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 realização 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 realização 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 proporcionalmente aumenta também o número de casos em que esse tipo de tecnologia é usada para fins ilícitos. Infelizmente, devido à natureza volátil da nuvem, a tarefa de coletar evidências para análise forense nesse ambiente tem esbarrado em desafios práticos e legais. Mais precisamente, a prática herdada da forense tradicional para coleta de evidências, por meio da qual se isola a cena do crime e coletam-se todas as evidências, foi traduzida para a forense digital como a cópia bit a bit da mídia que se deseja investigar. Tal prática leva à coleta de grandes volumes de informação para análise, impactando negativamente o tempo de investigação. Além disso, duas características das soluções em nuvem dificultam a obtenção de evidências válidas. A primeira é que o compartilhamento de recursos físicos entre vários usuários impede a sua remoção para análise, uma vez que isso violaria a privacidade de indivíduos não envolvidos na investigação. A segunda é que a localização do recurso físico em uma região geográfica diferente daquela 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 colocar em xeque a credibilidade das evidências. Este trabalho analisa propostas na literatura voltadas a resolver tais desafios na coleta evidências na nuvem, discutindo suas limitações. Propõe-se então uma solução que cobre coleta, transporte e armazenamento da evidência, visando suplantar as limitações existentes no estado da arte. A solução proposta, denominada Dizang , provê uma forma de correlacionar evidências e sua origem virtual, permitindo transportar e armazenar tais dados sem afetar sua credibilidade. Para tal, é feito o uso de contêineres e seus respectivos identificadores, técnica esta que tem a vantagem que conseguir preservar a relação evidência-origem mesmo que esta última não exista mais na solução sob investigação. Em resumo, Dizang tem como focos (1) a reprodutibilidade do processo de coleta, (2) o estabelecimento de um vínculo entre a evidência coletada e sua origem, (3) a preservação da jurisdição e da privacidade de usuários não envolvidos na investigação e (4) a garantia de custódia da evidência.

ABSTRACT

MARCOS: Reescreva considerando as revisões na parte em português - Hamilton: Feito The adoption of cloud architectures increases every day, and with it increases the number of cases in which this type of technology is used for illicit purposes. Unfortunately, due to the volatile nature of the cloud, the task of collecting evidences for forensic analysis in this environment has run into practical and legal challenges. The practice inherited from the traditional forensics, through which the crime scene is isolated and all the evidence is collected, when applied to digital forensics, led to the collection of large volumes of information for analysis. Such practice had a negative impact on the time of investigation. In addition, two characteristics of cloud solutions make it difficult to obtain valid evidence. The first is that the physical resources are shared among multiple users which prevents their removal for analysis, as this would violate the privacy of individuals not involved in the investigation. The second is the resource's physical location which can be in a different geographical region where the crime was committed and may prevent the collection of evidence if there are no cooperation agreements in place. These aspects, if not taken into consideration may call into question the credibility of the evidence. This work analyzes proposals in the literature aimed at solving such challenges, discussing its limitations and proposes a solution that covers collection, transportation and storage of evidence, aiming at overcoming existing limitations in the state of art. The proposed solution, called Dizang, provides a way of correlating evidence and its virtual origin, allowing the transport and storage of such data without affecting its credibility. For this, containers are used for their ability to preserve the relationship evidence-source even though the latter does not exist in the system under investigation. In summary, Dizang focuses on (1) the reproducibility of the collection process, (2)the establishment of a link between the evidence collected and its origin, (3) the preservation of jurisdiction and privacy of users not involved in the investigation and (4)the evidence's chain of custody.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

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

SUMÁRIO

1 INTRODUÇÃO

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 que a hospedam e balancear a carga entre elas. Isso permite ao sistema em nuvem atender à demanda dos usuários da aplicação 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 desperdício de recursos.

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, pode-se supor um cenário em que 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., em arquivos de *log*). Nesse caso, as evidências desse evento podem ser completamente perdidas após as instâncias em questão 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 multi-inquilinato dificulta a obtenção do *hardware* que executa as aplicações de interesse. Afinal, 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 em vários países, dificultando a sua obtenção, em especial quando não existem acordos de cooperação entre as entidades envolvidas (DYKSTRA; SHERMAN, 2012). Combinadas, tais características limitam a coleta de evidências com a credibilidade necessária para que elas possam ser usadas em processos legais, que exigem o respeito à privacidade, à jurisdição e à cadeia de custódia, bem como a reprodutibilidade do processo de coleta (RAHMAN; KHAN, 2015).

1.1 Motivação

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 em tempo de execução, enquanto trabalhos como (SANG, 2013) abordam processos de garantia da 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 máquina virtual) seja desalocado, haja condições de se reproduzir o processo de coleta de evidências.

1.2 Objetivos

O presente trabalho visa suplantar as limitações identificadas na literatura com relação à 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 um 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. Desta forma, a solução buscada deve prover uma forma de obter dados relevantes para investigações ao mesmo tempo que permita transportar e armazenar tais dados preservando sua credibilidade. Para isso, a proposta supõe que o sistema sendo monitorado é executado dentro de um ambiente virtualizado em nuvem, tendo como foco a tecnologia de contêineres.

1.3 Justificativa

O crescente volume de informações que as soluções em nuvem armazenam e trafegam atualmente, bem como os aspectos de multi-inquilinato e a multi-jurisdição dos provedores de nuvem, estão entre os principais obstáculos enfrentados pelos investigadores forenses. (QUICK; CHOO, 2014) (GILBERT; SUJEET, 2008). Desta forma, a criação de soluções que permitam melhorar o processo de investigação forense nesse cenário tem relevância não apenas teórica, mas também prática.

Ao mesmo tempo, virtualização tem sido amplamente adotada por empresas das mais diversas áreas. Segundo o ”State of the Cloud Report”, relatório produzido pela empresa Right Scale (Right Scale, 2017), 95% das organizações entrevistadas estão utilizando ou experimentando soluções em nuvem no modelo IaaS (*Infrastructure as a Service* – Infraestrutura como um Serviço). Embora a virtualização de recursos na nuvem seja tradicionalmente feita por meio de máquinas virtuais, tem ganhado importância também a tecnologia de contêineres (BERNSTEIN, 2014), uma forma de

virtualização bastante leve e flexível. De fato, conforme o "Container Market Adoption Survey 2016", realizado 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. Essa tendência motiva a construção de soluções que levem em consideração as particularidades inerentes a contêineres computacionais, e que sejam capazes de tirar vantagem de suas características.

1.4 Organização do documento

O restante deste documento está organizado conforme os 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 as atividades realizadas até o presente momento como parte do trabalho de pesquisa aqui apresentado, bem como os próximos passos esperados para a sua conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Desde 2001, diversos modelos para a condução de investigação digital foram propostos. O *Enhanced Digital Investigation Process Model*, proposto por Carrier e Stafford em 2003, é a última iteração na evolução do processo forense digital (GRISPOS; STORER; GLISSON, 2012). Entretanto, como tais modelos de investigação foram desenvolvidos antes da aparição de tecnologias de computação em nuvem, muitos partem do pressuposto que o investigador tem acesso e controle sobre o sistema sob investigação (GRISPOS; STORER; GLISSON, 2012). Esta defasagem é um dos desafios da forense digital atual.

Neste capítulo, são discutidos os principais conceitos que permeiam esse cenário. Especificamente, após uma discussão geral sobre computação em nuvem e suas particularidades, incluindo o uso de contêineres, são discutidas os características esperadas de um processo de forense digital robusto.

2.1 Nuvens computacionais e contêineres

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. Existem basicamente três modelos principais de comercialização de uso da nuvem (MELL; GRANCE, 2011):

- SaaS (*Software as a Service* – *Software* como serviço): Onde se provê o *software* que será utilizado; nesse caso, os clientes do serviço são os usuários finais o software.

- PaaS (*Platform as a Service* – Plataforma como serviço): Onde se provê o ambiente para o desenvolvimento, teste e execução do *software*; nesse caso, os clientes do serviço são desenvolvedores de aplicações.

- IaaS (*Infrastructure as a Service* – Infraestrutura como serviço): Onde são fornecidos recursos computacionais básicos, como processamento, memória e redes, em geral de forma virtualizada; os clientes desse tipo de serviço costumam ser arquitetos de sistemas.

O tipo de serviço de nuvem mais pertinente para este trabalho é o IaaS, uma solução muito usada atualmente pela sua capacidade de prover recursos sob demanda de forma auto-escalável. Nesse cenário, o uso intenso de tecnologias de virtualização costuma levar a recursos altamente voláteis, que são alocados e desalocados a qualquer momento pelo orquestrador da nuvem para suprir um suprir eventuais aumentos e reduções de demanda. É possível até mesmo construir scripts para a automatizar a construção da arquitetura desejada, permitindo a instanciação e interconexão de máquinas adequadas para a atividade fim do sistema. Esses scripts, escritos em linguagem específica como Puppet (Puppet,), Chef (Chef,) ou Vagrant (Vagrant,), podem conter instruções de como distribuir o tráfego de rede entre as diferentes instâncias de computação ou armazenamento. Dentre as vantagens de sua utilização, podem ser citadas a capacidade de usar os recursos de nuvem de uma forma mais eficiente, levar a uma menor necessidade de intervenção humana, e prover maior resiliência a variações de demanda do sistema.

Uma tecnologia de virtualização possível para cenários de nuvem, e cuja utilização vem crescendo nos últimos anos, são os chamados contêineres (BERNSTEIN, 2014).

Basicamente, um contêiner é um método de virtualização do sistema operacional que permite executar uma aplicação, bem como suas dependências, em um processo no qual recursos como disco, memória e rede permanecem isolados. Diferente das máquinas virtuais, a virtualização com contêineres é feita no nível do Sistema Operacional (SO) nativo. Como resultado, tem-se uma implementação de virtualização na qual eliminam-se camadas entre o aplicativo executado e o *hardware* físico, permitindo maior granularidade no controle sobre esses recursos e melhorando a eficiência da infraestrutura.

Uma implementação bastante utilizada para esse propósito são os LXC (*LinuX Conteiners* – Contêineres Linux) (LINUXCONTAINERS.ORG, 2015), que aproveitam-se de funcionalidades como *cgroups*, *kernel namespacing* e *chroot* do kernel do Linux para auxiliar no gerenciamento e isolamento de recursos virtuais. Mais precisamente, a funcionalidade de *cgroups* (*Control Groups* – Grupos de Controle) presente no Kernel do Linux limita e isola o uso de recursos como CPU, memória e disco de um conjunto de processos, além de organizá-los de forma hierárquica. O trabalho nessa funcionalidade começou em 2006 na Google, sob a denominação de *process container*. No final de 2007, seu nome foi alterado para *control groups*, e o resultado foi então adicionado à versão 2.6.24 do kernel lançado em 2008 (Unix Man Pages, a).

Já o *Namespacing* é uma funcionalidade do Kernel do Linux usada para isolar e virtualizar recursos do sistema operacional, como identificadores de processos, acessos à rede, comunicação inter-processos e sistema de arquivos. *Namespacing* 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. Desta forma, essa é a principal funcionalidade por trás da implementação de Contêineres Linux (Unix Man Pages, d).

Finalmente, *chroot* (*Change Root* – Mude Root) é uma funcionalidade do Kernel do Linux usada para mudar o diretório *root* enxergado pelo processo que está chamando a função, bem como por 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.

2.2 Forense digital e seus desafios

A área de forense digital (também conhecida por forense computacional) refe-se a 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, estabelecido pelo médico francês Edmond Locard da seguinte forma: "Quando um indivíduo entra em contato com outro objeto ou indivíduo, este sempre deixa vestígio deste contato" (RAMOS, 2011). De forma similar, 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 passados para que suas conclusões sejam validadas por terceiros e sejam aceitas em um processo legal.

A seguir são detalhados os principais desafios enfrentados pela forense digital quando aplicada a infraestruturas em nuvem.

2.2.1 Aceitabilidade da evidência em processo legal.

O processo de análise forense no evento de um crime digital é descrito no EDIPM (*Enhanced Digital Investigation Process Model* – Modelo Melhorado para Processo de Investigação Digital) na forma de 4 fases (GRISPOS; STORER; GLISSON, 2012):

identificar, preservar, examinar e apresentar. A fase mais pertinente a este trabalho é a de preservação da evidência, que deve ser conduzida de forma forensicamente aceitável. Ou seja, deve-se coletar as evidências de forma que elas sejam aceitas em um processo legal e não sejam invalidadas no curso do mesmo.

Para atingir tal objetivo, o primeiro passo é a garantia da cadeia de custódia relacionada a evidência. 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). Uma cadeia de custódia idealmente deve tornar visível o estado da evidência antes, durante e após a interação com o processo de investigação (LUIS; SANCHEZ; GIOVA, 2016). A razão para essa preocupação é que alguns processos investigativos podem causar alteração da evidência na forma que foi coletada, como, por exemplo, a consolidação em um único local dos arquivos que compõem uma base de dados distribuída.

O passo seguinte é a garantia da autenticidade e da integridade da evidência. Autenticidade pode ser definida como "o processo pelo qual se pode garantir a autoria do documento eletrônico" (RAMOS, 2011), ou seja, por meio do qual se não permite dúvida quanto à identificação do autor . Já a integridade pode ser definida como "o atestado da inteireza do documento eletrônico após sua transmissão, bem como apontar eventual alteração irregular de seu conteúdo" (RAMOS, 2011). Caso haja dúvida acerca de um desses requisitos, uma perícia técnica pode ser convocada. Nesse caso, durante a perícia é analisado o autor da evidência, ou seja, verifica-se sua fonte e se a mesma não foi alterada no processo.

Em uma infraestrutura física, a coleta de evidências pode ser feita de forma relativamente simples, bastando-se remover o recurso físico, transportar este para um laboratório e lá analisar os dados. Para limitar a exposição da evidência a manipulações indevidas, ela pode ser mantida em uma sala-cofre, à qual o acesso é controlado. A reprodutibilidade do processo de coleta e a manutenção da integridade da evidência

são, então, tarefas bem diretas.

Já em um cenário de computação em nuvem, especialmente as de infraestrutura auto-escalável, existe um conjunto de novos desafios. Primeiramente, em contraste com infraestruturas tradicionais, o recurso físico em princípio não pode ser removido: como os recursos são utilizados por outros usuários não relacionados à investigação, fazê-lo constituiria violação de privacidade. A volatilidade dos recursos também torna a verificação do seu autor um processo mais complexo, pois o recurso que gera certa evidência pode deixar de existir algum tempo depois de fazê-lo (SIMOU et al., 2014). A integridade da evidência também acaba sendo uma tarefa não trivial, pois ela precisa ser coletada, transportada e armazenada, o que caracteriza a necessidade de preservação da cadeia de custódia. Infelizmente, a violação de qualquer uma dessas características pode colocar em dúvida a credibilidade da evidência.

2.2.2 Volume de dados para coleta

O processo de coleta de evidências na forense digital herda suas práticas da forense tradicional, na qual isola-se cena do crime e coletam-se as evidências presentes. Transportando esse método para a forense digital, introduz-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 era considerada muito problemática. Entretanto, nas atuais soluções, aplicações e arquiteturas em nuvem, o volume de dados é consideravelmente maior (QUICK; CHOO, 2014). Afinal, o acesso fácil e dinâmico a recursos de armazenamento e de processamento, como máquinas virtuais, balanceadores de carga e *firewalls*, acaba também aumentando a quantidade elementos geradores de evidência. Esse problema é ilustrado em (QUICK; CHOO, 2014), onde se discute uma investigação na qual a quantidade de dados a serem analisados para fins de forense digital tomaria 6 meses.

Encontrar uma forma de armazenar menos informações e, assim, tornar a fase de

análise mais rápida e eficiente, é um passo importante para garantir a celeridade de investigações forenses.

2.2.3 Privacidade e jurisdição

MARCOS: Eu acho que já li essa frase... (sim, está repetitivo... resuma em uma frase curta, potencialmente fazendo referência à seção anterior) - Hamilton: Assim? A prática de remoção de equipamentos para coleta de evidências, mencionada na seção 2.2.2, também traz consequências negativas do ponto de vista de privacidade. Mais precisamente, em soluções envolvendo infraestruturas físicas, tal prática não costuma trazer grandes problemas porque os objetos ou indivíduos sob investigação normalmente estão diretamente relacionados ao equipamento removido. Nas soluções em nuvem, entretanto, tal prática não é recomendada porque o recurso físico é compartilhado por vários usuários, inclusive indivíduos não envolvidos na investigação. Logo, remover tais recursos configuraria violação de privacidade. Como um complicador adicional, o fato de os dados potencialmente não estarem armazenados no mesmo território em que a investigação é realizada acaba 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 ganham importância.

2.2.4 Coleta de evidências de memória volátil de máquinas em nuvem

A prática de armazenar histórico de tráfego de rede e alterações de dados armazenados em disco já é bem difundida na comunidade forense. Por outro lado, a memória volátil de computadores não costuma receber o mesmo tratamento: suas alterações quase nunca são armazenadas, seja por questões de desempenho ou por simples pra-

ticidade, dado a reduzida sobrevida dessas informações. Infelizmente, isso acaba dificultando a análise de uma classe específica de ataques, conhecidos como injeção de código em memória (CASE et al., 2014). Assim, quando usados contra uma arquitetura em nuvem, tais ataques não deixam rastros quando recursos de processamento virtuais são desativados e sua memória é liberada (VöMEL; STüTTGEN, 2013; CASE et al., 2014). Em particular, têm especial interesse quatro tipos particulares dessa família de ameaças (CASE et al., 2014):

- **Injeção remota de bibliotecas**: Um processo malicioso força o processo alvo a carregar uma biblioteca em seu espaço de memória. Como resultado, o código da biblioteca carregada executa com os mesmos privilégios do executável em que ela foi injetada. Tal estratégia, comumente usada para instalar malwares, pode fazer com que uma biblioteca maliciosa armazenada no sistema seja distribuída por vários processos de uma mesma máquina, dificultando sua remoção (MILLER; TURKULAINEN, 2004).

- **Inline Hooking**: Um processo malicioso escreve código como uma sequência de bytes diretamente no espaço de memória de um processo alvo, e então força este último a executar o código injetado. O código pode ser, por exemplo, um *script* de *shell*.

- **Injeção reflexiva de biblioteca**: Um processo malicioso acessa diretamente a memória do processo alvo, inserindo nela o código de uma biblioteca na forma de uma sequência de bytes, e então força o processo a executar essa biblioteca. Nessa forma de ataque, a biblioteca maliciosa não existe fisicamente; isso torna tal estratégia de injeção de código potencialmente mais atrativa, pois o carregamento da biblioteca não é registrado no sistema operacional (SO), dificultando a detecção do ataque (FEWER, 2008).

- **Injeção de processo vazio**: Um processo malicioso dispara uma instância de um

processo legítimo no estado “suspenso”; a área do executável é então liberada e realocada com código malicioso.

3 REVISÃO DA LITERATURA

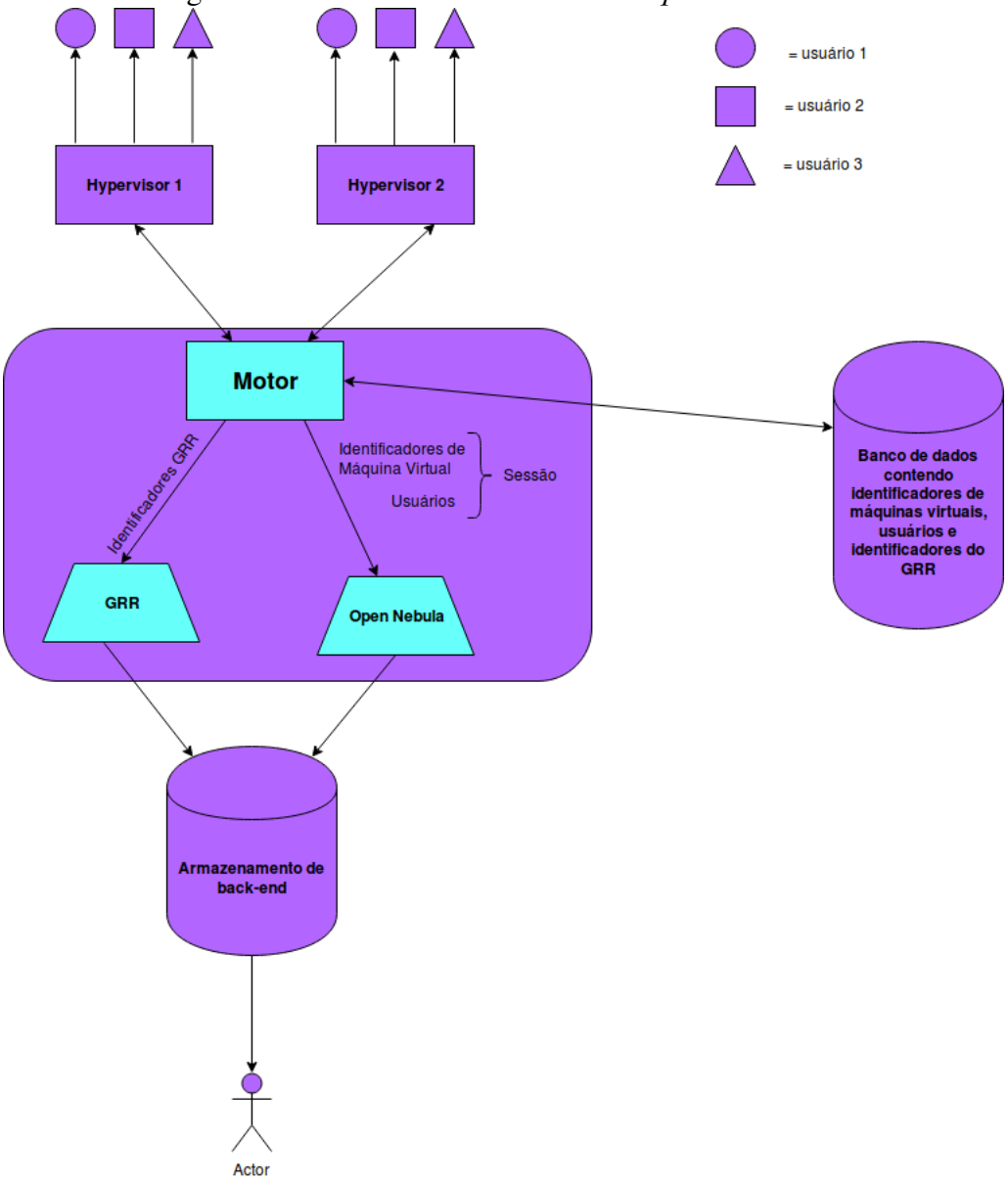
Os principais aspectos relacionados à coleta de evidências para análise forense em nuvem são: coleta, transporte, armazenamento, garantia da cadeia de custódia e reprodutibilidade do processo de coleta. Nesta seção são discutidas propostas encontradas na literatura que envolvem tais aspectos, com o objetivo de dar ao leitor uma visão geral do estado da arte na área.

MARCOS: Separe em seções essa discussão: isso facilita muito a leitura. Também precisa traduzir as figuras: seu texto está em português, então não faz sentido suas figuras estarem em inglês. Isso vale para qualquer figura, tabela, etc., mesmo se forem de outros trabalhos. Simplesmente diga que foi "Adaptado de REFERENCIA" quando for apresentar a figura

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 (*Google Rapid Response* – Resposta Rápida Google) 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. A figura 1 mostra o desenho da arquitetura proposta pelo autor.

Figura 1: *Automated Forensics Data Acquisition Model*



Adaptado de (REICHERT; RICHARDS; YOSHIGOE, 2015)

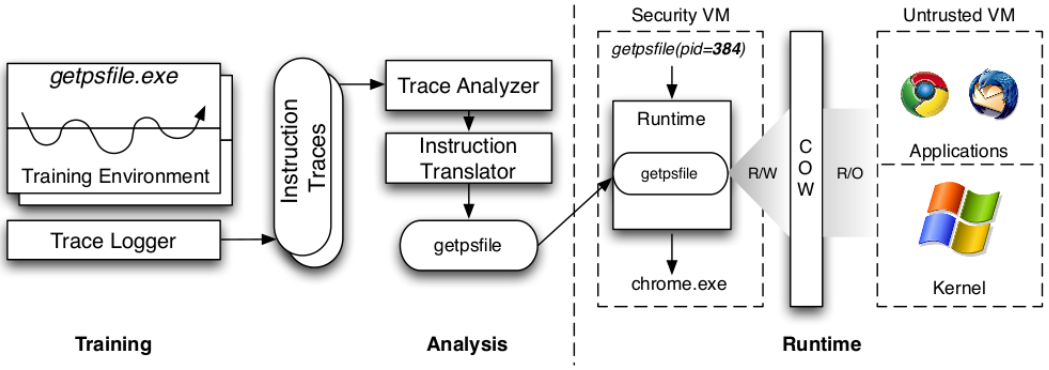
A proposta de (POISEL; MALZER; TJOA, 2013) é baseada na técnica de VMI

(*Virtual Machine Introspection* – Introspecção em Máquina Virtual) para coleta de memória volátil. Esta técnica se apoia na necessidade do VMM (*Virtual Machine Manager* – Gerenciador de Máquinas Virtuais) 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 portável 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

Figura 2: Virtuoso

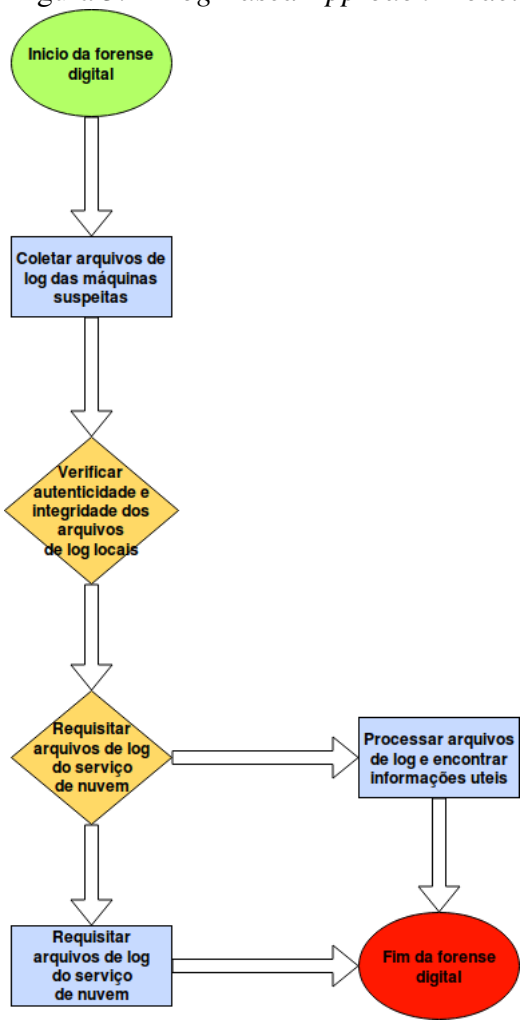


Adaptado de (DOLAN-GAVITT et al., 2011)

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 elas não foram alteradas durante o transporte. 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 (*Cloud Service Provider* – Provedor de Serviços de Nuvem) 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

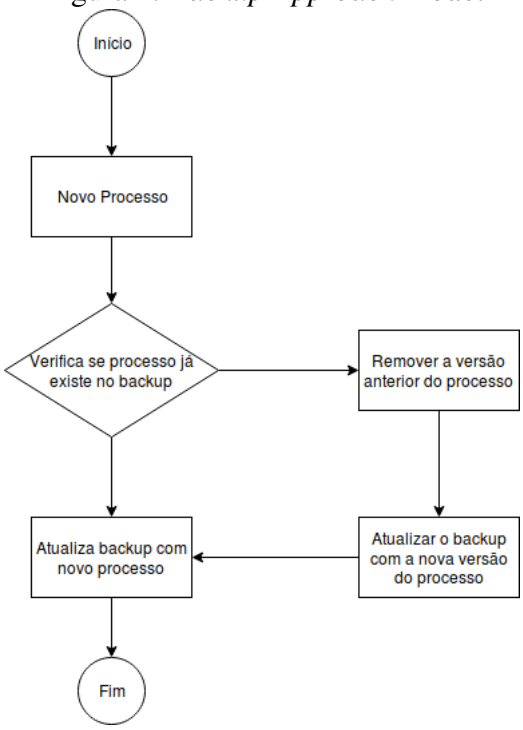
Figura 3: *A Log Based Approach Model*



Adaptado de (SANG, 2013)

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 4: *Backup Approach Model*

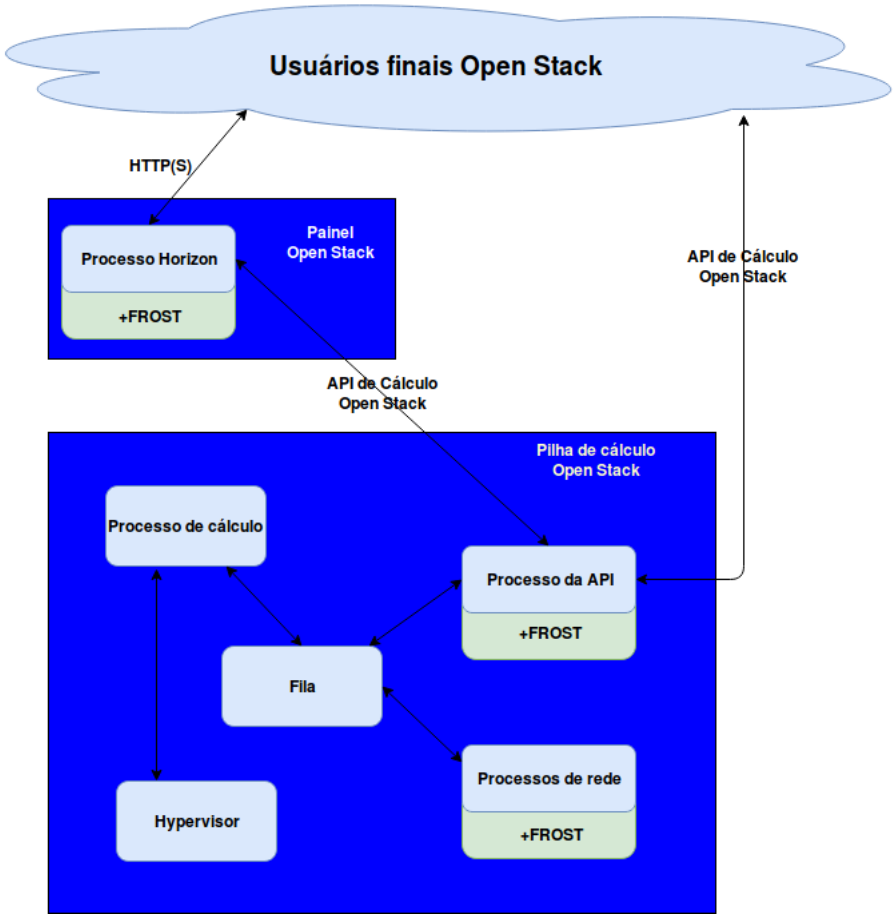


Adaptado de (DEZFOULI et al., 2012)

A ferramenta FROST (*FoRensic Open Stack Tools* – Ferramentas Forenses para Arcabouço Open Stack) 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 neste capítulo esta é a única que demonstra preocupação com adequação a questões legais e padrões já estabelecidos na industria forense. O autor declara que FROST segue as práticas definidas no SWGDE (*Scientific Working Group on Digital Evidence* – Grupo de Pesquisa Scientífica Em Evidência Digital) 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*

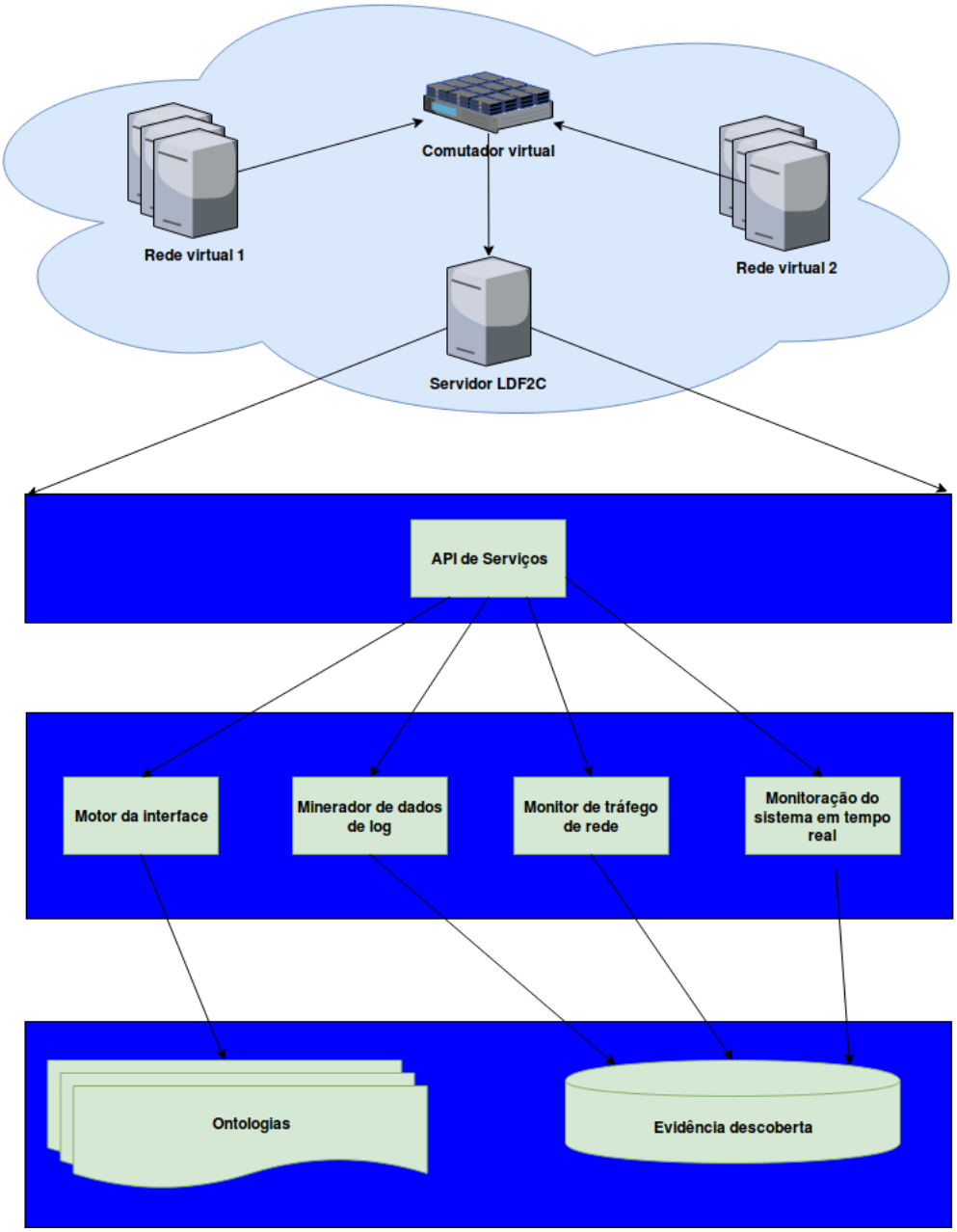
Figura 5: *FoRensic Open Stack Tools*



Adaptado de (DYKSTRA; SHERMAN, 2013)

O trabalho descrito em (GEORGE; VENTER; THOMAS, 2012) está focado em monitoração de rede e opera em uma arquitetura FaaS (*Forense as a Service* – Forense Como Serviço). 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 6: *Digital Forensic Framework for Cloud Environment*

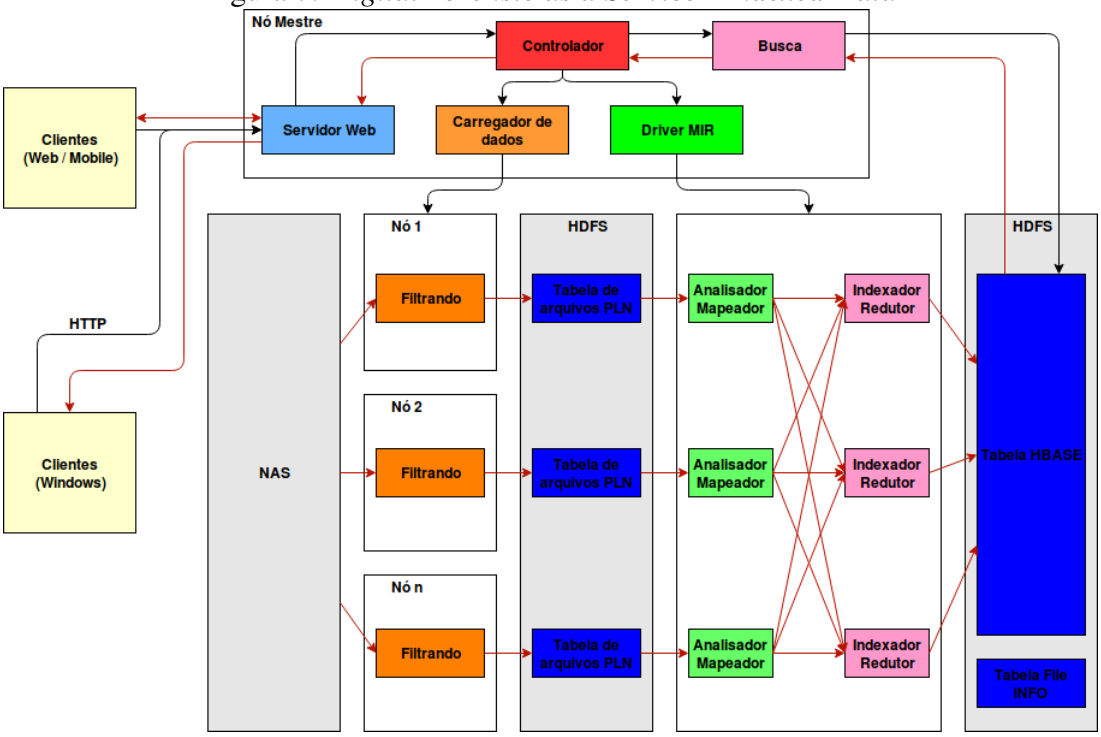


Adaptado de (GEORGE; VENTER; THOMAS, 2012)

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 (*Extract, Transform and*

Load – Extrair, Transformar e Carregar) e MapReduce (WIKIPEDIA, 2018), os 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 alteramos no processo. A figura 7 mostra o desenho da arquitetura do serviço proposta pelo autor

Figura 7: *Digital Forensic as a Service - Indexed Data*



Adaptado de (LEE; UN, 2012)

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

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 memória 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 aconteceu 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.

Tabela 1: Comparativo de soluções de coleta de informações de memória de máquinas em nuvem para análise forense

	Dizang (esta proposta)	(GEORGE; VENTER; THOMAS, 2012)	(POISEL; MALZER; TJOA, 2013)	(DYKSTRA; SHERMAN, 2013)	(LEE; UN, 2012)	(REICHERT; RICHARDS; YOSHIGOE, 2015)	(SANG, 2013)	(DOLAN-GAVITT et al., 2011)	(DEZFOULI et al., 2012)
Coleta é contínua?	✓	✘	✘	✘	✓	✘	✓	✘	✓
Reproduz o processo sem a VM?	✓	✘	✘	✘	✘	✘	✘	✘	✘
Garante cadeia de custódia?	✓	✘	✘	✘	✘	✓	✓	✘	✘
Preserva jurisdição e privacidade?	✓	✓	✓	✓	✓	✓	✓	✓	✓

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

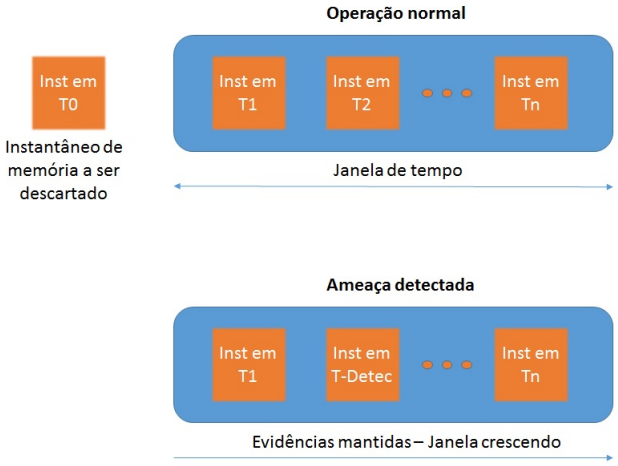
também permite identificar com mais precisão a fonte de uma evidência, uma vez que é possível dividir as partes de um sistema em contêineres: por exemplo, um contêiner para o motor de páginas dinâmicas (e.g., Apache), outro com a lógica de negócios (e.g., *golang*) e um terceiro para um banco de dados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

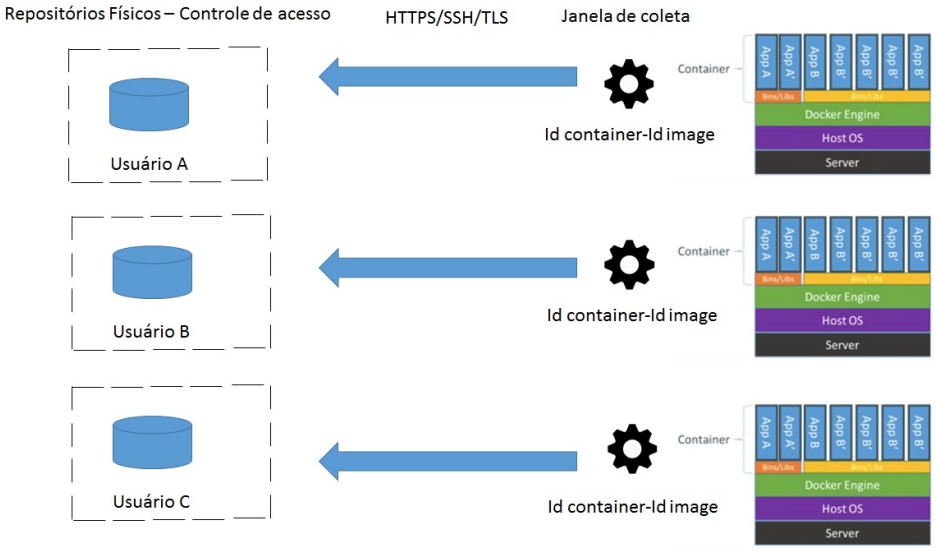
Figura 8: Janela deslizante de coleta de evidência



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 TLS (*Transport Layer Security* – Camada de Transporte Seguro) (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 (*Virtual Machine* – Máquina Virtual) usando o Oracle Virtual Box 5.0 (ORACLE,) 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 1.10e a API Docker 1.21, com os quais foram criados 3 contêineres executando o nginx 1.0em 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.

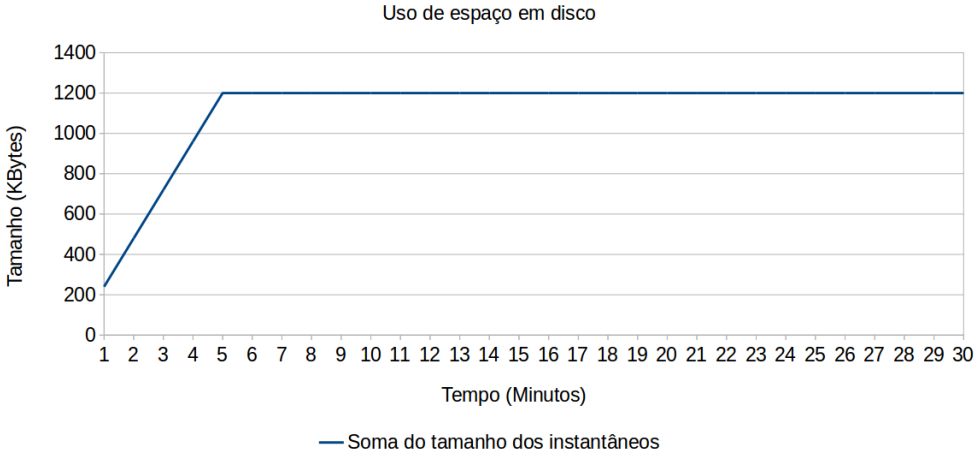
4.3 Resultados experimentais

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 delas. 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



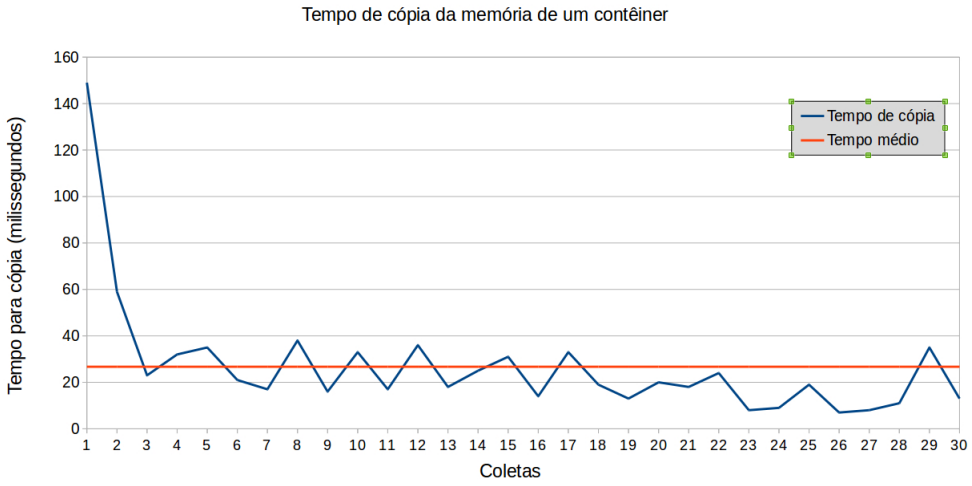
Fonte: Próprio autor

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*

Figura 11: Tempo de cópia da memória de um contêiner



Fonte: Próprio autor

space). Assim, Dizang em princípio não provê suporte a técnicas de investigação de malware que se baseiam em informações do *kernel space*, como, por exemplo, a comparação de informações do PEB (*Process Environment Block* – Bloco para o Ambiente dos Processos), que ficam no *user space*, com informações do VAD (*Virtual Address Descriptor* – Descritor de Endereços de Memória Virtual), que fica no *kernel space*. Análise de ameaças do tipo DKOM (*Direct Kernel Object Manipulation* – Manipulação Direta dos Objetos do Kernel) 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 reprodutível. 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.

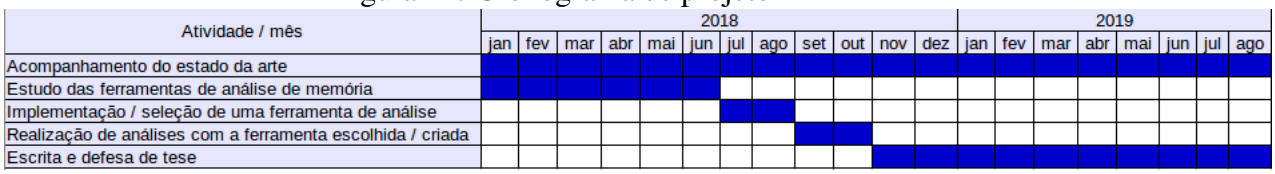
Para o restante do projeto, que consiste da parte 3, espera-se seguir o cronograma descrito na figura 12 onde serão executadas as seguintes atividades.

- **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 ferramente 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

Figura 12: Cronograma de projeto



Fonte: Próprio autor

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 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)*. Plymouth, UK: IEEE, 2015. p. 1–9.

BEM, D. et al. Computer forensics - past , present and future. *Journal of Information Science and Technology*, vol. 5, no. 3, p. 43–59, 2008.

BERNSTEIN, D. Containers and cloud: From LXC to Docker to Kubernetes. *IEEE Cloud Computing*, vol. 1, no. 3, p. 81–84, Sept 2014. ISSN 2325-6095.

CASE, A. et al. *The Art of Memory Forensics: Detecting malware and threats in Windows, Linux and Mac memory*. Hoboken, NJ: Wiley, 2014.

Chef. *Chef*. Https://www.chef.io. Acessado em: 16-04-2018.

DEZFOULI, F. et al. Volatile memory acquisition using backup for forensic investigation. In *Int. Conf. on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. Plymouth, UK: IEEE, 2012. p. 186–189.

Dierks T, R. E. *The Transport Layer Security (TLS) Protocol*. Fremont, CA: 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*. Plymouth, UK: IEEE, 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>.

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>.

FEWER, B. S. Reflective DLL Injection. no. October, 2008.

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>.

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/>.

LUIS, P.; SANCHEZ, P.; GIOVA, G. Digital Chain of Custody Quality Assessment. *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 4, p. 41–48, 2016. ISSN 17387906.

MELL, P.; GRANCE, T. *The NIST definition of cloud computing*. 2011. 7 p. NIST SP 800-145. Available from Internet: <csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.

MILLER, M.; TURKULAINEN, J. *Remote Library Injection*. 2004. Tech. Report: <www.nologin.org/Downloads/Papers/remote-library-injection.pdf>.

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

ORACLE. *Virtual Box 5.1*. Available from Internet: <https://www.virtualbox.org/>.

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>.

Puppet. *Puppet*. Https://puppet.com/. Acessado em: 16-04-2018.

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>.

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{\%}5CExploring-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/>.

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

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>. Acessado em: 15-09-2017.

____. *chroot - Change Root coomand*. <http://man7.org/linux/man-pages/man1/chroot.1.html>. Acessado em: 24-09-2017.

____. *dd - convert and copy a file*. <http://man7.org/linux/man-pages/man1/dd.1.html>. Acessado em: 01-12-2017.

____. *Namespacing*. <http://man7.org/linux/man-pages/man7/namespaces.7.html>. Acessado em: 15-09-2017.

____. *Numa Maps - Non Uniform Memory Architecture*. <man7.org/linux/man-pages/man7/numa.7.html>. Acessado em: 24-06-2017.

Vagrant. *Vagrant*. Https://www.vagrantup.com/. Acessado em: 16-04-2018.

Volatility Foundation. *Volatility Framework*. 2014. <www.volatilityfoundation.org/>. Acessado em: 24-06-2017.

VöMEL, S.; STüTTGEN, J. An evaluation platform for forensic memory acquisition software. *Digit. Investig.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, vol. 10, p. S30–S40, 2013. ISSN 1742-2876. Elsevier Science Publishers. Available from Internet: <http://dx.doi.org/10.1016/j.diin.2013.06.004>.

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