

**HAMILTON FONTE II**

**DIZANG: UMA SOLUÇÃO PARA COLETA DE  
EVIDÊNCIAS FORENSES PARA ATAQUES DE  
INJEÇÃO NA NUVEM**

Documento apresentado à Escola Politécnica  
da Universidade de São Paulo para a realiza-  
ção do Exame de Qualificação de Mestrado  
em Engenharia Elétrica.

São Paulo  
2018

**HAMILTON FONTE II**

**DIZANG: UMA SOLUÇÃO PARA COLETA DE  
EVIDÊNCIAS FORENSES PARA ATAQUES DE  
INJEÇÃO NA NUVEM**

Documento apresentado à Escola Politécnica  
da Universidade de São Paulo para a realiza-  
ção do Exame de Qualificação de Mestrado  
em Engenharia Elétrica.

Área de concentração:  
Engenharia da Computação

Orientador:  
Marcos Antonio Simplicio Junior

São Paulo  
2018

## RESUMO

A adoção de arquiteturas em nuvem aumenta a cada dia, e com ela também o número de casos em que esse tipo de tecnologia é usada para fins ilícitos. No ano de 2015 o número de registros de atas notariais comprovando abusos e crimes virtuais cresceu 87%. Infelizmente, devido à natureza volátil da nuvem, a tarefa de coletar evidências para análise forense nesse ambiente tem esbarrado em desafios práticos e legais. A prática herdada da forense tradicional para coleta de evidências, onde se isola e cena do crime e coleta-se todas as evidências, foi traduzida para a forense digital como a cópia bit a bit da mídia que se deseja investigar. Tal prática levou a coleta de grandes volumes de informação para análise impactando o tempo de investigação. Duas características das soluções em nuvem têm dificultado a obtenção de evidências válidas. O compartilhamento de recursos físicos entre vários usuários impede a remoção dos mesmos para análise uma vez que isso viola a privacidade de elementos não envolvidos na investigação. A localização do recurso físico em uma região geográfica diferente de onde o crime foi cometido pode impedir a coleta de evidências caso não haja acordos de cooperação estabelecidos. Estes aspectos, se não forem levados em consideração, podem por em xeque a credibilidade da evidência. Este trabalho analisa propostas na literatura voltadas a resolver os principais desafios existentes na coleta evidências na nuvem, discutindo suas limitações, e então propõe uma solução que cobre coleta, transporte e armazenamento da evidência visando suplantá-las. A solução aqui proposta provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados sem afetar sua credibilidade. Para tal faremos uso de contêineres e seu identificador para relacionar a evidência a sua origem, tal técnica tem a vantagem que conseguir preservar a relação evidência - origem mesmo que esta última não exista mais na solução sob investigação. Especificamente, ela tem como focos (1) a reprodutibilidade do processo de coleta, (2) não violar jurisdição e (3) não violar a privacidade dos não envolvidos na investigação. **MARCOS: Discuta como é feito e quais os benefícios da solução. Não seja tão sintético: não há limitação de espaço aqui! Até no artigo tem mais texto... Hamilton - Acho que foi**

## ABSTRACT

The adoption of cloud architectures increases every day and with it, also the number of cases where such technology is used for illicit purposes. In 2015, the number of notarial records acts related to digital abuses and virtual crimes grew by 87%. Unfortunately, due to the volatile nature of the cloud, the task of collecting evidences for forensic analysis frequently run into practical and legal issues. The practice, inherited from the traditional forensics, to isolate the crime scene and collect all evidences, has been translated to digital forensic as the bitwise copy of the media to be investigated. Although thorough, this practice led to the collection of large volumes of information for analysis which in its turn impacted investigation time. On top of that, two other features of cloud solutions have made it difficult to obtain valid evidence. The sharing of physical resources among multiple users which prevents the removal of hardware for analysis as this violates the privacy of parties not involved in the investigation. The geographic location of the physical resource which, if located in a country different from where the crime was committed, may prevent the collection of evidence if cooperation agreements are not in place. These aspects, if not taken into account, may call into question the credibility of the evidence in a court of law. This work analyzes other proposals aimed at solving the main challenges of collecting forensic evidence in the cloud, discusses their limitations and then proposes a solution that covers collection, transport and storage of evidence in order to overcome them. This proposal provides a way of correlating evidence and its virtual origin, transport and store such data without affecting its credibility. For this we make use of Linux Container and its identifier to relate the evidence to its origin. Such a technique has the advantage that it can preserve the relation evidence - origin even though the origin doesn't exists anymore. It focus on (1) the reproductibility of the collection process, (2) the non violation of juristiction, and (3) the non violation of privacy of those not involved in the investigation.

# LISTA DE ILUSTRAÇÕES

1	Automated Forensics Data Acquisition Model . . . . .	20
2	Virtuoso . . . . .	22
3	A Log Based Approach Model . . . . .	23
4	Backup Approach Model . . . . .	24
5	FoRensic Open Stack Tools . . . . .	25
6	Digital Forensic Framework for Cloud Environment . . . . .	26
7	Digital Forensic as a Service - Indexed Data . . . . .	27
8	Janela deslizante de coleta de evidência . . . . .	33
9	Arquitetura geral da solução Dizang . . . . .	34
10	Evolução do uso do espaço em disco com o Dizang . . . . .	36
11	Tempo de cópia da memória de um contêiner . . . . .	37
12	Cronograma de projeto . . . . .	40

## **LISTA DE TABELAS**

1	Comparativo de soluções de coleta de informações de memória de máquinas em nuvem para análise forense . . . . .	30
---	---	----

## LISTA DE ABREVIATURAS E SIGLAS

VM	<i>Virtual Machines</i>
SO	Sistema Operacional
SaaS	<i>Software as a Service</i>
PaaS	<i>Platform as a Service</i>
IaaS	<i>Infrastructure as a Service</i>
FaaS	<i>Forensics as a Serviço</i>
LXC	<i>Linux Containers</i>
VMI	<i>Virtual Machine Introspection</i>
VMM	<i>Virtual Machine Manager</i>
CSP	<i>Cloud Service Provider</i>
GRR	<i>Google Rapid Response</i>
FROST	<i>FoRensic Open Stack Tools</i>
API	<i>Aplication Programing Interface</i>
SWGDE	<i>Scientific Working Group on Digital Evidence</i>
ETL	<i>Extract, Transform and Load</i>
EDIPM	<i>Enhanced Digital Investigation Process Model</i>
SENASP	Secretaria Nacional de Segurança Pública

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>9</b>
1.1	Objetivos . . . . .	10
1.2	Motivação . . . . .	11
1.3	Organização do documento . . . . .	12
<b>2</b>	<b>Fundamentação Teórica</b>	<b>13</b>
2.1	Nuvem computacional . . . . .	13
2.2	Uso de contêineres . . . . .	14
2.3	Forense digital e seus desafios . . . . .	15
2.3.1	Aceitabilidade da evidência em processo legal. . . . .	16
2.3.2	Volume de dados para coleta . . . . .	17
2.3.3	Privacidade e jurisdição . . . . .	17
2.3.4	Coleta de evidências de memória volátil de máquinas em nuvem	18
<b>3</b>	<b>Revisão da literatura</b>	<b>19</b>
3.1	Acessar e coletar as informações de memória das máquinas virtuais em nuvem . . . . .	26
3.2	Capacidade de reproduzir o processo e obter os mesmos resultados . .	28
3.3	Não violar privacidade ou jurisdição das partes não envolvidas na in- vestigação . . . . .	28



3.4	Garantir a cadeia de custódia da evidência . . . . .	29
3.5	Resumo . . . . .	29
<b>4</b>	<b>Proposta de projeto: Dizang</b>	<b>31</b>
4.1	Descrição . . . . .	31
4.2	Implementação . . . . .	34
4.3	Resultados experimentais . . . . .	35
4.4	Limitações . . . . .	36
4.5	Conclusões e trabalhos futuros . . . . .	37
4.6	O que foi feito e cronograma esperado para os próximos passos . . . .	38
4.7	Contribuições . . . . .	39
	<b>Referências</b>	<b>41</b>

# 1 INTRODUÇÃO

**MARCOS: Erro comum 31: não coloque seção ou sub-seção sem texto no capítulo seção - Hamilton: Feito, falta de atenção minha**

Técnicas de virtualização, replicação de serviços e compartilhamento de recursos entre múltiplos usuários (multi-inquilinato) proveem a nuvens computacionais uma elevada escalabilidade (MORSY; GRUNDY; MULLER, 2010). Ao mesmo tempo, tais mecanismos também criam grande volatilidade dos recursos virtuais que executam aplicações em nuvem. Afinal, quando submetida a uma carga elevada, uma aplicação hospedada na nuvem pode criar clones das máquinas virtuais (*virtual machines* – VM) que a hospedam e balancear a carga entre elas, de modo a atender à demanda sem prejuízos na qualidade do serviço oferecido. Após esse pico, as máquinas que foram clonadas são normalmente desativadas, seus recursos liberados e o sistema retorna à capacidade anterior, evitando-se custos desnecessários.

Embora interessante do ponto de vista de eficiência e custos, do ponto de vista forense a volatilidade da nuvem traz problemas em caso de ataques. Por exemplo, caso uma das instâncias de processamento virtuais criadas temporariamente seja alvo de ameaças que atuam diretamente na sua memória, sem deixar rastros em discos (e.g., arquivos de *log*), as evidências desse evento podem ser completamente perdidas após elas serem desativadas e terem seus recursos liberados. Essa dificuldade é ainda agravada por aspectos como multi-inquilinato e multi-jurisdição típicas de soluções em nuvem pública ou híbrida (GILBERT; SUJEET, 2008). Especificamente, o aspecto

multi-inquilino dificulta a obtenção do *hardware* que executa as aplicações de interesse, pois, como ele é compartilhado por vários usuários, removê-los para análise poderia levar a uma violação de privacidade dos usuários não relacionados à investigação. Já a natureza distribuída da nuvem pode levar à alocação de informações relevantes à investigação em vários países, dificultando a obtenção das mesmas em especial quando não existem acordos de cooperação entre as entidades envolvidas (DYKSTRA; SHERMAN, 2012). Combinadas, tais características dificultam a coleta de evidências com a credibilidade necessária para que elas possam ser usadas em processos legais, o que exige o respeito à privacidade, à jurisdição e à cadeia de custódia, bem como a reprodutibilidade do processo de coleta (RAHMAN; KHAN, 2015).

Embora existam soluções na literatura que abordam a coleta de informações de nuvem com o propósito de análise forense, a grande maioria delas aborda a coleta, o transporte e o armazenamento de forma isolada. Por exemplo, trabalhos como (DYKSTRA; SHERMAN, 2013) e (REICHERT; RICHARDS; YOSHIGOE, 2015) tratam de fatores como multi-inquilinato e multi-jurisdição, discutindo formas de coleta e preservação da evidência fora da nuvem. Já estudos como (GEORGE; VENTER; THOMAS, 2012) se concentram na análise forense para a coleta de evidência de máquinas virtuais enquanto elas estão em execução, enquanto trabalhos como (SANG, 2013) abordam a questão de processos de garantia de cadeia de custódia em ambientes de nuvem para transporte da evidência. Por outro lado, não foram identificadas na literatura propostas que (1) descrevam como o dado é coletado e armazenado observando a cadeia de custódia, e (2) visem garantir que, mesmo que um recurso virtualizado (e.g., uma VM) seja desalocada, haja condições de se reproduzir o processo de coleta de evidências.

## 1.1 Objetivos

O presente trabalho visa suplantiar as limitações supracitadas, melhorando a capacidade de coleta de evidências de aplicações em nuvem. Mais precisamente, por meio

de uma proposta que tem como focos (1) a reprodutibilidade do processo de coleta, (2) o estabelecimento de vínculo entre a evidência coletada e sua origem, (3) a preservação da jurisdição e da privacidade dos não envolvidos na investigação e (4) a garantia de custódia da evidência. Em suma, a solução descrita provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados de modo a preservar sua credibilidade. Para isso, a proposta supõe que o sistema sendo monitorado é executado dentro de um contêiner em nuvem.

## 1.2 Motivação

Uma nuvem computacional é um modelo de infraestrutura no qual recursos compartilhados em quantidade configurável, acessíveis via rede, são alocados e desalocados com esforço mínimo de gerenciamento por parte de um provedor de serviços (MELL; GRANCE, 2011). Há três modelos principais de comercialização de uso da nuvem (MELL; GRANCE, 2011): *software* como serviço (*Software as a Service* – SaaS), na qual se provê o *software* que será usado pelo cliente; plataforma como serviço (*Platform as a Service* – PaaS), na qual se provê o ambiente para que o cliente desenvolva, teste e execute seu *software*; e, o tipo mais pertinente para este trabalho, Infraestrutura como serviço (*Infrastructure as a Service* – IaaS), na qual são fornecidos recursos computacionais básicos, como processamento e memória, em geral de forma virtualizada.

Virtualização tem sido amplamente adotada por empresas das mais diversas áreas. Segundo o "State of the Cloud Report" realizado pela empresa Right Scale, 95% das organizações entrevistadas estão utilizando ou experimentando soluções em nuvem no modelo IaaS (Right Scale, 2017). A virtualização de recursos na nuvem, embora tradicionalmente feita por meio de máquinas virtuais, vêm sendo crescentemente feita também na forma de contêineres. De fato, segundo o "Container Market Adoption Survey 2016", realizado pelas empresas DevOps.com (<https://devops.com/>) e ClusterHQ

(<https://clusterhq.com>) com 235 empresas que têm desenvolvimento de software como sua atividade fim ou como suporte à atividade fim, 76% dos respondentes utilizam contêineres para melhorar a eficiência do processo de desenvolvimento e em suas arquiteturas de microsserviços em nuvem.

O crescente volume de informações que as soluções em nuvem armazenam e trafegam hoje e os aspectos o multi-inquilino e a multi-jurisdição dos provedores de infraestrutura em nuvem estão entre os principais obstáculos enfrentados pelos investigadores forenses. (QUICK; CHOO, 2014) (GILBERT; SUJEET, 2008)

### 1.3 Organização do documento

O restante deste documento está organizado nos seguintes capítulos.

O Capítulo 2 apresenta o ambiente em que este trabalho está inserido, as tecnologias envolvidas e os desafios a serem superados. O Capítulo 3 analisa os trabalhos relacionados a forense de memória em nuvem. O Capítulo 4 apresenta o que foi feito até então e uma proposta para conclusão deste trabalho.

**MARCOS: liste todos os capítulos e diga o que eles discutem. O cap. 1 não precisa (o leitor já leu...) - Hamilton: Feito lista de capítulos e criação de labels para os mesmos**

## 2 FUNDAMENTAÇÃO TEÓRICA

Desde 2001, modelos de condução de investigação digital foram propostas. O *Enhanced Digital Investigation Process Model* proposto por Carrier e Stafford em 2003 é a última iteração na evolução do processo forense digital. Entretanto tais modelos de investigação foram desenvolvidos antes da aparição de tecnologias de computação em nuvem e partem do pressuposto que o investigador tem acesso e controle sobre o sistema sob investigação (GRISPOS; STORER; GLISSON, 2012). Esta defasagem esta no centro dos desafios da forense digital.

### 2.1 Nuvem computacional

Uma nuvem computacional é um modelo de infraestrutura no qual recursos compartilhados em quantidade configurável, acessíveis via rede, são alocados e desalocados com esforço mínimo de gerenciamento por parte de um provedor de serviços. Há três modelos principais de comercialização de uso da nuvem: *software* como serviço (*Software as a Service* – SaaS), na qual se provê o *software* que será usado pelo cliente; plataforma como serviço (*Platform as a Service* – PaaS), na qual se provê o ambiente para que o cliente desenvolva, teste e execute seu *software*; e, o tipo mais pertinente para este trabalho, infraestrutura como serviço (*Infrastructure as a Service* – IaaS), na qual são fornecidos recursos computacionais básicos, como processamento e memória, em geral de forma virtualizada (MELL; GRANCE, 2011).

Uma arquitetura usada hoje nas soluções em nuvem são as auto-escaláveis onde

recursos são altamente voláteis, sendo alocados e desalocados a qualquer momento. Estas implementações tem a vantagem de usar os recursos de nuvem de uma forma mais eficiente e menos necessidade de intervenção humana.

## 2.2 Uso de contêineres

Segundo (AMAZON, 2016), contêiner é um método de virtualização do sistema operacional que permite executar uma aplicação e suas dependências em um processo com os recursos como disco, memória e rede isolados. Diferente das máquinas virtuais, a virtualização com contêineres é feita no nível do SO nativo, tem uma implementação mais simples eliminando camadas entre o aplicativo executado e o *hardware* físico permitindo maior granularidade no controle sobre esses recursos melhorando a eficiência da infraestrutura. Uma tecnologia bastante utilizada para esse propósito são Contêineres Linux (LXC) (LINUXCONTAINERS.ORG, 2015), que aproveitam-se de funcionalidades como *cgroups*, *kernel namespaces* e *chroot* do kernel do Linux para auxiliar no gerenciamento e isolamento de recursos virtuais.

Control Groups (*cgroups*) é uma funcionalidade do Kernel do Linux que limita e isola o uso de recursos como CPU, memória e disco de um conjunto de processos e os organiza de forma hierárquica. O trabalho nesta funcionalidade começou em 2006 na Google sob a denominação de *process container*. No fim de 2007 seu nome mudou para *control groups* e foi adicionado a versão 2.6.24 do kernel lançado em 2008 (Unix Man Pages, a).

*Namespace* é uma funcionalidade do Kernel do Linux usado para isolar e virtualizar recursos do sistema operacional como Ids de processos, acessos a rede, comunicação inter-processos e sistema de arquivos. *Namespace* envolve os recursos do sistema operacional em uma abstração que faz parecer aos processos de um mesmo *namespace* que eles tem sua própria instância isolada de um recurso global, é a princi-

pal funcionalidade por trás da implementação de contêineres (Unix Man Pages, d).

Change Root (*chroot*) é uma funcionalidade do Kernel do Linux usado para mudar o diretório root do processo que está chamando a função e de todos os seus processos filhos. A chamada a *chroot* altera o processo de resolução de caminhos do sistema operacional para o processo que o chamou (Unix Man Pages, b). Desta forma pode-se instalar uma distribuição Linux secundária em uma pasta, ao invés de uma partição, e executar programas desta pasta sem perda significativa de desempenho.

## 2.3 Forense digital e seus desafios

Forense digital (também conhecida por forense computacional) é um conjunto de técnicas de coleta e análise da interação entre humanos e computadores de forma que esta seja aceita em um processo legal. Tal como a forense tradicional, a forense digital se baseia no princípio de Locard. Definido pelo médico francês Edmond Locard, estabelece que “Quando um indivíduo entra em contato com outro objeto ou indivíduo, este sempre deixa vestígio deste contato” (RAMOS, 2011).

A forense digital quando aplicada a investigação de incidentes em soluções na nuvem enfrenta desafios adicionais relacionados a coleta, transporte e análise da evidência. A primeira delas diz respeito a aceitabilidade de uma evidência. Para que uma evidência seja aceita em um processo legal é necessária que a cadeia de custódia relacionada a evidência tenha sido garantida. Cadeia de custódia é o processo de documentação da história cronológica da evidência de modo a saber onde a evidência esteve e quem teve acesso a ela (RAMOS, 2011). O SENASP 2013 define cadeia de custódia como “a sistemática de procedimentos que visa à preservação do valor probatório da prova pericial caracterizada.” Assim a Forense digital tem por objetivo a investigação de evidências digitais da interação entre homem e máquina de modo a reconstruir a cadeia de eventos no passado de forma que suas conclusões sejam validadas



por terceiros.

### **2.3.1 Aceitabilidade da evidência em processo legal.**

O processo de análise forense no evento de um crime digital é descrito no EDIPM por 4 fases: Identificar, preservar, examinar e apresentar (GRISPOS; STORER; GLISSON, 2012). Segundo os autores, a fase de preservação da evidência deve ser conduzida de forma “forensicamente aceitável” isto é, coletar as evidências de forma que as mesmas sejam aceitas em um processo legal e não sejam invalidadas no curso do mesmo. Segundo (RAMOS, 2011) a aceitabilidade de uma evidência digital em um processo legal deve atender a dois requisitos principais. Autenticidade, descrito como “o processo pelo qual se pode garantir a autoria do documento eletrônico”, ou seja, não permite dúvida quanto à identificação do autor e integridade, descrita como “atestado da inteireza do documento eletrônico após sua transmissão, bem como apontar eventual alteração irregular de seu conteúdo”. Caso haja dúvida a cerca de um destes requisitos, uma perícia técnica pode ser convocada, nesta será analisada o autor da evidência ou seja sua fonte e se a mesma não foi alterada no processo.

Em infraestrutura física esta coleta era relativamente simples, bastava-se remover o recurso físico, transportar este para um laboratório e lá analisar a evidência. A Evidência era mantida em uma sala-cofre onde o acesso era controlado. A reprodutibilidade do processo de coleta e a manutenção da integridade da evidência eram tarefas bem diretas. A computação em nuvem, especialmente as de infraestrutura auto-escalável trouxeram um conjunto de desafios novos. O recurso físico não pode mais ser removido pois o mesmo é utilizado por outros usuários não relacionados a investigação, fazê-lo constituiria violação de privacidade. A volatilidade dos recursos tornou a verificação do seu autor um processo mais complexo pois o recurso que a gerou pode não existir mais (SIMOU et al., 2014). A integridade da evidência também tornou-se mais complexa pois ela precisa ser coletada, transportada e armazenada. O processo de ca-

deia de custódia ganhou grande visibilidade neste quesito. Violação de qualquer uma das características citadas anteriormente põe em dúvida a credibilidade da evidência.

### **2.3.2 Volume de dados para coleta**

O processo de coleta da evidência na forense digital herdou suas práticas da forense tradicional onde isola-se cena do crime e coletam-se as evidências presentes. Transportando para a forense digital introduziu-se a realização da cópia bit a bit da informação que se deseja analisar. No passado, com as soluções manipulando quantidades bem menores de memória, disco e tráfego, tal prática não trazia problemas. Nas atuais soluções, aplicações e arquiteturas em nuvem, o volume de dados aumentou consideravelmente. Em 2014 investigadores forenses tinham em média 6 meses de backlog para analisar (QUICK; CHOO, 2014). Encontrar uma forma de armazenar menos informações de modo a tornar a fase de análise mais rápida e eficiente ajudará nas investigações.

### **2.3.3 Privacidade e jurisdição**

Na metodologia tradicional de coleta de evidências para análise, isola-se o ambiente e as evidências são removidas. Transportando para a forense digital, temos a prática de remover o equipamento para realização de cópia bit a bit da evidência. Nas soluções de infra estrutura física esta prática não trás problemas, os objetos ou indivíduos sob investigação estão diretamente relacionados ao equipamento removido. Nas soluções em nuvem esta prática não pode mais ser utilizada, pois como o recurso físico é compartilhado por vários usuários não envolvidos na investigação, removê-los configura violação de privacidade. Um complicador a mais é o fato de os dados não estarem armazenados no mesmo território em que a investigação é realizada demandando acordos de cooperação jurídica entre as partes o que nem sempre é possível (SIMOU et al., 2014). Neste cenário encontrar uma forma de coletar a evidência sem violar jurisdição

e privacidade são de grande importância num futuro próximo.

### **2.3.4 Coleta de evidências de memória volátil de máquinas em nuvem**

Na metodologia tradicional de coleta de evidências para análise isola-se o ambiente e as evidências são removidas. Transportado para a forense digital temos a prática de remover o equipamento para realização de cópia bit a bit da evidência. Nas soluções de infra estrutura física esta prática não trás problemas, os objeto ou indivíduos sob investigação estão diretamente relacionados ao equipamento removido. Nas soluções em nuvem esta prática não pode mais ser utilizada pois como o recurso físico é compartilhado por vários usuários não envolvidos na investigação, removê-los configura violação de privacidade. Um complicador a mais é o fato de os dados não estarem armazenados no mesmo território em que a investigação é realizada demandando acordos de cooperação jurídica entre as partes o que nem sempre é possível. Neste cenário encontrar uma forma de coletar a evidência sem violar jurisdição e privacidade são de grande importância num futuro próximo.

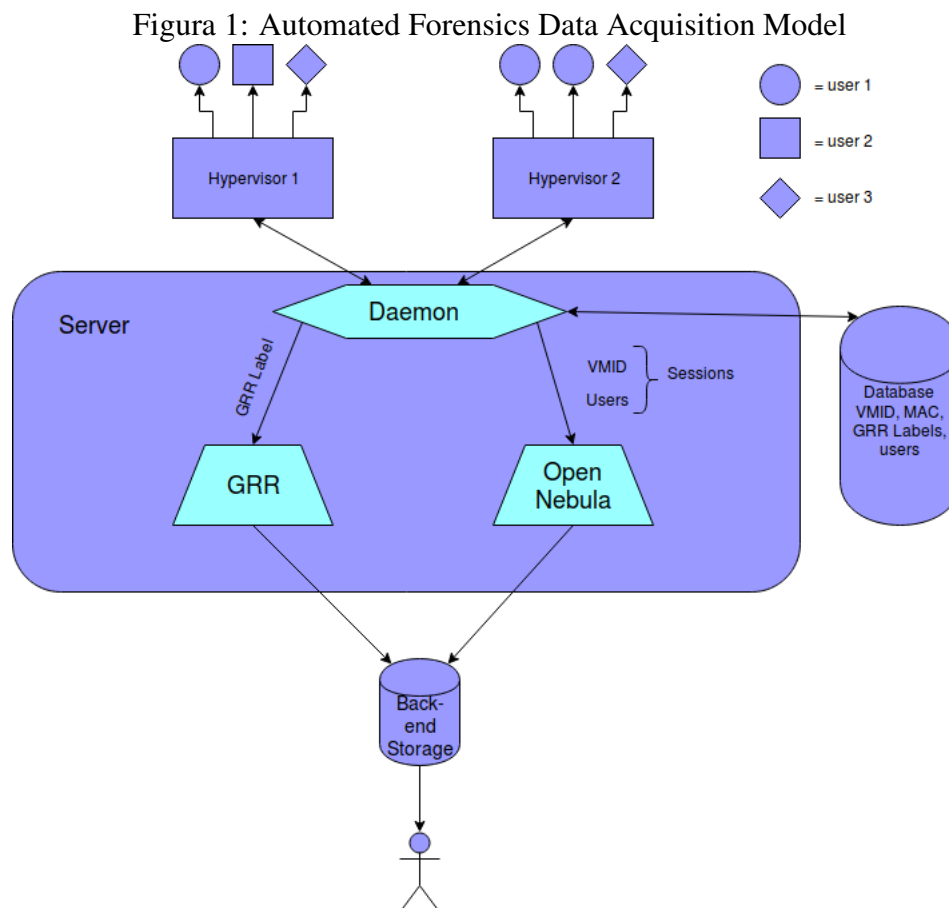
### 3 REVISÃO DA LITERATURA

**MARCOS: Explique melhor as soluções: você tem espaço, então pode deixar este capítulo muito mais autocontido - Hamilton: Feito**

Os principais aspectos relacionados a coleta de evidências para análise forense em nuvem são, coleta, transporte, armazenamento, garantia da cadeia de custódia e reprodutibilidade do processo de coleta. Das propostas encontradas em literatura oito foram selecionadas como melhor representação do estado da arte, a seguir uma breve descrição de cada uma delas.

O modelo proposto por (REICHERT; RICHARDS; YOSHIGOE, 2015) é um processo de coleta de evidências integrado ao *hypervisor* e disparado por algum sistema de detecção de intrusão. A partir do momento em que uma ameaça é detectada, o modelo tira instantâneos das máquinas virtuais comprometidas. O modelo toma cuidado para excluir informações de clientes não relacionados a investigação e armazena o restante em local seguro, tudo de forma automatizada. O autor não descreve os detalhes do armazenamento mas diz que é “forensicamente aceitável”. O modelo faz uso de GRR para agregar e analisar as evidências coletadas. O modelo possui um motor de regras que se baseia em um conjunto de descrições de ameaças conhecidas armazenadas em banco para, caso alguma evidência coletada coincida com ameaças armazenadas, ele alerta um usuário humano para uma avaliação mais detalhada. O intervalo de tempo em que os instantâneos de memória são gerados é configurável e são todos armazenados em persistência. Este modelo tem como principal vantagem a automação do processo

de coleta que dispensa intervenção humana tornando mais fácil garantir a cadeia de custódia. A figura 1 mostra o desenho da arquitetura proposta pelo autor.



Fonte: (REICHERT; RICHARDS; YOSHIGOE, 2015)

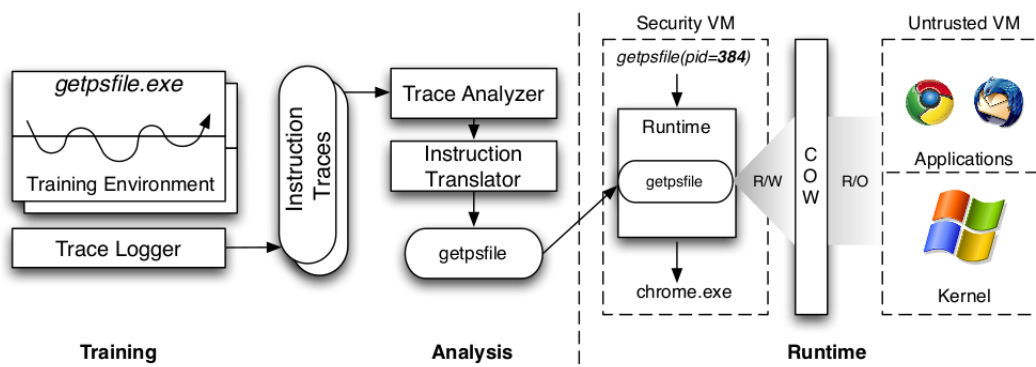
A proposta de (POISEL; MALZER; TJOA, 2013) é baseada na técnica de introspecção em máquina virtual (VMI) para coleta de memória volátil. Esta técnica se apoia na necessidade do gerenciador de máquinas virtuais (VMM) em mapear recursos físicos da máquina hospedeira a recursos alocados à máquina virtual cliente. Este mapeamento é usado para permitir que a memória volátil copiada da máquina virtual seja reconstruída em uma máquina física para realização da análise. A proposta utiliza de coleta contínua dos instantâneos de memória durante o funcionamento do sistema sem distinção do que aconteceu antes ou depois do fato de interesse, e todos os instantâneos de memória são armazenados para posterior análise. Para eliminar a chance de

inconsistências no instantâneo de memória volátil, a máquina virtual tem sua execução suspensa durante o processo de extração. Uma desvantagem da técnica de VMI mencionado pelo próprio autor é a necessidade de tradução de endereços de memória da máquina virtual em endereços de memória da máquina física hospedeira. Esta tradução depende de conhecimento do que está sendo executado na máquina virtual, logo uma solução baseada em VMI não é completamente portátil sendo necessário adequações para diferentes clientes além de ser computacionalmente custoso.

Também na vertente de introspecção de máquina virtual, (DOLAN-GAVITT et al., 2011) propõe o *Virtuoso*, um arcabouço de coleta de informações de processos específicos em uma máquina virtual. O arcabouço funciona em três fases, a primeira realiza um estudo em uma máquina virtual teste do processo que se deseja coletar dados de memória mapeando o conjunto de instruções executado. A fase dois traduz o conjunto de instruções mapeados na fase um e cria um executável para que o processo seja rodado fora da máquina virtual. A terceira fase usa um ambiente na máquina hospedeira para executar as instruções extraídas na fase dois. Este ambiente consegue acessar os endereços de memória do máquina virtual tornando possível coletar instantâneos de memória do processo em execução. A principal vantagem deste arcabouço é a capacidade de coletar instantâneos de memória de um processo específico. A desvantagem deste método é que sua atuação é após o fato. A figura 2 mostra um esquema do funcionamento do arcabouço

O arcabouço proposto por (SANG, 2013) é um sistema que funciona em parceria com o provedor de nuvem onde este último envia informações ao arcabouço que os armazena em um local centralizado. O conjunto de informações armazenadas é acordado antecipadamente com o provedor de nuvem e vão desde instantâneos de memória volátil até pacotes trafegados nas interfaces de rede da máquina virtual. O arcabouço coleta informações continuamente e usa cálculo de hash das evidências enviadas pelo provedor de nuvem para garantir que as mesmas não foram alteradas durante o transporte.

Figura 2: Virtuoso

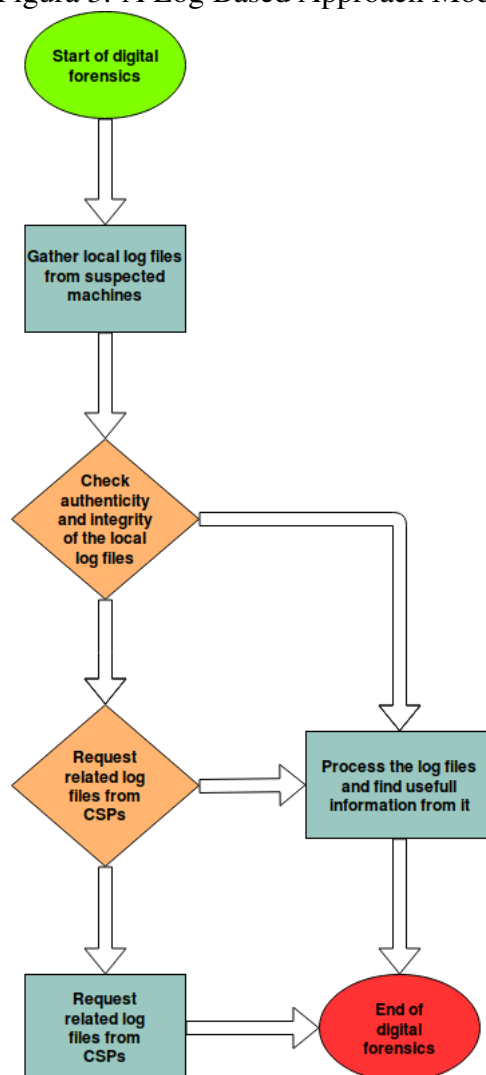


Fonte: (DOLAN-GAVITT et al., 2011)

Assim como as propostas anteriores, esta também não faz distinção do que aconteceu antes ou depois do fato de interesse coletando constantemente informações da máquina virtual. O próprio autor menciona que o arcabouço tem a desvantagem de depender da cooperação do provedor de nuvem. Esta dependência é uma estratégia considerada fraca pela comunidade forense pois a prioridade do CSP é o de garantir a disponibilidade do serviço não o de coletar evidências (ALQAHTANY et al., 2015). A figura 3 mostra um esquema do funcionamento da solução focada em um caso específico de log de rede como proposta pelo autor.

O trabalho descrito em (DEZFOULI et al., 2012) é voltada a dispositivos móveis e tem como principal vantagem a preocupação com as limitações de armazenamento do dispositivo. O processo de coleta do instantâneo de memória volátil do dispositivo separa as informações e as armazena por processo ativo, o autor usa esta técnica para melhor gerenciar o espaço que as informações estão ocupando em disco. Possui inteligência para descartar informações de processos que foram terminados e removidos da memória assim como inteligência para fazer uso ótimo do espaço de armazenamento do dispositivo. O processo de coleta de informações de memória volátil é feita continuamente independente de eventos de interesse como detecção de ameaças. A figura 4 o autor mostra um esquema macro de como o armazenamento da evidência é gerenciado.

Figura 3: A Log Based Approach Model

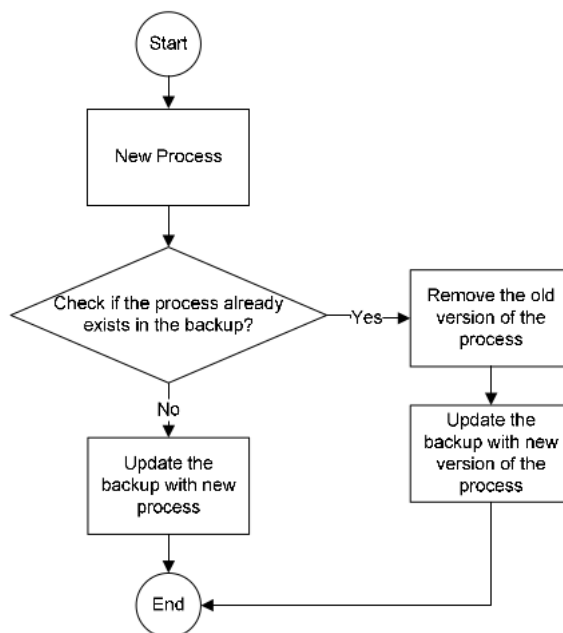


Fonte: (SANG, 2013)

A ferramenta FROST proposto por (DYKSTRA; SHERMAN, 2013) descreve um conjunto de bibliotecas integradas ao arcabouço *Open Stack Framework* de gerenciamento de infra estruturas virtualizadas. Através desta integração, FROST expõe um conjunto de API para serem usadas por aplicações de coleta de evidências forenses que dão acesso a recursos da máquina virtual sendo administrada como disco, logs de trafego de rede e memória volátil. A proposta descreve apenas o arcabouço, deixa a critério do usuário detalhes como periodicidade e tamanho da coleta assim como a forma de transporte da evidência e onde ela será armazenada. De todas as propostas descritas



Figura 4: Backup Approach Model

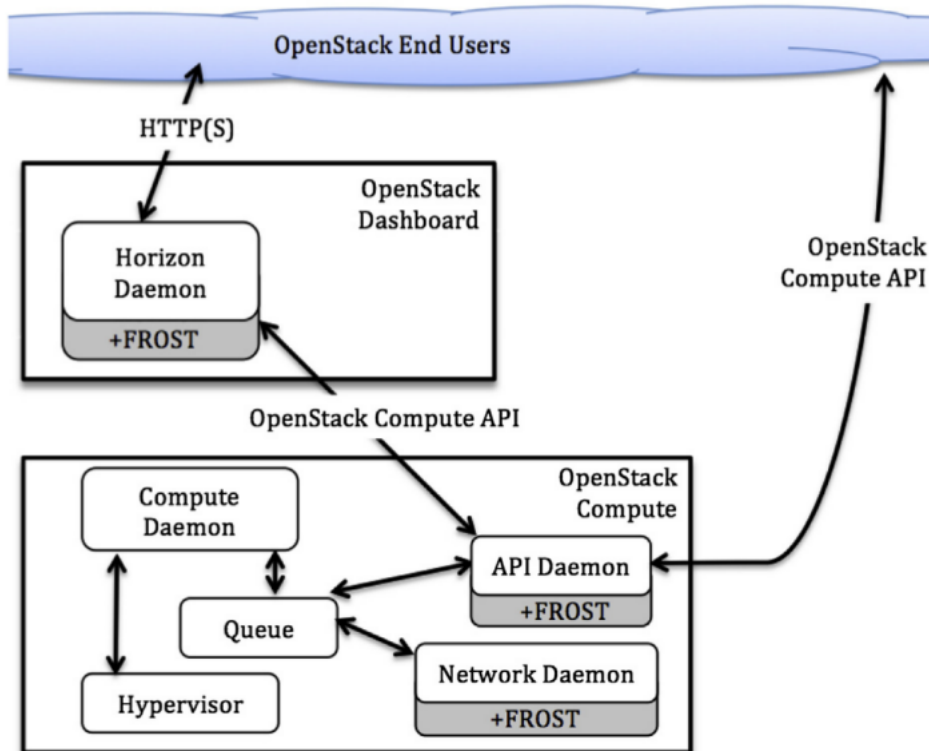


Fonte: (DEZFOULI et al., 2012)

neste capítulo esta é a única que demonstra preocupação com adequação a questões legais e padrões já estabelecidos na indústria forense. O autor declara que FROST segue as práticas definidas no Grupo de Trabalho Sobre Evidência Digital (SWGDE) e do Manual de Busca em Apreensão do Departamento de Justiça Norte-Americano. A figura 5 o autor mostra um esquema macro da integração entre FROST e o arcabouço *Open Stack Framework*

O trabalho descrito em (GEORGE; VENTER; THOMAS, 2012) está focado em monitoração de rede e opera em uma arquitetura de forense como serviço (FaaS). O autor propõe um conjunto de ferramentas que tem capacidade de realizar auto descoberta das interfaces sob monitoração, coletar evidências de tais máquinas e armazená-las. O processo de auto descoberta e associação das evidências com usuários de rede é realizado por um motor baseado em ontologias armazenadas em um banco de dados próprio. Esta proposta foca apenas no processo de coleta. A descrição do armazenamento e transporte é superficial. Na figura 6 mostra o desenho da arquitetura proposta pelo autor.

Figura 5: FoRensic Open Stack Tools

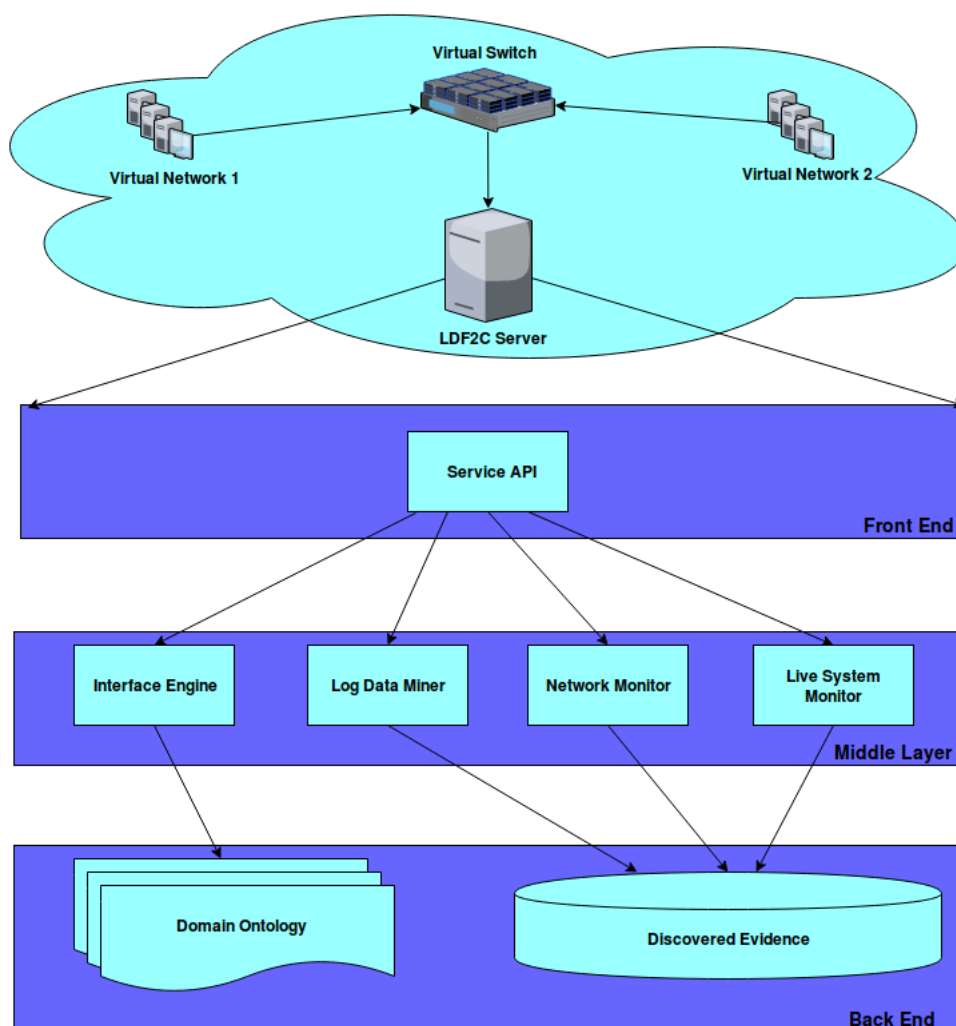


Fonte: (DYKSTRA; SHERMAN, 2013)

Na mesma vertente de forense como serviço, (LEE; UN, 2012) ataca o problema de grande volume de dados coletados propondo serviço de coleta e indexação de evidências. O serviço espera receber dados da execução do comando unix DD (Unix Man Pages, c) nas máquinas alvo onde, apoiado em processos ETL e MapReduce (WIKIPEDIA, 2018), esses dados serão disponibilizados para consulta pelos investigadores. A coleta ocorre continuamente a intervalos de tempo configuráveis. O autor não descreve onde os dados serão armazenados e quem será responsável pela infraestrutura de armazenamento. O autor também não fala como os dados serão transportados até o armazenamento e como ele garante que estes não foram alterados no processo. A figura 7 mostra o desenho da arquitetura do serviço proposta pelo autor

A seguir os trabalhos mencionados acima são agrupados e avaliados com base nos diferentes aspectos que abordam.

Figura 6: Digital Forensic Framework for Cloud Environment

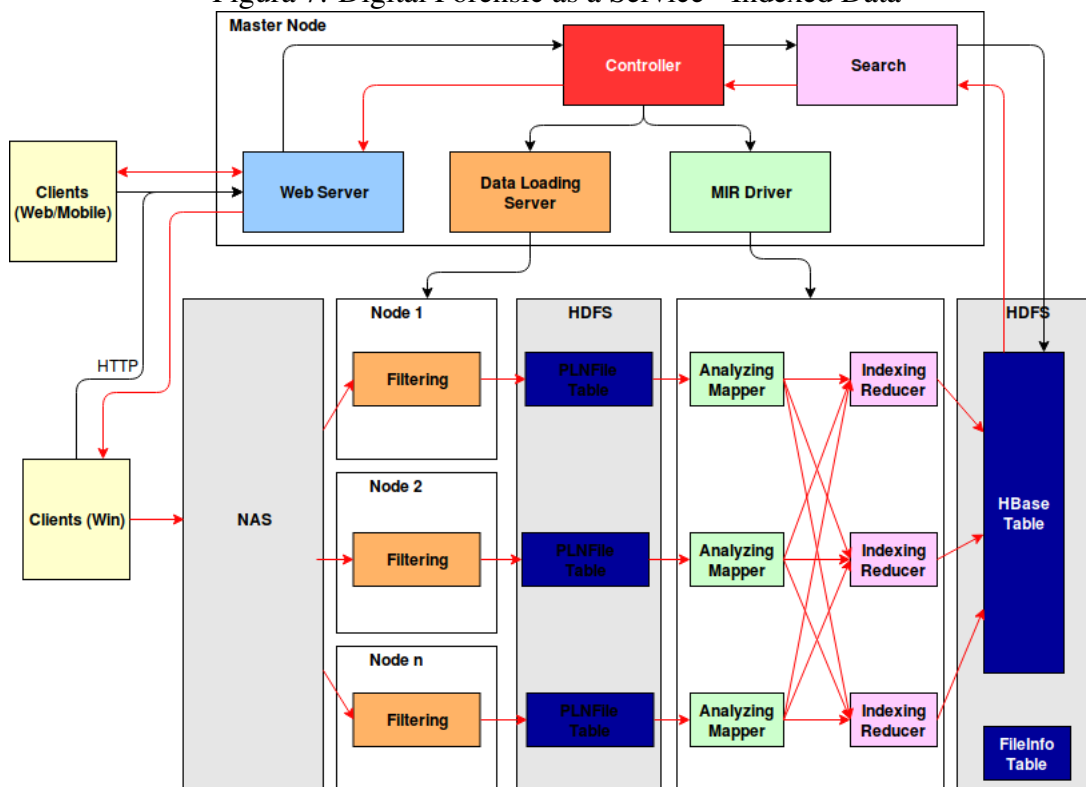


Fonte: (GEORGE; VENTER; THOMAS, 2012)

### 3.1 Acessar e coletar as informações de memória das máquinas virtuais em nuvem

Diversos trabalhos de análise forense na nuvem se concentram na coleta de dados “após o fato”, ou seja, após a intrusão ser detectada (REICHERT; RICHARDS; YOSHIGOE, 2015; POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013; GEORGE; VENTER; THOMAS, 2012; SANG, 2013). Os processos de coleta descritos nesses trabalhos podem ser iniciados de forma manual ou automaticamente, via integração com um mecanismo de detecção de intrusão. No caso específico de me-

Figura 7: Digital Forensic as a Service - Indexed Data



Fonte: (LEE; UN, 2012)

mória volátil, tal forma de coleta não consegue descrever como era a memória antes da intrusão, pois o processo só é acionado depois da detecção do ataque. Tal limitação pode trazer prejuízos à investigação, dado que algumas análises dependem exatamente da capacidade de se comparar dois momentos da memória (CASE et al., 2014). Entre os trabalhos estudados, a única proposta encontrada que leva tal necessidade em consideração é (DEZFOULI et al., 2012), que propõe que o dado seja armazenado no próprio equipamento sob análise. Infelizmente, entretanto, a aplicação de tal abordagem no cenário em nuvem é pouco viável, pois pode levar à perda de informações importantes caso a máquina virtual ou contêiner seja desativada, tendo seus recursos liberados.

Existem ainda trabalhos voltados à coleta de informações durante a execução do sistema, nos quais os dados são constantemente coletados sem distinção do que acon-

teceu antes ou depois do fato de interesse. Esse é o caso de trabalhos como (POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013; SANG, 2013; DOLAN-GAVITT et al., 2011), que adotam a estratégia de isolar e parar a máquina virtual para em seguida realizar o processo de coleta. Embora interessantes, as abordagens descritas nesses trabalhos podem levar a um elevado volume de dados coletados, além de também não tratarem o cenário em que é necessário coletar evidências quando os recursos virtuais contendo tais informações são liberados.

### **3.2 Capacidade de reproduzir o processo e obter os mesmos resultados**

Se, durante uma análise forense, analistas diferentes obtêm resultados distintos ao executar o mesmo procedimento de coleta, a evidência gerada não tem credibilidade, inviabilizando seu uso em um processo legal. Por essa razão, a reprodutibilidade do processo de coleta é uma parte importante da geração de evidências para análise forense. Infelizmente, entretanto, nenhuma das propostas encontradas na literatura atualmente permite tal reprodutibilidade em cenários de nuvem em que máquinas virtuais ou contêineres são desativados e seus recursos físicos liberados: todas elas dependem da existência do recurso virtual para a repetição do processo de coleta.

### **3.3 Não violar privacidade ou jurisdição das partes não envolvidas na investigação**

Em um ambiente de nuvem pública, remover o *hardware* para análise posterior pode levar à violação de privacidade de usuários, uma vez que o multi-inquilinato desse cenário faz com que uma mesma máquina física guarde informações de diversos clientes, alguns dos quais podem não estar envolvidos na investigação em curso. Diversos trabalhos na literatura tratam esse problema adequadamente, por meio das duas estratégias principais: a primeira, adotada em (REICHERT; RICHARDS; YOSHIGOE,

2015; GEORGE; VENTER; THOMAS, 2012; POISEL; MALZER; TJOA, 2013; DYKSTRA; SHERMAN, 2013; LEE; UN, 2012), consiste em coletar dados pertinentes à investigação e armazená-los fora da nuvem; a segunda, empregada em (SANG, 2013) e que constitui um caso específico de (GEORGE; VENTER; THOMAS, 2012), depende da cooperação do provedor de serviços de nuvem para conseguir as informações necessárias à investigação. Depender do provedor de serviços de nuvem é uma estratégia pouco recomendada, entretanto, pois (1) o volume de dados de usuários pode forçar os provedores a limitar o tamanho dos *logs* armazenados, e (2) caso ocorra uma indisponibilidade causada por um ataque, o objetivo do provedor será o de restabelecer o serviço, não necessariamente o de preservar evidências (ALQAHTANY et al., 2015).

### **3.4 Garantir a cadeia de custódia da evidência**

Dentre os trabalhos analisados, apenas (SANG, 2013) aborda a questão da garantia da cadeia de custódia. Especificamente, o trabalho emprega *hashes* para verificar a integridade da evidência, permitindo a detecção de alterações na mesma, embora não explique os mecanismos que poderiam ser utilizados para impedir acesso não autorizado (e, assim, potencial alteração) aos próprios *hashes*. As propostas dos outros autores concentram-se apenas no aspecto técnico da coleta, sem discutir claramente garantia de custódia mas apenas mencionando que as evidências devem ser coletadas de forma “forensicamente aceitável”.

### **3.5 Resumo**

A Tabela 1 mostra um comparativo das soluções estudadas, considerando os aspectos discutidos nesta seção, posicionando as contribuições da proposta apresentada neste trabalho.



## 4 PROPOSTA DE PROJETO: DIZANG

A presente proposta tem como objetivo principal coletar memória de recursos computacionais virtuais de modo a conseguir: (1) identificar a fonte da evidência, mesmo se o recurso virtual não existir mais; (2) descrever o sistema antes e depois do incidente; (3) transportar e armazenar a memória coletada de uma forma que garanta sua integridade e confidencialidade; e (4) não violar a jurisdição e a privacidade de outros usuários que porventura tenham recursos alocados no mesmo servidor físico. A solução aqui apresentada, denominada Dizang , é descrita em detalhes a seguir.

### 4.1 Descrição

Em sistemas computacionais executados sobre uma infraestrutura física (i.e., não virtualizada), pode-se fazer uma associação direta entre um recurso qualquer, como uma informação da memória, imagem de disco ou pacotes trafegando na rede, e sua origem correspondente. Já em sistemas construídos sobre uma infraestrutura virtual, em especial quando esta é auto-escalável, os recursos computacionais são altamente voláteis e, portanto, podem ser desalocados a qualquer momento. Para conseguir correlacionar uma evidência a sua origem volátil, é necessário utilizar outro elemento em que persista a relação fonte-evidência, o que na presente proposta é feito por meio de contêineres linux (LXC). Embora um contêiner seja um *software* e, portanto, também volátil, cada imagem compilada e sua execução na forma de contêiner são normalmente atrelados a um *hash* que identifica univocamente essa relação. O contêiner



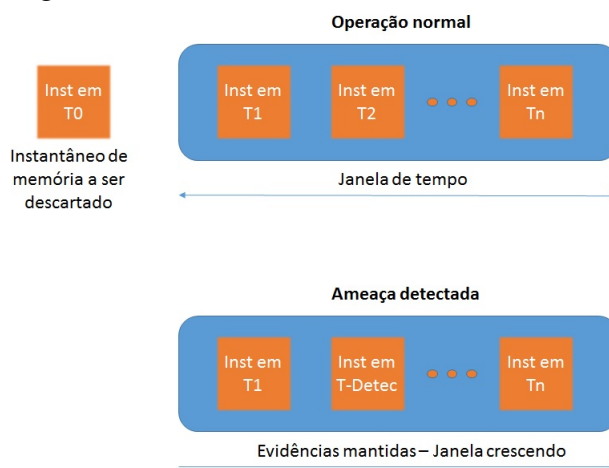
também permite identificar com mais precisão a fonte de uma evidência, uma vez que é possível dividir as partes de um sistema em contêineres: por exemplo, um contêiner para o motor de páginas dinâmicas (e.g., Apache), outro com a lógica de negócios (e.g., *golang*) e um terceiro para um banco de dados (e.g., *Cassandra*).

A cópia de memória não é uma atividade atômica, pois ela é executada em conjunto com outros processos. Portanto, caso um desses processos seja um código malicioso apagando traços de sua existência da memória do contêiner, informações possivelmente importantes para a investigação podem acabar sendo perdidas. Com o objetivo de deixar o processo de cópia da memória mais atômico, Dizang interrompe temporariamente a execução do contêiner, realiza a cópia de sua memória, e em seguida retoma sua execução. Essa técnica, que é semelhante àquela adotada em (RAFIQUE; KHAN, 2013) para VMs, produz um instantâneo da memória volátil do contêiner; isso permite sua análise em um estado de repouso, ou seja, sem a necessidade de ter o contêiner em execução. Ao realizar a coleta em intervalos de tempo adequados, é possível construir um histórico do estado da memória durante a execução no contêiner.

A maioria das técnicas forenses mais usadas atualmente são voltadas à obtenção da informação em sua totalidade, seja via cópia bit a bit, seja por meio da obtenção do *hardware* físico (SIMOU et al., 2014) (BEM et al., 2008). Embora tais técnicas possam parecer interessantes à primeira vista, elas muitas vezes acabam sendo responsáveis por um problema: o crescente volume de informações que os investigadores precisam analisar (QUICK; CHOO, 2014). Para mitigar essa dificuldade, em Dizang são adotadas duas estratégias: a primeira é a definição de um volume de dados que possa ser considerado *suficiente* para a realização de uma investigação; a segunda é a definição de uma *idade máxima* para a evidência enquanto o sistema trabalha em condições normais, isto é, quando não está sob ataque. Para detectar e analisar intrusões na memória de processos, é necessário ter uma cópia da memória antes e depois da intrusão (CASE et al., 2014). Assim, a solução proposta implementa uma janela

de instantâneos de memória cobrindo um intervalo de tempo pré-definido, como ilustrado na Fig. 8. Em condições normais de operação, as evidências são coletadas com certa periodicidade e coletas que atingem uma determinada idade são descartadas. Em contraste, após a detecção de um evento de ataque (e.g., por um sistema de detecção de intrusões), Dizang deixa de descartar as coletas mais antigas do *log* de monitoramento, sendo possível conhecer o sistema antes e depois do ataque e, assim, avaliar sua evolução.

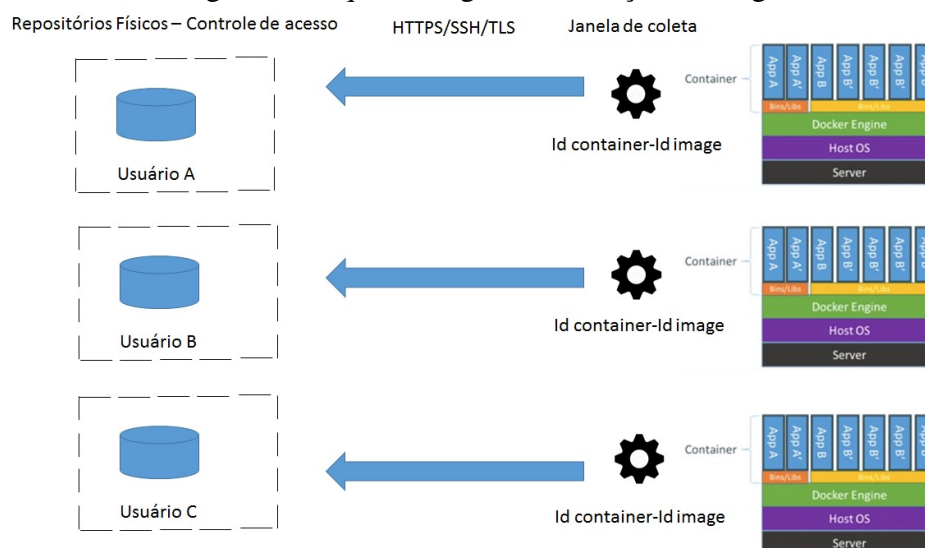
Figura 8: Janela deslizante de coleta de evidência



Fonte: Próprio autor

Para persistir a relação evidência-origem e garantir a integridade da mesma, a presente proposta calcula o hash do par [evidência, identificador da imagem do contêiner] e armazena a tripla [hash, identificador da imagem do contêiner, evidência]. Para evitar eventuais problemas com o armazenamento desses dados em países com jurisdições diferentes daquelas que devem ser aplicadas na investigação em questão, as evidências coletadas são armazenadas em um local físico fora da nuvem, após serem transportadas por meio de um canal seguro (e.g., via *Transport Layer Security* – TLS (Dierks T, 2008)).

Figura 9: Arquitetura geral da solução Dizang



Fonte: Próprio autor

## 4.2 Implementação

Os métodos propostos foram implementados em uma plataforma de testes visando avaliar a eficácia de Dizang em coletar as informações de memória dos contêineres de forma reprodutível, sem violar jurisdições ou a privacidade de usuários. A solução, ilustrada na Fig. 9, consistiu na criação de 1 VM usando o Oracle Virtual Box 5.0 em um notebook Intel i5 de 2.30Mhz e 4Gb de RAM com sistema operacional de 64 bits. Essas VMs possuem 2 Gb de memória RAM e emulam apenas 1 processador, e em cada uma delas foi instalado o Docker Engine 1.10 e a API Docker 1.21, com os quais foram criados 3 contêineres executando o nginx 1.0 em diferentes portas. Usando uma aplicação Java que descobre o identificador de processo associado a cada contêiner, pode-se copiar conteúdo do *descritor de alocação de memória não uniforme* (`/proc/pid/numa_maps`), o qual contém a alocação das páginas de memória, os nós que estão associados a essas páginas, o que está alocado e suas respectivas políticas de acesso (Unix Man Pages, e). A cópia e gravação do arquivo é tal que, a cada minuto, a aplicação (1) pausa o contêiner em questão, (2) copia a diretório **numa\_maps**, (3)

concatena os dados obtidos com o *hash* de identificação da imagem do contêiner, (4) calcula o *hash* do conjunto e (5) salva o resultado em um arquivo cujo nome é o identificador da imagem do contêiner e a extensão é **.mem**. Após a conclusão do processo de cópia, a mesma aplicação verifica se existem arquivos **.mem** em disco mais antigos que um certo intervalo de tempo “t”, descartando-os.

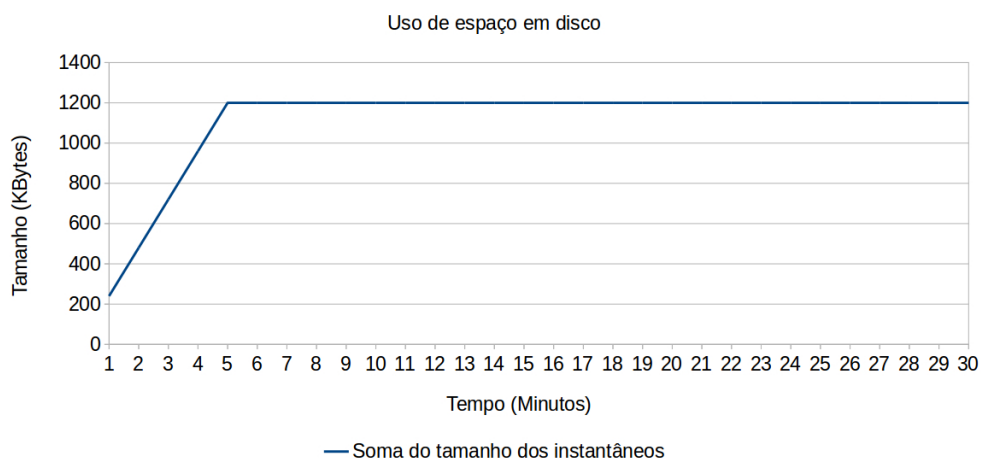
### 4.3 Resultados experimentais

Para avaliar a efetividade de Dizang na coleta de evidências, alguns experimentos foram realizados usando o ambiente implementado (descrito na Seção 4.2). Primeiramente, o sistema foi configurado para realizar coletas de memória em intervalos de 1 minuto, salvá-las em disco externo e apagar amostras coletadas há mais de 5 minutos. O sistema foi então executado por 30 minutos, tempo durante o qual foram coletadas como métricas (1) o uso de espaço em disco utilizado pelos instantâneos de memória salvos e (2) o tempo de pausa no contêiner necessário para a cópia das mesmas. A cada coleta, foi executado o comando `du -sh *.mem` do *Unix* no disco de armazenamento externo, para retornar a lista dos arquivos onde os instantâneos de memória foram armazenados e o espaço em disco ocupado pelos mesmos. Ao fim do experimento, os contêineres foram removidos.

A ocupação em disco devido aos instantâneos de memória capturados durante o experimento é mostrada no gráfico da Fig. 10. O gráfico mostra que o aumento do uso do espaço em disco é linear e o crescimento se interrompe quando é atingido o limite de tempo configurado para a janela, pois as coletas com tempo de vida maior que tal limite são apagadas do disco. Assim, que a solução mantém sob controle o espaço em disco ocupado pelas amostras coletadas. Ao mesmo tempo, instantâneos de memória salvos pela solução depois que os contêineres são removidos continuam no disco da máquina, podendo ser associados a sua origem (i.e., contêiner e imagem), conforme esperado para uma análise forense. Essa capacidade se mantém após a detecção de

uma ameaça, pois nesse caso coletas mais antigas deixam de ser apagadas. Logo, é possível descrever o estado do sistema antes e depois do incidente (CASE et al., 2014), permitindo-se, por exemplo, que ataques de injeção de código em memória sejam analisados.

Figura 10: Evolução do uso do espaço em disco com o Dizang



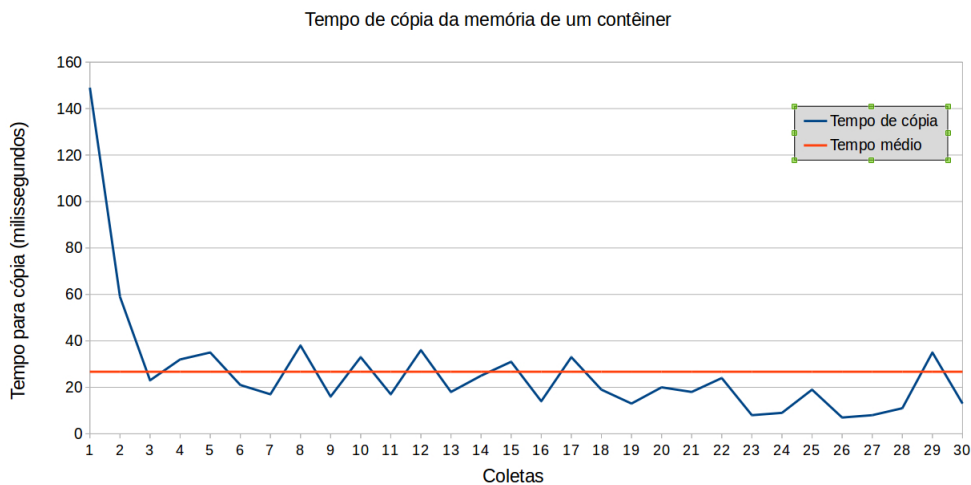
Fonte: Próprio autor

Uma potencial limitação da solução proposta é que a pausa de um contêiner para coleta de dados poder, em princípio, causar perdas no desempenho da aplicação sendo executada. Para avaliar esse impacto, durante o experimento foram medidos os tempos de cópia da memória do contêiner. Os resultados são mostrados no gráfico da Fig. 11. É possível notar que, após a inicialização da aplicação, o tempo para realizar a cópia é bastante reduzido, variando entre 20 e 40 milissegundos. Em especial, para contêineres executando um motor de páginas web dinâmicas, como é o caso do experimento em questão, essa latência deve ser pouco perceptível por usuários finais.

## 4.4 Limitações

Como a solução descrita tem como foco coletar informações de memória no espaço do usuário (*user space*), ela não consegue acessar o espaço de kernel (*kernel space*). Assim, Dizang em princípio não provê suporte a técnicas de investigação de

Figura 11: Tempo de cópia da memória de um contêiner



Fonte: Próprio autor

malware que se baseiam em informações do *kernel space*, como, por exemplo, a comparação de informações do bloco do ambiente do processo (*Process Environment Block* – *PEB*), que ficam no *user space*, com informações do descritor de endereços de memória virtual (*Virtual Address Descriptor* – *VAD*), que fica no *kernel space*. Análise de ameaças que realizam manipulação direta dos objetos do kernel (*D.K.O.M.* – *Direct Kernel Object Manipulation*) também não se beneficiam com a solução aqui proposta.

## 4.5 Conclusões e trabalhos futuros

Ameaças digitais que atuam diretamente na memória de sistema não costumam deixar rastros em disco após terem os recursos correspondentes desalocados, dificultando análises forenses posteriores. Esse problema é especialmente notável em sistemas de computação em nuvem, nos quais a alocação e desalocação de recursos virtualizados (e.g., VMs e contêineres) é frequente. Essa característica, aliada a aspectos como multi-inquilinato e multi-jurisdição de nuvens computacionais, dificulta a coleta de evidências para a investigação de incidentes.

Nesse cenário, a proposta apresentada visa relacionar o instantâneo de memória a

sua origem, utilizando o *hash* calculado da imagem do contêiner como identificador da evidência armazenada. Para evitar uso excessivo de memória, a quantidade de dados armazenados usa uma janela de armazenamento, o que permite descrever a memória antes e depois de um ataque (e.g., de injeção de memória). Combinada com uma ferramenta para identificação de ameaças, essas características de Dizang o transformam em uma solução poderosa para prover evidências e, assim, viabilizar análises forenses na nuvem.

Como trabalho futuro, pretende-se estudar o impacto provocado pela pausa do contêiner no desempenho de um sistema em produção e realizar uma análise da memória dos contêineres deste mesmo sistema utilizando as coletas feitas por Dizang . Com esses resultados pretende-se implementar uma ferramenta mais flexível, que não precise ter acesso a memória completa da máquina, requisito comum de ferramentas de análise de malware atuais (e.g., FROST (DYKSTRA; SHERMAN, 2013) e Volatility Framework (Volatility Foundation, 2014)).

## **4.6 O que foi feito e cronograma esperado para os próximos passos**

A presente proposta está planejada para ser executada em 3 fases. A primeira fase tem por objetivo encontrar uma forma de relacionar a evidência coletada de um contêiner a sua origem de forma que o processo seja reproduzível. Para atingir este objetivo será implementado um protótipo de coleta de memória de uma máquina virtual executando contêineres em um notebook. A segunda fase tem por objetivo encontrar uma forma de transportar a evidência coletada a um local de armazenamento garantindo a cadeia de custódia. Para atingir este objetivo, será definida uma cadeia de custódia. De posse desta será realizada em uma nuvem computacional e envolverá o transporte da evidência para uma máquina física fora da nuvem. A terceira parte tem por objetivo a realização da análise de alguma vulnerabilidade utilizando-se do material coletado

na primeira fase do projeto. As ações para se atingir este objetivo serão realizadas em um notebook com ferramental voltado a análise forense em memória. Até então foram realizados com sucesso a duas primeiras partes do projeto.

Parte 1: A associação da evidência de memória coletada do contêiner de uma máquina virtual foi associada a sua origem através do *hash* de identificação da imagem do contêiner. O processo de coleta foi reproduzido com sucesso diversas vezes.

Parte 2: O transporte da evidência para uma máquina física fora da nuvem utilizando camada de transporte seguro a assinatura da mensagem.

Para o restante do projeto, que consiste da parte 3, espera-se seguir o cronograma descrito na figura 12 onde serão executadas as seguintes atividades.

- **Estudo das ferramentas de análise de memória:** Onde serão avaliadas ferramentas de análise de memória na busca por alguma que consiga decodificar as informações coletadas
- **Implementação / seleção de uma ferramenta de análise:** Nesta atividade será selecionada uma das ferramentas estudadas na atividade anterior. Caso nenhuma se mostre adequada, será implementada uma ferramenta para o propósito da fase 3
- **Realização de análises com a ferramenta escolhida / criada:** Nesta fase ocorrerá a tentativa de analisar um malware encontrado em alguma coleta realizada na fase 1

## 4.7 Contribuições

A contribuição esperada da presente proposta é o de demonstrar que é possível coletar evidências de uma infraestrutura em nuvem de modo a atender os pré-requisitos de



Figura 12: Cronograma de projeto

Atividade / mês	2018												2019							
	jan	fev	mar	abr	mai	jun	jul	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul	ago
Acompanhamento do estado da arte																				
Estudo das ferramentas de análise de memória																				
Implementação / seleção de uma ferramenta de análise																				
Realização de análises com a ferramenta escolhida / criada																				
Escrita e defesa de tese																				

Fonte: Próprio autor

não violação de privacidade, não violação da jurisdição, reprodutibilidade do processo de coleta da evidência e garantia da cadeia de custódia. Demonstrar que é possível realizar a análise de evidência de memória com apenas a parte *user space* da memória coletada

## REFERÊNCIAS

- ALQAHTANY, S. et al. Cloud forensics: A review of challenges, solutions and open problems. In *Int. Conference on Cloud Computing (ICCC)*. [S.l.]: IEEE, 2015. p. 1–9.
- AMAZON. *What are Containers*. 2016. Available from Internet: <<https://aws.amazon.com/containers>>.
- BEM, D. et al. Computer forensics - past , present and future. *Journal of Information Science and Technology*, vol. 5, no. 3, p. 43–59, 2008.
- CASE, A. et al. *The Art of Memory Forensics: Detecting malware and threats in Windows, Linux and Mac memory*. [S.l.]: Wiley, 2014.
- DEZFOULI, F. et al. Volatile memory acquisition using backup for forensic investigation. In *Int. Conf. on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. [S.l.: s.n.], 2012. p. 186–189.
- Dierks T, R. E. *The Transport Layer Security (TLS) Protocol*. [S.l.]: IETF, 2008. IETF: <<https://tools.ietf.org/html/rfc5246>>.
- DOLAN-GAVITT, B. et al. Virtuoso: Narrowing the semantic gap in virtual machine introspection. In *IEEE Symposium on Security and Privacy*. [S.l.: s.n.], 2011. p. 297–312. ISSN 1081-6011.
- DYKSTRA, J.; SHERMAN, A. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation*, Elsevier Ltd, vol. 9, p. S90–S98, 2012. ISSN 17422876. (Proc. of the 12th Annual DFRWS Conference). Available from Internet: <[dx.doi.org/10.1016/j.diin.2012.05.001](https://doi.org/10.1016/j.diin.2012.05.001)>.
- DYKSTRA, J.; SHERMAN, A. T. Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform. *Digital Investigation*, Elsevier Ltd, vol. 10, p. S87–S95, 2013. ISSN 17422876. (Proc. of 13th Annual DFRWS Conference). Available from Internet: <[dx.doi.org/10.1016/j.diin.2013.06.010](https://doi.org/10.1016/j.diin.2013.06.010)>.
- GEORGE, S.; VENTER, H.; THOMAS, F. Digital Forensic Framework for a Cloud Environment. In *IST Africa*. Tanzania: IIMC, 2012. p. 1–8. ISBN 9781905824342.
- GILBERT, P.; SUJEET, S. *Advances in Digital Forensics IV*. 1. ed. Orlando: Springer-US, 2008. vol. 1. ISSN 1098-6596. ISBN 9788578110796.
- GRISPOS, G.; STORER, T.; GLISSON, W. Calm before the storm: the challenges of cloud computing in digital forensics. *International Journal of Digital Crime and Forensics*, vol. 4, no. 2, p. 28–48, 2012. ISSN 1466640073. Available from Internet: <[www.igi-global.com/article/calm-before-storm/68408](http://www.igi-global.com/article/calm-before-storm/68408)>.

LEE, J.; UN, S. Digital Forensics as a Service: A case study of forensic indexed search. p. 499–503, 2012.

LINUXCONTAINERS.ORG. *Linux Containers (LXC)*. 2015. Available from Internet: [<https://linuxcontainers.org/lxc/introduction/>](https://linuxcontainers.org/lxc/introduction/).

MELL, P.; GRANCE, T. *The NIST definition of cloud computing*. 2011. 7 p. NIST SP 800-145. [csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf](https://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf). Available from Internet: [www.mendeley.com/research/the-nist-definition-about-cloud-computing/](http://www.mendeley.com/research/the-nist-definition-about-cloud-computing/).

MORSY, A. M.; GRUNDY, J.; MULLER, I. An Analysis of the Cloud Computing Security Problem. In *APSEC Cloud Workshop*. Sydney, Australia: [s.n.], 2010. Available from Internet: <https://arxiv.org/abs/1609.01107>.

POISEL, R.; MALZER, E.; TJOA, S. Evidence and cloud computing: The virtual machine introspection approach. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1, p. 135–152, 2013. ISSN 20935374 (ISSN). Available from Internet: [citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.469.937](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.469.937).

QUICK, D.; CHOO, K. K. R. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, Elsevier Ltd, vol. 11, no. 4, p. 273–294, 2014. ISSN 17422876. Available from Internet: [dx.doi.org/10.1016/j.diin.2014.09.002](https://doi.org/10.1016/j.diin.2014.09.002).

RAFIQUE, M.; KHAN, M. N. A. Exploring Static and Live Digital Forensics: Methods, Practices and Tools. *IJSER*, vol. 4, no. 10, p. 1048–1056, 2013. Available from Internet: [www.ijser.org/researchpaper/5CEExploring-Static-and-Live-Digital-Forensic-Methods-Practices-and-Tools](http://www.ijser.org/researchpaper/5CEExploring-Static-and-Live-Digital-Forensic-Methods-Practices-and-Tools).

RAHMAN, S.; KHAN, M. N. A. Review of live forensic analysis techniques. *International Journal of Hybrid Information Technology*, vol. 8, no. 2, p. 379–388, 2015. Available from Internet: [www.sersc.org/journals/IJHIT](http://www.sersc.org/journals/IJHIT).

RAMOS, M. G. *Do Valor Probatório do Arquivo Digital*. PhD Thesis (PhD) — Universidade de Brasília, 2011.

REICHERT, Z.; RICHARDS, K.; YOSHIGOE, K. Automated forensic data acquisition in the cloud. *IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems*, p. 725–730, 2015.

Right Scale. *State of the Cloud Report*. 2017. [www.rightscale.com/lp/state-of-the-cloud](http://www.rightscale.com/lp/state-of-the-cloud).

SANG, T. A log-based approach to make digital forensics easier on cloud computing. *Intelligent System Design and Engineering Applications (ISDEA)*, p. 91–94, 2013.

SIMOU, S. et al. Cloud forensics: Identifying the major issues and challenges. In *Advanced Information Systems Engineering (CAiSE 2014)*. [S.l.: s.n.], 2014. vol. 8484, p. 271–284. ISBN 9783319078809. ISSN 16113349.

Unix Man Pages. *cgroups - Control Groups*. <<http://man7.org/linux/man-pages/man7/cgroups.7.html>>.

\_\_\_\_\_. *chroot - Change Root coomand*. <<http://man7.org/linux/man-pages/man1/chroot.1.html>>.

\_\_\_\_\_. *dd - convert and copy a file*. <<http://man7.org/linux/man-pages/man1/dd.1.html>>.

\_\_\_\_\_. *Namespacing*. <<http://man7.org/linux/man-pages/man7/namespaces.7.html>>.

\_\_\_\_\_. *Numa Maps - Non Uniform Memory Architecture*. <[man7.org/linux/man-pages/man7/numa.7.html](http://man7.org/linux/man-pages/man7/numa.7.html)>.

Volatility Foundation. *Volatility Framework*. 2014. <[www.volatilityfoundation.org/](http://www.volatilityfoundation.org/)>.

WIKIPEDIA. *MapReduce*. 2018. Available from Internet: <<https://en.wikipedia.org/wiki/MapReduce>>.