UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECUPERAÇÃO DE CONVERSAS DE MENSAGENS INSTANTÂNEAS REALIZADAS EM NAVEGADORES DE INTERNET

RONEI MAIA SALVATORI

ORIENTADOR: ANDERSON CLAYTON ALVES NASCIMENTO

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO**

PUBLICAÇÃO: PPGEE.DM – 104/12

BRASÍLIA / DF: FEVEREIRO/2012

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

RECUPERAÇÃO DE CONVERSAS DE MENSAGENS INSTANTÂNEAS REALIZADAS EM NAVEGADORES DE INTERNET

RONEI MAIA SALVATORI

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:

ANDERSON CLAYTON ALVES NASCIMENTO, Dr., ENE/UNB
(ORIENTADOR)

FLAVIO ELIAS GOMES DE DEUS, Dr., ENE/UNB
(EXAMINADOR INTERNO)

ROBSON DE OLIVEIRA ALBUQUERQUE, Dr., ABIN
(EXAMINADOR EXTERNO)

Brasília/DF, 29 de fevereiro de 2012.

FICHA CATALOGRÁFICA

SALVATORI, RONEI MAIA

Recuperação de Conversas de Mensagens Instantâneas Realizadas em Navegadores de Internet [Distrito Federal] 2012. xiii, 90p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2012).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Vestígios
2. Recuperação de conversas
3. Navegadores de Internet
4. Mensagens Instantâneas
5. *Instant Messaging web-based*

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

SALVATORI, R. M. (2012). Recuperação de Conversas de Mensagens Instantâneas Realizadas em Navegadores de Internet. Dissertação de Mestrado, Publicação PPGEE.DM – 104/12, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 90p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Ronei Maia Salvatori
TÍTULO DA DISSERTAÇÃO: Recuperação de Conversas de Mensagens Instantâneas Realizadas em Navegadores de Internet.
GRAU/ANO: Mestre/2012.

Ronei Maia Salvatori
UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica
70910-900
Brasília – DF - Brasil

A Deus, pela minha vida, saúde, sabedoria
e por mais essa oportunidade de superação e crescimento;

Aos meus pais Eneida e Edmo, presentes em todas as fases de minha vida,
pela educação de uma vida inteira e permanente incentivo aos estudo;

Aos meus irmãos Milena e Yves,
com quem compartilho essa educação recebida;

À minha amada Daniella e à nossa filhinha Isadora,
pela paciência, compreensão e motivação essenciais.

AGRADECIMENTOS

Aos idealizadores deste Mestrado que lutaram por essa iniciativa junto à Universidade de Brasília, concretizando um antigo anseio do Departamento de Polícia Federal, melhorando o nível de capacitação dos Peritos Criminais no Brasil.

Ao colega de trabalho, o Perito Galileu Batista de Sousa, por desenvolver ferramentas na área de conhecimento do presente trabalho no âmbito Departamento de Polícia Federal, encorajando-me a aprofundar os estudos nessa linha de pesquisa.

Ao Prof. Anderson, pela motivação e apoio fundamentais, desde a apresentação do tema até a finalização desta jornada, com serenidade e segurança necessárias à continuidade frente às dificuldades encontradas.

Aos colegas do Setor de Perícias em Informática da Polícia Federal, Perito Bruno Werneck, pelas ideias elucubradas no início dos trabalhos e o Perito Linhares pelas dicas de programação ao final da implementação do protótipo desenvolvido.

À chefia e subordinações da Diretoria-Técnico Científica do Departamento de Polícia Federal, que têm propiciado um excelente clima organizacional de trabalho, necessário ao bom desempenho das atividades policiais, nem sempre harmônicas, mas delicadas na maioria das vezes, por natureza.

Aos colegas de turma, com quem compartilhei os ensinamentos transmitidos pelos mestres em sala de aula, bem como os momentos de alegria e superação.

Àqueles que de forma direta ou indireta me ajudaram para que fosse possível chegar até a conclusão deste Mestrado.

O presente trabalho foi realizado com o apoio do Departamento Polícia Federal – DPF, com recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

RESUMO

RECUPERAÇÃO DE CONVERSAS DE MENSAGENS INSTANTÂNEAS REALIZADAS EM NAVEGADORES DE INTERNET

Autor: RONEI MAIA SALVATORI
Orientador: Anderson Clayton Alves Nascimento
Programa de Pós-graduação em Engenharia Elétrica
Brasília, fevereiro de 2012

A recuperação de vestígios provenientes de programas de mensagens instantâneas a partir de discos rígidos apreendidos é uma tarefa comumente demandada para a Perícia do Departamento de Polícia Federal. No entanto, essa recuperação é limitada devido às restrições das técnicas existentes. Estas são voltadas para extração de artefatos provenientes das versões instaladas dos comunicadores, e não categorizam os artefatos voláteis deixados pelas versões *web* implementadas com tecnologia Ajax, que limita a geração de cache pelos navegadores de Internet.

Como alternativa este trabalho demonstra que é possível recuperar artefatos referentes às listas de contatos e às conversações realizadas em ambiente *web*, a partir de procedimentos que dispensam análise aprofundada do protocolo de comunicação utilizado. As hipóteses são baseadas na identificação de padrões de caracteres armazenados no tráfego de rede, despejos de memória, arquivos de paginação e hibernação para criar um dicionário de palavras-chave, possibilitando buscas automáticas.

O método derivou-se de testes empíricos baseados na simulação de conversas entre usuários para extrações de artefatos com auxílio de um protótipo. Foi aplicado a estudos de caso nas versões *web* de quatro comunicadores instantâneos: *Windows Live Messenger – WLM, Gtalk, Yahoo!Messenger* e *Facebook chat*. Foram realizadas comparações entre os dicionários criados para cada comunicador, bem como extrações de listas de contatos e conversas entre usuários, demonstrando os procedimentos definidos e confirmando as hipóteses do trabalho. Os resultados foram considerados satisfatórios, podendo a técnica ser adotada para análises forenses *post-mortem* de discos rígidos, ainda que a volatilidade dos dados nesse ambiente seja um fator limitante de sua eficácia.

ABSTRACT

INSTANT MESSAGING *WEB-BASED* CHAT EXTRACTION

Author: RONEI MAIA SALVATORI
Supervisor: ANDERSON CLAYTON ALVES NASCIMENTO
Programa de Pós-graduação em Engenharia Elétrica
Brasília, feb/2012

The recovery of traces from instant messaging programs from hard drives seized is a task commonly demanded for the expertise of the Federal Police Department. However, the recovery is limited by the constraints of existing techniques. These are aimed at extraction of artifacts from the installed versions of communicators, not categorize the artifacts left by the volatile *web* versions implemented with Ajax technology, which limits the generation of the Internet browsers cache.

This work demonstrates that it is possible to recover artifacts relating to lists of contacts and talks held in a *web* environment, from procedures that do not require detailed analysis of the communication protocol used. The assumptions are based on identifying character patterns stored in network traffic, memory dumps, hibernation and paging files to create a dictionary of keywords, enabling automatic searches.

The method was derived from empirical tests based on simulation of conversations between users for extraction of artifacts with the help of a prototype. It was applied to case studies in the *web* versions of four IM: *Windows Live Messenger - WLM, Gtalk, Yahoo!Messenger and Facebook chat*. Comparisons between the dictionaries created for each communicator as well as extraction of lists of contacts and conversations between users, showing the procedures defined and confirmed the hypothesis of the work. The results were considered satisfactory, the technique can be adopted for *post-mortem* forensic analysis of hard disks, although the volatility of the data in this environment is a factor limiting its effectiveness.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

AJAX – Asynchronous Javascript and XML

DHTML – Dinamic Hypertex Markup Language

DTD – Document Type Definition

FTK – Forensic Toolkit

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

IM – Instant Messaging

IPL – Inquérito Policial Federal

MSN – *Windows* Messenger

P2P – Peer-to-peer

RFC – Request for Comment

XML – Extensible Markup Language

WLM – *Windows* Live Messenger

WMM – *Windows Mortem* Messenger

1 INTRODUÇÃO

A Polícia Federal têm apreendido um número muito grande de mídias de armazenamento de dados de diversas categorias. Inquéritos Policiais Federais – IPLs demandam para a Perícia Criminal Federal exames desses materiais com quesitos acerca da existência de comunicação entre determinados usuários, no sentido de se recuperar conversações realizadas a fim de que possam ajudar a esclarecer delitos em apuração.

Muitos vestígios de comunicação são encontrados. Dentre eles, destacam-se: conversas realizadas por meio de correio eletrônico (armazenadas em clientes de correio instalados ou em acessíveis via navegadores de Internet), conversas provenientes de salas bate-papo de sítios diversos, e conversas realizadas por meio de comunicadores de mensagens instantâneas (*Instant messaging – IMs*).

No que se refere à recuperação de vestígios de conversas realizadas por meio de *IMs*, dois paradigmas são encontrados com implementações e abrangências diferentes. Segundo Vicente (2011), os comunicadores de mensagens instantâneas podem ser classificados em *IMs program-based*, quando é necessária a instalação de um programa cliente para a conversação dos usuários ou em *IMs web-based* – comunicadores instantâneos baseados em navegadores de Internet, quando simplesmente se pode utilizar um navegador para o envio das mensagens para se comunicar com outras pessoas, não havendo a necessidade de se instalar outro programa específico para este fim. A proposta deste trabalho é demonstrar a recuperação de mensagens instantâneas realizadas nos navegadores a partir de discos rígidos apreendidos.

1.1 APRESENTAÇÃO DO PROBLEMA

Técnicas para recuperação das mensagens trocadas por meio do uso de *IMs progam-based* têm sido desenvolvidas, inclusive por Peritos da Polícia Federal. No entanto, o paradigma *web-based* de comunicações tem se tornado popular e não se constatam, com a mesma eficácia, procedimentos para recuperação de artefatos deixados por esse tipo de comunicação (Husain, 2009). Técnicas existentes são voltadas para a análise de protocolos de rede (Nunes, 2008) e cache dos navegadores. No entanto, *IMs web-based* podem utilizar comunicação segura baseada em *https*, o que, neste caso, impede a utilização de outras técnicas de investigação, como por exemplo, uma interceptação de tráfego de rede

para análise de conteúdo, além de utilizarem tecnologias como *Ajax* (*Asynchronous Javascript and XML)*, limitando a geração de cache de navegação para esse tipo de aplicação, restringindo as fontes de artefatos forense (Eleutério e Eleutério, 2011).

É a partir dessas limitações de se recuperar os vestígios produzidos pelos *IMs web-based* nos discos apreendidos que se encontra a motivação para se realizar este trabalho. Serão abordadas estratégias para se resolver o problema da recuperação de conversas realizadas nesse ambiente, propondo procedimentos para mostrar a possibilidade de se extrair os vestígios gerados, frente às dificuldades encontradas diante das características do uso desse tipo de comunicação.

1.2 OBJETIVO

O objetivo deste trabalho é demonstrar a possibilidade de recuperação de artefatos referentes a listas de contatos e conversas de mensagens instantâneas realizadas em navegadores de Internet (*IM web-based)*, a partir de procedimentos baseados na identificação de palavras-chave durante simulações de troca de mensagens, sem a necessidade de se conhecer o protocolo de comunicação utilizado. Os procedimentos definidos possibilitam buscas em discos rígidos, visando extrair os vestígios por meio de uma análise forense *post-mortem*, como alternativa para superação das limitações das técnicas existentes. Estas são voltadas para recuperação de mensagens provenientes de programas instalados ou desconsideram a limitação de geração de cache dos navegadores por aplicações *web* que utilizam a tecnologia Ajax.

Esse objetivo é divido em dois objetivos específicos:

- Definir um dicionário de palavras-chave a partir da análise dos artefatos forenses referentes à análise de tráfego de rede, despejos de memória, arquivos de paginação e hibernação dos sistemas operacionais *Windows* e *Linux*, em diferentes navegadores;
- Construir um protótipo para extrair conversas automaticamente, categorizando os atributos de comunicação (mensagem, remetente, destinatário, data e usuários das listas de contatos) referentes aos artefatos analisados.

1.3 JUSTIFICATIVA

Aplicações *web* abrem novas possibilidades para o desenvolvimento de *software*, uma vez que agregam a excelente usabilidade das aplicações convencionais *desktop* com o enorme potencial de serem acessadas pela Internet, sem a necessidade de usuários instalarem programas específicos. Dessa forma, o paradigma *web-based* dos *IMs* tem se tornado mais popular, representado desafios para pesquisadores forenses, devido à volatilidade dos dados nesse ambiente (Taivalsaari e Mikkonen, 2008).

Nesse sentido, também surgiu o paradigma denominado *Volatile Instant Messaging – VIM* (Husain, 2009), onde usuários podem trocar mensagens instantâneas a partir de navegadores de Internet, que por vezes não geram cache de navegação para esse tipo de atividade.

Com o crescimento do número de aplicações escritas nessa plataforma frente às limitações encontradas nas técnicas existentes, que são voltadas para a recuperação de mensagens deixadas por programas instalados, surge a necessidade de procedimentos para identificação e recuperação dos artefatos gerados pelos *IMs web-based*. Sanar a deficiência encontrada na recuperação das conversas realizadas por esse tipo de comunicação contribui com mais uma fonte de vestígios explorada. Comunicadores de mensagens instantâneas são imensamente populares, uma popularidade que parece transcender gerações (Carvey, 2007).

Em (Dankner, Kiley e Rogers, 2008) são apresentadas extrações de palavras-chave pré-definidas encontradas no cache do navegador utilizadas em sessões de comunicação instantânea no Internet Explorer 6.0. Trabalho de natureza similar ao aqui proposto, porém, com foco na análise de cache de navegação, não categorizando de forma automática os tipos de fragmentos encontrados, nem separando os atributos de acordo com o conteúdo das buscas realizadas ou demonstrando procedimentos para coleta dos padrões utilizados pelos protocolos.

O trabalho proposto também se justifica pelo volume demandado para análise de discos rígidos pelas áreas de perícias em informática na busca de vestígios de conversas frente a grande popularização no uso de comunicadores instantâneos.

1.3.1 Popularização de programas de Mensagens Instantâneas

A utilização programas de comunicação de mensagens instantâneas tem sido amplamente difundida. Na China, por exemplo, a taxa de utilização de *IMs* na Internet ultrapassa a de correio eletrônico, sendo quarta colocada no ranking geral de classificação de uso de aplicações e primeira na categoria "*comunicação*" (China Internet Network Information Center, 2010). O percentual de usuários de *IMs*, respectivamente, em dezembro de 2009 e junho de 2010 é de 70,9% e 72,4% frente a 56,8% e 56,5% do percentual de usuários de correio eletrônico no mesmo período, como pode ser visto na Figura 1.1:

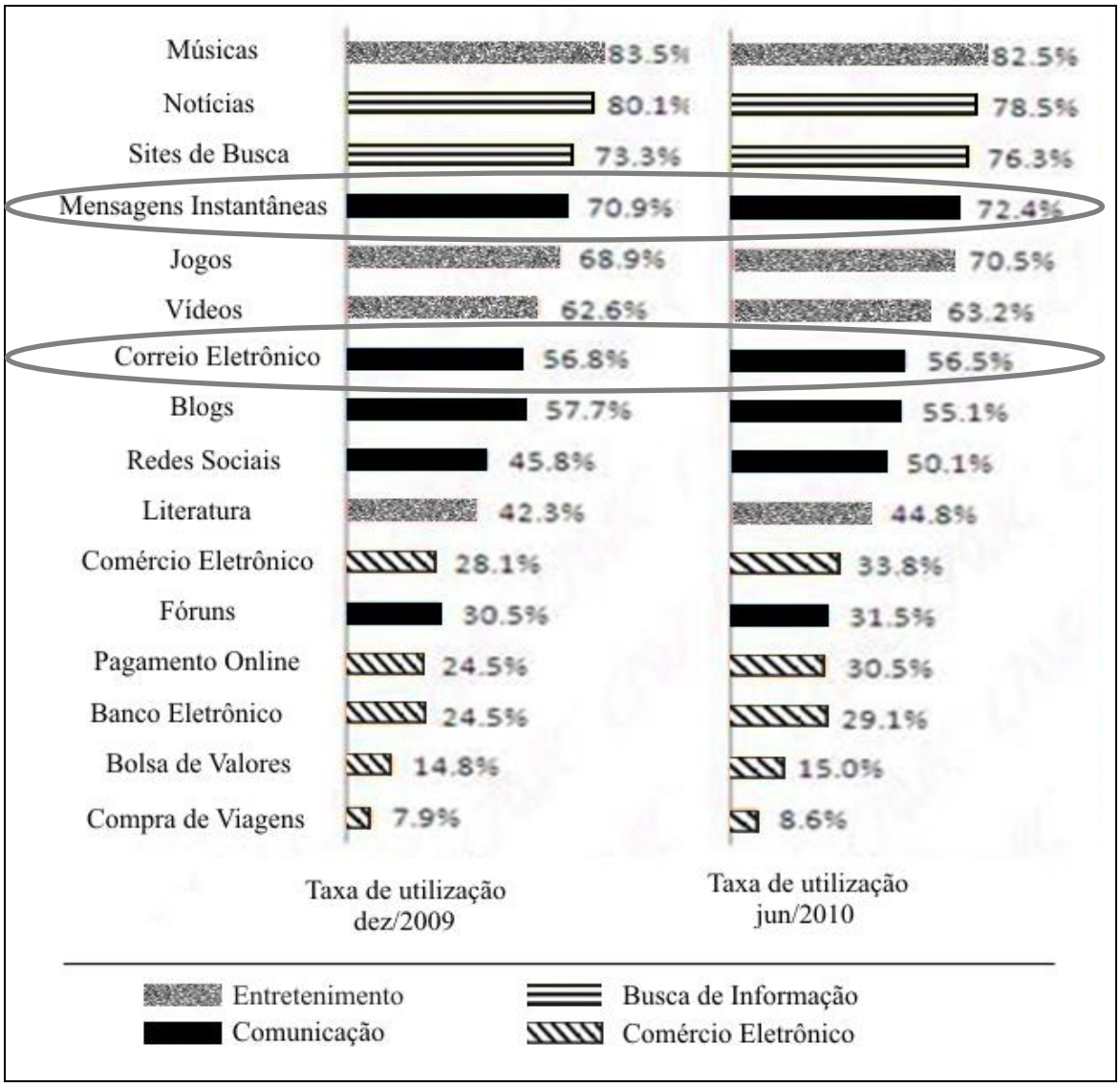

Figura 1.1 – Utilização de aplicações na Internet na China 2009/2010.
Fonte: China Internet Network Information Center (CNNIC). (2010), adaptado.

Segundo Laporte (2008, *apud* Vicente, 2011), não existe um só programa de comunicação instantânea dominante no mundo. A Figura 1.2 demonstra a diversidade dos comunicadores instantâneos em escala global:

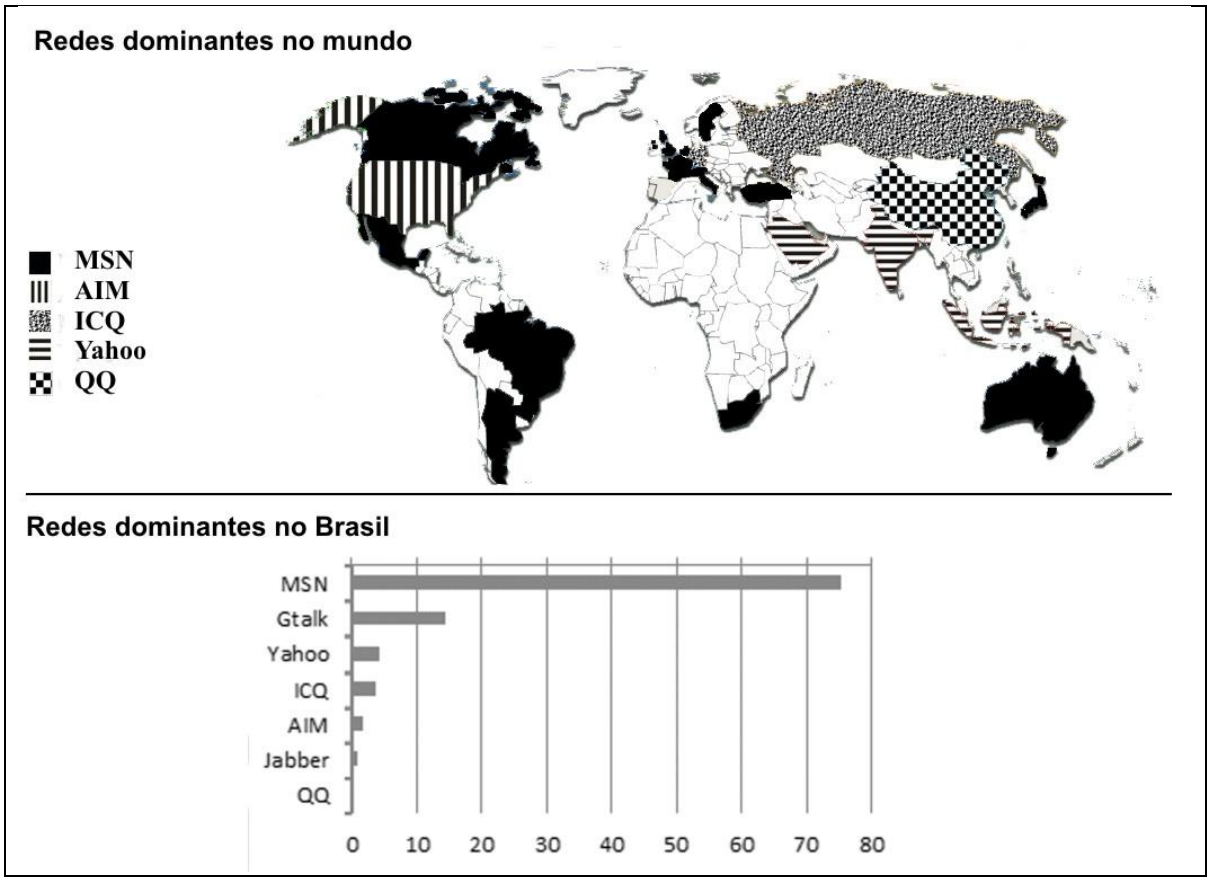

Figura 1.2 – Popularização de *IMs* no Brasil e no mundo (adaptada de Laporte, 2008)

Segundo Pingdom (2010), o número de usuários de *IMs* deve chegar a 1,7 bilhões até 2013. A Figura 1.3 demonstra o surgimento dos primeiros *IMs* e a previsão do aumento do número de usuários:

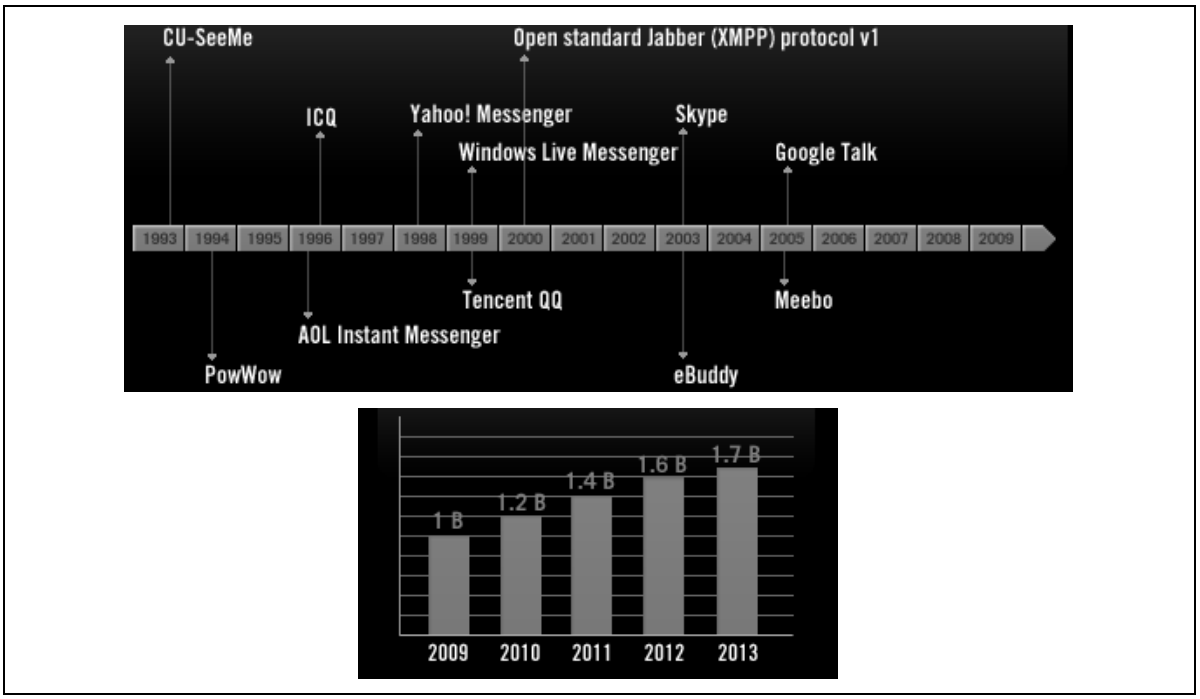

Figura 1.3 – Surgimento dos *IMs* e estimativa de usuários no mundo.
(Adaptada de Amazing Facts and Figures about Instant Messaging Infographic. Pingdom, 2010)

1.4 METODOLOGIA

O trabalho utiliza simulações de conversas entre usuários na Internet para identificação e extração de vestígios voláteis gerados pelos *IMs web-based*. São realizadas análise de artefatos forenses relacionados à memória deixados pelos comunicadores nos discos rígidos, sem a necessidade de um estudo detalhado dos diversos protocolos de comunicação associados a cada *IM*.

Por serem fonte de vestígios disponíveis em discos apreendidos, a memória referente aos arquivos de paginação e hibernação são os artefatos foco das análises, na busca de ocorrências de traços de comunicação correspondentes aos padrões utilizados nas conversas simuladas ou identificados nos protocolos.

A metodologia utilizada para recuperação dos vestígios provenientes de conversas por meio de *IMs web-based* consiste em três abordagens diferentes. São realizadas análises do tráfego de rede de conversas simuladas; análises *post-mortem* dos discos rígidos dos usuários envolvidos nas conversas; e análise *live*, utilizando-se um *proxy* de aplicação entre os navegadores e a memória principal, juntamente com análises de despejos de memória para confronto dos dados.

Os vestígios gerados nas simulações dos laboratórios são utilizados para identificar cadeias de caracteres utilizadas pelos comunicadores. Essas cadeias são identificadas por meio de observação direta dos resultados, procurando-se definir marcadores padrões ou expressões regulares, que poderão ser utilizados como palavras-chave para buscas indexadas nos discos analisados, a fim de se automatizar a recuperação das conversas.

Devido ao fato de os *IMs* terem definidos mecanismos de privacidade e segurança para seus usuários, haja vista os recursos de criptografia serem características cada vez mais presentes na implementação de soluções na Internet, aplicar unicamente a abordagem de se interceptar o tráfego de rede para análises pode não ser eficaz. Por isso, a segunda abordagem, abrangendo o tráfego de dados realizado entre os navegadores e a memória se faz necessária.

As três abordagens propostas resolvem inclusive a dificuldade encontrada na análise do cache do navegador. O uso de tecnologias de geração de páginas dinâmicas, como por exemplo, *Asynchronous Javascript and XML* – Ajax, limita a geração de páginas html em cache de navegação, prejudicando a análise desse artefato forense.

1.5 RESULTADOS ESPERADOS

Os procedimentos procuram demonstrar a possibilidade de se identificar os vestígios gerados pelos comunicadores de mensagens instantâneas baseados em navegadores de Internet, ainda que a volatilidade desses dados seja um aspecto limitante da sua eficácia.

Será possível identificar palavras-chave ou expressões regulares utilizadas por esses programas, a fim de se recuperar conversas de usuários que são acompanhadas desses marcadores, sem a necessidade de se realizar um estudo detalhado de cada protocolo de comunicação existente.

A técnica apresentada permite que sejam construídas ferramentas para automatizar a recuperação desses vestígios.

O conjunto palavras-chave identificado para cada *IM web-based* pode ser utilizado como parâmetros de entrada inclusive para uma análise pericial com a abordagem *live forensics*.

1.6 ORGANIZAÇÃO DO TRABALHO

Os demais capítulos são assim organizados:

No capítulo *2 – Fundamentos Teóricos* e *Revisão Bibliográfica* são apresentados alguns conceitos necessários ao entendimento dos procedimentos adotados nesta pesquisa que estão relacionados ao funcionamento da memória virtual, paginação, hibernação, volatilidade dos dados. São apresentados trabalhos correlatos com o assunto proposto, detalhando as áreas de concentração de outras pesquisas já realizadas, apontando suas limitações frente ao escopo deste trabalho.

No capítulo 3 – *Constatações preliminares* são apresentados testes empíricos que demonstram a possibilidade de se encontrar fragmentos remanescentes em discos rígidos de conversas realizadas em ambiente *web*. Os resultados destas constatações determinam a viabilidade do trabalho e dão origem à metodologia proposta.

No capítulo *4 – Metodologia* Proposta são detalhados os procedimentos e ferramentas utilizadas para se realizar identificação e extração de conversas em ambiente *web*, utilizando os conceitos apresentados.

No capítulo 5 – *Estudos de Caso* é aplicada a metodologia proposta como prova de conceito a estudos de caso realizados em navegadores e sistemas operacionais diferentes. São constatados conteúdos persistentes relativos aos artefatos objeto das análises.

No capítulo *6 – Resultados e Análises* são detalhados os resultados obtidos a partir dos procedimentos propostos, identificando as oportunidades e limitações. Uma lista de palavras-chave é definida a partir dos artefatos examinados nos estudos de casos. São apresentadas as análises das buscas realizadas com o uso de um protótipo desenvolvido para extração de conversas a partir das palavras-chave identificadas, validando a técnica a partir dos procedimentos sugeridos.

No capítulo 7 – *Conclusões* são consolidadas as deduções dos procedimentos adotados segundo a metodologia proposta. São apresentadas as lições aprendidas, ressalvando-se as oportunidades e limitações encontradas, propondo temas para trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS E REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados conceitos relacionados ao desenvolvimento do trabalho, envolvendo memória principal, arquivos de paginação, hibernação de sistemas operacionais e informática forense. Procura-se demonstrar a exploração desses artefatos, utilizando o conhecimento de técnicas e ferramentas, visando à coleta e preservação de possíveis provas.

São discutidos trabalhos correlatos com foco na recuperação de vestígios encontrados por meio de técnicas de análise de memória e tráfego de rede. As análises dessas soluções são importantes para o desenvolvimento do trabalho proposto, pois podem ser adaptadas e aplicadas à recuperação de artefatos relacionados aos comunicadores de mensagens instantâneas em ambiente *web*.

2.1 COMPUTAÇÃO FORENSE

A determinação da dinâmica, materialidade e autoria de ilícitos ligados à área de informática a partir do processamento de evidências digitais por meio de métodos técnico-científicos é o principal objetivo da computação forense (Eleutério e Machado, 2011).

Essa área do conhecimento visa produzir provas a fim de que possam ser apresentadas em juízo, a partir da coleta e análise de vestígios existentes no ambiente cibernético durante e após a prática de crimes (Mandia, Prosise e Pepe, 2003).

Para se produzir provas de crimes cometidos nesse ambiente, é necessário conhecer procedimentos para coleta e processamento dos vestígios encontrados de acordo com a natureza dos mesmos. A natureza do tipo de prova num ambiente computacional é bem peculiar, pois é dinâmica, volátil e de abrangência mundial. No entanto, procedimentos periciais podem provar a materialidade dos fatos, apresentar sua dinâmica e ainda indicar o possível autor (Nogueira e Campello, 2006).

As atividades de um sistema operacional geram muitos vestígios. Para que estes possam ser explorados de forma eficiente, devem ser utilizadas técnicas e ferramentas adequadas. É importante identificar artefatos forenses, de acordo com a relevância para cada caso, a fim de que se possa coletá-los de forma correta para permitir reprodutibilidade em laboratórios, e ser possível apresentar os resultados das provas em juízo.

Para obter informações sensíveis referentes à memória principal, Brezinski e Killalea, (apud Solomon *et al*, 2005) definem na RFC 3227 um padrão de coleta de vestígios, segundo uma ordem de volatilidade estabelecida:

1. Registradores;
2. Cache;
3. Tabela de roteamento;
4. Cache do protocolo ARP;
5. Tabela de processos;
6. Estatísticas realizadas pelo Kernel do sistema operacional;
7. Memória principal;
8. Arquivos temporários de sistema;
9. Disco;
10. Logs remotos e dados monitorados que são relevantes para o sistema em questão;
11. Configuração física;
12. Topologia de rede;
13. Mídias de backup.

No trabalho aqui desenvolvido, por exemplo, o método apresentado para recuperação de vestígios pressupõe a aplicação da ordem da coleta de dados referentes aos itens 7 e 8 elencados, respectivamente, memória principal e arquivos temporários de sistema.

A coleta de vestígios representa apenas uma etapa da análise pericial. Segundo Carroll, Brannon e Song (2008), após consultar peritos em computação forense de agências federais americanas, o Laboratório de Crimes Cibernéticos da *Computer Crime and Intellectual Property Section* do Departamento de Justiça dos Estados Unidos descreveu uma metodologia para padronizar as perícias de informática. Essa metodologia é constituída de 07 (sete) etapas:

- Obtenção e cópia forense dos dados: pressupõe a apreensão necessária de equipamentos e duplicação bit a bit dos dados armazenados para preservação do conteúdo original e permitir a reprodutibilidade dos exames;

- Requisição forense: são realizados os quesitos a que o perito deve responder, de modo a ser possível dimensionar a complexidade dos exames;
- Preparação ou extração, identificação e análise: constituem o exame pericial propriamente dito. Geralmente são executadas em sequência por um mesmo grupo de peritos;
- Relatório forense: fase em que o laudo pericial é emitido. Exames necessitam reproduzir de alguma maneira os vestígios relevantes. Geralmente os arquivos são exportados para uma mídia anexa ao laudo, com os devidos códigos de integridade referenciados em seu corpo;
- Análise (nível de caso): os resultados são analisados perante a investigação que pode demandar análises de novos artefatos, com as devidas buscas e apreensões ou outros dados necessários à nova iteração do processo.

O foco deste trabalho se concentra nas etapas de identificação e análise dos vestígios para os quais é aplicada uma metodologia de análise de memória, visando à recuperação de mensagens instantâneas realizadas em navegadores de Internet.

2.1.1 Análise de Memória

A recuperação de artefatos para investigações de crimes cibernéticos a partir de dispositivos eletrônicos de computador tem sido abordada por diversos autores da área forense. Com relação à análise de memória, as soluções compreendem desde a simples recuperação de arquivos intactos em sua totalidade até a recuperação de fragmentos de informações fossilizadas em espaço não alocado em disco pelo sistema de arquivos, bem como contidas em processos em execução na memória, neste caso, por meio de uma abordagem *live forensics*.

Em muitos casos, a melhor fonte de informações ou evidências está disponível na memória do computador: conexões de rede, conteúdo de texto contido em janelas de programas de mensagens instantâneas e memória usada pelos seus processos, uma vez que alguns clientes não geram logs de conversação (Carvey, 2007).

Segundo Zhao e Cao (2009), a coleta de dados sensíveis a partir da memória principal de computadores não se encontra no mesmo estágio de maturidade, quando comparada aos procedimentos de outras áreas forenses. No entanto, existem métodos que têm por objetivo realizar despejos de memória por meio de artifícios que visam evitar, ou pelo menos

minimizar, as interações com a fonte de dados, a fim de se evitar o *Princípio da Troca de Locard*[1]. Procedimentos diferentes são sugeridos para coleta de dados em memória, cada um com suas vantagens, desvantagens e aplicações a casos:

- Utilização de dispositivos de *hardware* para acessar a memória física por meio de comunicação dedicada. Dessa forma, é possível obter os dados voláteis sem a necessidade de carregar um programa adicional no sistema e introduzir códigos para realizar a extração. Esses dispositivos geralmente são utilizados para a realização de procedimentos de *debug* em sistemas de *hardware*, mas podem ser utilizados para a análise forense. Nesse caso, teriam que ser instalados antes da ocorrência de incidentes. Outra desvantagem na utilização de dispositivos de *hardware* é que não são amplamente encontrados no mercado;
- Cópia total da memória física por meio da utilização da interface do barramento *firewire*, quando disponível, como forma de não interferir no conteúdo da RAM a ser copiada. Esse barramento suporta o acesso direto à memória e o mapeamento dos dados é realizado em nível de *hardware* com alta velocidade e baixa latência, sem a necessidade de execução de programas pelo sistema operacional, assim como acontece como os dispositivos de *hardware*. A vantagem dessa abordagem em relação a anterior é que muitas placas-mãe já são fabricadas com interfaces *firewire*. A desvantagem é que o barramento dessa interface apresenta problemas com a *Upper Memory Area* (UMA) e em alguns casos provoca mau funcionamento de *hardware,* apresentando a exceção de tela azul para sistemas operacionais *Windows*;
- Realização de cópias forenses por meio do utilitário *Data Dumper – DD* existente em sistemas *Linux*. Esse programa tem sido utilizado como um padrão para produzir imagens forenses, além disso, ferramentas forenses também interpretam o formato gerado pelo DD. Para sistemas operacionais *Windows* existem versões modificadas do utilitário. A desvantagem é que a realização do despejo de memória realizado por meio desse procedimento pode levar algum tempo e o sistema operacional vai modificando o estado de memória à medida que o tempo vai passando;

[1] Qualquer um ou qualquer coisa que adentra local de crime leva consigo algo ao chegar e deixa alguma coisa

- Utilização de ferramentas de virtualização. Essas ferramentas permitem criar máquinas virtuais e realizar *snapshots*, que são cópias do estado de execução dos sistemas operacionais virtualizados, permitindo a restauração do sistema para estados anteriores. O *software* para virtualização *Vmware*[2] é capaz, dentre outras operações, de realizar o procedimento de suspensão do sistema e armazenar a cópia da RAM. Toda a atividade do sistema virtualizado é congelada e o procedimento de suspensão é rápido, minimizando as chances de interação entre o investigador e a memória a ser copiada. A desvantagem é que a virtualização não é muito encontrada em sistemas para serem analisados;

- BodySnatcher (Schatz, 2007). Prova de conceito que visava coletar a memória física, tendo como alvo um sistema operacional *Windows* 2000. Consistia em injetar a imagem de um sistema operacional independente em num computador hospedeiro com o kernel modificado. Dessa forma, o controle do sistema injetado é despachado para o sistema hospedeiro. A desvantagem é que o sistema operacional alternativo deve suportar o *hardware* existente do sistema operacional alvo.

Além das técnicas apresentadas, os autores indicam ferramentas para coletar a memória física e arquivos protegidos pelo sistema operacional: Disk Explorer[3], Forensic Toolkit – FTK[4], WinHex/X-Ways Forensics[5] e iLook[6] e sugerem como objeto de estudo:

- Memória RAM;
- Despejos de memória ocasionados por exceções do sistema operacional, *crash dump;*
- Arquivos de paginação;
- Arquivos de hibernação.

[2] Vmware – http://www.vmware.com/
[3] Disk Explorer – http://www.runtime.org/
[4] Forensic Toolkit-FTK – http://www.accessdata.com
[5] WinHex/X-Ways Forensics – http://www.x-ways.net/forensics
[6] iLook – http://www.ilook-forensics.org

2.1.2 Memória Virtual

Um dos artefatos forenses que pode ser explorado em busca de vestígios é a memória virtual. Farmer e Venema (2006) apresentam os princípios do funcionamento de memória, discorrendo que todos os sistemas operacionais modernos usam a memória virtual como uma abstração que combina a memória principal, *Random Access Mémory* – RAM, com o arquivo de paginação, memórias de somente leitura *Read Only Memory* – ROM, a memória RAM não volátil, *Non-Volatile Random Access Memory* – NVRAM e ainda outros tipos de dispositivos. São apresentados os conceitos básicos de páginas de memória, fazendo-se uma associação entre arquivos de usuários abertos na memória e essas páginas.

Uma página de memória é um arquivo com um atributo especial para manipulações excessivas pelo gerenciador de memória virtual. Como esse arquivo também é mapeado na memória principal, isso significa que as alterações de arquivos na memória realizadas pelos usuários permanecerão mapeadas por um determinado tempo, dependendo do quanto sobrecarregado se encontra o sistema operacional.

A possibilidade de recuperação de senhas, que frequentemente são encontradas como texto em claro em arquivos do sistema operacional, como por exemplo, arquivo de paginação e o arquivo de hibernação, além de serem encontrados também na memória RAM é apresentada em (Anson e Bunting, 2007), pág. 246.

> "Finding Clear-Text Password in the swap file
> Would you now be surprised why it is that clear-text-passwords are often found in the swap file and the hiberfil.sys file? The swap file is used to store RAM contents when RAM space is full. Thus RAM data, complete with clear –text-password, is often written to the swap file, resulting in clear-text passwords being written to disk. When the computer is placed in the hibernate mode, the entire contents of RAM are written to the hiberfil.sys. Would that file be yet another source of plain-text passwords? Enough said?"

Essa oportunidade de recuperação contribui para o presente trabalho de forma significante, uma vez que as conversas podem também ser encontradas nesses arquivos, apesar de os autores não terem abordado técnicas específicas para recuperação de dados advindos de áreas de texto de navegadores de Internet.

Marshall (2008) apresenta diversas fontes de artefatos forenses. O autor discorre sobre os locais de evidências, tais como, arquivos disponíveis no próprio sistema de arquivos, arquivos apagados e arquivos de paginação. Ao abordar os arquivos de paginação, o autor

ressalva que o processo de paginação é totalmente automatizado pelo sistema operacional, não cabendo ao usuário interferir em que tipo de informação é ou não paginada.

No entanto, usuários podem definir quanto de memória em disco poderá ser utilizada para a realização da paginação. Pelo fato de os usuários não terem controle sobre o que é gravado nesse arquivo, o autor afirma que é possível encontrar dados que nunca foram escritos em qualquer outro dispositivo de armazenamento. Esse arquivo é frequentemente um ótimo repositório de senhas, dados sensíveis, como arquivos decriptados em memória, por exemplo. O autor não afirma como é possível mapear onde estão armazenados esses dados no referido arquivo. Devido ao fato do arquivo de paginação ser constantemente utilizado, e não haver referências a nenhum metadado para identificação de conteúdo, é difícil determinar quando e onde um determinada informação é gravada.

2.1.2.1 Arquivos de Paginação

O arquivo de paginação consiste de dados anônimos que alguma vez estiveram carregados na memória principal, mas, devido a alguma falta de recursos do sistema operacional, foram salvos em disco. Os arquivos em si, não aparecem no arquivo de paginação. Eles já foram gravados no sistema de arquivos e não existe razão para serem salvos novamente (Farmer e Venema, 2006).

O arquivo de paginação é importante para a investigação forense por armazenar um volume de informações que muitas vezes não estão disponíveis nos arquivos de usuário (Schweitzer, 2003).

Computadores têm cada vez mais capacidade de memória RAM, devido a maior utilização de recursos demandados pelos *softwares* modernos, o que consequentemente leva a existência de arquivos de paginação maiores. Essa característica tem aumentado as chances de se explorar arquivos de paginação com sucesso, uma vez que as informações armazenadas temporariamente nesses arquivos têm chances menores de serem sobrescritas.

Identificar o conteúdo que possa ser de interesse para uma investigação, a partir da análise de dados remanescentes no arquivo de paginação não é uma tarefa fácil. Existem fragmentos armazenados em formato binário e simplesmente vascular o arquivo com editores hexadecimais é tedioso frente ao enorme volume de dados.

2.1.3 Hibernação

A hibernação é um artifício que os sistemas operacionais modernos utilizam para desligar o computador com a possibilidade de preservação do seu estado de funcionamento, mantendo os dados dos programas em execução inalterados, para que após o reinício do sistema, o estado de funcionamento anterior seja restaurado. Esse artifício permite que informações residentes na memória do computador, processos e seus estados, e dados de usuários manipulados pelos programas em execução sejam armazenados em disco, o que implica um despejo de memória automático gerado pelo próprio sistema operacional (Anson e Bunting, 2007).

Sistemas operacionais *Windows* realizam a hibernação por meio de um arquivo denominado *hiberfil.sys*, residente no diretório raiz do sistema de arquivos, que contém o estado do sistema no momento de seu desligamento. Esse arquivo tem formato compactado não documentado pela Microsoft, no entanto, informações podem ser recuperadas mesmo a partir do seu formato compactado. (Lee, *et al*, 2009)

Utilitários para análises de artefatos referentes à memória podem ser ferramentas eficazes para analisar o arquivo de hibernação do *Windows*. O utilitário denominado *Volatility* [7] é um *framework* de código aberto escrito em *Python* que capaz de manipular o arquivo de hibernação para ser interpretado como um despejo de memória. A conversão pode ser realizada por meio do *plug-in imagecopy* utilizado com parâmetros para especificar o tipo de arquivo de hibernação, com relação à versão do *Windows*, que se pretende converter. A Tabela 2.1 mostra um exemplo de utilização do comando para realizar a conversão:

Tabela 2.1 – Conversão do arquivo hiberfil.sys (adaptado de volatitity, 2011)

c:\volatility\volatility.exe imagecopy -f c:*hiberfil.sys* --profile=Win7SP0x86 -O *arquivoDeDestino*

No caso de sistemas operacionais *Linux*, a hibernação é realizada por meio da partição de *swap*. Algumas distribuições denominam esse recurso *suspend to disk*. No entanto, além da compactação dos dados referentes à memória, criptografia também pode ser utilizada, protegendo os dados e o estado hibernado persistido no disco.

Existe na distribuição OpenSuse *Linux* uma característica que permite a hibernação ocorrer sem criptografia ou compactação. Essa funcionalidade é útil para realização de *debugs* e para o trabalho proposto, pode ser utilizada para verificação do formato do conteúdo das

7 *Volatility* - https://www.volatilesystems.com/default/volatility

mensagens trocadas em simulações e compará-las com as mensagens contidas nos arquivos de hibernação do *Windows*. Para isso, os parâmetros "*compress=n*" e "*encrypt=n*" devem ser configurados no arquivo */etc/suspend.conf* e para desligar o computador deve ser invocado o comando *pm-hibernate* (Opensuse, 2011).

Dessa forma, é possível verificar o conteúdo da memória hibernada, tanto a partir de sistemas *Windows* como a partir de sistemas *Linux*, como um ponto em comum, sem necessidade de procedimentos adicionais diferenciados.

2.1.4 Volatilidade

Informações contidas em dispositivos eletrônicos não voláteis, como por exemplo, discos rígidos e *Pen Drives*, podem ser recuperadas após o desligamento do sistema. Arquivos de logs, códigos maliciosos, cache de navegação e arquivos apagados são exemplo de informações armazenadas em dispositivos não voláteis. Criminosos com maior expertise podem destruir essas informações persistentes, no entanto, um bom investigador pode recuperar o estado de funcionamento de um sistema para que possa simular seu funcionamento quando o incidente ocorreu (Maclean, 2006).

Devido ao fato de os sistemas operacionais salvarem pedaços de informações de arquivos de usuários com certa frequência, a depender do quanto são demandados os recursos existentes, os dados escritos no arquivo de paginação permanecem preservados de forma fossilizada. Porém, não há como afirmar que o dado se deteriora devagar ou regularmente. Essa afirmação também é verdadeira para a persistência de pedaços de informação anônima na memória principal (Farmer e Venema, 2006).

Os dados gravados nos arquivos de paginação ainda podem ser referenciados novamente na memória principal. Para isso, basta que uma nova requisição faça referência a uma determinada página não carregada na memória. De acordo com o algoritmo de paginação, o sistema operacional decidirá qual das páginas residentes na memória principal dará lugar à nova página referenciada que estava armazenada no disco. O seu tempo de permanência na memória principal dependerá do quanto essa página ainda é utilizada pelos programas em execução e do tipo de algoritmo utilizado pelo sistema operacional para realizar o processo de paginação. Segundo Schweitzer (2003), o *Windows*, por exemplo, move a página mais tempo não referenciada para o arquivo de paginação.

Isso significa que uma página pode permanecer por muito tempo armazenada, tanto no arquivo de paginação, quanto na memória principal, enquanto outras podem durar minutos ou segundos.

Farmer e Venema (2006) realizaram um experimento com um servidor de Internet para mostrar a frequência do número de páginas alteradas em um determinado período. A análise dinâmica da persistência em memória levou em consideração um servidor com 01 (um) Gigabyte de memória moderadamente ocupado. Ao longo do período de observação de duas semanas e meia, o servidor manipulou por dia cerca de 65.000 requisições de Internet, incluindo requisições de páginas de sítios e verificações de correio. Verificou-se em um determinado momento que 40 a 45 por cento da memória principal era consumida pelo *Kernel* do sistema operacional e processos que estavam executando. O restante correspondia à memória livre dedicada ao cache de arquivos. As medições foram realizadas a cada hora.

Observou-se que algumas páginas mudaram muito mais vezes entre as medições realizadas. No entanto, uma página mudou 76 vezes durante o período de testes. Cerca de 2.350 páginas de memória, de um total de 256.000, não mudaram ou mudaram e retornaram à memória principal e 1.400 mudaram a cada leitura. A Figura 2.3 ilustra as variações do número de páginas alteradas durante o período de medições.

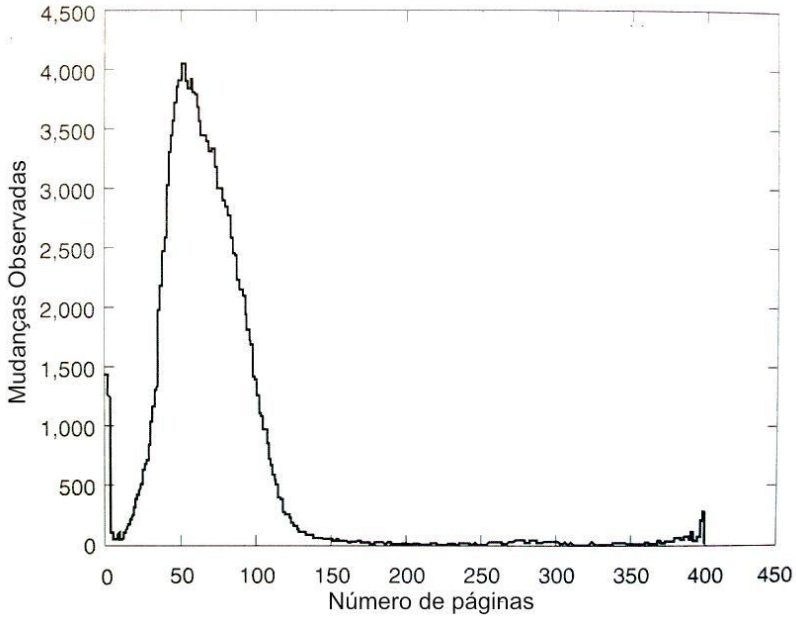

Figura 2.1 – Contagem de mudanças de páginas de memória.
Período de análise: duas semanas e meia, 402 horas (adaptada de Farmer e Venema, 2006).

Um segundo experimento demonstra a duração de dados não referenciados mais por processos na memória principal. Dependendo da atividade do sistema, os dados podem permanecer intactos por algum período de tempo. O experimento foi realizado utilizando-se dois servidores moderadamente ocupados executando FreeBSD 4.1 e *Linux* Red Hat. Um programa foi escrito para realizar a medição da deterioração de 01(um) Megabyte de dados alocado a um processo que finalizou a sua execução, repetindo o experimento muitas vezes e computando-se média dos valores obtidos.

Em todas as medições, em aproximadamente 10 (dez) minutos, 90% (noventa por cento) da memória monitorada havia mudado. Foi observado também que apesar de se tratarem de dois sistemas operacionais distintos, a degradação de memória ao longo do tempo é semelhante.

Sob condições normais de uso, sem stress no equipamento, foi observado uma rápida e inevitável degradação das páginas anônimas de memória, isto é, das páginas que estavam alocadas ao processo de 01 Megabyte. Essa volatilidade depende fortemente da configuração utilizada, no entanto, quando um computador não está fazendo nada, páginas não referenciadas podem persistir por um período mais longo de tempo. Por exemplo, em alguns computadores, senhas e outros dados pré-calculados foram facilmente recuperados dias após serem digitados ou carregados em memória. Em outros casos, embora não estivessem demandando nada, os mesmo dados foram perdidos em minutos ou horas. A Figura 2.2 ilustra a deterioração da memória no experimento realizado:

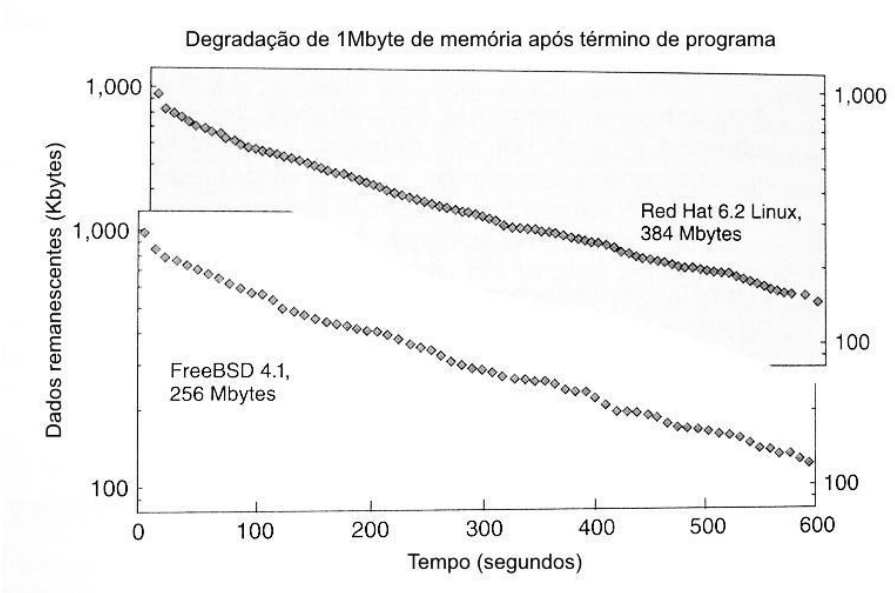

Figura 2.2 – Degradação de páginas não referenciadas em memória.
Período de análise: duas semanas e meia, 402 horas (adaptada de Farmer e Venema, 2006).

2.2 MENSAGENS INSTANTÂNEAS - *IMS*

A comunicação por mensagens instantâneas é baseada na troca de mensagens de texto entre dois ou mais usuários de forma *on-line*. Com a evolução das aplicações, recursos como, por exemplo, troca de arquivos, conversações por áudio e videoconferências estão cada vez mais presentes nas suas implementações. Estruturas de servidores específicos para esse serviço estão espalhadas pela Internet para atender aos requisitos dos protocolos de *IMs* definidos pelas suas RFCs.

As comunicações por meio de mensagens instantâneas estão dentre as atividades de Internet de interesse da informática forense (Marshall, 2008). *IMs* são citados como exemplos de programas que podem operar como de redes *peer-to-peer* (P2P) ou cliente-servidor. Segundo o pesquisador a análise tem enfoque no rastreio de usuários na rede por meio de identificação de endereços IP e na recuperação de logs gerados por essas ferramentas. Não há referência, no entanto, à recuperação específica das mensagens a partir das versões *web* desses comunicadores. Essa limitação pode ser parcialmente sanada com a adoção de novos procedimentos, cuja técnica a ser desenvolvida procure suprir a falta de identificação de metadados do arquivo de paginação, no que se refere à identificação de cadeias de caracteres que podem ser utilizadas como palavras-chave para recuperação de conversas proveniente dos *IMs web-based*.

Com a diversidade e popularização dos comunicadores existentes, interfaces que integram mais de um cliente de comunicação instantânea têm surgido, permitindo aos usuários combinar outras contas de *IM* numa única aplicação (Carvey, 2007). Assim os usuários podem realizar *login* e acessar múltiplas redes de *IM*. Meebo[8] e ebbudy[9] fornecem um serviço similar para versões *web* dos comunicadores. Redes sociais como *Facebook* [10] têm desenvolvido interfaces de comunicação, permitindo aos seus membros a realização de conversas por meio da troca de mensagens instantâneas a partir da própria rede, sem a necessidade de se instalar um programa específico para este fim. A comunicação ocorre entre os membros participantes do bate-papo e é possível adicionar contatos de outros clientes de *IM*, como por exemplo, usuários do *Windows Live Messenger – WLM* [11]e *Yahoo!Messenger*[12].

[8] https://www.meebo.com
[9] http://www.ebuddy.com
[10] http://www.facebook.com
[11] http://www.live.com
[12] http://www.yahoo.com

A possibilidade do envio de mensagens instantâneas por meio das redes sociais expande as chances de se encontrar vestígios provenientes de conversas de bate-papo com as mesmas características voláteis presentes na implementação *web* deste tipo de comunicação.

De acordo com Steel (2006) a análise forense de artefatos referentes aos *IMs* se dá em três etapas:

- Identificar perfis de usuários;
- Identificar lista de contatos;
- Visualizar logs de conversações.

Essa abordagem também pode ser aplicada na análise de *IMs web-based,* porém com diferenças nos procedimentos de localização e recuperação desses vestígios. Por exemplo, nas simulações realizadas para geração e coleta de padrões deixados no disco por meio do uso de *IMs web-based*, quando o usuário opta por não manter o histórico de conversação, não é gerado nenhum arquivo em disco. Outro exemplo são as mensagens com o conteúdo do histórico das conversações que são requisitadas por meio do navegador de Internet para o servidor, onde estão armazenados os históricos das mensagens. Alternativamente, o histórico pode ser encontrado por meio da análise de cache de navegador, uma vez, que nesse caso, há requisição de página *web* com o histórico da conversa.

2.2.1 Recuperação de Mensagens Instantâneas

A extração de vestígios deixados pelos programas de conversação de mensagens instantâneas é abordada por autores da área forense, inclusive por Peritos Criminais Federais do Departamento de Polícia Federal. Estudos têm sido desenvolvidos para recuperação desses vestígios, no entanto voltados para a versão *program-based* dos comunicadores.

A utilização do *Windows* Live Messenger – WLM versão 8 deixa vestígios no disco rígido referentes às configurações de usuário para o programa, lista de contatos, registros de conversas e entradas nas chaves de registro do *Windows*, dentre outros (Dogen, 2007).

Souza (2008) apresenta a ferramenta *Windows* Mortem Messenger – WMM para extração de artefatos do WLM 8. São exibidos artefatos forenses de interesse que a ferramenta é capaz de recuperar, tais como as conversas e as listas de contatos, além de outras informações, como imagens dos contatos e seus grupos. São explicados em detalhes a

localização do armazenamento das mensagens em disco e a decriptografia dos dados gerenciados pelo comunicador. Trata-se de uma ferramenta de linha de comando, porém uma interface visual já estava sendo desenvolvida a época de publicação de seu trabalho. É sugerida uma reescrita do seu código como parte de uma ferramenta genérica forense, o que daria um caráter versátil favorecendo sua reutilização.

Posteriormente no trabalho de Medeiros e Sousa (2009), outra versão da ferramenta WMM é apresentada para extrair os artefatos provenientes do *Windows* Live Messenger em sua nova versão – WLM 2009. Nessa versão, a maneira que os contatos são armazenados no disco difere bastante da versão anterior. Os autores propuseram outra forma de se recuperar as informações baseada na estrutura de armazenamento dos dados na nova versão. Não há mais cifragem de dados e, seguindo a tendência de outras ferramentas da *Microsoft*, os dados estão armazenados em estrutura de banco de dados denominada *ESENT* (*Extensible Storage Engine*), também conhecida como *Jet Blue*, que é uma implementação de base de dados nativa do *Windows* e permite que aplicações armazenem e recuperem dados. É a partir da engenharia reversa dessa base de dados que os autores definem os procedimentos consolidados na nova ferramenta para leitura dos dados armazenados.

As contribuições trazidas por esses trabalhos podem ser utilizadas como parâmetros para nortear a recuperação dos tipos de vestígios equivalentes na versão *web* do comunicador WLM, tais como lista de usuários e histórico de comunicações.

Jerônimo (2011) realizou uma comparação entre a versão 2008 do WMM e um script para a ferramenta EnCase[13], denominado *MSN_Extractor*, desenvolvido pelo Perito Criminal Federal do Setor Técnico científico da Polícia Federal na Bahia, Rogério Dourado. Ambas as ferramentas analisadas são voltadas para recuperação de artefatos deixados pela versão *program-based* do WLM. O *script* executa a recuperação dos fragmentos contendo mensagens em arquivos perdidos a partir de buscas por palavra-chave. Para isso cadeias de caracteres são usadas como assinaturas de mensagens de saída e entrada, representadas como expressões regulares assim codificadas, respectivamente:

- MSG #+ [A-Z] #+[^\x0D\x0A]+
- MSG [a-z#~_\.!\#$%\^&*\(\)\-]+@[a-z#_\-]+\.[a-z#_\-\.]{2,3} [^\x0D\x0A]

[13] EnCase – http://www.guidancesoftware.com/forensic.htm

O trabalho não aborda como foram definidas as palavras-chave, nem como foram colhidos os padrões para realizar a identificação das mesmas, como é feito em Parsonage (2008), que, no entanto, também restringe o estudo somente ao *Windows* Live Messenger – WLM e ao extinto *Windows* Messenger – MSN. A comparação entre as ferramentas WMM e o *MSN_Extractor* não recomenda a utilização de uma específica em detrimento de outra, uma vez que seus níveis de eficácia se alternavam conforme a massa de dados testada. Foram realizadas comparações entre extrações de seis casos reais de análise de discos rígidos apreendidos. A Figura 2.3 ilustra o total de mensagens recuperadas por ambas as ferramentas:

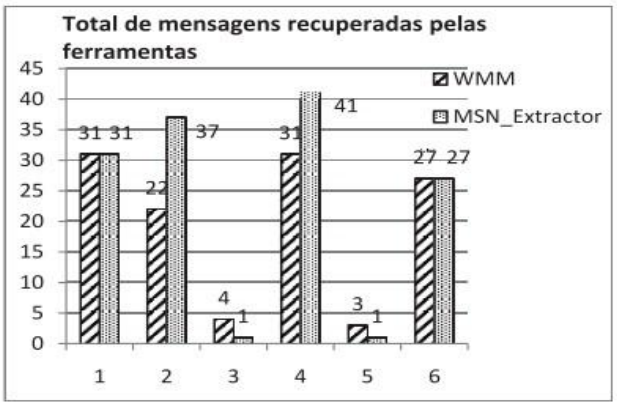

Figura 2.3 - Comparação do total de mensagens recuperadas. *(*Jerônimo, 2011)

Segundo Sousa (2011), analisar vestígios deixados por uma ferramenta é sempre uma atividade suscetível a erros ou incompletudes. No entanto, o pesquisador aborda a recuperação de vestígios deixados por praticamente todas as operações suportadas pelo Skype 5.x. Foram identificadas tabelas de banco de dados e uma visão geral dos seus relacionamentos é apresentada, ressaltando que a concentração dos vestígios num mesmo banco facilita bastante a extração realizada com a utilização da ferramenta *SQLiteExplorer*[14] versão 3.04 e a DLL *SQLite*[15] versão *3.7.7.1*.

O surgimento de ferramentas específicas para extração de vestígios deixados por comunicadores diferentes trás a necessidade de consolidar as informações obtidas das várias fontes de dados. Visando suprir essa necessidade, Vicente (2011) propôs uma arquitetura de referência para agregar informações provenientes de diversos programas extratores de vestígios de *IMs* de forma extensível. Assim, novos extratores podem ter seus

[14] SQLiteExplorer - www.hackerfactor.com/blog/ index.php?/archives/231-Skype-Logs.htm
[15] SQLite - http://www.sqlite.org/download.html

dados agregados a essa plataforma, desde que implementem as interfaces previstas pela arquitetura. Essa abordagem de se construir um *framework* poderia ser levada em consideração para o trabalho proposto, sendo flexível e agregado à interface.

Segundo Nunes (2008), é possível realizar uma classificação dos protocolos de comunicação dos *IMs*. Em seu trabalho são apresentados diversos tipos de protocolo associados a comunicadores. O autor propõe um método científico para servir de base para análise desses protocolos. Seu trabalho cita duas fontes de coleta de informações: uma com a abordagem de recuperação de dados local, com acesso a arquivos de logs dos comunicadores e outra com o enfoque sobre o tráfego de rede, apresentando como resultado, para essa última, a análise da ferramenta chamada *MSN Shadow*, baseada na análise de fluxos TCP/IP.

A desvantagem da primeira abordagem sugerida é que são poucos os *IMs* que criam arquivos de logs por padrão e, dessa forma, as mensagens de troca de status entre os clientes e os servidores não são gravadas. Não é demonstrada a recuperação de artefatos a partir do próprio disco rígido onde executam os comunicadores. Seu trabalho se aprofunda na segunda abordagem, baseada num sistema de captura de rede que necessita de uma infraestrutura de interceptação para análise do protocolo em que está focada a sua pesquisa, o *Microsoft Notification Protocol*. Sua técnica poderia ser aperfeiçoada para ser estendida a outros protocolos, como por exemplo, o utilizado pelo *Yahoo!Messenger*. No entanto, o número de usuários em atividade em relação ao WLM é menor (Fei, *et al*, 2009). São sugeridos outros tipos de pesquisa para novos protocolos analisados pelo perito. Porém, não demonstra como identificar quais parâmetros são utilizados, nem como podem ser obtidos.

Como trabalhos futuros, sua técnica poderia ser adaptada para análises em nível de host, uma vez que mecanismos de criptografia inviabilizam suas análises em nível de rede. Apesar de não propor nenhum método para identificar os dados trafegados dessa forma, essa é uma importante contribuição, pois poderia ser estendida para tentar identificar rastreios dos protocolos utilizados por outros comunicadores e que utilizem também o mecanismo de criptografia por padrão, assim como acontece com *IMs web-based.*

Com relação à análise de memória de hosts em busca de traços do protocolo, uma abordagem diferente é apresentada no trabalho de Gao e Cao (2010). O estudo tem como objeto o comunicador Tecent QQ, o *IM* mais popular na China. Diferentemente do MSN Messenger, que permite a troca de mensagens instantâneas em texto em claro, o QQ realiza

encriptação das mensagens trocadas pela rede e de todos os arquivos armazenados em disco provenientes de transmissões, não sendo possível a leitura dos mesmos a partir de discos rígidos sem a validação de um usuário pelo programa.

A abordagem adotada consiste em analisar a memoria física do computador e reconstruir o espaço de memória alocado pelo processo executor do *IM*. São feitas ressalvas de outros locais de onde podem ser recuperados vestígios, com considerações das limitações encontradas para se realizar a extração. São indicadas três fontes de artefatos forenses que armazenam os vestígios deixados pelo comunicador:

- O disco rígido, onde o QQ cria diretórios com o número de identificação de cada usuário. Os registros de conversação são localizados nesse mesmo diretório em um arquivo denominado "Msg.Ex.db". Esse arquivo é um banco de dados implementado pelo próprio comunicador que não reorganiza os registros armazenados quando conversas são apagadas. Ainda que o usuário apague todas as conversas pelo painel de controle da interface da aplicação, as conversas podem ser recuperadas, uma vez que permanecem armazenadas na base de dados, mesmo sem serem referenciadas. No entanto, essa base é encriptada, o que torna impossível a leitura dos registros sem o conhecimento da forma de decriptografia;
- O protocolo de comunicação utilizado para a troca de mensagens, *Open ICQ*. Os dados também trafegam de forma encriptada e os registros de conversas são codificados utilizando-se um algoritmo denominado *Tiny Encryption Algorithm – TEA*;
- A memória física do computador onde executa o *IM*. Essa é a fonte apontada como a mais eficaz para a aplicação dos procedimentos de recuperação de registros de conversas. Todas as mensagens enviadas e recebidas estão armazenadas em texto em claro, portanto, acessíveis ao investigador sem a necessidade de procedimentos adicionais para contornar o artifício de criptográfica encontrado nas fontes anteriores. Ao reconstruir a memória alocada ao processo utilizado, é possível recuperar os registros das conversas remanescentes.

Durante os experimentos para análise de memória, foram encontradas referências ao nome da conta e às identificações de lista de usuários, data do envio das mensagens e conteúdo das mesmas.

A reconstrução do espaço de memória onde estava alocado o processo responsável pela execução do programa QQ permitiu reunir a maioria das informações sensíveis necessárias às análises. Dessa forma, os examinadores não precisariam realizar pesquisas em larga escala por toda a memória do computador, na busca de informações em particular de conversas realizadas.

Os trabalhos abordam técnicas diferentes para se recuperar e agrupar artefatos gerados pelos *IMs program-based.* Os conceitos apresentados e a forma como são organizadas as informações contribuem para o desenvolvimento do presente trabalho, que se baseia em outra abordagem: identificar vestígios deixados pelo paradigma *web-based* dos comunicadores e realizar suas extrações, sem a necessidade de conhecer todos os protocolos de comunicação.

2.3 ASYNCHRONOUS JAVASCRIPT AND XML – AJAX

Com a evolução dos navegadores, aplicações na Internet utilizam recursos de programação para geração de HTML dinâmico. É possível escrever aplicações que fornecem maior interatividade e que não são restritas para executarem apenas por um navegador específico.

Ajax não é uma nova linguagem de programação. Representa uma mudança de paradigma na forma como páginas *web* são interpretadas pelos navegadores. A possibilidade de se realizar requisições *http*, contendo códigos *Javascript* carregados dinamicamente, sem a necessidade de recarregar todo o código HTML da página atual visitada, representa um menor tráfego de rede com maior interatividade entre os usuários e as aplicações *web* (Garrett, 2005).

No modelo tradicional dinâmico de aplicação *web*, os usuários fazem requisições para servidores que realizam processamentos, acessam sistemas legados e retornam outras páginas HTML para serem carregadas pelos navegadores clientes. Artifícios como a utilização de *iframes*, camadas e demais recursos suportados pelas versões mais recentes de HTML são utilizados para manter dados disponíveis para os usuários, enquanto eventos de mouse ocultam e mostram dados já interpretados no carregamento inicial da página.

Na abordagem Ajax, as aplicações pressupõem a existência de uma *engine* escrita em *javascript* que executa geralmente num frame oculto para intermediar as requisições *web* e carregar os dados. O navegador carrega a *engine* no início da sessão antes da página requisitada.

A *engine* fornece o código proveniente das respostas das requisições *http* realizadas para ser interpretado pelo navegador, ao mesmo tempo em que realiza a comunicação com o servidor *web*, permitindo a interação do usuário com a aplicação acontecer de forma assíncrona, independente da comunicação com o servidor (Garrett, 2005). A Figura 2.4 ilustra diferenças entre o comportamento de requisições *web* tradicionais e as que utilizam tecnologia Ajax.

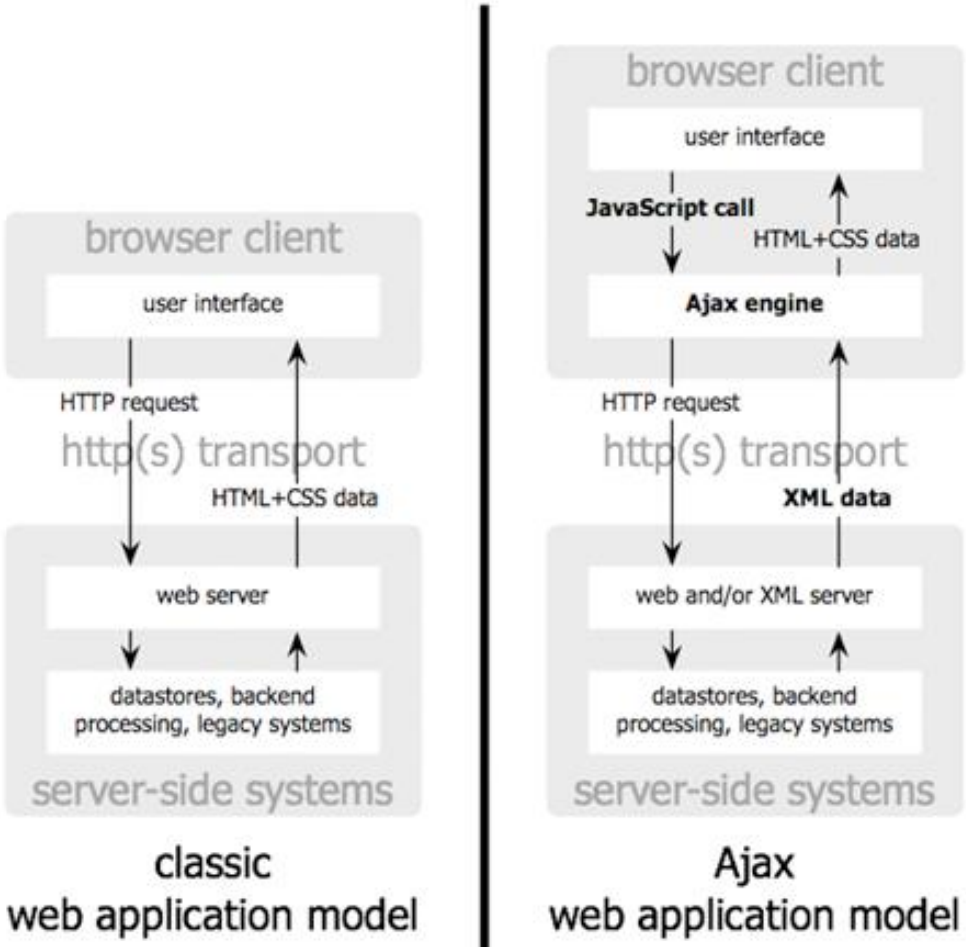

Figura 2.4 – Diferença entre requisições *web* tradicionais e Ajax. (Garrett, 2005)

O navegador hospeda uma aplicação baseada em *scripts*, cujos dados são atualizados por meio de requisições baseadas em *XMLHttpRequest*, uma API utilizada para enviar requisições *web* por meio de *scripts* e carregar a resposta de volta para o *script* que origina a requisição.

Cada evento gerado pelo usuário na página *web* corresponde a uma chamada *javascript* para a *engine* Ajax e não a uma submissão de um formulário HTML, que requer novo

carregamento da página inteira. Se a *engine* necessitar de dados remotos armazenados no servidor ou precisar carregar códigos complementares da interface de usuário, novas requisições serão realizadas de forma assíncrona por meio do uso de XML, sem a necessidade de interromper a interação do usuário com a aplicação. A Figura 2.5 ilustra a diferença de processamento interativo síncrono e assíncrono da interface cliente com as submissões e processamento de dados.

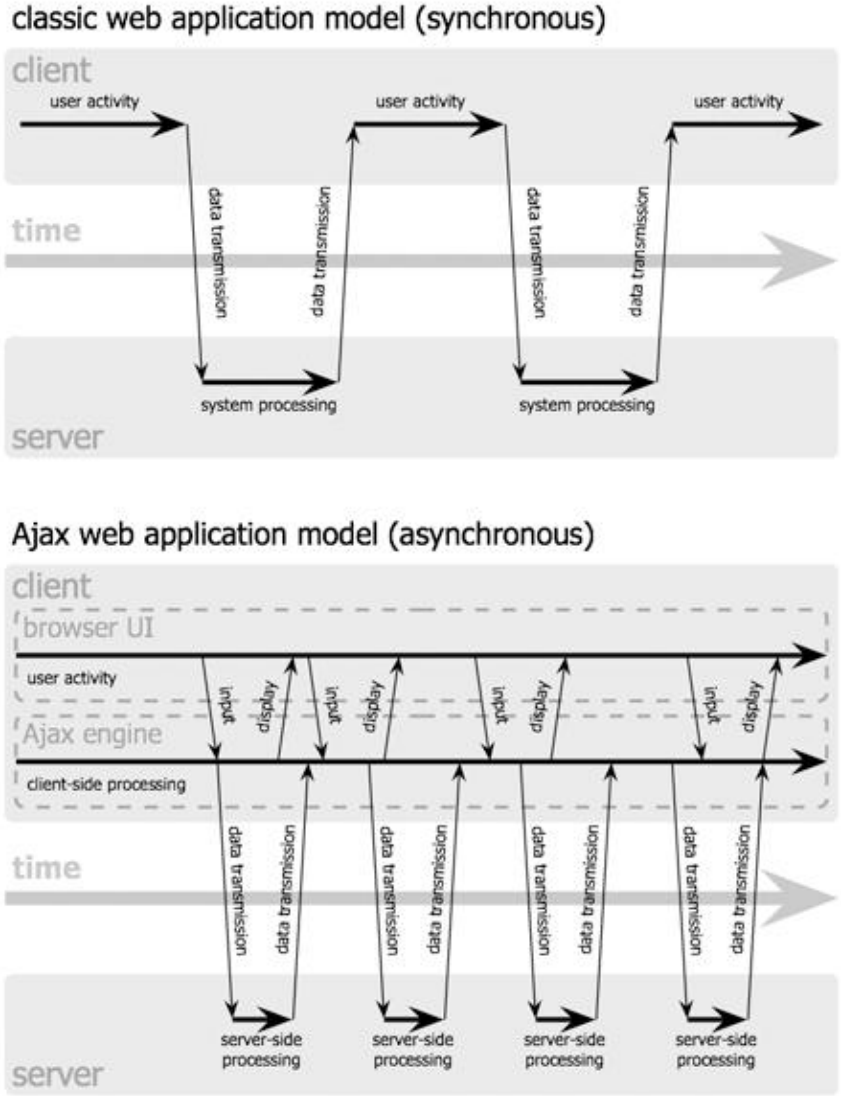

Figura 2.5 – Comparação entre requisições síncronas e assíncronas. (Garrett, 2005)

Uma das implicações do uso de Ajax é a limitação do cache dos navegadores. Como não há uma nova página sendo carregada, o navegador não atualiza os dados dinâmicos em um novo HTML. Isso afeta o armazenamento local de dados de aplicações *web*, como por exemplo, *webmails* (Eleutério e Eleutério, 2011).

2.3.1 JavaScript object notation – JSON

As requisições Ajax em *XMLHttpRequest* podem ser realizadas utilizando outra formatação de objetos além de XML. Nas aplicações *web* para troca de dados encapsulados em protocolos de comunicação, a notação para objetos *Javascripts – Javascript object notation – JSON* tem sido adotada em detrimento ao XML por uma série de vantagens (Crockford, 2011):

- Fornece objetos em formato nativo *Javascript*. Essa vantagem elimina a necessidade de realização de conversores de formato de texto para objeto;
- Os objetos em formato de texto são menores que os codificados em notação XML;
- Os objetos são expressos literalmente como uma coleção de pares de chaves com respectivos valores não ordenados e podem conter vetores com listas ordenadas de valores.

São exemplos de aplicações *web* que utilizam Ajax com dados encapsulados em objetos *JSON*: *Google Suggest*[16], *Gmail*[17], versão *web* do *Windows Live Messenger – WLM*[18], *Facebook Chat*[19], dentre outros.

Neste capítulo foram discutidos os principais conceitos e trabalhos correlatos para o entendimento da metodologia proposta. No próximo capítulo, serão apresentadas constatações preliminares do conteúdo de conversas de mensagens instantâneas realizadas em ambiente *web* remanescentes na memória e no disco rígido de computadores.

[16] http://www.google.com
[17] http://www.gmail.com
[18] http://www.live.com
[19] http://www.facebook.com

3 CONSTATAÇÕES PRELIMINARES

Neste capítulo serão apresentadas constatações de vestígios de conversas realizadas em ambiente *web* remanescentes em arquivos de hibernação, paginação e despejos de memória. Essas constatações determinaram a viabilidade do presente estudo e possibilitaram a derivação da metodologia proposta.

Com base nos conceitos de análise de memória e volatilidade dos dados apresentados no capítulo 2, foram realizados testes que simulavam ambientes *web* de conversação entre usuários. Esses testes consistiram na realização de buscas nos artefatos forenses foco das análises e contribuíram para definição dos procedimentos para se identificar e extrair o conteúdo de conversas realizadas.

Foram realizados testes utilizando as seguintes ferramentas forenses para análise de tráfego de rede e memória:

- Wireshark[20] para análise de tráfego de rede;
- Forensic Toolkit[21] para indexação e buscas em arquivos;
- Charles *Web* Proxy[22] para análise de tráfego entre navegador e memória.

Dois comunicadores de mensagens instantâneas foram utilizados nos testes em simulações de conversas para análise preliminar do comportamento das mensagens trocadas entre usuários criados para esta finalidade. Foram selecionados o *Gtalk*[23] e o *Windows Live Messenger*[24] por serem mais populares no Brasil, conforme seção *1.3.1 – Popularização de programas de Mensagens Instantâneas.*

[20] http://www.wireshark.org/
[21] http://www.accessdata.com
[22] http://www.charlesproxy.com/
[23] http://www.gmail.com/
[24] http://www.live.com/

3.1 ENCERRAMENTO DO NAVEGADOR E PERSISTÊNCIA EM MEMÓRIA

Este primeiro teste visou averiguar a persistência dos dados carregados na memória principal após o fechamento da janela do navegador. Para isso, foram realizados os seguintes passos:

1. Troca de mensagens entre os usuários Alice e Bob pela versão *web* do *IMs*;
2. Fechamento da janela do navegador;
3. Captura de memória após o fechamento da janela;
4. Realizada busca no conteúdo de memória capturado pelo nome "Bob" ou qualquer cadeia de caractere trocada durante a conversação.

Esses passos foram repetidos para diferentes navegadores executando em sistemas operacionais distintos. A Tabela 3.1 mostra que dos ambientes testados, utilizando-se os navegadores *Internet Explorer*[25], *Firefox*[26] e *Chrome*[27], somente o navegador *Chrome* retirou os dados em memória quando do fechamento da janela, não possibilitando recuperação das mensagens ao se executar o passo 4:

Tabela 3.1 – Persistência dos dados em memória após encerramento do navegador

Navegador / S.O	IE8	Firefox	Chrome 17.0.963.46
Windows XP	Sim	Sim	Não
Windows 7	Sim	Sim	Não
OpenSuse 11.2	Não Aplicável	Sim	Não Aplicável

[25] http://www.microsoft.com
[26] http://www.mozilla.org
[27] https://www.google.com/chrome

3.2 MENSAGENS NOS ARQUIVOS PAGEFILE.SYS E HIBERFIL.SYS

Foram constatadas mensagens remanescentes nos arquivos de paginação e hibernação de uma estação de trabalho onde haviam sido realizadas conversações reais em ambiente *Windows*, utilizando-se o *Gtalk* por meio do Internet Explorer 8. As cadeias de caracteres conhecidas das comunicações reais foram utilizadas como argumento de pesquisa nas buscas realizadas por meio do programa *FTK – Forensic Toolkit.*

A cadeia de caracteres "TESTERONEIMAIA" era conhecida por ter sido utilizada em sessão de conversação anterior. Os parâmetros utilizados na realização deste teste se encontram na Tabela 3.2:

Tabela 3.2–Persistência dos dados nos arquivos pagefile.sys e hiberfil.sys

Sistema Operacional	*Windows* 7
Virtualizado	NÃO
Navegador	Internet Explorer 8
HTTPS	SIM
Evento	1. Têm-se dois usuários do *Gtalk* trocando mensagens entre si por meio de navegadores em máquinas distintas. 2. Já havia sido utilizada a cadeia de caracteres "TESTERONEIMAIA" em sessão de comunicação anterior e esta foi utilizada para verificar possibilidade de recuperação. 3. O computador havia sido hibernado anteriormente.
Análise	Analisados os arquivos hiberfil.sys e pagefile.sys diretamente em busca da cadeia de caracteres.
Resultado	Verificado que foi possível encontrar cadeia de caractere utilizada, a partir do *hiberfil.sys*, porém não foi verificada sua existência no pagefile.sys.
Exemplo	33&req0_type=m\o=ronei.maia%40g l.coQ>1FF2D763B7_ext=**TESTERONEIMAIA** chatstate=aPctive iconset=squarDTTUYH

Ressalta-se que o arquivo de hibernação se encontra compactado originalmente pelo sistema operacional e, mesmo assim, foi possível recuperar a cadeia de caracteres utilizada como argumento de pesquisa. A Figura 3.1 ilustra o formato da cadeia encontrada na busca indexada:

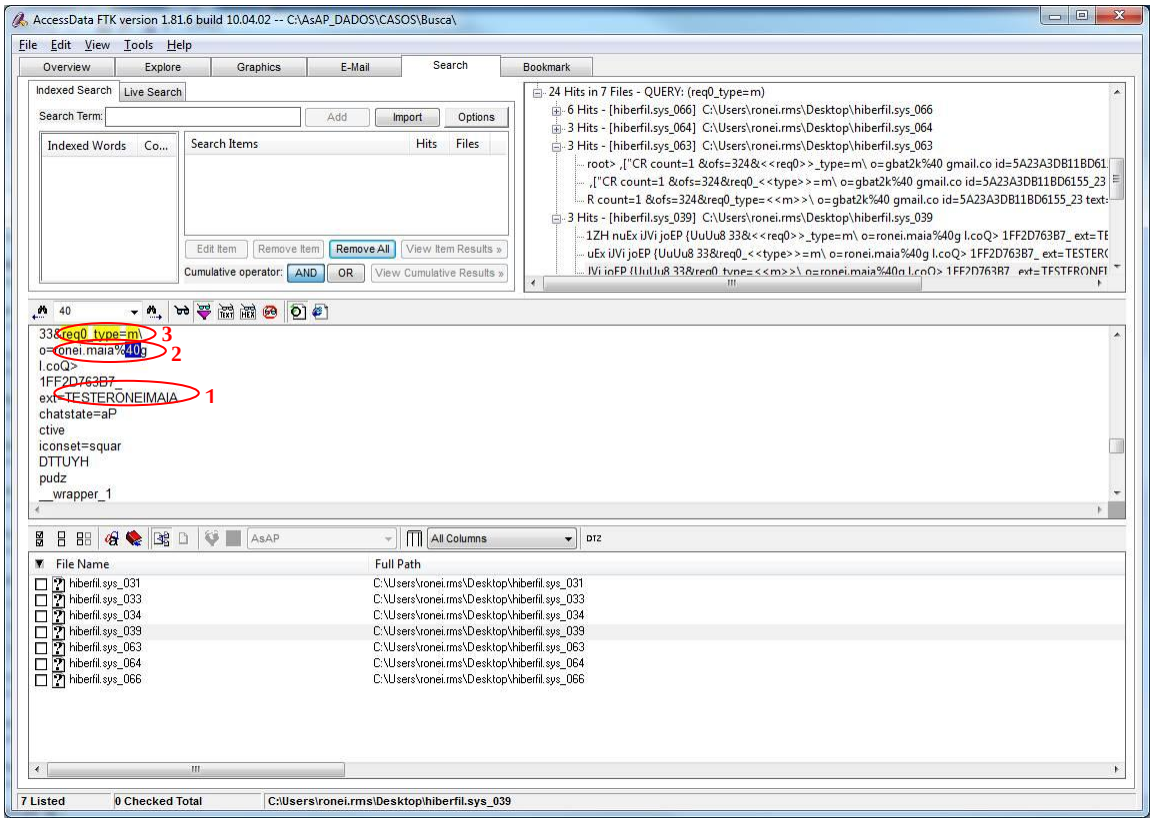

Figura 3.1 – Busca de sequência de caracteres previamente conhecida.

Conforme indicado em “1” na Figura 3.1, foi possível verificar a existência da cadeia de caracteres *TESTERONEIMAIA* no arquivo de hibernação.

Em “2”, identificou-se o nome *ronei.maia* que era um dos nomes de usuário utilizados na comunicação e que estava conectado, sendo um dos destinatários das mensagens enviadas. Até então, o nome dos usuários envolvidos não tinham importância para os resultados, uma vez que este experimento não foi realizado em ambiente controlado, não sendo necessário identificar, incialmente, os usuários que realizavam a conversação. No entanto, essa foi uma verificação importante, por mostrar a possibilidade de se recuperar outros parâmetros utilizados na comunicação.

Próximo ao nome de usuário identificado em “3”, foi encontrada a seguinte cadeia de caracteres *req0_type=m*. Essa cadeia se assemelha a um parâmetro de cabeçalho que ocorreu imediatamente antes de T*ESTERONEIMAIA*.

Uma nova busca indexada foi realizada utilizando-se *req0_type=m* como argumento de pesquisa. Foram observadas 24 ocorrências desta palavra em vários fragmentos do arquivo de hibernação analisado.

Ao analisar as outras ocorrências de *req0_type=m*, identificou-se outras conversas anteriores, que também não foram realizadas no ambiente de teste controlado. Isso prova que as conversas podem permanecer armazenadas por tempo indeterminado.

A Figura 3.2 mostra outro trecho de conversa realizada anteriormente para usuário identificado a partir da busca de *req0_type=m.* É possível observar a cadeia destacada em "1" - "*Meu%20irm%C3%A3o tem acontecid oisa!*" enviada na ocasião para um dos contatos do usuário utilizado. Também é possível verificar o nome do usuário destinatário, como pode ser confirmado em "2" – "*savio.zaidan%40gmail.co"*:

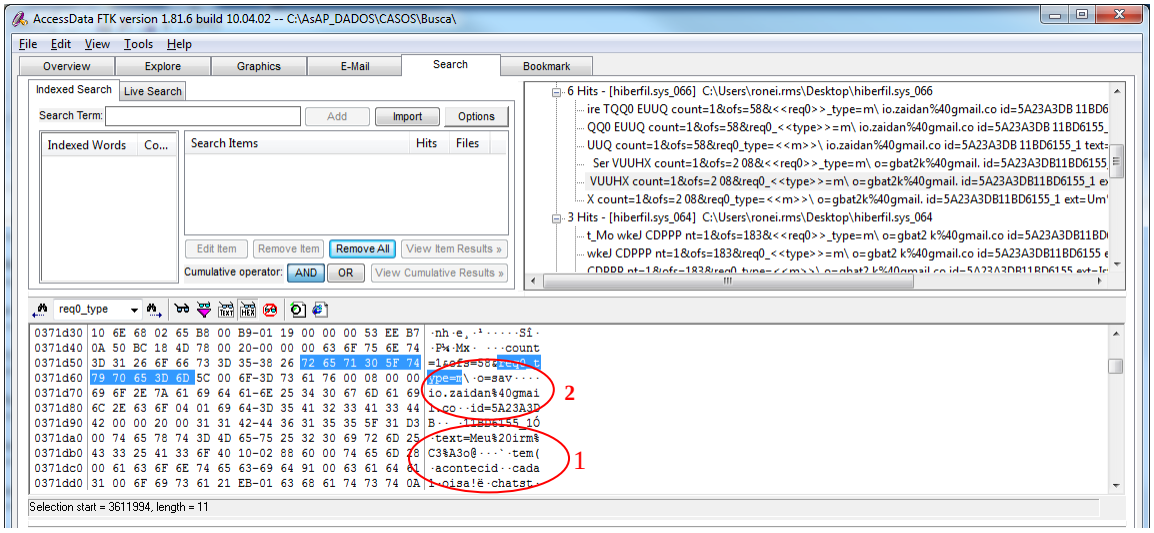

Figura 3.2 – Trechos de conversas recuperadas da máquina local.

Outra ocorrência de *req0_type=m* revelou mais um trecho de conversa. Foi identificada a cadeia em "1" "*Ent%C3%A3o%20 1 vamos8 trabalhar em cima disso%3F"* que na ocasião da conversação tinha sido enviada para o contato gbat2k@gmail.com, conforme constatado em "2" na Figura 3.3:

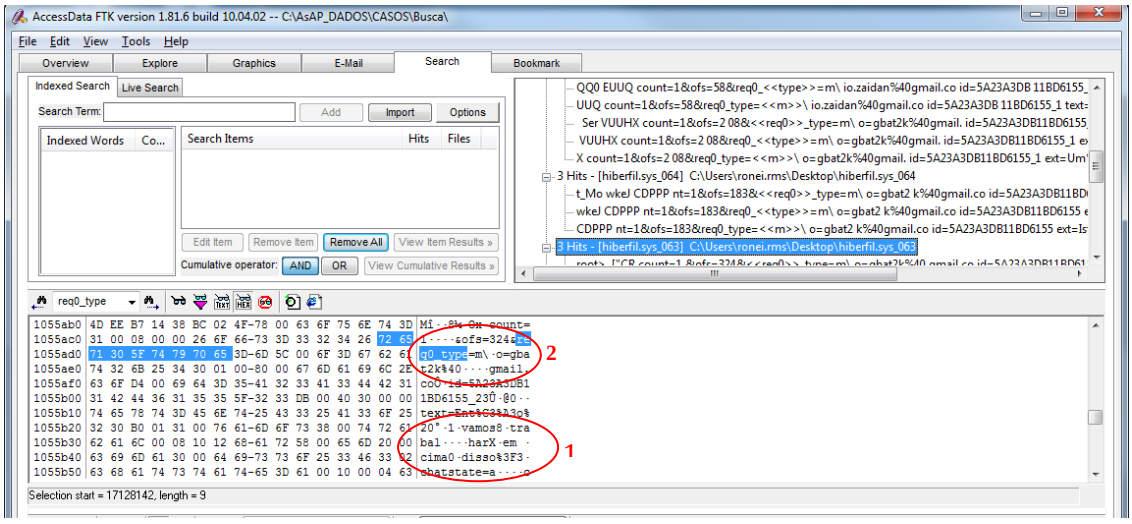

Figura 3.3 – Trechos de conversas recuperadas

É possível concluir a partir destes testes que atributos de comunicação, como por exemplo, nomes de usuários para os quais eram enviadas mensagens, bem como as próprias mensagens em si, podem ser recuperados a partir do arquivo de hiberfil.sys, ainda que essas mensagens tenham sido enviadas em data diversa da realização das buscas e do procedimento de hibernação pelo usuário.

3.3 MENSAGENS NO TRÁFEGO DE REDE

Os endereços eletrônicos alice.unb2011@gmail.com e bob.dpf2011@gmail.com foram criados para analisar os dados a partir de simulações de conversações realizadas. Os parâmetros dessas simulações são apresentados na Tabela 3.3:

Tabela 3.3– Cenário para análise do tráfego de rede

Sistema Operacional	*Windows* 7
Virtualizado	SIM
Navegador	Internet Explorer 8
HTTPS	NÃO
Evento	Dois usuários do *Gtalk* (alice.unb2011 e bob.dpf2011). Bob manda para Alice: *"Ola Alice! Aki eh Bob!!!"*
Fonte de Dados	Tráfego de rede via Wireshark.
Resultado	Identificadas variáveis definidas pelo programador do protocolo de comunicação contidas na requisição *http*.
Exemplo	count=2&ofs=63&req0_type=c&req0_cmd=a&req0_jid=alice.unb2011%40gmail.com&req0__sc=c&req1_type=m&req1_to=alice.unb2011%40gmail.com&req1_id=46E67DA00C8F05BD_4&req1_text=**Ola%20Alice!%20Aki%20eh%20Bob!!!**&req1_chatstate=active&req1_iconset=classic&req1

A Figura 3.4 ilustra a captura de tráfego de rede realizada por meio do Wireshark. É possível identificar na requisição *http* as variáveis utilizadas no protocolo de comunicação, conforme definidas pelo programador:

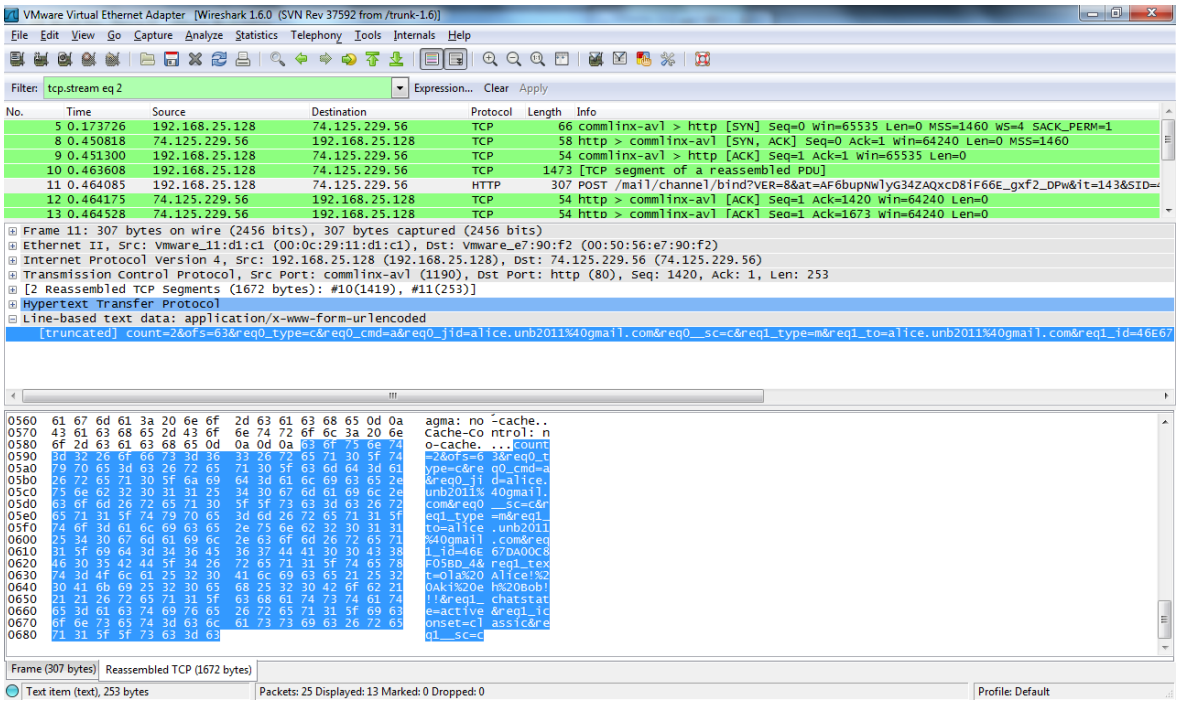

Figura 3.4 – Tráfego de rede capturado

3.4 MENSAGENS NO TRÁFEGO NAVEGADOR-MEMÓRIA

Para verificação do conteúdo trafegado existente na memória do computador, utilizou-se um *proxy* de aplicação que auxilia a classificação dos parâmetros contidos nas requisições *web*. A Tabela 3.4 resume o ambiente utilizado para constatação de conteúdo:

Tabela 3.4 – Cenário para análise do tráfego de dados entre navegador e memória

Sistema Operacional	*Windows* 7
Virtualizado	SIM
Navegador	Internet Explorer 8
Evento	Usuários do *Gtalk* (alice.unb2011 e bob.dpf2011) Bob manda para Alice: *"Ola Alice! Aki eh Bob!!!"*
Fonte de Dados	Captura de tráfego em memória por meio do *proxy* de aplicação – *Charles web proxy*.
Resultado	Identificadas, próximo à cadeia de caracteres enviada de Bob para Alice, as mesmas variáveis definidas pelo programador em experimento anterior.
Exemplo	count=2&ofs=63&req0_type=c&req0_cmd=a&req0_jid=alice.unb2011%40gmail.com&req0__sc=c&req1_type=m&**req1_to**=alice.unb2011%40gmail.com&req1_id=46E67DA00C8F05BD_4&**req1_text=Ola%20Alice!%20Aki%20eh%20Bob!!!**&req1_chatstate=active&req1_iconset=classic

Os testes realizados com o uso do *proxy* de aplicação *web*, *Charles*, mostraram que o mesmo conteúdo capturado do tráfego de rede pelo *Wireshark*, na constatação anterior, também pode ser obtido a partir da captura do tráfego de dados do navegador na memória.

A Figura 3.5 mostra interfaces do *proxy* de aplicação. À esquerda como são esperados encontrar os dados em memória. À direita, a funcionalidade da ferramenta para se formatar variáveis utilizadas na requisição *http*. É possível escolher modos de visualização, separando parâmetros identificados automaticamente em campos. Essa funcionalidade facilita a identificação dos parâmetros do protocolo.

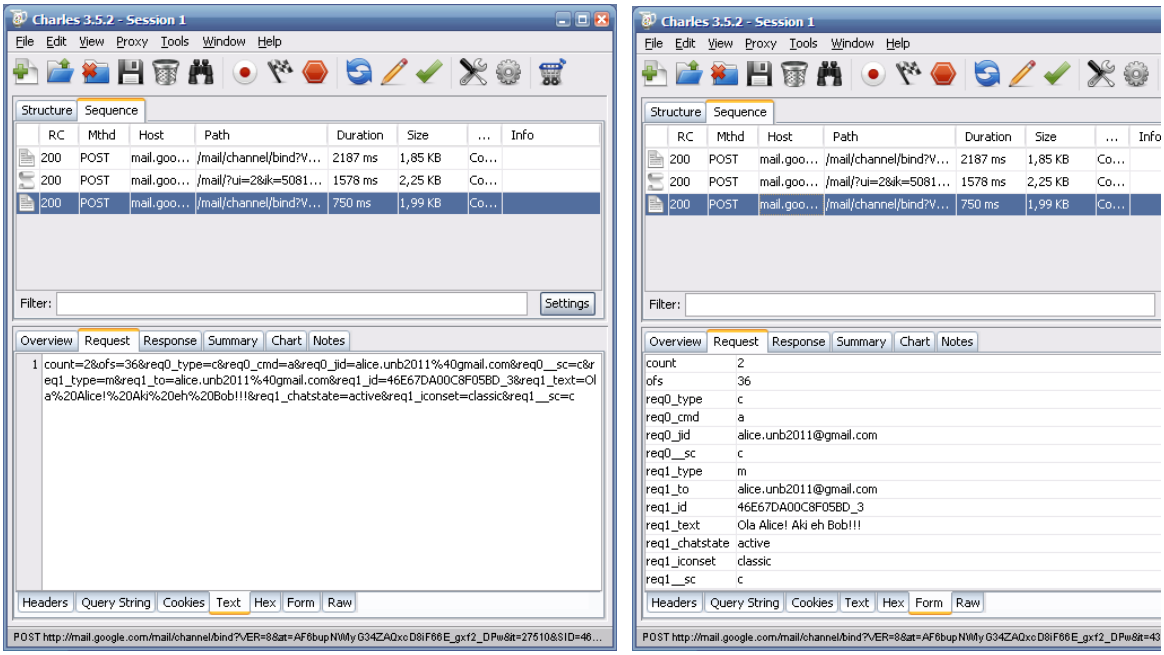

Figura 3.5 – Captura de evento do Gtalk por meio do programa *Charles web proxy*

Com o objetivo de realizar as mesmas constatações dos testes anteriores, porém no ambiente *web* do *Windows* Live Messenger – WLM, foram criados dois novos usuários: alice-unb@live.com e bob-dpf@live.com. A Tabela 3.5 ilustra os parâmetros utilizados nessa simulação.

Tabela 3.5– Parâmetros utilizados na simulação com o WLM

Sistema Operacional	*Windows* 7, Chrome
Sessões de comunicação	Não cifradas em ambos os usuários do WLM (alice-unb e bob-dpf) Bob manda para Alice: *"Olá Alice UNB aqui é Bob DPF!!!"*
Cenário	Captura de dados em memória por meio do *proxy* de aplicação, *Charles*.
Resultado	Não foram verificadas variáveis na requisição *http* em comparação com os experimentos realizados com o G*talk*

Exemplo de captura:	SDG 16 305 Routing: 1.0 To: 1:alice-unb@live.com From:1:bob-dpf@live.com;epid={82b641da-5fa4-419b-b44e-87003ad5a0ae} Reliability: 1.0 Messaging: 2.0 Content-Type: text/plain; charset=UTF-8 Message-Type: Text X-MMS-IM-Format: FN=Segoe%20UI; EF=; CO=0 Content-Length: 31 Olá Alice-UNB! Aki é Bob-DPF!

A Figura 3.6 mostra como se deu a captura dos parâmetros utilizados pelo WLM. Diferentemente dos padrões identificados nos testes com o G*talk*, não foi observada a utilização de variáveis na requisição *http*:

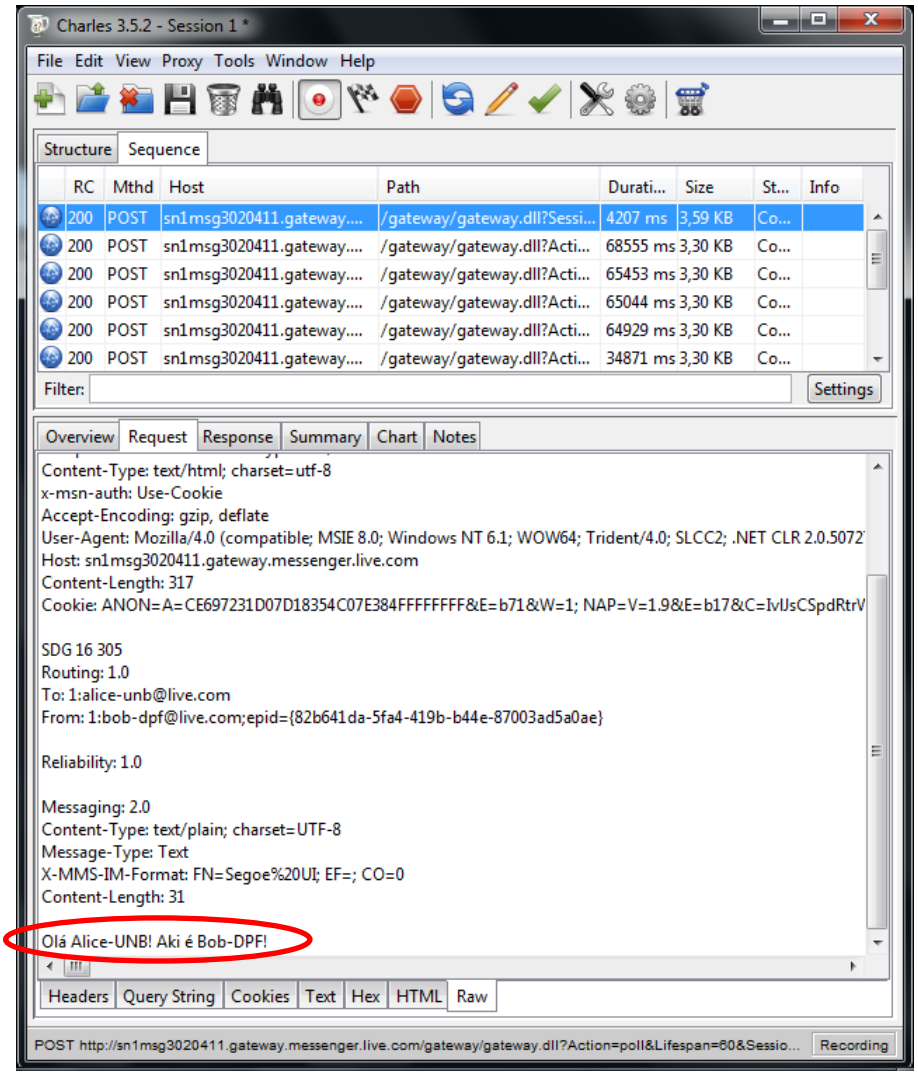

Figura 3.6 – Dados trafegados em memória

3.5 MENSAGENS NO TRÁFEGO NAVEGADOR-MEMÓRIA CIFRADO

Para confrontar o conteúdo trafegado existente na memória do computador de forma criptografada com o tráfego sem criptografia capturado no experimento anterior, repetiu-se a captura utilizando-se comunicação em *https*: A Tabela 3.6 resume o ambiente utilizado para constatação de conteúdo:

Tabela 3.6 – Cenário para análise do tráfego de dados criptografado

Sistema Operacional	*Windows* 7
Virtualizado	NÃO
Navegador	Chrome 15.0.874.121
HTTPS	SIM
Sessões de comunicação	Cifradas em ambos usuários do *Gtalk* (alice.unb2011 e bob.dpf2011) Alice manda para Bob: *"Ola Bob! Aki eh Alice!!!"*
Cenário	Captura de dados em memória por meio do *proxy* de aplicação, *Charles*.
Resultado	Identificadas, próximo à cadeia de caracteres enviada de Alice para Bob, as mesmas variáveis definidas pelo programador em experimento anterior.
Exemplo	count=2&ofs=21&req0_type=c&req0_cmd=a&req0_jid=bob.dpf2011%40gmail.com&req0__sc=c&req1_type=m&**req1_to**=bob.dpf2011%40gmail.com&req1_id=2BCC1E3952EF157C_0&**req1_text**=Oi%20Bob!!%20Aki%20eh%20Alice!!!&req1_chatstate=active&req1_iconset=classic&req1
Captura *https*	POST /mail/channel/bind?VER=8&at=AF6bupNXZXBLX8eBi3bTLkzvtAIYRIpRvw&it=21&SID=2BCC1E3952EF157C&RID=60515&AID=36&zx=ccx4eeoc3ov7&t=1 HTTP/1.1 Host: mail.google.com Connection: keep-alive Referer: **https://mail.google.com/mail/?ui=2&view=js&name=main,tlist&ver=V0pVgbtIxfk.pt_BR.&am=!QOoT6kuBunO7Qv3h9Nwowg_ktLZGNU8rmNaktdZHn9VyQarmVk6GWxMZ0_dS&fri** Content-Length: 249 Origin: **https://mail.google.com** User-Agent: Mozilla/5.0 (*Windows* NT 6.1; WOW64) Apple*Web*Kit/534.30 (KHTML, like Gecko) Chrome/12.0.742.112 Safari/534.30 Content-Type: application/x-www-form-urlencoded Accept: */* Accept-Encoding: gzip,deflate,sdch Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3 Cookie: S=gmail=PjSt1saPzbj4-pzntUGFtg:gmproxy=-iU08m-

	dmLe9R5pDG6owNA; GXSP=S; **GX=DQAAAIkAAAAS9ZrM9CXd4VDZJoXWkQ9Hzw_eXHNyftcYR3sHq1Qw9Y-dMSbqgaNZHt_SfX1f5BA7QgNjy6hhi38ClaB9vHNdeCiyZN-UrBw3UsqnwuAfBIZDl2ETSgPUJQ0-QHA8k5UBns1H671o2eAcXY6UEn-wJsbIKGxmgAVOCJuHhBnKv-1dZe9uW_yzJIDBWRIgTS4**; GMAIL_AT=AF6bupNXZXBLX8eBi3bTLkzvtAIYRIpRvw; gmailchat=alice.unb2011@gmail.com/672438; PREF=ID=cf897a238b3dbe5e:U=8b21066fc527a4b1:FF=0:TM=1304515282:LM=1304547499:GM=1:S=wXbFlKZe_pkXvUlh; rememberme=false; NID=48=YVyAiaVLU8W9lh0gr42DVcWAJP1nTFlXBkmA2U-gpAXVFvvvr5ByOT5yoDPrn71mzykTt8tZSkeMG1DgQt5V-XP-HSID=Aliy9UVJPy340URvW; SSID=AQBLRsD0mH16Bymjk

A Figura 3.7 ilustra os mesmos parâmetros identificados no teste anterior, desta vez, capturados em memória. Verifica-se que a técnica pode ser utilizada para identificar o conteúdo trocado pelo *Gtalk*, mesmo quando a comunicação é cifrada.

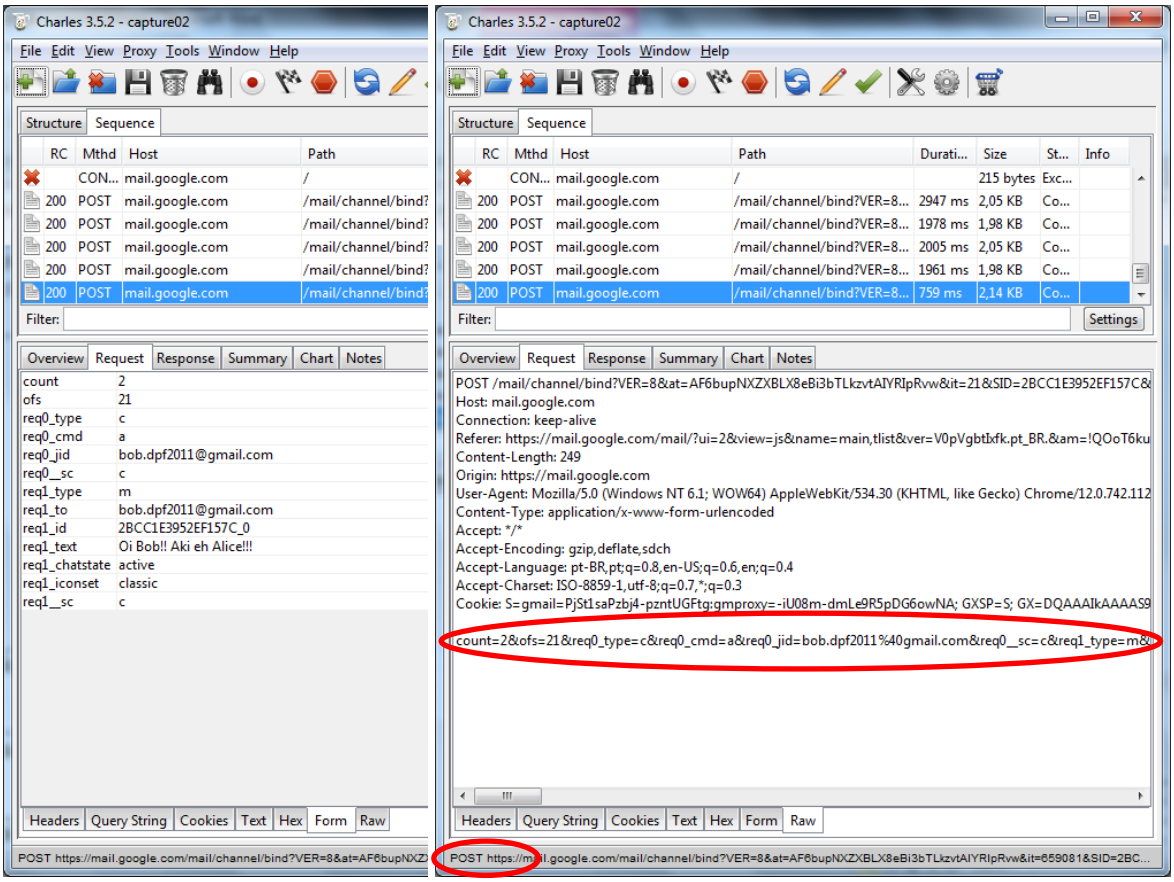

Figura 3.7 – Captura de tráfego em memória antes da cifragem dos dados.

3.6 MENSAGENS GERADAS SEM CONVERSAS ARMAZENADAS

Diante dos testes anteriores, percebeu-se que muito fluxo de requisições *http* era gerado, mesmo que apenas uma frase ou uma pequena cadeia de caracteres fosse trocada. Os tráfegos de conversação entre os usuários corresponderam à minoria do tráfego total capturado durante as análises.

Com base nas capturas realizadas, tanto a partir do tráfego de rede, como a partir do tráfego em memória, verificou-se que o grande volume de transações se dá pelo fato de o comunicador *Gtalk* realizar requisições correspondentes aos eventos ocorridos na janela de comunicação, como por exemplo, quando o *mouse* é movimentado na área da janela, para demonstrar atividade do usuário, quando uma determinada tecla é pressionada ou também quando ocorre algum período de inatividade, para atualizar o status do usuário.

Segundo estudo realizado por Xiao, Guo e Tracey (2007), sobre as características do tráfego de *IMs*, é comum a geração de muito tráfego de rede mesmo que os usuários envolvidos na sessão de comunicação não troquem mensagens entre si. Nesse tráfego involuntário, ocorrem sinalizações de *status* de atividade dos usuários por parte do *IM* de forma a notificar o servidor que a outra parte envolvida na comunicação realizou alguma atividade.

Essas simulações mostraram as solicitações *https* referentes a mensagens de eventos gerados e que não correspondem a troca de mensagens entre usuários. A Figura 3.8 ilustra os diferentes valores armazenados nas variáveis utilizadas pelo protocolo de comunicação identificadas como *req0_chatstate* e *req0_evtype*, como definidas pelo programador:

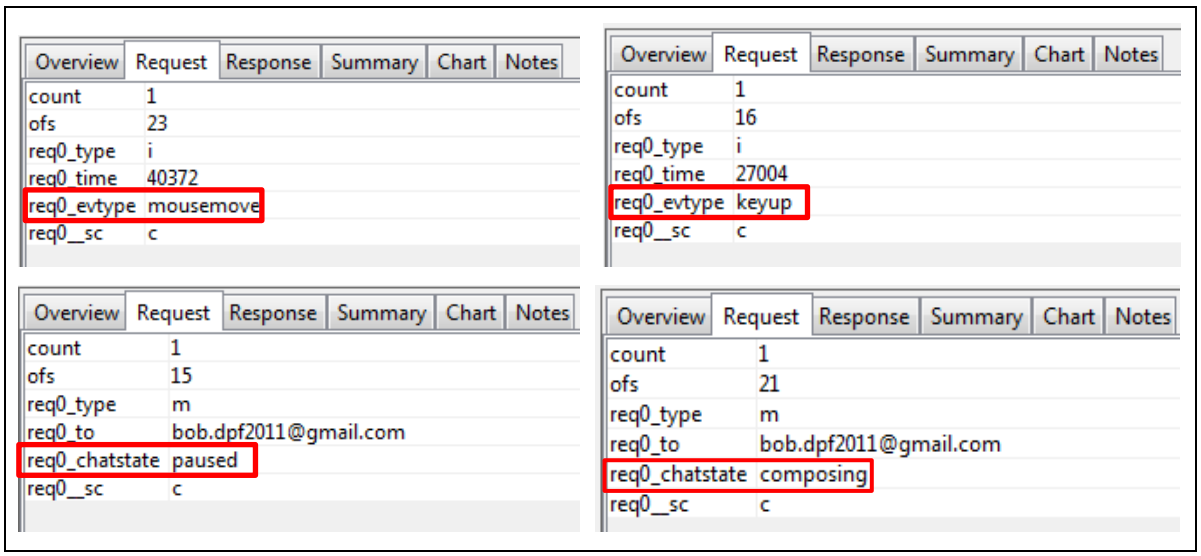

Figura 3.8 – Eventos do *Gtalk* capturados em memória a partir do *Charles web proxy*.

4 METODOLOGIA PROPOSTA

Neste capítulo é apresentada a metodologia proposta derivada das constatações preliminares descritas no Capítulo 3. São definidas as etapas do processo e identificados os artefatos que constituem fonte de informações.

Os procedimentos para recuperação de conversas geradas pelos *IMs web-based* foram divididos em duas etapas: identificação e extração dos vestígios deixados pelas requisições de comunicação *web*. Esses procedimentos visam validar as seguintes hipóteses: que é possível identificar padrões utilizados em eventos pré-definidos por cada comunicador, sem a necessidade de aprofundar a análise de cada protocolo utilizado; e que é possível reconhecer ocorrências desses padrões nos artefatos forenses relacionados aos arquivos de paginação, hibernação e despejos de memória.

Esses três artefatos são os objetos das análises como alternativa a não geração de registros de logs de conversas em disco e à limitação da geração de cache pelos navegadores de Internet em decorrência de implementações que utilizam tecnologias como Ajax.

4.1 IDENTIFICAÇÃO DE VESTÍGIOS

O processo de identificação de vestígios consiste em gerar volume de tráfego de comunicações simuladas entre usuários criados para este fim, para os diversos comunicadores que se pretende estudar. O objetivo é identificar repetidas cadeias de caracteres específicas utilizadas nas requisições *web* que encapsulam de forma adaptada o protocolo de troca de mensagens. Essas cadeias podem ser agrupadas para serem utilizadas como marcadores, palavras-chave ou expressões regulares para realização de buscas na extração dos vestígios. O resultado desta primeira etapa é a definição de um dicionário que servirá como parâmetro de entrada para a etapa seguinte.

Para obter o padrão das palavras-chave utilizadas pelos comunicadores e constatação de seu conteúdo em memória, é realizada análise do fluxo do tráfego de rede em simulações de conversas entre usuários, bem como análise do fluxo de dados entre os navegadores de Internet e a memória. Para isso, são utilizadas máquinas virtuais com os recursos de *snapshot* para permitir que as repetições das simulações sejam realizadas a partir de um mesmo estado inicial e que não sejam influenciadas pela execução de testes anteriores.

Essas análises são baseadas na observação direta dos fluxos trocados, onde o agrupamento das palavras-chave é feito pela quantidade de ocorrências e no contexto em que ocorrem nas sucessivas simulações das comunicações entre os usuários envolvidos nos eventos controlados.

4.1.1 Definição de eventos controlados para coleta de padrões

As simulações são concentradas em eventos predefinidos que foram elencados por serem intrínsecos ao uso de programas de *IM*, tendo grandes chances de serem encontrados vestígios gerados por esses eventos nos computadores analisados em casos reais. Foram gerados eventos referentes a:

4.1.1.1 Carregamento da lista de contatos

Para obter o padrão de carregamento da lista de contatos são utilizados cenários para a repetição dos testes para cada *IM* submetido a estudo, convencionando-se a máquina virtual de um dos usuários a fonte de dados a ser coletada. Esse comportamento se repete para os demais usuários:

- *Login* e todos seus contatos estão ***off-line***;
- *Login* e todos seus contatos estão ***on-line***.

Os estados de disponibilidade dos usuários da lista de contato, após realização de *login*, são analisados para verificar possíveis diferenças existentes no formato dos vestígios gerados pelo protocolo de comunicação.

4.1.1.2 Troca de mensagens instantâneas entre usuários

Foram convencionadas conversas baseadas na troca de cadeias de caracteres predefinidas, acrescidas de um contador, por exemplo: “MENSAGEM01, MENSAGEM02, ...” de forma que um dos usuários foco das análises sempre envie mensagens com sequenciais ímpares e sempre receba mensagens com sequenciais pares. Essa padronização visa identificar eventuais conversas não extraídas, permitindo confronto estatístico entre mensagens recuperadas e perdidas, bem como a identificação do emissor e receptor das mensagens.

Esses eventos são repetidos procurando-se gerar uma quantidade significativa de amostras no contexto das análises dos dados trafegados em rede e em memória, bem como na análise dos arquivos de paginação e hibernação.

4.1.2 Parâmetros para os testes e identificação das palavras-chave

Os parâmetros utilizados nas simulações são documentados à medida que os experimentos eram realizados. Os resultados eram colhidos e o ambiente de monitoramento era documentado, descrevendo-se:

- Sistema Operacional utilizado para os testes;
- Utilização ou não de máquina virtual na simulação;
- Tipo e versão do navegador de Internet utilizado;
- Utilização ou não de sessões *https* nas sessões de comunicação;
- Evento gerado: nomes de usuários dos *IMs* utilizados para realizar a comunicação e cadeia de caracteres enviada;
- Fonte de dados capturada (Tráfego de dados em rede, memória, ou ambos);
- Resultado da análise da captura: as cadeias de caracteres comuns utilizadas por um mesmo *IM* encontradas repetidamente entre as simulações.

A repetição das mesmas cadeias de caracteres anteriormente já identificadas nos rastros do protocolo e adjacentes aos *offsets* das mensagens de teste conhecidas nas simulações caracteriza palavra-chave candidata para compor o dicionário do comunicador testado.

4.1.3 Coleta de padrões a partir dos tráfegos de rede e memória

Os padrões para compor o dicionário podem ser obtidos a partir do tráfego de rede ou da memória. No caso de um determinado comunicador utilizar somente sessões de comunicação com cifragem de dados, por meio do uso de *https*, impedindo a identificação das palavras-chave a partir da captura e análise do tráfego de rede, é utilizado um *proxy* de aplicação capaz de interceptar dados enviados à memória principal pelos navegadores de Internet, antes do mecanismo de criptografia ocorrer, contornando o problema da cifragem dos dados. Os procedimentos para coleta dos fluxos de dados a partir do tráfego de rede e memória para identificação do conteúdo proveniente do protocolo de troca de mensagens dos *IMs* são ilustrados na Figura 4.1.

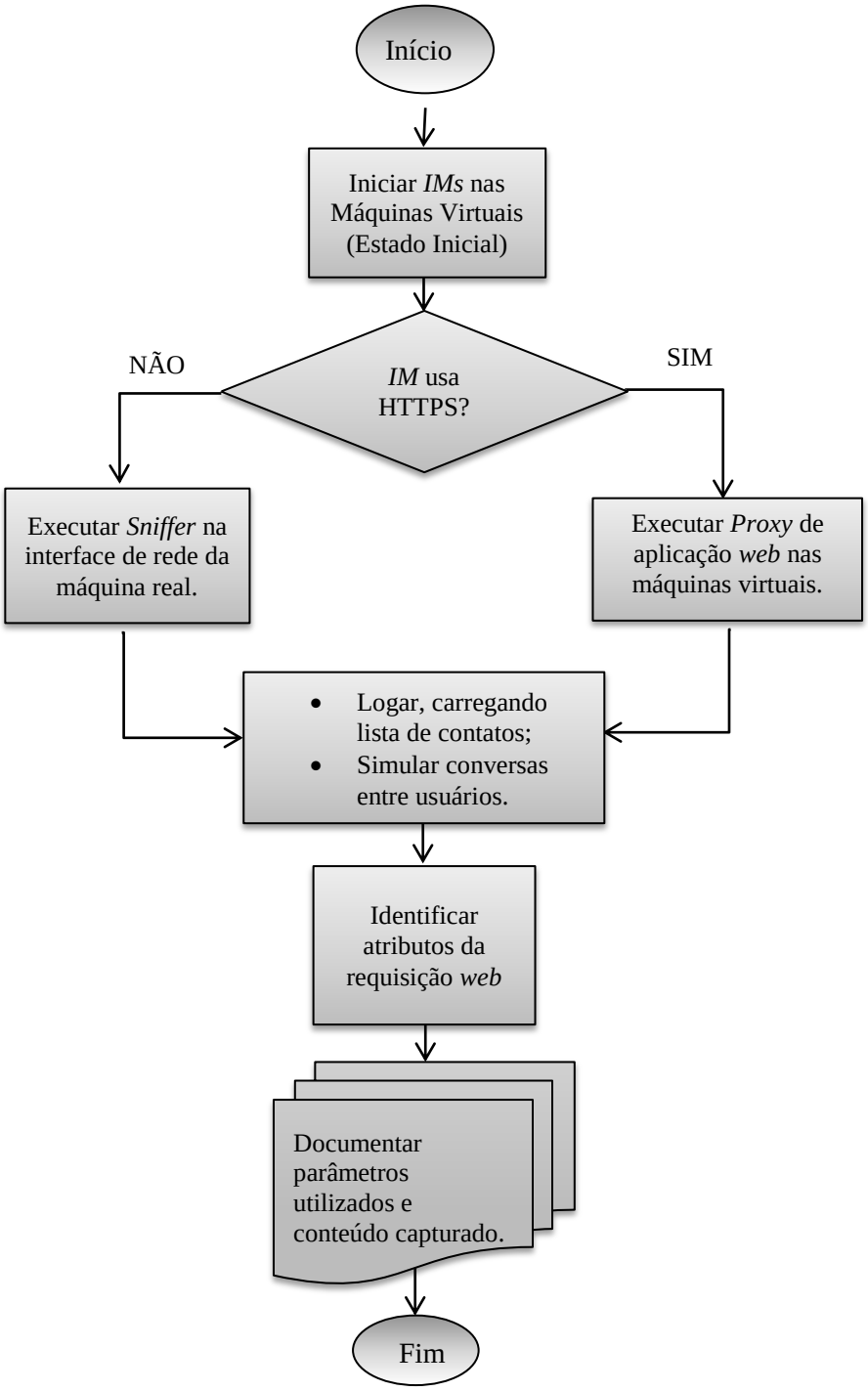

Figura 4.1 – Coleta de conversas a partir do tráfego de rede e memória.

4.1.3.1 Analise de tráfego de rede

Os testes foram realizados com comunicações entre duas sessões de conversação não criptografadas e utilizando-se o fluxo de comunicação pela Internet entre máquinas virtuais e máquinas reais. O monitoramento foi realizado utilizando a ferramenta de análise de

protocolo de rede *wireshark*[28], com o objetivo de verificar o conteúdo das mensagens trocadas pelos *IMs* que é possível recuperar e como foram definidos os parâmetros de requisição *http* pelo programador do protocolo de comunicação. A partir dessas informações é possível identificar caracteres delimitadores candidatos que podem ser utilizados como palavra-chave.

4.1.3.2 Análise de tráfego em memória (Proxy de aplicação *web*)

Despejos de memória podem ser realizados para contornar o problema da análise do tráfego de rede com dados criptografados. A Figura 4.2 ilustra a obtenção dos dados de forma alternativa ao canal criptografado:

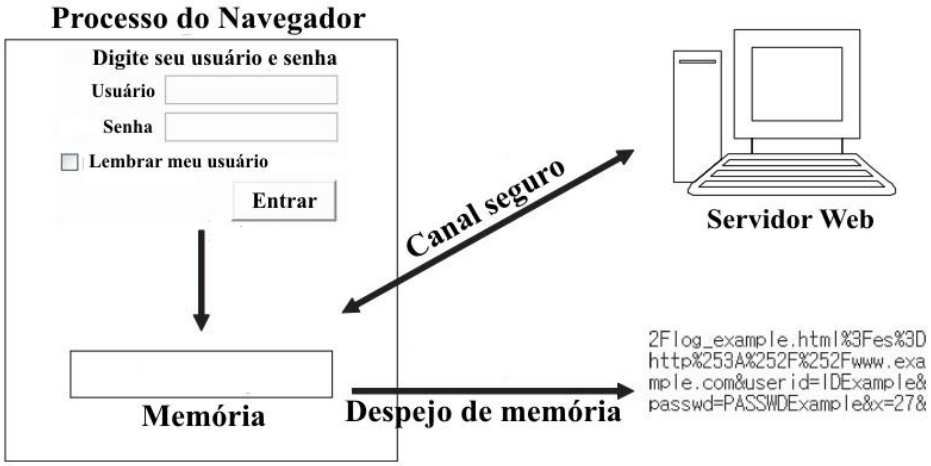

Figura 4.2 – Captura de dados em memória. (adaptada de Lee, et al, 2007)

Um *proxy* de aplicação, no entanto, pode ser utilizado para facilitar a identificação do conteúdo trafegado entre o navegador e a memória, pois funciona como um agente que intermedeia as requisições realizadas. Nas simulações foi utilizado o *software Charles*[29], que permite a captura dos códigos HTML, bem como do tráfego gerado por requisições Ajax, contidos nas requisições *http* ou *https*. Esse *software* fornece um de despejo de memória com as seguintes funcionalidades (Randow, 2011):

- O tráfego capturado pode ser armazenado em arquivos;
- Interpretação do conteúdo capturado;
- Identificação de existência de formulários *web*;

[28] Wireshark – http://www.wireshark.org
[29] Charles – http://www.charlesproxy.com

- Identificação e separação de parâmetros utilizados em requisições *http*;
- Divisão dos campos do cabeçalho *http*;
- Possibilidade de configuração de certificados para realização da captura de tráfego *https* em texto em claro;
- Discriminação do protocolo utilizado *http* ou *https*.

Dessa forma, seu uso permite realizar análises do tráfego gerado pelo protocolo de comunicação de *IMs web-based* antes da encriptação dos seus dados e submissão cifrada pela rede, sem a necessidade de se realizar pesquisas em toda a área de memória capturada por um despejo de memória padrão.

Como os *IMs* permitem configurar as sessões de comunicação para utilizarem o protocolo *https* na realização da troca de mensagens e arquivos, foram utilizados ambientes com conversas cifradas e não cifradas em ambos os navegadores dos usuários envolvidos na comunicação. Esse procedimento visa verificar a convergência das informações fornecidas por meio das capturas realizadas, a partir do tráfego de rede, e as realizadas a partir do *proxy* de aplicação, constatando que as cadeias de caracteres identificadas pela comunicação cifrada são as mesmas realizadas em comunicações abertas. Essa constatação define um mesmo padrão de busca por comunicador.

4.1.4 Coleta de padrões a partir de arquivos de paginação e hibernação

Esta etapa de geração de padrões consiste em criar versões diferentes de arquivos de paginação e hibernação, contendo conversas simuladas entre usuários, com o objetivo de constatar traços conhecidos do protocolo remanescentes no computador. A Figura 4.3 mostra o fluxo de procedimentos para a geração de padrões e identificação dos traços armazenados:

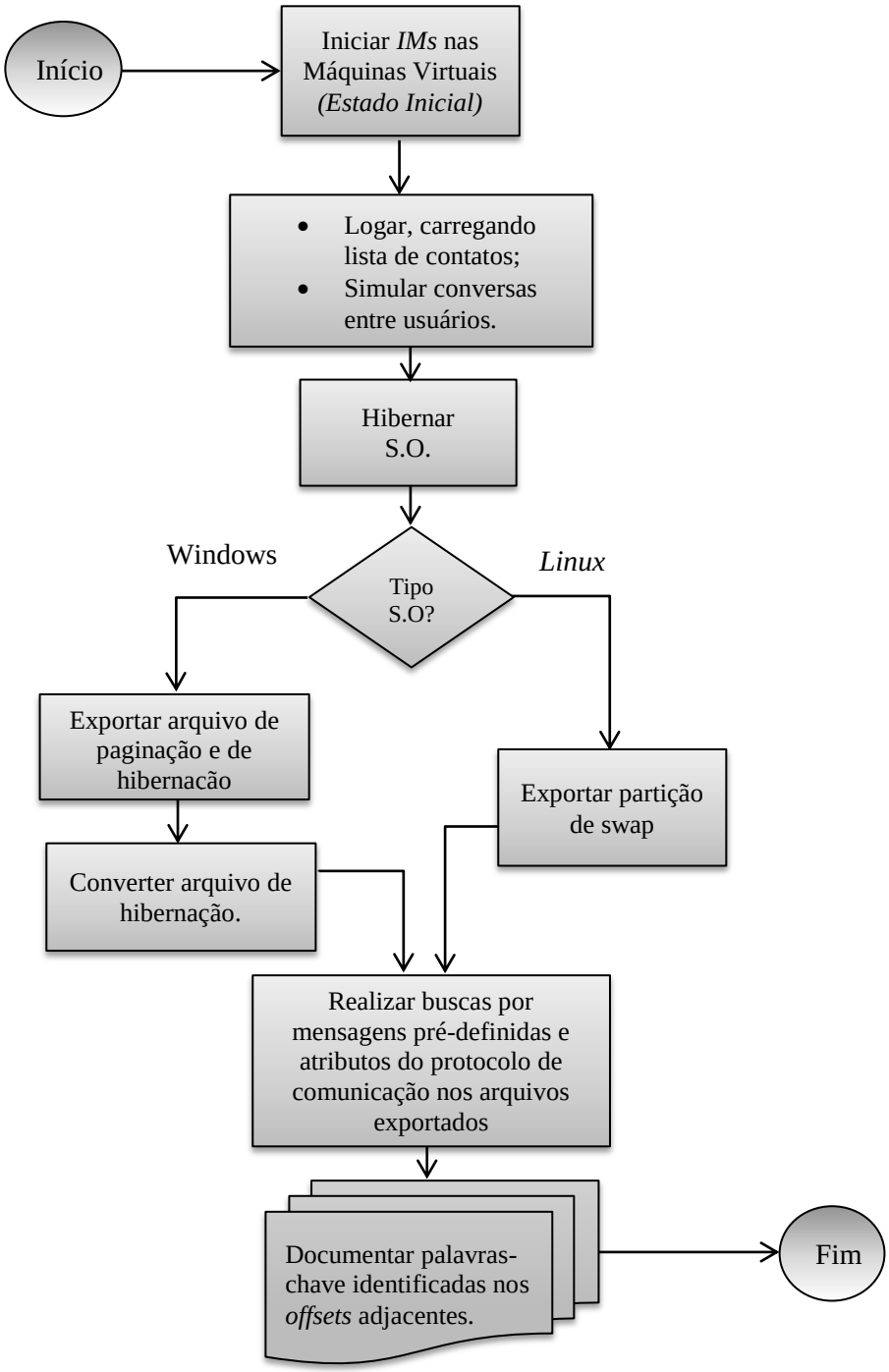

Figura 4.3 – Verificação de padrões de requisições *web* em arquivos extraídos.

Pode-se considerar o diagrama ilustrado na Figura 4.3 uma continuação do diagrama da Figura 4.1. No entanto, devido à utilização do *proxy* de aplicação *web*, que executa na memória dos sistema operacionais virtualizados, há a necessidade de simulação das conversas em ambiente sem execução de programas de captura, para que a memória não fique "contaminada" com dados constantes capturados e formatados pelo proxy. Dessa forma, ao hibernar o sistema ou ocorrer o processo de paginação, têm-se os dados livres de qualquer interferência por formatação de *software* externos.

Para permitir reprodução e comparação entre os testes, é necessário que os arquivos referentes à memória do sistema operacional estejam "limpos", isto é, sem dados remanescentes de testes anteriores. O recurso de *snapshots* presente nas máquinas virtuais permite que se tenha sempre um mesmo ponto de partida para realização dos testes.

A fim de aumentar as chances de se encontrar conteúdo utilizado nas conversas das primeiras simulações, procura-se forçar o processo de paginação, utilizando-se máquinas virtuais com diferentes configurações de tamanho de memória RAM, variando entre 256MBytes a 1GBytes.

De acordo com as primeiras buscas realizadas pelas cadeias de caracteres conhecidas para constatação de conteúdo nos arquivos de paginação e hibernação, foram identificadas mensagens armazenadas em ambos os formatos ASCII e Unicode em *offsets* não consecutivos. O fato de existir uma mesma informação duplicada em pontos distintos dos arquivos examinados aumenta as chances de recuperação das conversas, pois ainda que parte do arquivo tenha sido sobrescrita, um "backup" dessa informação, em outro formato, tem chances ainda de ser recuperado a partir de outras posições nos arquivos.

Devido a essa característica de armazenamento em mais de um formato nos arquivos examinados, visando maximizar as chances de recuperação de vestígios nos artefatos, é relevante realizar busca por palavras-chave, considerando os formatos ASCII e Unicode.

4.2 EXTRAÇÃO DE VESTÍGIOS

A segunda etapa da metodologia proposta consiste na recuperação propriamente dita. São realizadas buscas nos arquivos de paginação, hibernação e despejos de memória, utilizando-se como argumento de pesquisa os elementos definidos no dicionário criado na etapa anterior. Os elementos encontrados podem representar vestígios fossilizados de protocolos de comunicação, cujo conteúdo das mensagens pode ser extraído e classificado, de acordo com a origem em cada comunicador utilizado. Essa fase conta com um protótipo que requer como parâmetros de entrada de dados o dicionário gerado para automatizar as buscas e separar o conteúdo, e o caminho do arquivo que se deseja analisar, validando as hipóteses.

As buscas nos arquivos de memória constatam a hipótese de que a informação residente advinda das requisições *web* tem possibilidade de recuperação considerada. Como a fase anterior consistiu em analisar artefatos forenses além da memória, para verificar a existência das palavras-chave identificadas, são objetos de análise para extração de vestígios:

- Despejos de memória RAM;
- Arquivo de paginação;
- Arquivo de hibernação.

A partir de ferramentas forenses de indexação e editores hexadecimais, o processo de extração pode ser realizado por meio da busca por palavras-chave e expressões regulares geradas pelos comunicadores para cada tipo de vestígio de interesse. Nesse trabalho foram considerados vestígios a serem processados as conversas e as listas de contato.

4.2.1 Buscas manuais

De posse das palavras-chave identificadas na fase anterior, é possível utilizar o dicionário criado para realizar as buscas nos artefatos forenses identificados como fontes de vestígios. São utilizados editores hexadecimais para se visualizar conteúdo referente à memória, como, por exemplo, o *software Winhex* [30] *e FTKImager*[31]. A simples busca por palavras-chave nos arquivos coletados exige do Perito atenção. Deve ser feita uma análise detalhada dos *offsets* adjacentes às ocorrências encontradas para identificar o conteúdo das conversas

[30] http://accessdata.com/support/adownloads
[31] http://www.winhex.com/winhex/

propriamente ditas, visto que as buscas não retornam exatamente as conversas realizadas, apenas ocorrências das palavras-chave.

Conforme tamanho dos arquivos referentes à memória, o processo de busca pode ser um pouco demorado. Opcionalmente, pode-se indexar os arquivos referentes à memória para realização das buscas em um tempo menor.

4.2.2 Protótipo

Uma vez que a análise de arquivos de memória tende a ser um processo tedioso e por vezes informações podem passar despercebidas, visando automatizar o processo de extração dos dados armazenados, um protótipo foi desenvolvido com inteligência para encontrar as palavras-chave definidas na fase anterior a partir do dicionário gerado, e categorizar os atributos identificados na conversação.

O protótipo foi escrito em linguagem Java visando à portabilidade entre plataformas, uma vez que sistemas *Windows* e sistemas baseados em *Linux* foram objetos de análise na fase anterior de identificação de palavras-chave.

Ao vasculhar os artefatos forenses alvos das buscas e encontrar palavras-chave remanescentes, o protótipo executa métodos que tratam *offsets* adjacentes, onde as informações alvo das análises realmente se encontram.

A separação e categorização dos tipos de vestígios encontrados, tais como, lista de contatos e mensagens trocadas é realizada automaticamente, sem a necessidade de o Perito analisar *offsets* adjacentes em sua totalidade.

4.2.2.1 Parâmetros de entrada

- Caminho do arquivo a ser analisado (hibernação, paginação ou despejos de memória);
- Caminho do arquivo contendo dicionário de palavras-chave.

4.2.2.2 Processamento

- Fragmentos que possam conter traços de lista de contatos ou conversações realizadas entre usuários, conforme definido no dicionário.

4.2.2.3 Saída

- Lista de fragmentos extraídos contendo listas de contatos identificadas;
- Lista de fragmentos extraídos contendo conversações entre usuários;
- Relatório discriminando atributos encontrados referentes às listas e as conversações.

Os arquivos de hibernação devem estar em formato descompactado para otimizar as chances de recuperação das conversas encontradas. Para descompactação do arquivo de hibernação de sistemas *Windows* foi utilizado o *software volatility* capaz de converter o referido arquivo em formato de despejo de memória.

Apenas para validar a leitura dos arquivos de hibernação de sistemas *Linux*, foi utilizado o artifício de hibernar o sistema operacional com a opção "*suspend to disk"* com os parâmetros "*compress=n*" e "*encrypt=n*" configurados nos arquivos */etc/suspend.conf*, desligando-se o computador invocando-se o comando *pm-hibernate.* Maiores detalhes sobre a manipulação de arquivos de hibernação estão descritos na seção *2.1.3 – Hibernação*.

O dicionário contendo as palavras-chave consiste em um arquivo em formato XML, conforme DTD detalhado na Tabela 4.1:

Tabela 4.1 – Formato do arquivo XML contendo o dicionário das palavras-chave

```
-----------------------------------------------------
Exemplo DTD
-----------------------------------------------------
<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT Dictionary (key-words)*>

<!--- Instant Messaging de onde foram extraídas as palavras-chave. -->
<!ELEMENT key-words (friend-list|message)*>
<!ATTLIST key-words
   im CDATA #IMPLIED
  >
<!--- Lista de palavras-chave para recuperação de conversas -->
<!ELEMENT message (#PCDATA)>

<!--- Lista de palavras-chave para recuperação da lista de contatos. -->
<!ELEMENT friend-list (#PCDATA)>
```

```
-----------------------------------------------------
Exemplo XML
-----------------------------------------------------
<Dictionary>

    <key-words im="GTALK">
        <chat>palavra-chave-01</chat>
        <chat>palavra-chave-02</chat>
        <friend-list> palavra-chave-03</friend-list>
        <friend-list> palavra-chave-04</friend-list>
    </key-words>

</Dictionary>
```

A possível similaridade das palavras-chave utilizadas pelos *IMs* com demais cadeias de caracteres constantes em memória, porém sem nenhum significado no contexto de conversas de *IM*, implica na possibilidade de recuperação de *offsets* que denotam falsos positivos. O protótipo é capaz de realizar identificação de parte desses falsos positivos realizando a uma filtragem por ocorrência do caractere "@" presente nos endereções de correio eletrônico dos usuários envolvidos na transação. Nas simulações verificou-se a existência desse caractere na maioria das mensagens de *IMs* trocadas, o que pode ser uma característica utilizada como filtro para tentar eliminar a quantidade de falsos positivos que possam ocorrer.

O tamanho dos arquivos a serem processados depende da quantidade de memória RAM existente nas máquinas virtuais analisadas. De qualquer forma, por se tratar do processamento de arquivos de grandes tamanhos, seriam necessárias máquinas com mais memória do que o tamanho dos arquivos. Para evitar exceções do tipo falha de memória, os arquivos são lidos em blocos menores, os quais são processados separadamente. Isso significa que se um mesmo fragmento de conversa estiver contido na leitura de dois blocos diferentes, estando parte da conversa na leitura do bloco anterior à leitura do novo bloco, a conversa pode não ser extraída completamente e alguns atributos podem ficar prejudicados ou, no caso de a própria palavra-chave utilizada se encontrar em blocos de leitura diferentes, o fragmento de conversa não será extraído. Essa limitação, no entanto, refere-se apenas à forma como foi implementado o protótipo e não tem maiores implicações na demonstração da técnica utilizada.

Com relação ao desempenho do protótipo na recuperação dos fragmentos de conversas, não houve diferença entre a realização da pesquisa com um único padrão de expressão

regular, contendo concatenação de todas as cadeias de caracteres utilizadas como palavra-chave e da pesquisa com um padrão por vez.

Com a implementação do protótipo, é possível concluir que mesmo conhecendo as palavras-chave para a recuperação das conversas, adaptações são necessárias para construções dos *parses* para realizar a categorização dos atributos contidos nos fragmentos extraídos de novos comunicadores.

Na implementação do protótipo foi considerada uma interface gráfica para ilustrar as funcionalidades de extrações dos vestígios, conforme ilustra a Figura 4.4.

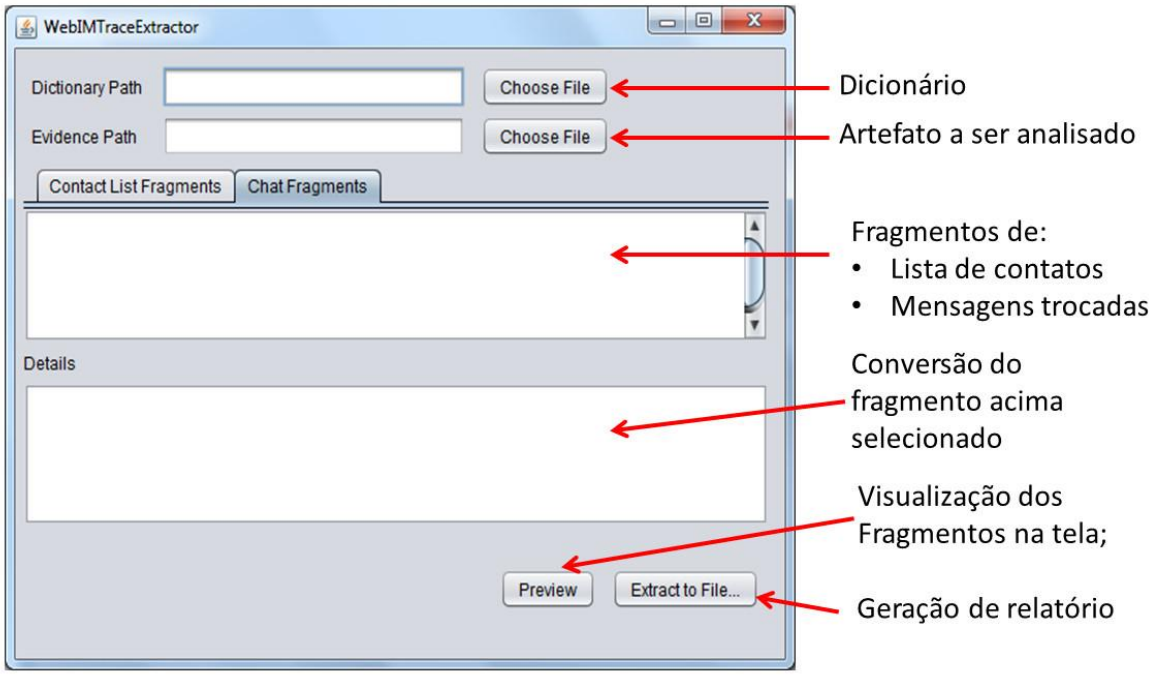

Figura 4.4 – Interface desenvolvida para ilustrar as funcionalidades do protótipo

5 ESTUDOS DE CASO

Neste capítulo são apresentados os estudos de caso de recuperação de conversas ocorridas em quatro *IMs web-based.* Aplicou-se a metodologia proposta como prova de conceito dos procedimentos adotados para identificação e extração das conversas instantâneas realizadas nesse ambiente. Os estudos de caso se basearam em simulações de eventos de conversas entre usuários, procurando gerar uma quantidade significativa de amostras a serem analisadas. O objetivo de cada estudo de caso é identificar padrões, criar um dicionário de palavras-chave e realizar a extração das conversas simuladas nos ambientes, conforme definido na metodologia.

Foram analisados os tráfegos de dados das conversas entre a memória e os navegadores de Internet, bem como o tráfego de rede. Essas análises permitem identificar visualmente as cadeias de caracteres existentes nos protocolos formatados nas requisições *http.* O objetivo é identificar as cadeias que podem ser utilizadas como palavras-chave ou expressões regulares na recuperação de mensagens nos arquivos de memória virtual.

O protótipo desenvolvido foi utilizado para realizar a recuperação automática das mensagens, de acordo com os parâmetros de entrada, e para confrontar os resultados obtidos na análise visual das conversas realizadas.

Dada à diversidade dos comunicadores existentes, é comum que suas implementações possam diferir na forma como os dados são submetidos, tanto na memória, quanto no tráfego de rede. A partir das análises desses fluxos, constatou-se que o padrão de submissão dos protocolos de comunicação encapsulados nas requisições *http* varia, conforme definidas as requisições de cada tipo de comunicador.

Por isso, as simulações foram padronizadas para que fosse analisado um mesmo conjunto de variáveis que se repetem para um mesmo *IM*, e agrupá-las de acordo com cada tipo de comunicador analisado.

O método proposto é aplicado a um ambiente controlado baseado no uso de máquinas virtuais e os resultados agrupados por tipo de comunicador. Serão utilizados os usuários criados para este fim, analisadores de tráfego de rede, *proxy* de aplicação e análises de despejos de memória.

Para permitir a reprodutibilidade dos testes, o recurso de *snapshots* fornecido pelas máquinas virtuais foi utilizado para preservar o estado inicial dos sistemas operacionais.

Baseado nas informações de utilização e popularidade dos *IMs* descritas na seção *1.3.1 – Popularização de programas de Mensagens Instantâneas*, foram criados diferentes usuários para cada um dos *IMs* objetos de estudo: *Gtalk*, *Windows Live Messenger e Yahoo!Messenger*. E como redes sociais permitirem aos usuários comunicar-se por meio de mensagens instantâneas, foram criados usuários também para o *Facebook*.

Foram simulados eventos de conversação entre os usuários nas plataformas *Windows*, executando os navegadores Internet Explorer, Firefox e Chrome e na plataforma OpenSuse *Linux*, utilizando apenas o navegador Firefox, conforme Tabela 5.1:

Tabela 5.1 – Usuários de Teste

Contas de usuários	**Comunicadores *Web-based* testados**	**Ambientes**
 alice.unb2011	@live.com	
 bob.dpf2011	@yahoo.com	
 charlie.dpf2011	@gmail.com	
 dave.dpf2011	@facebook.com	
 eve.dpf2011		

Os cenários foram baseados na análise de dois eventos, conforme definido na metodologia:

1. Carregamento da lista de contatos do usuário Alice;
2. Troca de mensagens realizadas entre Alice e Bob.

Foram utilizadas as mesmas ferramentas empregadas nos testes para as constatações preliminares descritas no Capítulo 3, com foco na análise dos dados colhidos da máquina de Alice. Para cada evento, foram realizadas coletas: dos dados capturados do conteúdo trafegado em rede; do proxy de aplicação *web*, quando as simulações eram realizada em

https; dos despejos de memória e dos arquivos de hibernação e paginação. A Tabela 5.2 discrimina o ambiente das versões dos navegadores e sistemas operacionais utilizados:

Tabela 5.2 – Configuração de ambientes e fontes de dados coletadas

Sistemas Operacionais e Navegadores	WinXPSp02 => IE. 8.0, Firefox10, Chrome 17.0.963.46m, *Windows* 7 => IE. 8.0, Firefox 5.0, Firefox10, Chrome 17.0.963.46m, OpenSuse*Linux* 11.2 => Firefox 3.5
Virtualização	Virtualizado e não virtualizado
Criptografia	Http e Https
Fonte de Dados	Hiberfil.sys, Pagefile.sys, tráfego de rede e memória

Para o primeiro evento definido, como é esperado o carregamento dos contatos de Alice no ato do *login*, foram realizadas buscas nos dados coletados, utilizando-se como argumento de pesquisa a cadeia de caracteres *bob.dpf2011*.

Para o segundo evento utilizaram-se as mensagens trocadas entre os usuários Alice e Bob padronizadas com as cadeias: *Mensagem01, Mensagem02, Mensagem03 e Mensagem04* de forma que as mensagens com número ímpares eram sempre enviadas por Alice, e as com números pares sempre recebidas de outros usuários. O cenário simulado para identificação das mensagens é ilustrado na Figura 5.1:

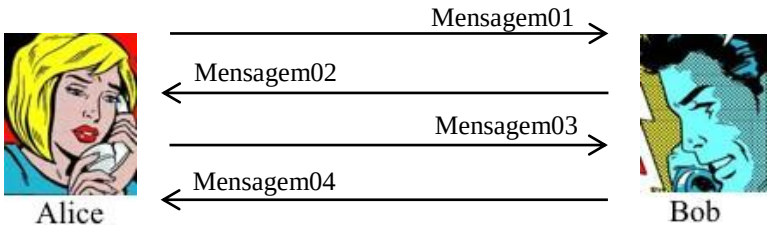

Figura 5.1 – Padrão do conteúdo das mensagens trocadas nas simulações.

Após as capturas, são realizadas buscas preliminares pelas palavras utilizadas durante os eventos (bob.dpf2011, Mensage01, Mensagem02...) e identificadas outras cadeias de caracteres que se repetem nos eventos em *offsets* adjacentes nos artefatos analisados. Assim são obtidas as palavras-chave discriminadas em um dicionário para recuperação dos eventos, a partir de buscas executadas automaticamente pelo protótipo.

Os resultados de cada etapa da aplicação da metodologia proposta foram descrito nos itens "Identificação de palavras-chave", "análises" e "extração de vestígios", contidos em cada estudo de caso apresentado nas seções a seguir.

5.1 CASO 01: GTALK

Nesta seção estão reunidas simulações realizadas por meio das conversas utilizando-se a versão *web* do *Gtalk*. Algumas simulações se deram de forma cifrada automaticamente. Esse comunicador, no entanto, permite acesso às configurações de sessão *https* de forma que o usuário pode habilitar ou desabilitar o seu uso. Porém, com relação ao navegador *Chrome*, não foi possível realizar o *login* para ter acesso ao bate-papo sem o uso de criptografia, visto que a própria aplicação redirecionava o usuário para o serviço em *https*, após o *login*. Mas, com a aplicação dos procedimentos definidos, foi possível recuperar padrões mesmo quando a comunicação era cifrada. Nas seções seguintes são apresentadas as análises realizadas durante os procedimentos de identificação e extração de vestígios, definidos na metodologia proposta:

5.1.1 Identificação de palavras-chave

Ao aplicar a primeira fase do método, foram identificadas formatações diferentes para mensagens enviadas ou recebidas. Mais de um formato pode ser reconhecido para uma mesma mensagem. A Tabela 5.3 resume exemplos da identificação dos campos das mensagens e formato gerado pelos diferentes navegadores testados:

Tabela 5.3 – Padrões gerados pelo *Gtalk*.

Lista de contatos:

Internet Explorer 8

,[\42c\42,[\42r\42,0,[[0,\42bob.dpf2011@gmail.com\42,\0420\42,\42Bob Dpf\42,4,0,0,0,,\42\42,0,2,2,0,1,\42Bob Dpf\42,\42Bob\42,0,\42\42,0,0,0,\42bob.dpf2011@gmail.com\42,0,0,\42\42,\42\42,\0422cc4e1798a78db50\42,[]\n,[]\n,0,0,-1,0,\42https://plus.google.com/109381310432393877303\42,\42\42,\42\42,\42\42,\042109381310432393877303\42,0,0,0]\n,[0,\42charlie.dpf2011@gmail.com\42,\0421\42,\42Charlie Dpf\42,0,0,0,0,,\42\42,0,2,2,0,1,\42Charlie Dpf\42,\42Charlie\42,0,\42\42,0,0,0,\42charlie.dpf2011@gmail.com\42,0,0,\42\42,\42\42,\0424da0946c89875ab6\42,[]\n,[]\n,0,0,-

Demais Navegadores

gmailchat=alice.unb2011@gmail.com/700039; jid=alice.unb2011@gmail.com/
[[9,**["c","r",**0,[**[0,"**bob.dpf2011@gmail.com","0","Bob Dpf",4,0,0,0,,"",0,2,2,0,1,"Bob Dpf","Bob",0,"",0,0,0,"bob.dpf2011@gmail.com",0,0,"","","2cc4e1798a78db50",[]
,[]
,**[0,"**charlie.dpf2011@gmail.com","1","Charlie Dpf",0,0,0,0,,"",0,2,2,0,1,"Charlie Dpf","Charlie",0,"",0,0,0,"charlie.dpf2011@gmail.com",0,0,"","","4da0946c89875ab6",[]
,[]
,**[0,"**dave.dpf2011@gmail.com","2","Dave Dpf",0,0,0,0,,"",0,2,2,0,1,"Dave Dpf","Dave",0,"",0,0,0,"dave.dpf2011@gmail.com",0,0,"","","594af7c80a7d40b9",[]
,[]
,**[0,"**eve.dpf2011@gmail.com","3","Eve Dpf",0,0,0,0,,"",0,2,2,0,0,"Eve Dpf","Eve",0,"",0,0,0,"eve.dpf2011@gmail.com",0,0,"","","71e9a0a10e411a95",[]

Mensagens enviadas para Bob (Formato 1)

["c",["csu","bob.dpf2011@gmail.com",1,"o",[["B13F41A726A2FFF5_1","c",1325046352588,1,"alice.unb2011@gmail.com/gmail.5B988C5E","Mensagem01","Mensagem01",",

Origem **Destino** **Mensagem** **Data e hora em milisegundos**

Mensagens enviadas para Bob (Formato 2)

count=1&ofs=10&req0_type=m&req0_to=bob.dpf2011%40gmail.com&req0_id=B13F41A726A2FFF5_0&req0_text=Mensagem01&req0_chatstate=active&req0_iconset=classic&req0

Mensagens recebidas de Bob:

["c",["m","bob.dpf2011@gmail.com","DC09759C3D2BFE68_2","active","Mensagem02","Mensagem02

Exemplo de Palavras-chave Identificadas:

Usuário que realiza *login*: **gmailchat=**
Início da lista de contatos: **,["c",["r",**
Início de cada contato no bloco identificado: **,[0,"**
Mensagens enviadas ou recebidas: **["c",["csu","**
Mensagens recebidas: **["c",["m","**

5.1.2 Análises

Não foi identificado campo vinculando a lista de contato encontrada ao usuário Alice. No entanto foram encontradas muitas ocorrências da cadeia gmailchat=alice.unb2011@gmail.com. Somente é possível associar Alice a lista de contatos, caso não haja ocorrências de outros usuários associados à cadeia *gmailchat=*.

A diferença entre o estado on-line/off-line dos contatos na realização do *login* é observada na mudança de valor referente aos offsets destacados nos exemplos a seguir:

[0,"bob.dpf2011@gmail.com","0","Bob Dpf",4,0,0,**0**,"",0,2,2,**0**,1 *(off-line)*
[0,"bob.dpf2011@gmail.com","0","Bob Dpf",4,0,0,**1**,"",0,2,2,**1**,1 *(on-line)*

[0,"charlie.dpf2011@gmail.com","1","Charlie Dpf",0,0,0,**0**,"",0,2,2,**0**,1 *(off-line)*
[0,"charlie.dpf2011@gmail.com","1","Charlie Dpf",0,0,0,**1**,"",0,2,2,**1**,1 *(on-line)*

[0,"dave.dpf2011@gmail.com","2","Dave Dpf",0,0,0,**0**,"",0,2,2,**0**,1 *(off-line)*
[0,"dave.dpf2011@gmail.com","2","Dave Dpf",0,0,0,**1**,"",0,2,2,**1**,1 *(on-line)*

[0,"eve.dpf2011@gmail.com","2","Eve Dpf",0,0,0,**0**,"",0,2,2,**0**,1 *(off-line)*
[0,"eve.dpf2011@gmail.com","2","Eve Dpf",0,0,0,**1**,"",0,2,2,**1**,1 *(on-line)*

Para o Internet Explorer 8, em todas as simulações, o caractere aspas duplas (") se encontrava codificado como /42 ou /042.

Nem sempre foi possível associar mensagem ao usuário de origem. Alguns fragmentos só foram encontrados contendo atributos referentes a destinatário, mensagem e data.

5.1.3 Extração de vestígios

As Figuras Figura **5.2** e Figura **5.3** ilustram, respectivamente, extrações automatizadas de listas de contatos e conversações realizadas pelo protótipo a partir do dicionário contendo palavras-chave identificadas na fase anterior. Para demonstração foi utilizado o arquivo de hibernação do *Windows* em formato descompactado:

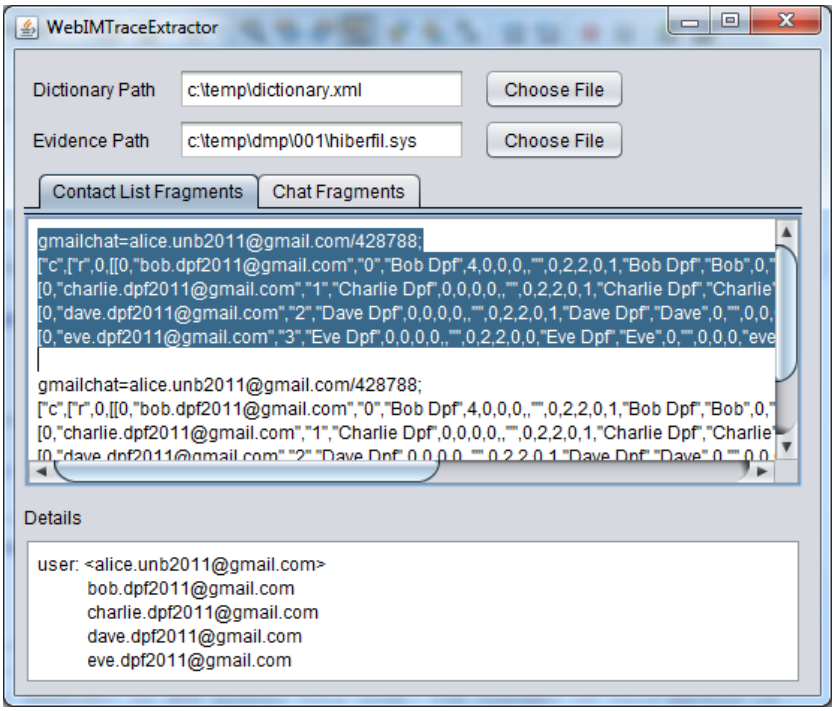

Figura 5.2 – Extração de lista de contatos - *Gtalk*

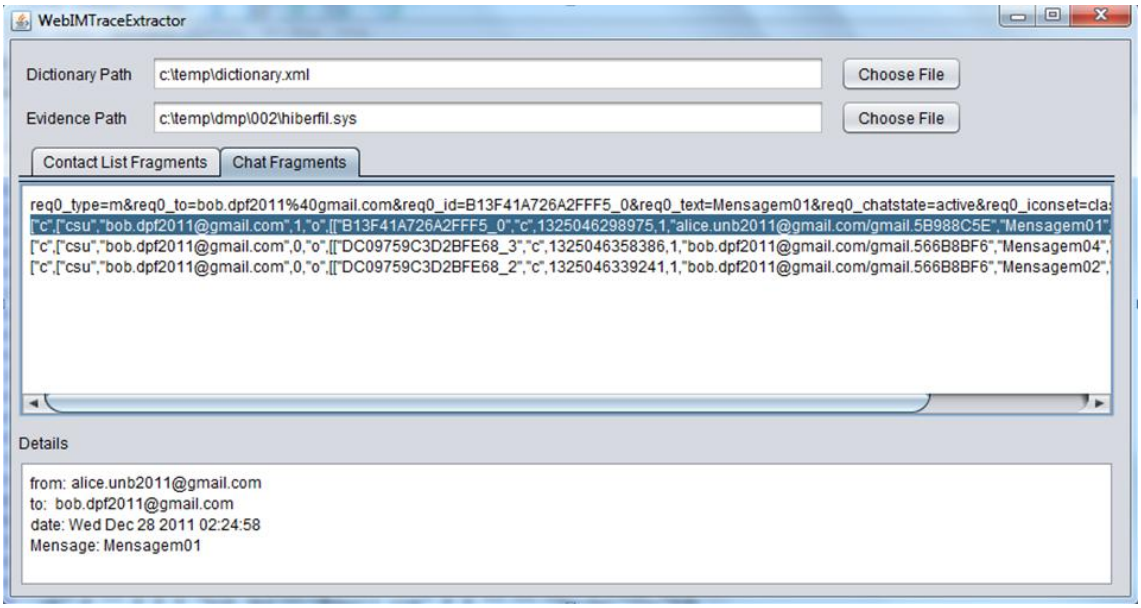

Figura 5.3 – Extração de conversas realizadas - *Gtalk*

5.2 CASO 02: *WINDOWS* LIVE MESSENGER – WLM

Conforme verificado nos testes da seção anterior com o G*talk*, a configuração de cifragem de sessão também pode ser habilitada ou desabilitada pelo usuário nas comunicações com o WLM. Nesses experimentos foram utilizados os usuários do WLM para identificar o padrão de cadeia de caracteres produzidos e compará-los com os identificados anteriormente no estudo de caso do G*talk*, agregando novas palavras-chave para identificar uma conversação.

5.2.1 Identificação de palavras-chave

De forma similar às simulações realizadas com o *Gtalk*, formatações diferentes são encontradas para mensagens enviadas ou recebidas e mais de um formato pode ser reconhecido para uma mesma mensagem. A Tabela 5.4 resume exemplos da identificação de atributos da conversação:

Tabela 5.4 – Padrões gerados pelo Windows Live Messenger

Lista de Contatos:
Não foram encontrados nomes de usuários da lista de contato de Alice, quando realizado *login* com seus usuários *off-line*. No entanto, quando os usuários da sua lista realizam *login*, notificações do protocolo são enviadas associando-se Alice ao nome do contato que foi logado:

HTTP/1.1 200 OK Content-Length: 258Content-Type: text/htmlX-MSN-Messenger: essionID=1523997507.933729243; GW-IP=65.54.61.210X-MSNSERVER: SN1MSG3030116X-MSN-ost: SN1MSG3030116.gateway.messenger.live.comDate: Tue, 14 Feb 2012 03:41:07 GMT **NFY PUT** 45Routing: 1.0To: 1:alice.unb2011@live.com From: 1:bob.dpf2011@live.com

NFY é comando utilizado pelo protocolo do WLM MSNP[32] para enviar notificações de *status* dos usuários. É útil para agregar usuários à lista de contatos.

Mensagens Enviadas para Bob:
SDG 14 292Routing: 1.0 To: 1:bob.dpf2011@live.com From: alice.unb2011@live.com; epid={6ed62f90-0bb0-421a-ab7c-679988dfb436}Reliability: .0Messaging: 2.0Content-Type: text/plain; charset=UTF-8Message-Type: TextX-MMS-M-Format: FN=Segoe%20UI; EF=; CO=0Content-Length: 10 **Mensagem01**

"sender":{"address":"alice.unb2011@live.com","type":1,"roleLists":0,"cid":"-5675366045147591562","id":"alice.unb2011@live.com","isFan":false},"timestamp":{"$date":1329194424064},"id":null,"isOfflineMessage":false,"isHistoryMessage":false,"info":{"text":"**Mensagem01",**

Mensagens Recebidas de Bob:
"sender":{"address":"bob.dpf2011@live.com","type":1,"roleLists":0,"cid":"","id":"bob.dpf2011@live.com","isFan":false},"timestamp":{"$date":1325094465000},"id":"1329194465880","isOfflineMessage":false,"isHistoryMessage":false,"info":{"text":"**Mensagem04"**

[32] MSNP - http://code.google.com/p/msnp-sharp/wiki/KB_MSNP21

Exemplo de Palavras-chave Identificadas:
Lista de contatos: **NFY**
Mensagens enviadas: **SDG + caracter espaço EF=;**
Mensagens enviadas ou recebidas: **"sender":{"address":"**

5.2.2 Análises

A lista de usuários pode ser vinculada a Alice quando usuários realizam *login* posteriormente ao *login* de Alice.

As mensagens podem ser enviadas e recebidas com formatos diferentes, no entanto, nem todos os formatos identificam o destinatário.

A cadeia de caracteres EF=; pode se utilizada para extrair mensagens enviadas, porém as conversões terão que considerar *offset* anteriores do fragmento.

Foram identificadas cadeias de caracteres que também são utilizadas pelo protocolo de comunicação nas versões *program-based* do comunicador.

Foram identificados na análise de tráfego de rede objetos JSON utilizados nas requisições *web*. A Figura 5.4 ilustra a decomposição do objeto trafegado, contendo nomes dos usuários envolvidos na comunicação:

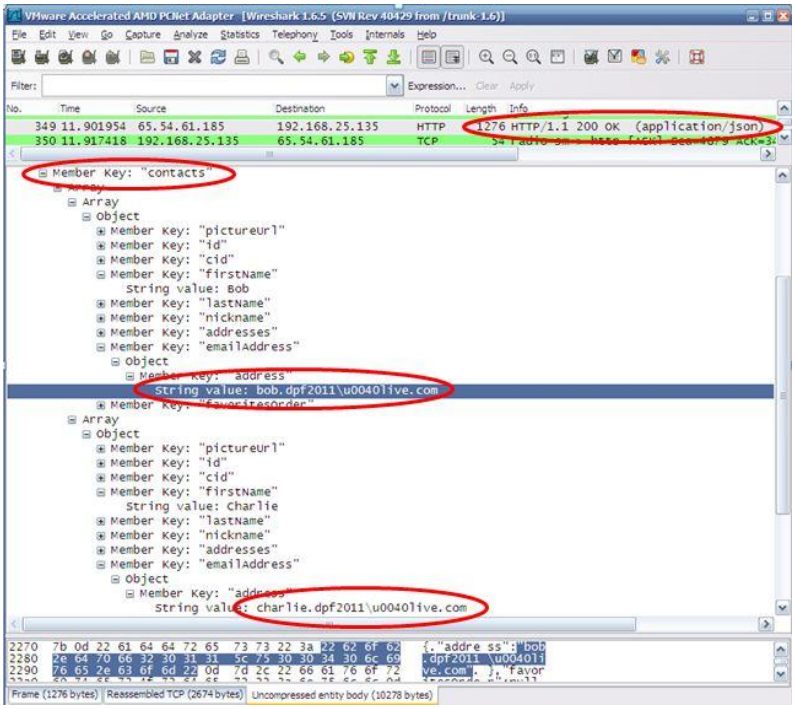

Figura 5.4 – Lista de contatos codificada em requisições com objetos JSON

5.2.3 Extração de vestígios

A Figura 5.5 ilustra extrações automatizadas de conversações realizadas pelo protótipo a partir do dicionário contendo palavras-chave do dicionário criado na fase de identificação de vestígios. Para demonstração foi utilizado o arquivo de paginação do *Windows*, *pagefile.sys*.

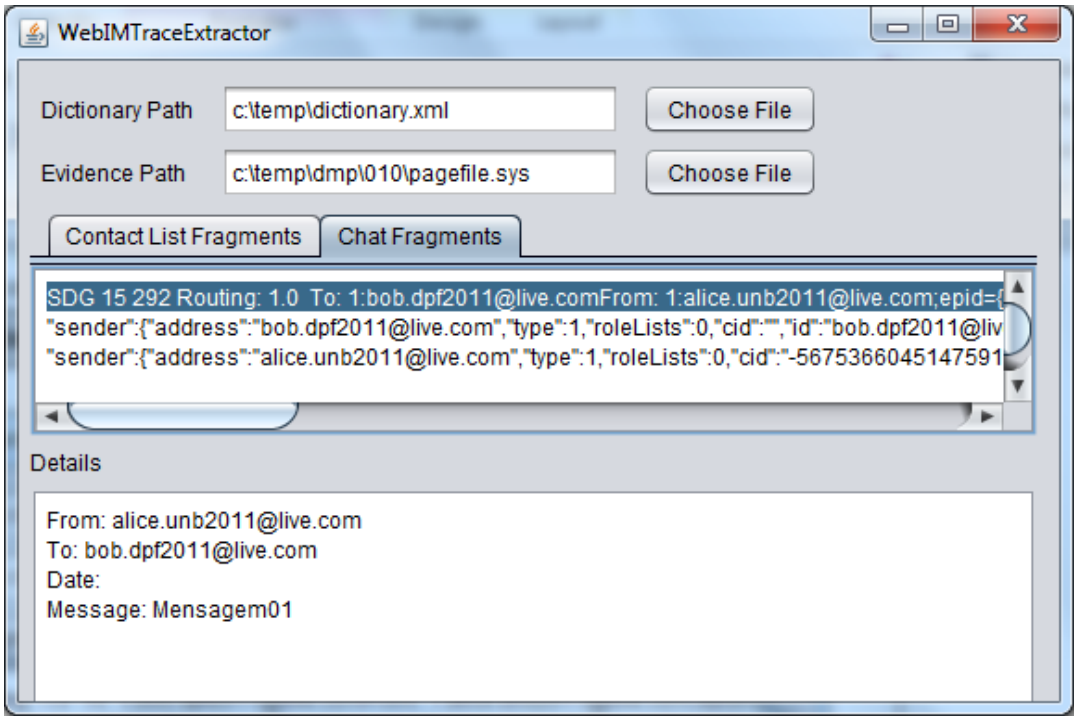

Figura 5.5 – Extração de conversas realizadas – WLM

A fim de ressaltar que nem todos os atributos da comunicação (origem, destino, data e mensagem) estão presentes num determinado fragmento extraído, foi selecionado um fragmento onde a data da comunicação não existe. Outros fragmentos, no entanto, puderam ser extraídos com todos os atributos envolvidos na comunicação.

A completude dos atributos em determinado fragmento depende da palavra-chave utilizada para se realizar a extração e da degradação dos dados nos arquivos manipulados pelo sistema operacional.

5.3 CASO 03: YAHOO! MESSENGER

Seguindo os procedimentos aplicados nos testes anteriores com o G*talk* e *WLM*, foram realizados testes com os usuários do *Yahoo!Messenger* para identificar o padrão de cadeia de caracteres produzidos e compará-los com os identificados, agregando novas palavras-chave que podem ser utilizadas para identificar uma conversação.

5.3.1 Identificação de palavras-chave

De forma similar às simulações descritas nas seções anteriores, formatações diferentes foram encontradas para mensagens enviadas ou recebidas. Mais de um formato pode ser reconhecido para uma mesma mensagem. A Tabela 5.5 resume exemplos da identificação de atributos de conversação:

Tabela 5.5 – Padrões gerados pelo *Yahoo!Messenger*

Lista de Contatos:

pushchannel/alice.unb2011

{ "buddyInfo" : { "sequence" : 0, "contact" : [{ "sender" : "charlie.dpf2011" , "presenceState" : 0, "avatarUser" : 0, "avatarPreference" : "0" , "clientCapabilities" : 8915971, "clientUserGUID" : "GAHSOWL2KPZCRSBRYHFT2NAOKE" } , { "sender" : "bob.dpf2011" , "presenceState" : 0, "avatarUser" : 0, "avatarPreference" : "2" , "clientCapabilities" : 8915971, "clientUserGUID" : "R7FBF3DT2U4QUIOIKRFCF243V4" }] } }

{ "@mpopState" : 0, "@pendingMsg" : 0, "@syncStatus" : 0, **"responses" : [{ "typing" :** { "sequence" : 6, "sender" : "bob.dpf2011" , "receiver" : "alice.unb2011" , "activity" : 1, "network" : "yahoo" } }
] }

Mensagens Enviadas para Bob:

{"convId":"alice.unb2011~bob.dpf2011~yahoo","txnId":"alice.unb2011~1325202709131~4","message" :"<font face=\"Arial\" size=\"10\">Mensagem01</

{"convId":"alice.unb2011~bob.dpf2011~yahoo","txnId":"alice.unb2011~1325202743423~4","message" :"<font face=\"Arial\" size=\"10\">Mensagem03</

Mensagens Recebidas de Bob:

{ "@mpopState" : 0, "@pendingMsg" : 0, "@syncStatus" : 0, **"responses" : [{ "message"** : { "status" : 1, "sequence" : 7, "sender" : "bob.dpf2011" , "receiver" : "alice.unb2011" , "msg" : "<font face=\"Arial\" size=\"10\">Mensagem02<\/font>" , "timeStamp" : 1325202730, "hash" :

Exemplo de Palavras-chave Identificadas:

Lista de contatos: **pushchannel/** e **{ "buddyInfo" : { "**

Mensagens enviadas: **{"convId":"**

Mensagens recebidas: **[{ "message" : {**

5.3.2 Análises

A lista de usuários pode ser vinculada a Alice quando usuários realizam *login* posteriormente ou quando realizam eventos de conversas com Alice. De forma similar ao *Gtalk* não é possível associar um usuário a lista identificada, caso haja mais de um usuário associado ao atributo pushchannel/.

As mensagens são enviadas e recebidas com formatos diferentes, no entanto, nem todos os formatos identificam o destinatário.

As mensagens enviadas contêm nomes de usuários, no entanto, o domínio a que pertencem se encontra em *offset* não adjacente ao nome identificado e não seguido do caractere @, como por exemplo: "alice.unb2011~bob.dpf2011~yahoo". Nesse caso, é possível associar o domínio "yahoo" aos usuários.

5.3.3 Extração de vestígios

As Figuras Figura **5.6** e Figura **5.7** ilustram, respectivamente, extrações automatizadas de listas de contatos e conversações realizadas pelo protótipo a partir do dicionário contendo palavras-chave identificadas na fase de identificação de vestígios. Para demonstração foi utilizado arquivo contendo despejo de memória capturado após finalização das conversas simuladas.

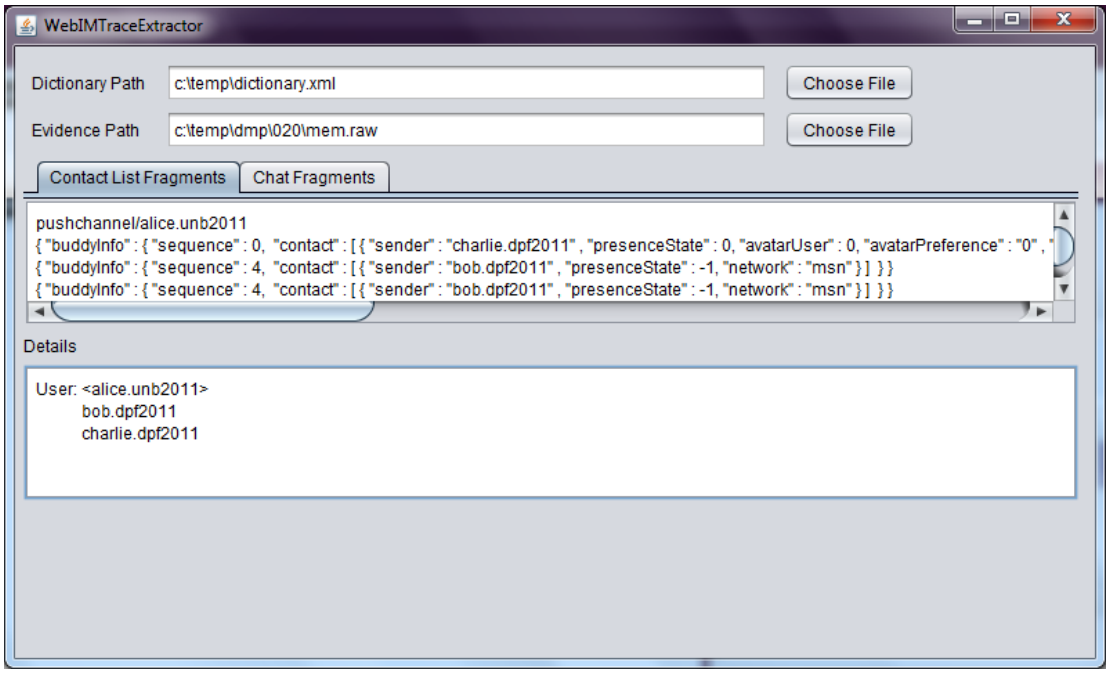

Figura 5.6 – Extração de listas de contatos - *Yahoo!Messenger*

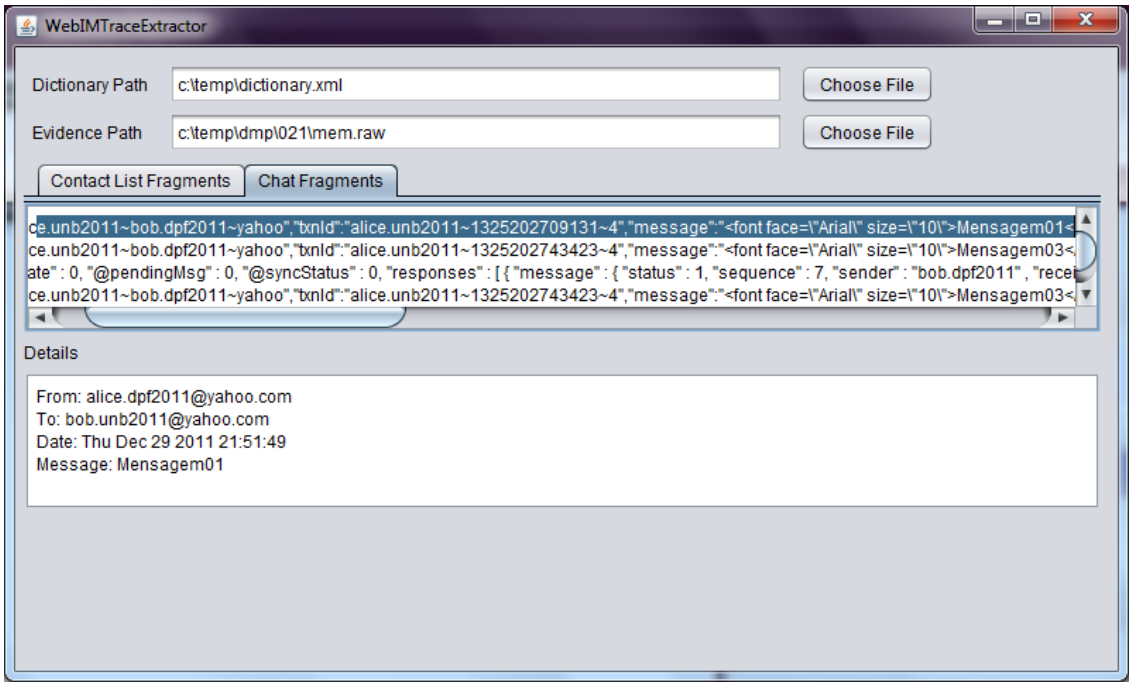

Figura 5.7 – Extração de conversas realizadas - *Yahoo!Messenger*

Devido à característica do comunicador de não associar o domínio ao nome de usuários por meio do caractere “@”, o protótipo foi programado para identificar o domínio a que pertencem os usuários a partir da informação contida após o nome dos usuários envolvidos por meio do caracter “~”.

5.4 CASO 04: FACEBOOK

O *Facebook* permite que seus usuários se conectem a um programa de bate-papo onde estão disponíveis para troca de mensagens instantâneas contatos pertencentes a clientes de diferentes *IM*.

As requisições *http* têm formato padrão diferente das originais ilustradas nos estudos de casos anteriores, ainda que o destinatário da mensagem esteja conectado diretamente ao serviço originário de *IM*.

Essa característica é estendida ao comportamento de outras interfaces concentradoras de clientes de *IMs*, como por exemplo, *meebo* e *ebuddy*, usados também para integrar em uma única interface contatos referentes aos comunicadores de *IMs* diversos.

O cenário deste estudo de caso difere um pouco dos anteriores, na medida em que foram simuladas conversas entre usuários conectados em serviços de mensagens instantâneas distintos: Alice conectada ao bate-papo do *Facebook* e Bob *on-line* no bate-papo do *Yahoo!Messenger*. Ambos participaram da mesma lista de contatos compartilhada para os dois clientes de *IM* e dessa forma podem se comunicar.

5.4.1 Identificação de palavras-chave

Diferentemente dos estudos de caso anteriores, não foram encontrados para o comunicador formatos distintos para mensagens enviadas ou recebidas. A Tabela 5.6 mostra o formato adotado nas requisições *http* provenientes do bate-papo do *Facebook* e do *Yahoo!Menssenger*, obtidos utilizando-se a metodologia proposta.

Tabela 5.6 – Padrões gerados pelo *Facebook*

Exemplo de padrões identificados
Lista de Contatos: **Lista de contatos de Bob** \"InitialChatUserInfos\",[],{\"100003508664692\":{\"name\":\"Bob Dpf\",\"firstName\":\"Bob\",\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/static-ak\\\/rsrc.php\\\/v1\\\/yo\\\/r\\\/UlIqmHJn-SK.gif\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003508664692\",\"gender\":2,\"**type\":\"user**\",\"exist\":true,\"showVideoSheet\":false},\"100003516971476\":{\"name\":\"Alice Unb\",\"firstName\":\"Alice\",\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/hprofile-ak-snc4\\\/161664_100003516971476_505534554_q.jpg\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003516971476\",\"gender\":1,\"**type\":\"friend**\",\"exist\":true,\"showVideoSheet\":false

Lista de contatos de Alice

\"InitialChatUserInfos\",[],{\"100003516971476\":{\"name\":\"Alice Unb\",\"firstName\":\"Alice\",\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/hprofile-ak-snc4\\\/161664_100003516971476_505534554_q.jpg\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003516971476\",\"gender\":1,\"type\":\"user\",\"exist\":true,\"showVideoSheet\":false},\"100003508664692\":{\"name\":\"Bob Dpf\",\"firstName\":\"Bob\",\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/static-ak\\\/rsrc.php\\\/v1\\\/yo\\\/r\\\/UlIqmHJn-SK.gif\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003508664692\",\"gender\":2,\"type\":\"friend\",\"exist\":true,\"showVideoSheet\":false}});\n__c(\"ChannelConnection

Mensagens

Máquina de Alice (Facebook):

Mensagens Enviadas para Bob:

for(;;);{"t":"msg","seq":8,"ms":**[{"msg":{"text":**"**Mensagem01**","time":**1329668239814**,"clientTime":1329668237825,"msgID":"1329668227085:3737843392","coordinates":null,"messageId":"id.112179775577226"},"from":100003516971476,"to":100003508664692,"window_id":1700697610,"from_name":"**Alice Unb**","from_gender":1,"sender_offline":false,"to_name":"**Bob Dpf**","to_gender":2,"tab_type":"friend","type":"msg"}

{"history":[{"from":100003516971476,"to":100003508664692,"time":1329889748861,"msgId":"1329889743648:62807842","window_id":"1966794921",**"msg":{"text":**"**Mensagem01**","messageId":"id.327963310589558"},"type":"msg"},{"from":100003508664692,"to":100003516971476,"time":1329889773211,"msgId":"351985681","window_id":null,**"msg":{"text":**"**Mensagem02**","messageId":"id.237413103018952"},"type":"msg"}],"userInfo":{"name":"Bob Dpf","firstName":"Bob

Mensagens Recebidas de Bob:

for(;;);{"t":"msg","seq":15,"ms"**:[{"msg":{"text":**"**Mensagem02**","time":**1329668883288**,"clientTime":1329668882269,"msgID":"1329668878428:1197592284","coordinates":null,"messageId":"id.116296691829066"},"from":100003508664692,"to":100003516971476,"window_id":3537633229,"from_name":"Bob Dpf","from_gender":2,"sender_offline":false,"to_name":"Alice Unb","to_gender":1,"tab_type":"friend","show_orca_callout":false,"type":"msg"}

Máquina de Bob (Yahoo!Messenger):

Mensagens Enviadas por Bob

{"convId":"bob.dpf2011~-100003516971476@chat.facebook.com~facebook",
"txnId":"bob.dpf2011~1329889758608~3","message":"Mensagem02","sendAs":"bob.dpf2011"}

Mensagens Recebidas de Alice

{ "@mpopState" : 0, "@pendingMsg" : 0, "@syncStatus" : 0, "responses" : **[{ "message" : {** "status" : 1, "sequence" : 11, "sender" : "-100003516971476@chat.facebook.com" , "receiver" : "bob.dpf2011" , "msg" : "Mensagem01" , "timeStamp" : 1329889750, "network" : "facebook" , "hash" : "I5WCO0xikH0vml+LQZGTayXLZVScfA==" , "msgContext" : "I5WCO0xikH0vml+LQZGTayXLZVScfA==" , "displayName" : "Alice Unb" } }
] }

Exemplo de Palavras-chave Identificadas:

Lista de contatos: **\"InitialChatUserInfos\"**

Mensagens enviadas ou recebidas: **[{"msg":{"text":**

Histórico de Mensagens enviadas e recebidas **{"history":[{"from":**

5.4.2 Análises

É possível associar sempre o usuário conectado a sua lista de contatos, ao contrário dos exemplos anteriores. Após a palavra-chave identificada \"InitialChatUserInfos\" o primeiro usuário listado é o proprietário da lista, com atributo específico que o identifica (type\":\"user), diferente dos demais contatos (type\":\"friend);

As mensagens são enviadas e recebidas com requisições *http* que respeitam o mesmo formato, identificando sempre os códigos identificadores de usuário (*ids*) e nomes de origem e destino.

A recuperação de mensagens se mostrou mais eficiente para esse comunicador devido a maior completude dos atributos existentes nas solicitações, diferentemente do observado nos outros estudos de caso.

Foram encontradas mais de uma palavra-chave para recuperação de conversas. Essa característica também foi observada nos demais estudos de caso realizados. No entanto, a completude dos atributos não depende da palavra-chave utilizada na extração dos dados, uma vez que todos os atributos considerados na comunicação (origem, destino, data e mensagem) estão presentes nos fragmentos recuperados.

5.4.3 Extração de vestígios

As Figuras Figura **5.8** e Figura **5.9** ilustram, respectivamente, extrações automatizadas de listas de contatos e conversações realizadas pelo protótipo a partir do dicionário contendo palavras-chave identificadas na fase de identificação de vestígios. Para demonstração foi utilizado arquivo de hibernação do *Windows* em formato descompactado.

Foi verificado que os usuários envolvidos na comunicação são identificados por um código (*id*). No entanto, é possível associar a esses códigos os nomes dos usuários, uma vez que estão presentes nos *offsets* do fragmento recuperados. O protótipo foi programado para concatenar os nomes de usuários aos seus respectivos códigos de identificação.

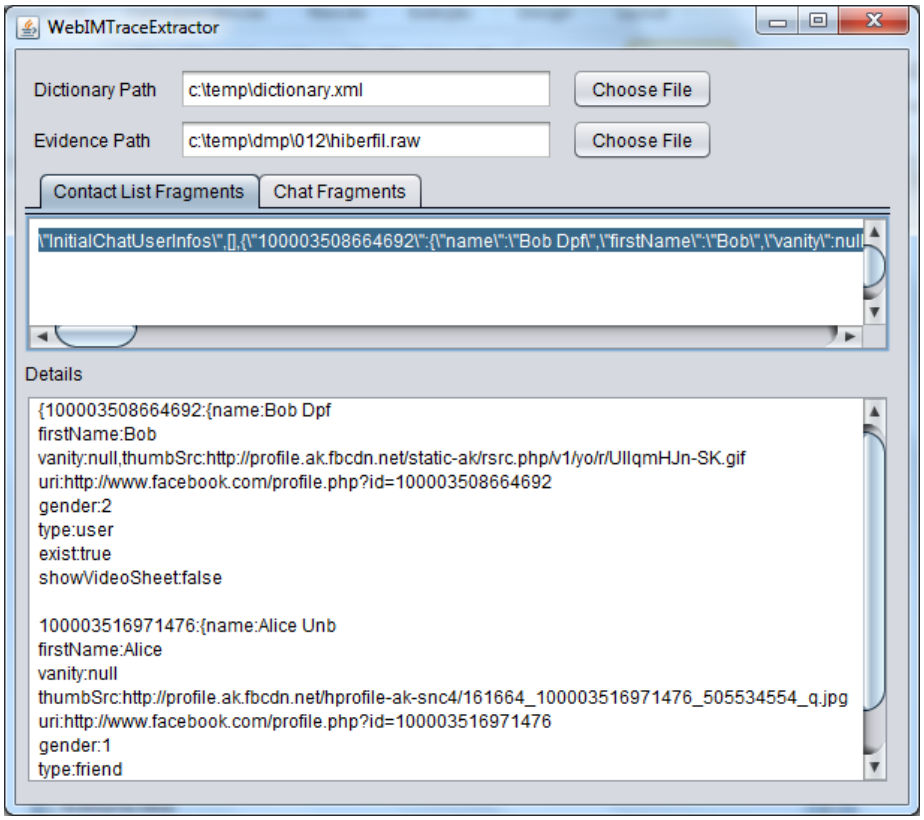

Figura 5.8 – Extração de listas de contatos - *Facebook*

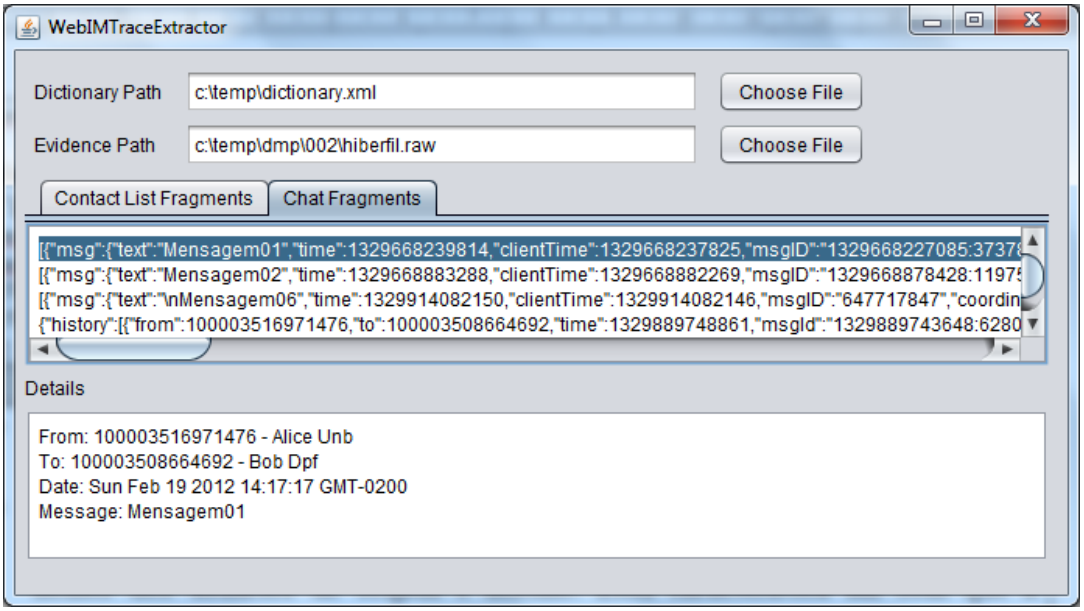

Figura 5.9 – Extração de conversas de bate-papo realizadas - *Facebook*.

Este capítulo mostrou a realização de estudos de casos, considerando a aplicação da metodologia proposta em simulações de conversas realizadas. Foram utilizadas as versões *web* de quatro comunicadores instantâneos, demonstrando fragmentos de conversas identificados e extraídos por meio do protótipo. No capítulo seguinte serão apresentados os resultados e as análises consolidadas.

6 RESULTADOS E ANÁLISES

O presente capítulo condensa os resultados obtidos dos estudos de caso realizados. São apresentados o dicionário com a lista de palavras-chave para recuperação de conversas, bem como os resultados das análises quantitativa e qualitativa dessas palavras, de acordo com os vestígios recuperados por cada uma.

Observou-se que as mensagens trocadas a partir dos navegadores de Internet permanecem remanescentes nos artefatos forenses por períodos indeterminados de tempo. Os resultados mostraram que foi possível recuperar mensagens, mesmo após dias as conversas terem ocorrido.

Verificou-se que as conversas não são armazenadas em arquivos estruturados, não podendo ser recuperadas por meio de buscas por assinaturas, procedimento adotado para análise de arquivos de despejos de memória em busca de estrutura de dados conhecidas.

Os procedimentos propostos demonstraram a possibilidade de se recuperar os vestígios provenientes das conversas simuladas a partir da identificação de traços do protocolo utilizados pelos comunicadores, sem a necessidade de se realizar estudo aprofundado de cada protocolo utilizado nas simulações. Esses traços foram encontrados nos arquivos de despejos de memória, paginação e hibernação.

As simulações das conversas mostram que é possível identificar, a partir da análise do tráfego de rede e captura de memória, cadeias de caracteres deixadas pelos comunicadores em diversos artefatos forenses. Essas cadeias podem ser utilizadas para recuperação de eventos relacionados ao uso dos *IMs Web-Based*. A aplicação da metodologia proposta identificou diferentes codificações de caracteres ao repetir as simulações com diferentes navegadores.

No entanto a extração das conversas a partir dos artefatos forenses referentes à memória de Sistemas operacionais *Windows* ou *Linux* podem ser realizadas com os mesmos procedimentos.

Foi verificado nos ambientes de testes que os *IMs* deixam no disco vestígios com o mesmo formato independente das versões dos sistemas operacionais testados. O formato de armazenamento dos caracteres, no entanto, pode variar a depender do navegador utilizado, como demonstrado no estudo de caso 01 apresentado na seção 5.1. As Figuras Figura **6.1** e 6.2 ilustram o mesmo formato de armazenamento de trechos de mensagens instantâneas recuperadas da área de paginação do sistema OpenSuse *Linux* 11.2 e do arquivo de paginação do *Windows* XP.

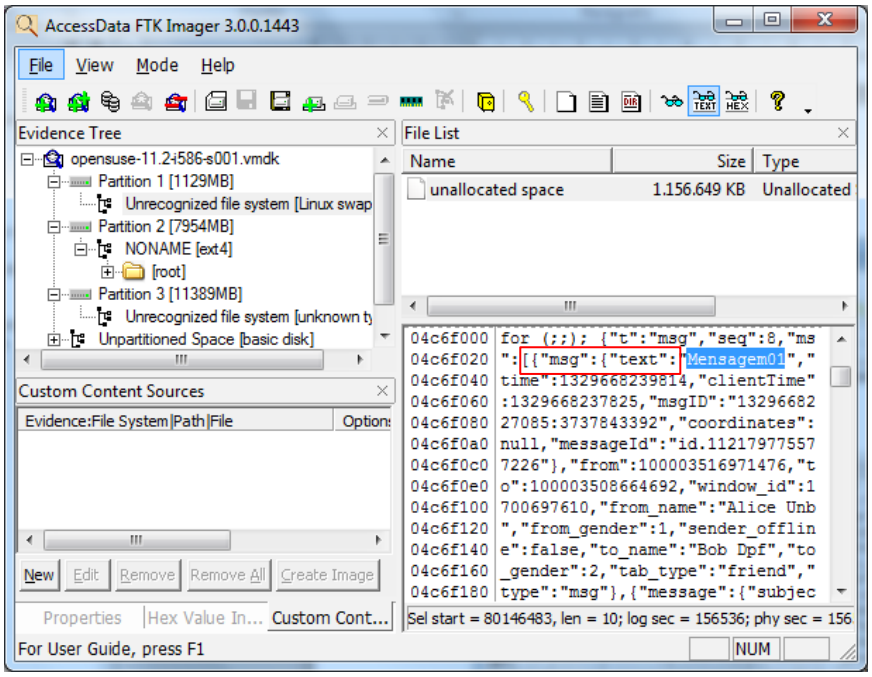

Figura 6.1 – Fragmento extraído da partição de swap do OpenSuse *Linux* 11.2

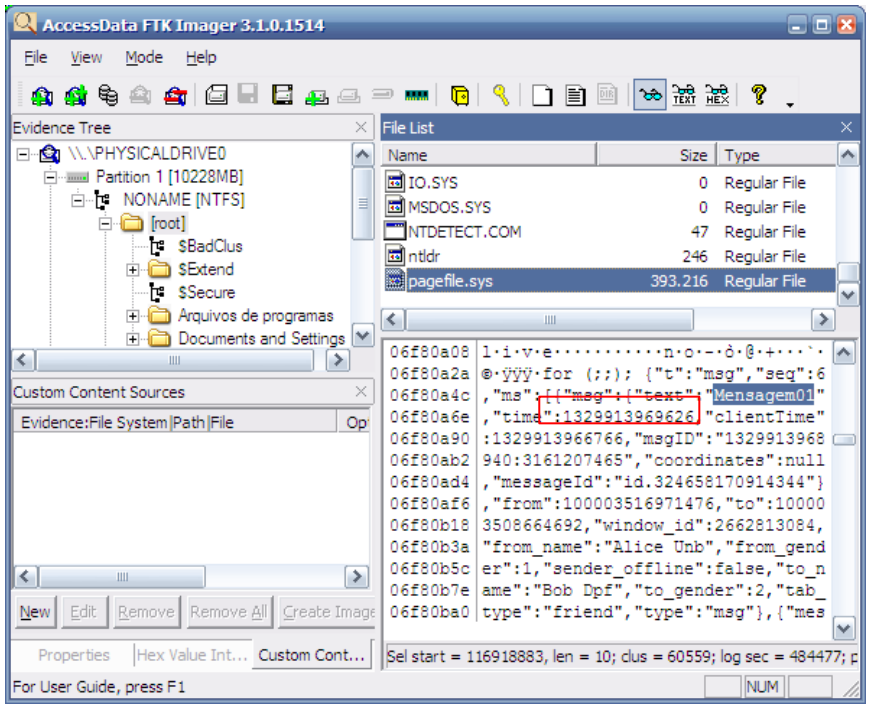

Figura 6.2 – Fragmento extraído do arquivo de paginação do *Windows* XP Sp2

6.1 RECUPERAÇÃO DE LISTAS DE CONTATO

Para se recuperar as listas de contato, é necessário associar o usuário conectado a cada contato encontrado. No entanto, foi identificada uma limitação para se realizar esta associação para os comunicadores *Windows Live Messenger*, *Yahoo!Messenger* e *Gtalk*, pois palavras-chave diferentes são utilizadas para recuperar o usuário conectado e os diversos contatos encontrados. Essa associação é possível apenas, se somente um usuário conectado for encontrado nas extrações. Assim, são relacionadas as seguintes implicações:

- Nem sempre será possível associar os contatos encontrados a um determinado usuário de *IM*. Essa possibilidade foi levada em consideração na definição do dicionário utilizado para se extrair as listas de contatos;
- Nem sempre é possível identificar o estado *on-line/off-line* dos usuários da lista de contatos no momento de realização do *login*. Essa característica é intrínseca a definição de variáveis existentes no protocolo;
- O carregamento da lista de contatos do *Facebook* sempre identificou o proprietário da lista, discriminando inclusive o proprietário e os contatos por um atributo específico.

6.2 RECUPERAÇÃO DE CONVERSAS DE MENSAGENS INSTANTÂNEAS

Na recuperação dos fragmentos foi considerada a existência dos atributos: origem, destino, data e mensagem. Verificou-se que sem sempre todos os atributos eram recuperados. A recuperação do conjunto de atributos depende da palavra-chave utilizada para extração. Esses atributos podem não estar presentes em um mesmo fragmento extraído. A completude desses atributos depende do tipo de requisição utilizada a partir de diferentes eventos gerados na janela de conversação. Por exemplo, ao se passar o mouse sobre a janela, requisições são realizadas para anunciar a presença ou a atividade do usuário conectado ao bate-papo. Essas requisições independem do usuário e compõe a maior parte do tráfego de rede.

Existe possibilidade de mais de uma palavra-chave identificar um mesmo evento, mensagem enviada ou recebida. Essa característica possibilita recuperação de mais fragmentos frente à degradação dos dados. Porém, tem a desvantagem de se recuperar conversas repetidas encontradas em *offsets* diferentes num mesmo artefato analisado.

6.3 DICIONÁRIO DE PALAVRAS-CHAVE

As palavras-chave candidatas identificadas nos estudos de caso foram documentadas em um dicionário, discriminando-se para cada *IM* testado, o tipo de vestígio que pode ser recuperado. O dicionário de palavras-chave identificadas para cada um dos eventos simulados é discriminado na Tabela 6.1:

Tabela 6.1 – Dicionário de palavras-chave

IM	Palavra-chave	Vestígio recuperado	Atributos Presentes	Atributos Ausentes
	["c",["r",	Início da lista de contatos. Cada contato é separado por [0,"	- Contatos	Proprietário -
	gmailchat=	Usuário que se conecta ao bate-papo na máquina local	Proprietário -	- Contatos
	["c",["csu",	Mensagens enviadas ou recebidas	Origem Destino Mensagem Data	- - - -
	["c",["m",	Mensagens recebidas e estado dos usuários que as enviaram: ativo, inativo, digitando...	Origem, - Mensagem Data	- Destino - -
	["c",["e",	Mensagens enviadas pelo usuário conectado à máquina	- Destino, Mensagem Data	Origem - - -
	req0_type=m	Mensagens enviadas pelo usuário conectado à máquina local. Usar expressão regular para substituir o 0	- Destino, Mensagem -	Origem - - Data
	"sender":{"address":	Mensagens enviadas ou recebidas	Origem, - Mensagem Data	- Destino - -
	SDG (SDG mais o caractere espaço)	Bate-Papo entre usuários	Origem Destino Mensagem Data	- - - -
	EF=;	Bate-Papo entre usuários	Origem Destino Mensagem Data	- - - -

	NFY	Lista de contatos. Apenas quando usuários são notificados de eventos	Proprietário Contato	- -
	/pushchannel/	Usuário que se conecta ao bate-papo na máquina local	Proprietário -	- Contato
	{ "buddyInfo" : {	Usuários de Lista de Contatos	- Contato	Proprietário -
	[{ "message" : {	Mensagens recebidas	Origem Destino Mensagem Data	- - - -
	{"convId":"	Mensagens enviadas	Origem Destino Mensagem Data	- - - -
	[{"msg":{"text":	Mensagens enviadas e recebidas	Origem Destino Mensagem Data	- - - -
	[{"history":[{"from":	Históricos de mensagens	Origem Destino Mensagem Data	- - - -
	\"InitialChatUserInfos\"	Lista de Contatos	Proprietário Contato	- -

Para cada palavra-chave discriminada foi extraído um exemplo de fragmento e identificados os atributos para a construção de *parsers* com a finalidade de se categorizar esses atributos.

Os exemplos de fragmentos encontrados estão relacionados no Anexo A – Formato dos Fragmentos.

6.4 ANÁLISE QUANTITATIVA E QUALITATIVA DOS VESTÍGIOS

Para cada comunicador testado, foram encontradas mais de uma palavra-chave referente ao contexto da troca de mensagem e carregamento de lista de usuários. Essa diversidade fornece alternativas para recuperar conversas, caso não sejam encontrados fragmentos com uma determinada palavra utilizada. A Figura 6.3 mostra comparação da quantidade de palavras que podem ser utilizadas para recuperar vestígios gerados pelos comunicadores testados:

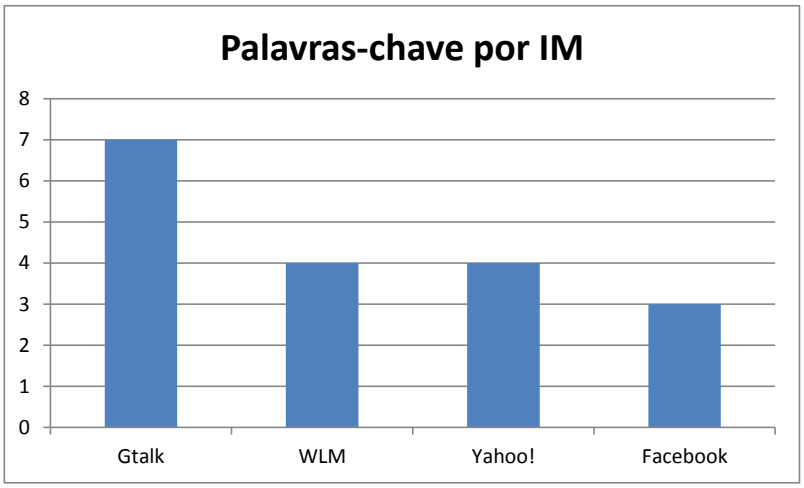

Figura 6.3 – Quantidade de palavras-chave identificadas por comunicador

No entanto, nem todas as palavras identificadas possibilitaram buscas por fragmentos contendo todos os atributos para identificar completamente os envolvidos numa conversação. Algumas palavras-chave do *Gtalk* não identificaram o remetente, por exemplo. Já as palavras provenientes do protocolo do *Facebook* recuperaram fragmentos com o conjunto mais completo de atributos. Foi possível identificar inclusive se o destinatário está conectado ao bate-papo externamente, sem utilizar a rede social. A Figura 6.4 mostra comparação da quantidade de atributos existentes nos fragmentos extraídos pelas palavras-chave de cada comunicador, levando em consideração a completude necessária para identificar uma conversa:

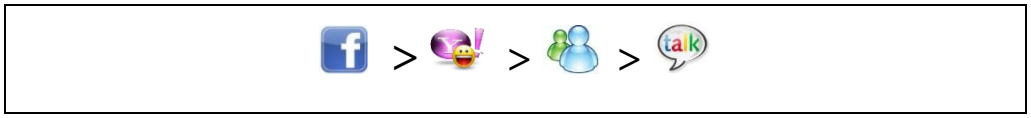

Figura 6.4 – Quantidade de atributos recuperados por conjunto de palavras-chave

O comparativo do número de atributos recuperados por palavra-chave identificada, considerando 04 (quatro) o número máximo de atributos para fragmentos contendo troca de mensagens e 02 (dois) para fragmentos contendo lista de contatos é demonstrado na Figura 6.5:

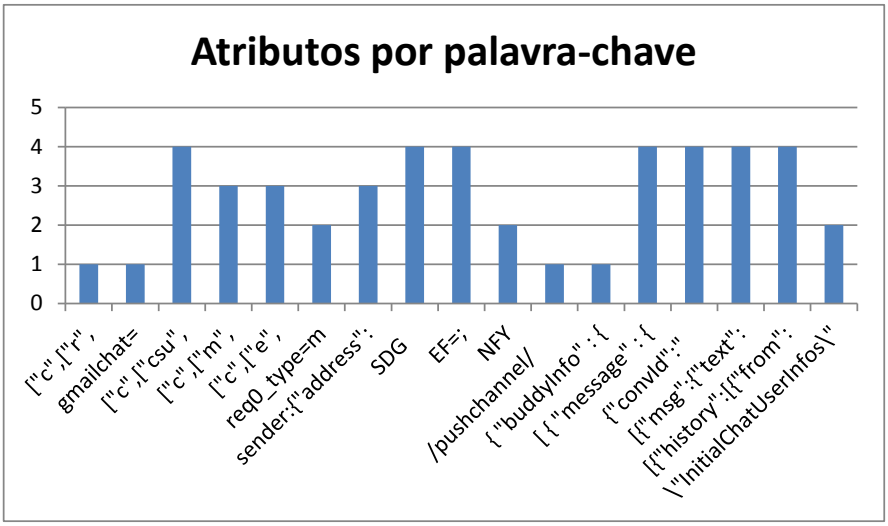

Figura 6.5 – Quantidade de atributos recuperados por palavra-chave

Foi realizada estimativa de percentual referente ao número de falsos positivos atribuído a cada palavra-chave identificada. O percentual foi estimado dividindo-se o número de ocorrências da palavra encontrada fora do contexto do dicionário, pelo número total de ocorrências encontradas, conforme detalhado na Tabela 6.2:

Tabela 6.2 – Fórmula para determinar percentual de falsos positivos

$$\% \text{ Falsos Positivos} = \frac{\text{Nr de ocorrências fora do contexto do dicionário}}{\text{Nr total de ocorrências encontradas}}$$

Foram analisados 03 (três) despejos de memória, contendo 04 (quatro) mensagens instantâneas trocadas entre Alice e Bob. As mensagens foram realizadas em navegadores com abas abertas acessando sítios diversos a partir dos sistemas operacionais *Windows* e OpenSuse *Linux.*

Todos os eventos relacionados ao dicionário de palavras-chave foram simulados. A Figura 6.6 ilustra a estimativa do percentual de falsos positivos encontrados:

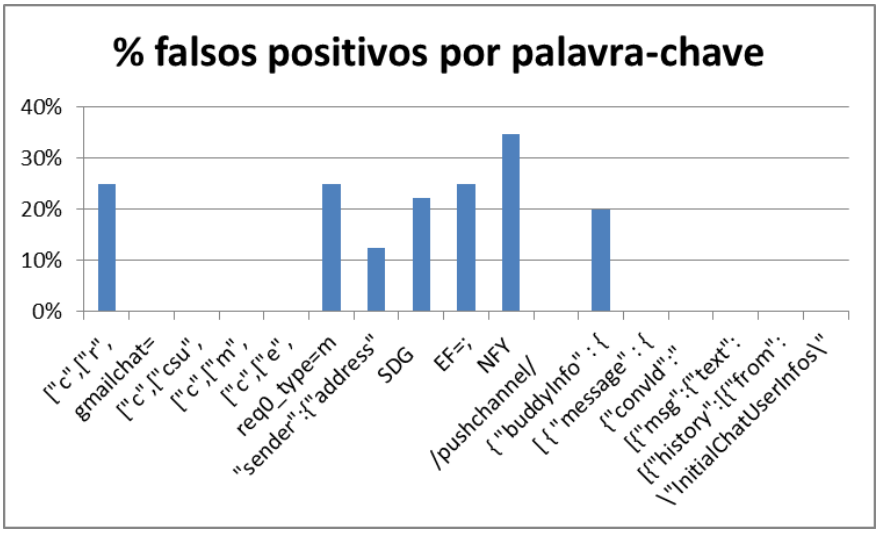

Figura 6.6 – Estimativa de falsos positivos nas simulações

Os percentuais de falsos positivos foram ocasionados em decorrência das palavras-chave no disco que não puderam ter seu contexto identificado e pela ocorrência em contexto diverso do definido no dicionário. Pode-se concluir que estimativas em relação à existência de palavras relacionadas a contextos diferentes é influenciada pela degradação dos dados.

Nesse capítulo foram consolidadas as observações dos estudos de caso realizados, destacando-se os resultados obtidos em relação à recuperação de listas de contatos e mensagens simuladas. Foi demonstrado o dicionário de palavras-chave criado e realizada análise comparativa dessas palavras em relação à completude dos atributos recuperados. Também foi destacada estimativa de falsos positivos em relação à identificação de fragmentos a partir das palavras contidas no dicionário, no entanto, que não correspondem ao contexto definido.

7 CONCLUSÕES

A recuperação de mensagens instantâneas realizadas em ambiente *web* tem sido limitada nas perícias com quesitos acerca da extração de conversas em discos rígidos apreendidos pela Polícia Federal. As técnicas existentes são voltadas para a extração de vestígios deixados pelas versões de programas instalados ou desconsideram a limitação do cache de navegação, devido à utilização de tecnologias como Ajax pelos navegadores.

O trabalho realizado demonstra que é possível recuperar vestígios de conversações deixados por *IMs web-based*, a partir das análises de discos rígidos, sem a necessidade de um estudo detalhado dos protocolos de comunicação utilizado pelos mesmos, como alternativa à limitação das técnicas existentes. O problema apresentado na seção 1.1 pode ser resolvido como demonstrado nos capítulos 4 e 5 em que foi apresentada uma metodologia que define procedimentos para identificar padrões de cadeias de caracteres utilizados pelos comunicadores, com base na análise dos artefatos coletados dos despejos de memória, da memória virtual, dos arquivos de hibernação e tráfego de rede. Esses padrões servem como marcadores existentes nos protocolos de troca de mensagens dos *IMs web-based* e são utilizados como palavras-chave ou expressões regulares, com a finalidade de permitir buscas automáticas de conversas de usuários, a partir dos artefatos forenses identificados. Para isso, foi desenvolvido um protótipo que é capaz de extrair fragmentos de conversas desses artefatos, realizando a conversão de formato dos vestígios encontrados e categorizando os atributos existentes da conversação, tais como, mensagem, usuário de origem, destino, data e usuários pertencentes a listas de contatos.

A metodologia proposta também permite identificar padrões utilizados pelos *IMs* mesmo que a comunicação entre usuários seja criptografada. Com o auxílio de um *proxy* de aplicação *web*, foi possível recuperar as mensagens trocadas entre *IMs web-based* e a memória, antes da submissão cifrada pelo navegador. Assim, pode-se estudar padrões de nomes de variáveis utilizadas em requisições *http* desenvolvidas para um determinado *IM*.

A análise de memória mostrou ser mais eficiente do que a análise dos pacotes de rede. Isso se deve pela possibilidade de se identificar cadeias de caracteres, mesmo quando a comunicação se dá de forma criptografada, e pela comparação direta com o conteúdo remanescente nos arquivos de hibernação e paginação. Conclusões semelhantes foram citadas por outros autores referenciados ao longo deste trabalho.

Análises de memória física e virtual necessitam de um ambiente estéril para repetição dos testes, uma vez que testes sucessíveis contaminam a memória com dados de testes anteriores.

A comparação dos fragmentos recuperados mostra que a quantidade de atributos extraídos de uma conversação não depende da quantidade de palavras-chave identificadas para um determinado *IM*, mas da completude dos atributos provenientes da requisição *web* em que está contida a palavra adaptada ao protocolo.

7.1 CONTRIBUIÇÕES

A recuperação de mensagens instantâneas realizadas em navegadores de Internet, a partir dos procedimentos definidos, representa uma limitação a menos nos ramos de investigação da Polícia Federal. É possível realizar exames com maior eficácia ao extrair também vestígios provenientes desse tipo de comunicação. Dessa forma, é possível responder aos quesitos relacionados à recuperação de conversas de uma forma geral, com uma análise mais completa dos discos rígidos apreendidos. Nesse sentido:

- Demonstraram-se procedimentos eficazes para a recuperação de mensagens instantâneas aplicados ao paradigma *web-based* de 04 (quatro) comunicadores: *WLM, Gtalk, Yahoo!Messenger e Facebook*. Esses procedimentos não dependem da análise de cache do navegador, contornando o problema das requisições realizadas por meio de implementações Ajax, que limitam a geração de cache de navegação em disco;
- Demonstrou-se um método para coleta e identificação de padrões utilizados pelas requisições *web* dos comunicadores aplicado a mais de um *IM web-based*. A solução foi apresentada frente à limitação das técnicas existentes, voltadas à recuperação de vestígios deixados por programas instalados;
- Definiu-se um dicionário para realização de buscas por palavras-chave e expressões regulares para detectar eventos remanescentes em discos rígidos gerados pelo uso de *web IM*;
- Demonstraram-se ambas as análises dos fluxos de rede e de memória para identificação de padrões de rastros do protocolo, contornando o problema da criptografia do fluxo de comunicação;

- Realizou-se a comparação dos traços do protocolo utilizado pelos comunicadores residentes na memória de sistemas operacionais *Windows* e *Linux*, demostrando a convergência do conteúdo existente na memória de ambos;
- Desenvolveu-se um protótipo para extração de vestígios, categorizando usuários de *IM web-based*, atributos de conversas e lista de contatos;
- Demonstrou-se a descompactação de arquivos de hibernação para permitir a realização de buscas mais eficientes nesse artefato forense.

7.2 LIMITAÇÕES

A metodologia utilizada recuperou a maior parte dos vestígios gerados nas simulações realizadas, no entanto foram verificadas limitações que não puderam ser superadas devido à natureza da geração dos vestígios por parte das próprias aplicações, que por vezes não vinculam dados trafegados ao usuário conectado:

- Não foi possível associar sempre uma lista de contatos a um usuário específico identificado nos artefatos analisados. Algumas simulações permitiram vincular a conta de Alice à lista de contatos apenas no momento do *login* dos usuários presentes na lista;
- Não foi possível associar sempre uma mensagem recebida ao usuário destinatário porque nem todos os atributos de uma conversação estão presentes nos fragmento gerados pelos *IMs*;
- O uso dos *sniffers* de rede e memória, *Wireshark* e *Charles web proxy*, respectivamente, não se mostraram eficazes na captura das mensagens de bate-papo recebidas por usuários do *Gtalk*, uma vez que não foram capazes de capturar os pacotes recebidos por esse *IM*. Para identificar o formato das mensagens recebidas, foram realizados despejos de memória, hibernações e forçado o processo de paginação, demandando-se o sistema configurado com pequena quantidade de RAM, para que se fossem analisados os conteúdos dos arquivos referentes a esses artefatos, logo em seguida ao evento de recebimento da mensagem;
- Os procedimentos definidos não são voltados para a análise dos caches dos navegadores. Porém, quando usuários requisitam históricos de conversas

armazenados nos servidores, essas páginas podem ser analisadas pelo protótipo, porém com prejuízo na interpretação de outros formatos não mapeados.

7.3 TRABALHOS FUTUROS

O escopo do presente trabalho incluiu o estudo eventos relacionados a listas de contatos e mensagens instantâneas trocadas entre usuários em ambiente *web*. É possível expandir a metodologia proposta para analisar os demais vestígios originados por esse tipo de comunicação, bem como automatizar etapas da metodologia sugerida. Assim, surge a oportunidade dos seguintes trabalhos futuros:

- A automatização da primeira etapa do método proposto, fase de identificação das palavras-chave utilizadas por um determinado comunicador. Para isso, um algoritmo necessita ser desenvolvido para reconhecimento de padrões que precedem às cadeias de caracteres conhecidas e utilizadas nas simulações de mensagens trocadas;
- O protótipo desenvolvido pode ser evoluído para que sejam construídos extratores na forma de *plug-ins*, para flexibilizar a sua utilização junto às interfaces de agregação de dados, buscando as conversas provenientes de outros comunicadores específicos e dar maior eficiência ao processo de extração;
- Expandir o método para identificar e recuperar arquivos trocados pelos usuários durante as sessões de comunicação;
- Realizar a identificação e recuperação de imagens dos perfis definidas pelos usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

ANSON, S.; BUNTING S. (2007). *Mastering Windows Network Forensics and Investigation*. Wiley Publishing, Inc. Indianópolis. United States.

BREZINSKI, D. ; Killalea, T. (2002). RFC 3227 – Guidelines for Evidence Collection and Archiving. 2002

China Internet Network Information Center (CNNIC). (2010). Statistical Report on Internet Development in China July 2010. Disponível em: <http://www.cnnic.net.cn/uploadfiles/pdf/2010/8/24/93145.pdf>. Acesso em 30 jan. 2011.

CARVEY. H. (2007). *Windows Forensic Analysis*. Syngress Publishing, Inc. Burlington. 2007. United States

CROCKFORD, D. (2011). Introducting Json. Disponível em <http://www.json.org>. Acesso em 21 nov. 2011.

DANKNER, S.; Kiley, M.; Rogers, M. (2008). Forensic Analysis of Volatile Instant Messaging. In: *IFIP International for Federation of Information Processing*, vol. 285, Advances in Digital Forensics IV, Ray, I.; Shenoi, S. Boston: Springer pp. 129-138.

DOGEN, W. V. (2007). Forensic artefacts left by Windows Live Messenger 8.0. *Digital Investigation, 4*(2), 73-87. doi:10.1016/j.diin.2007.06.019

ELEUTÉRIO, P. , ELEUTÉRIO, J. (2011). Webmail evidence recovery: a comparison among the most used *Web* browsers and webmail services. In: *Proceeding of the 6th International Conference on Forensic Computer Science (ICoFCS 2011),* Florianópolis, SC, Brazil. DOI: http://dx.doi.org/10.5769/C2011021

ELEUTÉRIO, P.; Machado, M. (2011). *Desvendando a Computação Forense*. Novatec. Brasil . ISBN 978-85-7522-260-7.

FARMER, Dan; VENEMA, Wietse. (2006*). Forensic Discovery*. Addison-Wesley. United States.

FEI, H., RUI, L., & LITING, H. (2009). Analysis and Characteristic at the Chat Session Level in Instant Message Traffic. In: *The 1st International Conference on Information Science and Engineering (ICISE2009)*, Nanjing

GAO, Y., & CAO, T. (2010). Memory Forensics for QQ from a Live System. In: *Journal of Computers*, *5*(4), 541-548. doi:10.4304/jcp.5.4.541-548

GARRETT, J. J. (2005). Ajax: A New Approach to *Web* Applications. Disponível em: <http://www.adaptivepath.com/ideas/ajax-new-approach-*web*-applications>. Acesso em 25 set. 2011

HUSAIN, M. I.; Sridhar R. (2009). iForensics: Forensic Analysis of Instant Messaging Smart Phones. In: *Proceedings of the International Conference on Digital Forensics & Cyber Crime (ICDF2C'09)*, Albany, United States.

JERÔNIMO, K. de Souza. (2011). Busca de conversas do MSN utilizando os softwares WMM e EnCase. In: *Proceeding of the 6th International Conference on Forensic Computer Science (ICoFCS 2011),* Florianópolis, SC, Brazil. DOI: http://dx.doi.org/10.5769/C2011017

LAPORTE, J. (2008). Billions Conected. Disponível em: <http://billionsconnected.com>. Acesso em 19 fev. 2011.

LEE, Seokhee, Savoldi A., Lee, Sangjin, Lim, J. (2007). *Windows* Pagefile Collection and Analysis for a Live Forensics Context. In: *Future Generation Communication and Communication and Networking (FGCN 2007),* doi: 10.1109/FGCN.2007.236, Seoul, Korea.

MACLEAN, N. P. (2006). "Acquisition and Analysis of *Windows* Memory" Dissertação de Mestrado, Universidade de Strathclyde, Glasgow.

MANDIA, K; Prosise, C; Pepe, M. (2003). *Incident Response and Computer Forensics*, McGraw-Hill Osborne Media.

MARSHALL, A. M. (2008). *Digital Forensics: Digital Evidence in Criminal Investigation*. Sybex. New Delhi, India.

MEDEIROS, M. H. F; SOUSA, G. B. (2009). Extração de vestígios do *Windows* Live Messenger 2009. In: *Proceeding of the 4th International Conference on Forensic Computer Science (ICoFCS 2009),* Natal, Brazil.

NOGUEIRA, J.H.M; Campello, R.S. (2006). *Informática Forense* Brasília: Academia Nacional de Polícia do Departamento de Polícia Federal.

NUNES, G. M. (2008). Instant Messaging Forensics. In: *Proceeding of the 3rd International Conference on Forensic Computer Science (ICoFCS 2008),* Rio de Janeiro, Brazil.

PARSONAGE ,H. (2008). Computer Forensics Miscellany. The Forensic Recovery of Instant Messages from MSN Messenger and Windows Live Messenger. Disponível em: <http://computerforensics.parsonage.co.uk/downloads/MSNandLiveMessengerArtefactsof Conversations.pdf>. Acesso em 25 fev. 2011.

PINGDOM. (2010) Amazing Facts and Figures about Instant Messaging Infographic. Disponível em: <http://royal.pingdom.com/2010/04/23/amazing-facts-and-figures-about-instant-messaging-infographic/>. Acesso em 19 fev. 2011.

RANDOW, K. (2011) Charles, Web Debugging Proxy Application. Disponível em: <http://www.charlesproxy.com> Acesso em 26 mar. 2011.

SCHATZ, B. (2007). BodySnatcher : Towards reliable volatile memory acquisition by software. *Digital Investigation*, *BIOS*, 126-134. doi:10.1016/j.diin.2007.06.009

SCHWEITZER, D. (2003). *Incident Response: Computer Forensics Toolkit*. Wiley Publishing, Inc. Indianópolis. United States.

SOLOMON, M. G., BARRET, D.; BROOM, N (2005). *Computer Forensics*. *Jump Start*. Sibex. United States.

SOUSA, G. B. (2011). Identificação e análise de vestígios deixados pelo Skype 5. X. In Proceeding of the 6th International Conference on Forensic Computer Science. (2011) Florianópolis, Brazil. DOI: http://dx.doi.org/10.5769/C2011005

SOUSA, G. B. (2008). WMM - Uma ferramenta de extração de vestígios deixados pelo *Windows* Live Messenger. In: *Proceeding of the 3rd International Conference on Forensic Computer Science (ICoFCS 2008)*, Rio de Janeiro, Brazil.

STEEL, C. (2006). *Windows Forensics The Field Guide for Conducting Corporate Computer Investigations*. Wiley Publishing, Inc. Indianópolis. United States.

OPENSUSE (2011). SDB:Suspend to disk. Dsponível em: <http://en.opensuse.org/SDB:Suspend_to_disk> Acesso em 24/out/2011.

TAIVALSAARI, A., Mikkonen, T., Ingalls, D.; Palacz, K. (2008). *Web* Browser as an Application Platform. *2008.* In: *34th Euromicro Conference Software Engineering and Advanced Applications*, *553*(January), 293-302. IEEE. doi:10.1109/SEAA.2008.17

VOLATILITY (2011). Volatility: An advanced memory forensics framework. Disponível em: <http://code.google.com/p/volatility/wiki/CommandReference#imagecopy>. Acesso em 31 ago. 2011

XIAO, Z.; GUO, L.; TRACEY, J. (2007.). Understanding Instant Messaging Traffic Characteristics. In: *27th International Conference on Distributed Computing Systems (ICDCS'07)*, Toronto, Canadá.

ZHAO, Q. ; CAO, T. (2009). Collecting Sensitive Information from Windows Physical Memory. *Journal of Computers*, *4*(1), 3-10. doi:10.4304/jcp.4.1.3-10

ANEXOS

A - FORMATO DOS FRAGMENTOS

Palavra-chave: ["c",["csu",

Palavra-chave Destino Data
["c",["csu","bob.dpf2011@gmail.com" ,0,"o",[["F879EE39E71B16E6_0","c",1328848273906,1,
"alice.unb2011@gmail.com/gmail.5B988C5E","Mensagem01
Origem Mensagem

Palavra-chave: gmailchat=

gmailchat= alice.unb2011@gmail.com/740119
Palavras-chave Proprietário

Palavra-chave: ["c",["r",

Palavras-chave Contatos da Lista
["c",["r",0,[[0,"bob.dpf2011@gmail.com","0","Bob Dpf",4,0,0,0,,"",0,2,2,0,1,"Bob
Dpf","Bob",0,"",0,0,0,"bob.dpf2011@gmail.com",0,0,"","","2cc4e1798a78db50",[],[]

,[0,"charlie.dpf2011@gmail.com","1","Charlie Dpf",0,0,0,0,,"",0,2,2,0,1,"Charlie
Dpf","Charlie",0,"",0,0,0,"charlie.dpf2011@gmail.com",0,0,"","","4da0946c89875ab6

,[0,"dave.dpf2011@gmail.com","2","Dave Dpf",0,0,0,0,,"",0,2,2,0,1,"Dave
Dpf","Dave",0,"",0,0,0,"dave.dpf2011@gmail.com",0,0,"","","594af7c80a7d40b9",[],[]

Palavra-chave: ["c",["e",

Palavras-chave Destino Mensagem Data
["c",["e","dave.dpf2011@gmail.com","F879E1B16E6_2","Mensagem05","Mensagem05",13288483442
37

Palavra-chave: req0_type=m

Palavra-chave Destino
req0_type=m&req0_to=bob.dpf2011%40gmail.com&req0_id=F4B37E2450825EDC_1&
req0_text=Mensagem03&req0_chatstate=active&req0_iconset=classic&req0__sc
Mensagem

Palavra-chave: ["c",["m",

Palavra-chave Origem Mensagem
["c",["m","charlie.dpf2011@gmail.com","97E68E1480EBF67F_0","active","Mensagem04","Mensagem
04",1328848330653,,,0,0,0,0,[]
Data

Palavra-chave: "sender":{"address":

Palavra-chave Origem
"sender":{"address":"alice.unb2011@live.com","type":1,"roleLists":0,"cid":"-5675366045147591562",
"id":"alice.unb2011@live.com","isFan":false},"timestamp":{"$date":1329194424064},"id":null,"isOfflin
eMessage":false,"isHistoryMessage":false,"info":{"text":"Mensagem01", Data
Mensagem

Palavra-chave: SDG

Palavra-chave Destino Origem
SDG 0368Routing:1.0 To: 1:alice.unb2011@live.com From: 1:bob.dpf2011@live.com;epid={e393bc35-
6ca4-4cb4-a505-fe538c3d641d}Reliability: 1.0Messaging: 2.0 Message-ID: 1329084863912 Message-
Type: TextOriginal-Arrival-ime: 012-02-12T22:14:24.152ZContent-Type: text/plain; charset=UTF-
8Content-Length: 10X-MS-IM-Format: FN=Segoe%20UI; EF=; CO=0 Mensagem04
Mensagem Data

Palavra-chave: EF=;

Destino Origem
SDG 0368Routing:1.0 To: 1:alice.unb2011@live.com From: 1:bob.dpf2011@live.com;epid={e393bc35-
6ca4-4cb4-a505-fe538c3d641d}Reliability: 1.0Messaging: 2.0 Message-ID: 1329084863912 Message-
Type: TextOriginal-Arrival-ime: 012-02-12T22:14:24.152ZContent-Type: text/plain; charset=UTF-
8Content-Length: 10X-MS-IM-Format: FN=Segoe%20UI; EF=; CO=0 Mensagem04
Data
Palavra-chave Mensagem

Palavra-chave: NFY

Palavra-chave Destino Origem
NFY PUT 45Routing: 1.0To: 1:alice.unb2011@live.com From: 1:bob.dpf2011@live.com

Palavra-chave: /pushchannel/

oprietário da Lista de contatos
Palavra-chave
GET /v1/pushchannel/alice.unb2011?

Palavra-chave: { "buddyInfo" : { "

{ "buddyInfo" : { "sequence" : 0, "contact" : [{ "sender" : "charlie.dpf2011" , "presenceState" : 0, "avatarUser" : 0, "avatarPreference" : "0" , "clientCapabilities" : 8915971, "clientUserGUID" : "GAHSOWL2KPZCRSBRYHFT2NAOKE" } , { "sender" : "bob.dpf2011" , "presenceState" : 0, "avatarUser" : 0, "avatarPreference" : "2" , "clientCapabilities" : 8915971, "clientUserGUID" : "R7FBF3DT2U4QUIOIKRFCF243V4" }] } }

Palavra-chave: "responses" : [{ "typing" :

"responses" : [{ "typing" : { "sequence" : 6, "sender" : "bob.dpf2011" , "receiver" : "alice.unb2011" , "activity" : 1, "network" : "yahoo" } }
] }

Palavra-chave: "responses" : [{ "message" :

"responses" : [{ "message" : { "status" : 1, "sequence" : 7, "sender" : "bob.dpf2011" , "receiver" : "alice.unb2011" , "msg" : "<font face=\"Arial\" size=\"10\">Mensagem02<\/font>" , "timeStamp" : 1325202730, "hash" :

Palavra-chave: "responses" : [{ "message" :

"responses" : [{ "message" : { "status" : 1, "sequence" : 7, "sender" : "bob.dpf2011" , "receiver" : "alice.unb2011" , "msg" : "<font face=\"Arial\" size=\"10\">Mensagem02<\/font>" , "timeStamp" : 1325202730, "hash" :

Palavra-chave: { "msg" :{ "text":

[{"msg":{"text":"Mensagem02","time":1329668883288,"clientTime":1329668882269,"msgID":"1329668878428:1197592284","coordinates":null,"messageId":"id.116296691829066"},"from":100003508664692,"to":100003516971476,"window_id":3537633229,"from_name":"Bob Dpf","from_gender":2, "sender_offline":false,"to_name":"Alice Unb","to_gender":1,"tab_type": "friend","show_orca_callout":false,"type":"msg"}

Palavra-chave: {"history":[{"from":

{"history":[{"from": 100003516971476,"to":100003508664692,"time":1329889748861, "msgId":"1329889743648:62807842","window_id":"1966794921","msg":{"text":"Mensagem01","messageId":"id.327963310589558"},"type":"msg"},{"from":100003508664692,"to":100003516971476,"time"

:1329889773211,"msgId":"351985681","window_id":null,"msg":{"text":"Mensagem02","messageId":"id.237413103018952"},"type":"msg"}]

Palavra-chave: \"InitialChatUserInfos\"

\"InitialChatUserInfos\",[],{\"100003508664692\":{\"name\":\"Bob Dpf\",\"firstName\":\"Bob\",
\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/static-ak\\\/rsrc.php\\\/v1\\\/yo\\\/r\\\/UlIqmHJn-SK.gif\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003508664692\",\"gender\":2,\"type\":\"user\",\"exist\":true,\"showVideoSheet\":false},\"100003516971476\":{\"name\":\"Alice Unb\",\"firstName\":\"Alice\",\"vanity\":null,\"thumbSrc\":\"http:\\\/\\\/profile.ak.fbcdn.net\\\/hprofile-ak-snc4\\\/161664_100003516971476_505534554_q.jpg\",\"uri\":\"http:\\\/\\\/www.facebook.com\\\/profile.php?id=100003516971476\",\"gender\":1,\"type\":\"friend\",\"exist\":true,\"showVideoSheet\":false