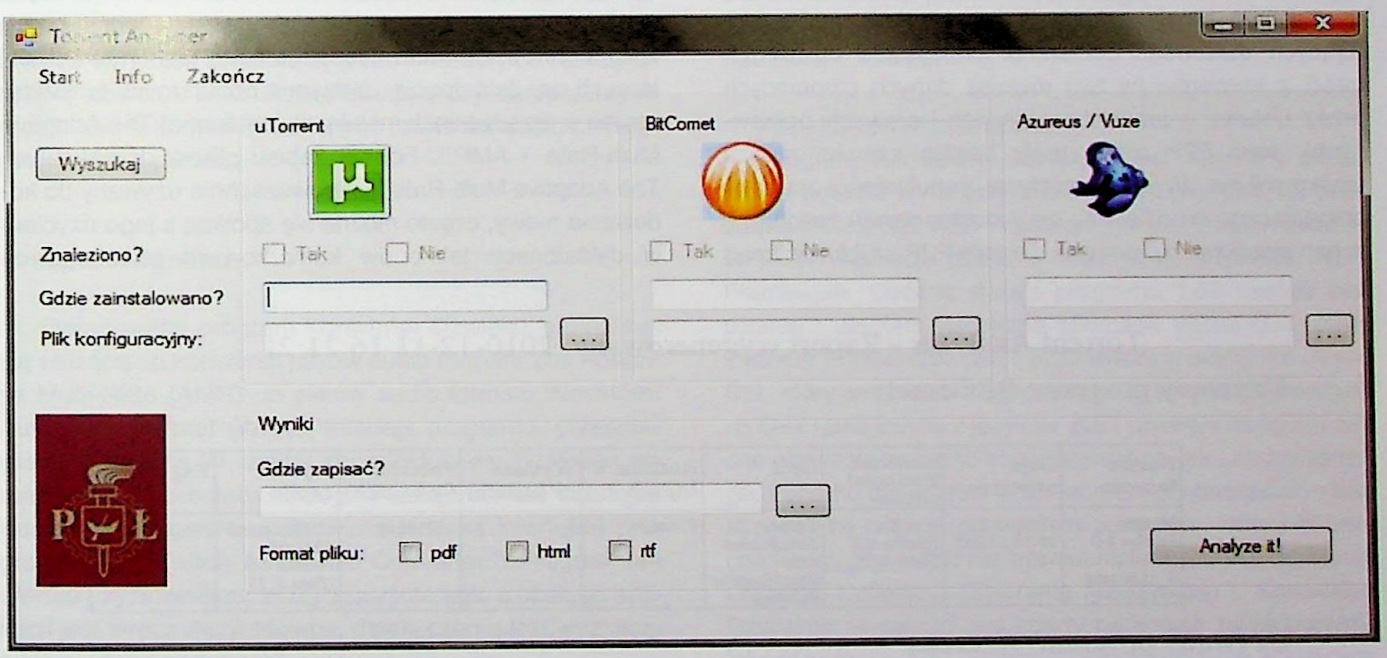
Własność	Fragment	Wartość
Ścieżka zapisu	*8:save_dir60:C:\Documents and Settings\arek\Moje dokumenty\Vuze Downloads*	*C:\Documents and Settings\arek\Moje dokumenty\Vuze Downloads*
Nazwa pliku	*9:save_file30:Mozilla Firefox 4.0 [Beta 4_2]*	*Mozilla Firefox 4.0 [Beta 4_2]*
Data dodania	*creationTimei1292049169619e*	*1292049169619*
Ile danych pobrano?	*10:downloadedi10729731e*	*10729731* [B]
Ile danych wysłano	*8:uploadedi1784e*	*1784* [B]

Funkcjonalność programu Torrent Analyzer

Program powstał w roku 2009 pod kierunkiem dr inż. Andrzeja Romanowskiego i mgr inż. Tomasza Pawlickiego w ramach przedmiotu Projekt Kompetencyjny wykładanego na studiach stacjonarnych I stopnia, na kierunku Informatyka, na Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej. Już pierwsza wersja autorstwa Arkadiusz Bolewskiego, Tomasza Marciniaka, Michała Dubla oraz Kamila Sobańskiego zaowocowała w pełni funkcjonalnym programem do analizy plików konfiguracyjnych wybranych programów klienckich sieci BitTorrent o nazwie Torrent Analyzer (interfejs przedstawiono na rycinie 19). Program został przetestowany w następujących systemach operacyjnych: Windows XP, Windows Vista, Windows 7. Ze względu na to, że został napisany z użyciem technologii .NET C# firmy Microsoft konieczne jest zainstalowanie .NET Framework w wersji 3.5, co z kolei wymaga minimum platformy Windows XP SP3.

Jednym z podstawowych założeń programu jest bardzo intuicyjna i prosta obsługa, która ogranicza się do wyszukania plików konfiguracyjnych (automatycznie lub ręcznie) oraz wyboru ścieżki zapisu raportu i jego formatu. Najbardziej istotnym mechanizmem jest proces wyszukiwania. Analiza, czy dany program kliencki P2P został zainstalowany w systemie, odbywa się przez sprawdzenie wybranych kluczy rejestru systemu oraz ich wartości przedstawionych w tabeli 5.

Decyzja o sprawdzaniu wartości kluczy rejestru (przedstawionych w tabeli 5) została oparta na analizie wstecznej wersji programów klienckich. Autorzy są świadomi niedoskonałości tej metody, ale przeszukiwanie całego rejestru systemowego pod kątem wystąpienia słów kluczowych byłoby wysoce nieefektywne. Ze względu na wspomnianą niedoskonałość, oprócz automatycznego sprawdzania, czy dany program został zainstalowany w systemie, zaimplementowano także możliwość ręcznego wskazania pliku do analizy dla każdego z programów klienckich BitTorrent.



Ryc. 19. Główne okno programu Torrent Analyzer
Fig. 19. *Main window of Torrent Analyzer*

Tabela 5

Klucze rejestru i odpowiadające im wartości użyte w procesie wyszukiwania

Registry keys and corresponding values used in the search process

Program	Klucz rejestru	Wartość
µTorrent	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\ HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\	uTorrent
BitComet	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\ HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\	BitComet
Vuze	HKEY_LOCAL_MACHINE\SOFTWARE\ HKEY_CURRENT_USER\SOFTWARE\	Azureus

W związku z tym, jeżeli w przyszłości wpisy do rejestru omawianych programów klienckich będą się znajdować w innych miejscach, nie będzie to oznaczało automatycznego zmniejszenia funkcjonalności programu Torrent Analyzer. Doświadczona osoba, która wie jakich wpisów i plików należy szukać, bez problemu poradzi sobie bez stuprocentowo automatycznego wyszukiwania.

Raport końcowy z przeprowadzonej analizy można zapisać w trzech formatach: PDF, HTML, RTF. Przykładowy wynik pracy programu Torrent Analyzer pokazano na rycinie 20. Rezultat analizy przedstawiony jest w formie czytelnej tabeli, w której uwzględniono najważniejsze informacje takie jak: data i czas wygenerowania raportu, data dodania torrenta, nazwa pliku i ścieżka zapisu, ilość pobranych danych, ilość wysłanych danych.

Kierunki dalszego rozwoju aplikacji Torrent Analyzer

Pomimo silnej konkurencji ze strony serwisów oferujących możliwość pobierania praktycznie dowolnych treści z serwerów (w tym również danych chronionych przez Ustawę o prawach autorskich i prawach pokrewnych), sieci P2P ciągle mają bardzo szeroką rzeszę użytkowników. W opinii autorów popularność zjawiska polegającego na dzieleniu się (udostępnianiu) zasobami, w tym pirackimi, nie pozwoli sieciom P2P szybko zniknąć z Internetu. W związku z tym trwa rozbudowa funkcjonalności programu Torrent Analyzer, który w swojej finalnej wersji będzie obejmował nie tylko protokół BitTorrent, ale też inne popularne sieci P2P, m.in. eDonkey, Gnutella, FastTrack. Projekt aplikacji powstaje w ramach pracy inżynierskiej Arkadiusza Bolewskiego pod kierunkiem dr inż. Andrzeja Romanowskiego oraz we współpracy merytorycznej z mgr inż. Tomaszem Pawlickim. Planowane jest w opracowywanej aplikacji udoskonalenie procesów przeszukiwania oraz wzbogacenia programu o funkcje umożliwiające przetwarzania plików najpopularniejszych programów klienckich sieci P2P.

Z praktyki – Konwerter Dźwięku

W niniejszej, ostatniej części artykułu opisany jest opracowany program do konwersji zapisów dźwiękowych – Konwerter Dźwięku.

Problem konwersji różnych formatów zapisu dźwięku występuje często przy analizie dowodów pochodzących z komputerów, ale coraz częściej także z telefonów komórkowych czy dyktafonów. Jednym z popularnych formatów zapisu w urządzeniach mobilnych jest format The Adaptive Multi-Rate – AMR[11]. Format zapisu plików dźwiękowych The Adaptive Multi-Rate jest powszechnie używany do kodowania mowy, często można się spotkać z jego użyciem w dyktafonach telefonów komórkowych rejestrujących

Torrent Analyzer - Raport wygenerowany: 2010-12-11 16:21:21

Używany program: BitComet

Nazwa torrenta	Data utworzenia	Nazwa pod którą został zapisany	Opis	Rozmiar w bajtach	Wysłano (w bajtach)	Pobrano (w bajtach)	Ścieżka zapisu	Tagi
Mozilla Firefox 4.0 [Beta 4_2].torrent	2010-12-11 06:36:37.895 147	Mozilla Firefox 4.0 [Beta 4_2]	Torrent downloaded from http://thepiratebay.org	9764624	0	10157840	C:\Downloads\Mozilla Firefox 4.0 [Beta 4_2]	Programy

Używany program: Azureus

Data utworzenia	Nazwa pod którą został zapisany	Pobrano (w bajtach)	Wysłano (w bajtach)	Ścieżka zapisu
11 grudnia 2010	Mozilla Firefox 4.0 [Beta 4_2]	10729731	270849	C:\Documents and Settings\arek\Moje dokumenty\Vuze Downloads

Używany program: uTorrent

Data utworzenia	Nazwa pod którą został zapisany	Pobrano (w bajtach)	Wysłano (w bajtach)	Ścieżka zapisu
11 grudnia 2010	Mozilla Firefox 4.0 [Beta 4_2]	9764624	16384	C:\Documents and Settings\arek\Moje dokumenty\Downloads\Mozilla Firefox 4.0 [Beta 4_2]

Ryc. 20. Przykładowy raport wygenerowany przez program Torrent Analyzer, zapisany w formacie PDF
Fig. 20. *A sample report generated using Torrent Analyzer in PDF format*

mowę. Stąd też często jest wykorzystywany w laboratoriach kryminalistycznych. Gros zadań, jakie są wykonywane z materiałem dowodowym w tym formacie, sprowadza się do jego konwersji na format Waveform Audio – WAV[12], co stosowane jest choćby w celu umożliwienia prezentacji takiego materiału na różnych komputerach, także takich niewyposażonych w specjalistyczne oprogramowanie do odczytu plików AMR. Sytuacja ta bardzo często zdarza się np. w sądach, gdzie komputery nie mają zainstalowanych bibliotek do odtwarzania plików AMR.

Obecnie na rynku dostępnych jest wiele programów służących do konwersji dźwięku. Są to aplikacje mniej lub bardziej złożone oferujące konwersje do najróżniejszych formatów, najczęściej do popularnego formatu MP3, przy czym większość z nich to drogie aplikacje mające szereg zaawansowanych opcji, które nie są wykorzystywane przez przeciętnego użytkownika. Dlatego też prezentowane oprogramowanie jest wyspecjalizowane do pracy w laboratorium kryminalistycznym z konkretnym formatem dźwiękowym – AMR. Format Adaptive Multi-Rate cechuje się wysokim stopniem kompresji, niską przepustowością, a co za tym idzie prostotą sygnału oraz zwykle niższą jakością dźwięku, co znacznie utrudnia uzyskanie dobrych rezultatów w konwersji do innych formatów audio oferujących wyższą jakość.

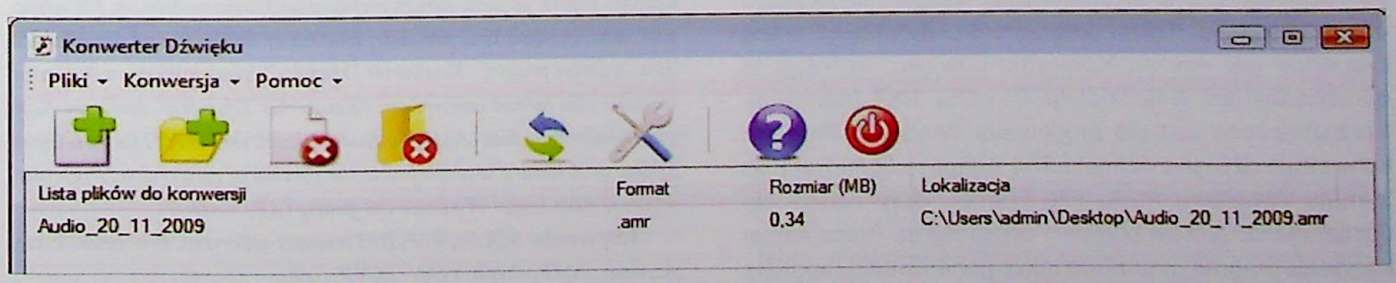
Opis Programu

Opracowany program Konwerter Dźwięku jest aplikacją służącą do konwersji plików audio formatu The Adaptive Multi-Rate (AMR) do plików audio formatu Waveform Audio File Format (WAV). Interfejs programu, przedstawiony na rycinie 21, został zaprojektowany z założeniem maksymalnej prostoty obsługi. Funkcjonalność interfejsu Konwertera Dźwięku wzorowana była na interfejsie programu NCH Switch Audio File Converter[13]. Program ten oferuje użytkownikowi przejrzysty interfejs, a obsługa aplikacji jest wręcz instynktowna, dzięki czemu jest wyznacznikiem wśród podobnych aplikacji.

Procedura konwersji plików dźwiękowych odbywa się w trzech etapach. Najpierw użytkownik dodaje plik/pliki, tudzież folder z plikami przeznaczonymi do konwersji. Dodane pliki są wyświetlane w oknie aplikacji w czterech kolumnach, tj. nazwa pliku, format pliku, rozmiar pliku i lokalizacja pliku. Kolejnym krokiem jest możliwość zmiany folderu docelowego, w którym zapisane zostaną pliki po konwersji, a także możliwość zmiany opcji używanego kodeka WAV, tj. próbkowania, formatu. Ostatnią czynnością jest przeprowadzenie samej konwersji pliku. W wyniku konwersji otrzymywany jest plik w formacie Waveform Audio File Format[14]. Pliki WAV mogą być zapisywane z użyciem kodeków, możliwe jest także zastosowanie nieskompresowanego formatu Pulse Code Modulation – PCM. PCM to metoda reprezentacji sygnału cyfrowego, która polega na przedstawieniu wartości chwilowej sygnału w określonych odstępach czasu, czyli z określoną częstością[15]. Wadą formatu WAV jest z jednej strony duży rozmiar tworzonych plików, a z drugiej ograniczenie jego maksymalnego rozmiaru (maksymalnie do 4 GB), pomimo to format ten nadaje się doskonale do edycji dźwięków. Po udanej konwersji generowany jest także raport jako plik tekstowy. Zawiera on dane na temat nazwy, formatu, rozmiaru i lokalizacji plików wejściowego oraz wynikowego. W raporcie podane są także dokładne data i czas przeprowadzonej konwersji.

Aspekty techniczne aplikacji

Program Konwerter Dźwięków jest tworzony przez Pawła Kubicza w ramach jego pracy inżynierskiej na podstawie technologii .NET Framwork i języka programowania C++/CLI. Aplikacja działa w środowisku Windows .NET Framework. Obecna wersja programu 1.00 została napisana z użyciem narzędzia Microsoft Visual C++ 2008 Express Edition oraz .NET Framework w wersji 3.5. C++/CLI, który jest nowym językiem programowania opartym na C++ i jako jedyny z języków .NET pozwala na jednoczesne używanie zarówno kodu natywnego, jak i zarządzanego. Ponadto ma wygodny dla programisty finalizator – jest to specjalny niedeterministyczny destruktor, który jest uruchamiany, gdy obiekt ma być usunięty w ramach działania Garbage Collection (Zbieranie Nieużytków)[16]. Konwerter Dźwięków w całości jest oparty na kodzie zarządzanym. Wykorzystanie .NET Framework przejawia się w każdym aspekcie aplikacji, zaczynając od interfejsu, przez zarządzanie plikami i folderami, a kończąc na zapisie plików wynikowych. Kolejnym ważnym aspektem technicznym Konwertera Dźwięku jest konwersja plików audio. Obecnie



Ryc. 21. Główne okno programu Konwerter Dźwięku
Fig. 21. *Konwerter Dźwięku main window*

zaimplementowany jest format Adaptive Multi-Rate Narrowband, ale w przyszłości Konwerter Dźwięku będzie także obsługiwał nowszy format Adaptive Multi-Rate Wideband wykorzystywany w sieciach UMTS. Obydwa formaty AMR i AWB są kodekami mowy kompresji stratnej używanymi w trzeciej generacji telefonów komórkowych i zdefiniowanymi przez 3rd Generation Partnership Project (3GPP)[17]. Różnią się one przepustowością, dla pliku w formacie AMR jest to 12,2 kbit/s, a dla pliku w formacie AWB 23,85 kbit/s. W samym procesie konwersji wykorzystane są biblioteki edytora dźwięku Sound eXchange (SoX)[18] dostępne na licencji *open source*. Wykorzystany kodek to 3GPP Adaptive Multi-Rate Floating-point (AMR) Speech Codec, który został opracowany w ramach projektu 3GPP[19].

Podsumowanie

Artykuł prezentuje wycinek prac badawczych prowadzonych w LK KWP w Łodzi we współpracy z Politechniką Łódzką. Przedstawione zagadnienia zostały podzielone na trzy części. W pierwszej części autorzy wskazali procedurę odtwarzania wybranych danych dotyczących działalności użytkowników systemu komputerowego, tj. czatów internetowych prowadzonych za pośrednictwem komunikatora internetowego Gadu-Gadu. Dane takie mogą być dodatkowym źródłem informacji o istotnych faktach dotyczących prowadzonego postępowania. Codzienna praca ekspertów kryminalistyki/biegłych to również konieczność analizy programów klienckich sieci P2P. W sieci istnieje wiele wersji oprogramowania do wymiany i udostępniania plików P2P, co wiąże się z dużą ilością i rodzajem plików konfiguracyjnych. Jak zauważono na łamach niniejszej publikacji w przypadku programów µTorrent i Vuze wpisy generowane są za pomocą metody kodowania (Bencode), a inny program – BitComet wykorzystuje format XML. Jednak ta sama informacja, w zależności od użytego klienta, różni się od siebie słowami kluczowymi. Warto w tym miejscu podkreślić, że nie istnieje żaden standard opisujący, w jakiej formie dane mają być zapisywane do plików konfiguracyjnych (wyjątek stanowi kodowanie poszczególnych plików). W tym wypadku determinującym elementem jest wyłącznie kreatywność programistów, dlatego aby ułatwić i skrócić czas na wykonanie badania wydaje się zasadne korzystanie z proponowanego, wyspecjalizowanego oprogramowania. Takim programem ułatwiającym pracę ekspertów jest również Konwerter Dźwięku. Aplikacja ta opracowana jest w technologii C++/CLI .NET framework i przeznaczona jest dla środowiska Windows. Program konwertuje dźwięk z formatu The Adaptive Multi-Rate do formatu Waveform Audio File Format, który nadaje się doskonale do dalszej obróbki i edycji audio. Prócz samej konwersji program umożliwia także generowanie raportów dotyczących każdego przeprowadzonego zadania, co jest niezbędne dla potrzeb kryminalistycznych.

Streszczenie

Niniejszy artykuł prezentuje kilka istotnych, z punktu widzenia informatyki śledczej, zagadnień praktycznych, które zostały podzielone na trzy części. Część pierwsza zawiera opis możliwości analizy informacji przechowywanych w bazach danych SQLite. Na przykładzie odtwarzania danych dotyczących czatów prowadzonych za pomocą komunikatora internetowego Gadu-Gadu przedstawiono procedurę uzyskiwania informacji z kontenera bazodanowego. SQLite jest obecnie jednym z najpopularniejszych rozwiązań bazodanowych, szczególnie dla systemów wbudowanych i jest używany przez wiele popularnych programów; m.in. Skype, Firefox czy też omawiany komunikator GG. Kolejne dwa zagadnienia przedstawiają część wyników prac badawczych dotyczących opracowywania nowoczesnych narzędzi informatycznych na potrzeby kryminalistyki prowadzonych we współpracy z LK KWP Łódź oraz Politechniką Łódzką. Pierwszy z opisanych projektów to Torrent Analyzer, który jest aplikacją pozwalającą na analizę plików konfiguracyjnych najpopularniejszych klientów sieci BitTorrent: µTorrent, BitComet oraz Vuze (dawniej Azureus). Wynikiem działania programu jest wygenerowanie raportu uwzględniającego najważniejsze, z punktu widzenia eksperta, informacje takie jak: nazwa pliku, ścieżka zapisu, data dodania, ilość danych wysłanych i pobranych. Kolejny projekt to Konwerter Dźwięku, który służy do konwersji plików dźwiękowych. Dokonywana jest ona z formatu Adaptive Multi-Rate (AMR) do formatu Audio Waveform (WAV). Format AMR jest często spotykany w analizie nagrań pochodzących z dyktafonów cyfrowych i telefonów komórkowych. W procesie konwersji wykorzystywana jest biblioteka Open Source SoX. Prezentowane oprogramowanie działa pod kontrolą systemów z rodziny Windows (XP, Vista, 7).

Słowa kluczowe: *SQLite, Gadu-Gadu, komunikator internetowy, P2P, BitTorrent, klient BitTorrent, torrent, µTorrent, BitComet, Vuze, Azureus, AMR, AWB, WAV, The Adaptive Multi-Rate, 3GPP, SoX, .NET*

Abstract

This article presents three computer forensics problems. The first part is devoted to analysis of SQLite contained database file. SQLite is currently employed by many popular programs such as Skype, Firefox or Gadu Gadu instant messenger. The latter one is used as an example for proposed data retrieving procedure. Other problems include results of research on computer forensics tools conducted in cooperation between Lodz Voivodship Police Forensic Laboratory and Technical University of Lodz. The paper presents two software projects developed to support forensic experts. The first one is "Torrent Analyzer" – an application for analysis of configuration files of the most popular BitTorrent clients: µTorrent, BitComet and Vuze (formerly Azureus). Program generates of analysis report, as well. Report includes such information as: file name, save path, creation time and date, amount of uploaded and downloaded data. Another project, "Konwerter Dźwięku" is an application dedicated to audio files format conversion. "Konwerter Dźwięku" converts Adaptive Multi-Rate files (AMR) to Audio Waveform (WAV) format. Open Source SoX library is employed is the process of conversion. Presented software runs under Windows OS family (XP, Vista, 7).

Keywords: *SQLite, P2P, BitTorrentts, µTorrent, BitComet, Vuze, Azureus, AMR, AWB, WAV, 3GPP, SoX,*

PRZYPISY

[1] SQL (*Structured Query Language)* – strukturalny język zapytań.

[2] RDBMS (*Relational Database Management System*) – system zarządzania relacyjną bazą danych to zestaw programów służących do korzystania z bazy danych opartej na modelu relacyjnym.

[3] Licencja Public Domain oznacza dobra, dla których wygasły już prawa autorskie, bądź takie, które od samego powstania nie były objęte takim prawem. Dzięki czemu „produkt" jest dostępny dla każdego, do wszystkich zastosowań.

[4] *Atomicity, Consistency, Isolation, Durability* – atomowość, spójność, izolacja, trwałość – oznacza to, iż zbiór operacji na bazie danych jest nierozerwalną całością i zostają wykonane wszystkie bądź żadna.

[5] Podany numer GG przykładowy (niezwiązany z prowadzonymi badaniami)

Wyniki przedstawione w niniejszym artykule opierają się na analizie danych niezaszyfrowanych.

[6] Pierwsza popularna sieć wymiany i udostępniania plików; działalność serwisu w latach 1999–2001 zapoczątkowała dyskusję dotyczącą łamania praw własności intelektualnej w sieci Internet.

[7] Dane szacunkowe na luty 2009 r.

[8] **Fonseca J., Reza B., Fjeldsted L.**: *BitTorrent Protocol version 1.0.* Department of Computer Science, University of Copenhagen, April 2005.

[9] W tym formacie zapisu czasu podawana jest liczba sekund, która upłynęła od 1 stycznia 1970 roku godz. 00:00:00 czasu UTC.

[10] 3GPP, 3GPP TS 26.090 – Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions.

[11] IBM Corporation and Microsoft Corporation, Multimedia Programming Interface and Data Specifications 1.0 (Sierpień 1991).

[12] Dodatkowe informacje na stronie: http://www.nch.com.au/switch/index.html.

[13] Jest to format plików dźwiękowych stworzony przez firmy Microsoft i IBM.

[14] Dodatkowe informacje na stronie: http://en.wikipedia.org/wiki/Pulse-code_modulation.

[15] Fraser Stephen R. G., Pro Visual C++/CLI and the .NET 3.5 Platform (2008).

[16] RFC 4867 – RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs.

[17] Dodatkowe informacje na stronie: http://sox.sourceforge.net/.

[18] Dodatkowe informacje na stronie: http://www.3gpp.org/ftp/Specs/html-info/26104.htm.

BIBLIOGRAFIA

1. **Owents M.:** The Definitive Guide to SQLite, Copyright (C) by Michael Owents (2006).

2. **Fonseca J., Reza B., Fjeldsted L.**: BitTorrent Protocol version 1.0. Department of Computer Science, University of Copenhagen, April (2005).

3. **Wai-Sing Loo A.**: Peer-to-Peer Computing, Copyright (C) Alfred Wai-Sing Loo (2007).

4. **Markus Hofmann, Leland R. Beaumont.:** Content networking: architecture, protocols, and practice Copyright (C) by Lucent Technology and Leland R. Beaumont. (2005).

5. **Fraser Stephen R. G.**, Pro Visual C++/CLI and the .NET 3.5 Platform (2008).

6. **Hogenson G.**: Fundations of C++/CLI Copyright (C) Gordon Hogenson (2008).

7. USCERT 2006, http://www.us-cert.gov/reading_room/forensics.pdf, dostępne 2010.

8. Wong LW, Forensics analysis of the windows registry (2007), dostępne online http://www.forensicfocus.com/forensic-analysis-windows-registry, dostępne 2010.

Autorzy pragną podziękować kierownikowi Katedry Informatyki Stosowanej PŁ, prof. dr. hab. inż. Dominikowi Sankowskiemu – Prezesowi Oddziału Łódzkiego PTI, za możliwość podjęcia i nieustanne wsparcie badań dotyczących informatyki śledczej. Specjalne podziękowania kierujemy także do Naczelnika Laboratorium Kryminalistycznego KWP w Łodzi, mł. insp. Ryszarda Ziemeckiego.