RT: PESC 754 04/2018

RadNet-S: Um Mecanismo para Transmissão Segura e Secreta de Registros *Syslog*

Leonardo Lima [1], Paulo Cabral Filho [1],
Diego L. C. Dutra [2], Claudio L. Amorim [2],
Evandro Luiz Cardoso Macedo [3], Renato Souza Silva [3],
Marco Antonio Coutinho [3], Luís Felipe M. de Moraes [3]

[1] Laboratório Nacional de Computação Científica (LNCC/MCTI)

[2]Laboratório de Computação Paralela e Sistemas Móveis (COMPASSO/COPPE/UFRJ)

[3]Laboratório de Redes de Alta Velocidade (RAVEL/COPPE/UFRJ)

{lgomes, cabral}@lncc.br, {ddutra, amorim}@cos.ufrj.br

{evandro, renato, marco.coutinho, moraes}@ravel.ufrj.br

Abstract. *Forensic analysis is based on security event logs provided by the Syslog servers. Hence to protect these servers from intruders that are interested in hiding their traces is a critical mission for the management of information security. This work introduces the RadNet-S, a novel protection mechanism for secure and obfuscate transmissions of logs to Syslog servers through unsafe channels, to protect logs against tempering and keeping them intact for auditing purposes. Our experimental results indicate that RadNet-S does not compromise the network operation, even if the logs rate is similar to that of a moderate DDoS attack of* $1000\ EPS$*, while makes it impossible to any intruder on the same network to localize the Syslog server.*

Resumo. *Análises forenses são baseadas nos registros de eventos de segurança fornecidos pelo servidor de logs. Assim, a proteção destes servidores contra invasores interessados em esconder os seus rastros é missão crítica para a gerência de segurança da informação. Este trabalho introduz o RadNet-S, um mecanismo original de proteção para transmissão segura e ofuscada de logs para servidores Syslog através de canais não-seguros, que protege esses registros contra adulterações e os mantêm íntegros para fins de auditoria. Os resultados experimentais indicam que o RadNet-S não compromete a operação da rede, mesmo quando a taxa de logs atinge o patamar de um ataque DDoS moderado de* $1000\ Eventos/s$*, enquanto inviabiliza qualquer atacante na mesma rede de localizar o servidor de logs.*

1. Introdução

Qualquer dispositivo conectado à uma rede IP é alcançável pelos demais dispositivos na rede [Kurose and Ross 2012], devido ao endereçamento fim-a-fim do protocolo. Além de permitir alcançabilidade entre os membros da rede, o endereço IP também agrega informações relacionadas à identificação do dispositivo e sua localização

na rede. Os atacantes se aproveitam de tais fatores, oferecidos pelo endereço IP, para atacar os seus alvos independente de suas localizações, geralmente sem que sejam notados, tendo em vista que a maioria dos ataques ocorre de maneira despercebida pelos seus alvos [François et al. 2012, Jamdagni et al. 2013].

Existem várias formas de se proteger destas ameaças, seja através do uso de sistemas de detecção/prevenção de intrusões, ou por antivírus, *firewalls*, entre outros. Uma das formas de proteção mais eficazes contra novos ataques é através da auditoria dos rastros deixados por um atacante, a fim de entender sua metodologia e descobrir seus objetivos. Este tipo de análise é conhecida como análise forense e os rastros são na verdade os *logs*, que registram os eventos ocorridos e que podem ser gravados nos próprios sistemas que os geram. Contudo, é importante ressaltar que, apesar dos benefícios, esta abordagem oferece alguns riscos inerentes ao processo, tais como:

- **Destruição de *logs*:** Durante uma invasão do sistema, os *logs* podem ser destruídos pelo atacante, o qual realiza tal ação para ocultar os rastros da invasão. Em alguns sistemas, isso pode ser contornado através da instalação de um *loghost* centralizado. Nesta configuração, é disponibilizado um sistema dedicado à coleta e ao armazenamento de *logs* de outros sistemas em uma rede, servindo como um repositório redundante de *logs*. Via de regra, o *loghost* não disponibiliza nenhum outro serviço, nem mesmo acesso remoto para os administradores, para minimizar a possibilidade de que o mesmo seja comprometido. Outra vantagem de *loghosts* centralizados é que eles facilitam a análise dos *logs* e correlação de eventos ocorridos em sistemas distintos. Sempre que possível, o uso de *loghosts* centralizados é fortemente recomendado;
- **Ataque de negação de serviço:** Um atacante pode usar o sistema de *logging* para executar um ataque de negação de serviço (DoS) contra um determinado sistema, gerando eventos em excesso até que o disco onde são armazenados os *logs* fique cheio e o sistema trave em consequência disto. O uso de uma partição separada para armazenar os *logs* pode minimizar o impacto deste problema;
- **Rotação automática de *logs*:** Quando este recurso é utilizado, deve-se garantir que os *logs* sejam movidos para o armazenamento *offline* antes que eles sejam removidos do sistema pela rotação, evitando assim a perda de registros. Alguns sistemas trazem a rotação automática habilitada na sua configuração padrão;
- **Falha de *hardware*:** Como os *logs* são armazenados em discos, estes passam a ser pontos críticos, visto que falhas de *hardware* podem incorrer nos discos, sendo necessário que medidas de replicação sejam utilizadas, como utilização de RAID e *backups* extras em localizações diferentes;
- **Identificação e localização dos servidores de *logs* na rede:** Os servidores que hospedam os serviços de armazenamento de *logs* por vezes apresentam alguma conectividade de rede com endereços IPs associados em suas interfaces de rede. Isto os torna alcançáveis, o que habilita oportunidades de ataques, como o próprio ataque de negação de serviço e a destruição de *logs*.

Estes e outros riscos podem ser listados, contudo, o que torna os sistemas de *log* mais fragilizados e vulneráveis é o fato dos servidores de *log* serem acessível através de um endereço IP. Isto os torna também um ponto crítico na rede de gerência de segurança, em virtude das possibilidades oferecidas a um usuário malicioso, seja por permitir que

ações não autorizadas sejam removidas dos registros, seja por descobrir detalhes sobre a infraestrutura da rede. Com isso, faz-se necessária a implementação de mecanismos que permitam a ofuscação das informações de registro enviadas para o servidor de *logs*, bem como a ocultação de sua localização no ambiente de rede. Desta forma é possível fornecer a confiabilidade, a integridade e a disponibilidade das informações de *log*, que são premissas para a segurança da informação [Stallings 2013].

Para mitigar estes riscos e aumentar o nível de segurança sobre os *logs*, o *Internet Engineering Task Force* (IETF) padronizou a arquitetura *Syslog* [Gerhards 2009a]. A arquitetura *Syslog* se caracteriza principalmente por agentes locais, os quais emitem as mensagens padronizadas dos eventos monitorados; pelo protocolo, que opera sobre TCP ou UDP; e pela gravação destes *logs*, seja local ou remotamente (servidor *Syslog*).

O servidor *Syslog* passa a ser um ativo fundamental de rede, que precisa ser protegido contra atacantes, devido ao interesse destes pelas informações salvas, com o intuito de descobrir comportamentos na rede alvo e ocultar seus rastros. Um servidor *Syslog* corrompido pelo atacante é ainda mais perigoso do que não ter tal servidor, pois as informações adulteradas podem induzir o perito auditor a tomar decisões que facilitam o atacante, ao invés de proteger o sistema. Proteger o servidor *Syslog* é portanto um dos grandes objetivos de qualquer sistema de defesa.

Existem várias formas de proteger um servidor *Syslog*: i) disfarçar a sua presença dentro da rede, escondendo-o de possíveis atacantes; ii) tornar o mesmo inalcançável para os atacantes. Considerando a própria essência do protocolo IP, que assume endereçamento único e fim-a-fim para todos os dispositivos da rede, tornar seletivamente um dispositivo inalcançável é uma tarefa complexa, que requer configurações específicas. Essas configurações têm maior probabilidade de causar falhas na rede, além de poderem revelar detalhes do processo de gravação dos *logs*.

Apesar da esmagadora utilização do protocolo IP na Internet, existem outros protocolos e arquiteturas específicas que não se baseiam em endereçamento para proporcionar conectividade entre os dispositivos em rede. Uma destas arquiteturas mais estudadas atualmente são as Redes Centradas em Interesse, também conhecidas como Redes Centradas em Informação (*Information-Centric Networks - ICN*) [Ahlgren et al. 2012, Gonçalves et al. 2016]. Nas Redes Centradas em Interesse o paradigma tradicional de endereçar os dispositivos de origem e destino para criar uma conexão fim-a-fim é substituído pelo roteamento baseado no conteúdo que se deseja disponibilizar e no interesse neste conteúdo.

Este artigo apresenta o RadNet-S, um mecanismo de transmissão de *logs* de maneira segura através de um canal não-seguro em uma rede de computadores, construído sobre o protocolo RadNet [Dutra et al. 2012], com o objetivo de proteger os *logs* contra adulterações, mantendo-os imaculados para fins de auditoria. O RadNet-S permite o envio dos registros de *log* para o servidor de *log* de modo furtivo (*stealth*), considerando que nenhum endereço IP é associado ao servidor, mas sim um interesse mútuo é estabelecido entre os agentes que geram os *logs* e o servidor. São mostrados os resultados sobre o impacto do protocolo RadNet-S em um ambiente simulado com *logs* sendo gerados concomitantemente a uma aplicação de vídeo sob demanda sendo executada. É possível mostrar que a solução proposta só afeta o tráfego de rede da aplicação quando uma carga

de *logs* extremamente grande é gerada, tendo como embasamento a carga de *logs* gerada em um ambiente real de produção durante um ataque.

O restante do artigo está organizado da seguinte forma: na Seção 2 trabalhos relacionados ao tema são explorados. Na Seção 3 abordam-se os conceitos fundamentais da proposta, incluindo uma breve descrição do protocolo RadNet. Uma avaliação experimental é apresentada na Seção 4 mostrando o impacto que a proposta tem em uma rede já operacional. Por fim, na Seção 5 são colocadas as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Diversas propostas de proteger servidores de possíveis invasões em uma rede privada podem ser encontradas na literatura. As mais clássicas abordagens contemplam o uso de Sistemas de Detecção de Intrusão (*Intrusion Detection Systems – IDS*) [Lunt 1993, Roesch 1999, Depren et al. 2005, Silva and Macedo 2017], e Sistemas de Prevenção de Intrusão (*Intrusion Prevention Systems – IPS*) [Koller et al. 2008, Stiawan et al. 2010].

Uma maneira também comum de ocultar servidores é através de servidores *proxy* [Kruegel and Vigna 2003]. Desta forma os serviços que precisam ser protegidos ficam atrás do servidor *proxy*, com todo o acesso sendo monitorado por este. Contudo, nesta abordagem, o servidor *proxy* é susceptível a ataques baseados em IP, o qual pode ser facilmente localizado na rede através de um escaneamento de portas, por exemplo. Uma vez invadido, o servidor *proxy* é porta de entrada para os outros servidores que deveriam estar protegidos.

Através do uso da ferramenta Snort [Roesch 1999], o autor em [Bauer 2002] propõe a ocultação de serviços de rede, dentre eles o servidor de registros de *log* de uma rede de computadores. O artigo apresenta uma solução para o problema mencionado através da configuração do IDS Snort colocando-o em modo promíscuo, no qual todo tráfego de rede é capturado pela ferramenta. Neste caso, o servidor que hospeda o Snort não tem um endereço IP configurado em sua interface de rede, o que dificulta que ataques baseados em IP sejam explorados. Contudo, a pilha do protocolo IP ainda é necessária, o que proporciona ataques de injeção de pacotes permitindo que um endereço IP seja configurado na interface de rede.

Em [Popa et al. 2011] os autores propõem um sistema para proteção de armazenamento em nuvem através do qual os usuários do sistema podem provar a ocorrência de violação da integridade dos dados armazenados na nuvem. Com esse sistema, é possível garantir os níveis de segurança dentro dos SLAs estabelecidos para os serviços contratados. Contudo, o sistema incorre de um aumento de 15% nos retardos de gravação dos dados, bem como na redução em 10% da vazão de leitura/escrita. Nesse mesmo tema, os autores em [de Carvalho et al. 2017] propõe uma melhoria nas verificações de segurança dos ambientes de nuvem, permitindo que a detecção de violações sejam feitas em tempo real. Um exemplo de aplicação de serviços de *log* em ambientes de nuvem foi proposto em [Ray et al. 2013]. Nesse trabalho, os autores defendem o armazenamento de *logs* na nuvem a fim de proporcionar acessos aos registros por longos períodos de tempo e com redução de custos por parte das organizações.

Os autores em [Park et al. 2017] apresentam um sistema que propõe a redução da

superfície de ataque, ou seja, das possibilidades de ataque que um servidor pode sofrer, tornando os endereços IP e MAC invisíveis através de uma configuração da interface de rede em modo promíscuo. A proposta atua como um ataque *man-in-the-middle* (MITM), onde o servidor secreto (invisível) sequestra requisições autênticas encaminhadas ao servidor público, tratando-as com o serviço pretendido que é protegido, e descartando as requisições que não forem autenticadas. Contudo, a proposta ainda utiliza a pilha de protocolos IP, o que permite que ataques sejam realizados ao servidor invisível, como por exemplo, um *buffer overflow*.

Como proposto no presente artigo, o RadNet-S também não considera o uso de endereçamento IP. Contudo o RadNet-S pode operar inclusive sem a pilha de protocolo IP instalada, o que não é considerado nas propostas apresentadas. É interessante mencionar que até o presente momento não foram encontrados trabalhos que utilizem uma rede orientada a interesse como abordagem para proteção da transmissão de *logs* através da ocultação dos servidores de *log* sem a utilização da pilha de protocolos IP.

3. RadNet-S

A proposta RadNet-S busca mitigar a vulnerabilidade a ataques nos servidores de *logs* da rede de gerência de segurança, a fim de manter a integridade dos *logs* armazenados. Tradicionalmente, a comunicação entre o servidor que armazena o *log* e os dispositivos que os enviam é feita usando um endereço IP de destino (servidor) e a porta 514 sob o protocolo de transporte UDP, através da aplicação Syslog [Gerhards 2009b]. Neste cenário, é possível ao atacante comprometer a integridade do servidor de *logs*, visto que existe um caminho pelo qual o servidor pode ser acessado.

A RadNEt-S é uma solução para propagação de mensagens de *log* em uma rede de computadores utilizando o protocolo RadNet [Dutra et al. 2012], um protocolo baseado em interesse sem endereçamento fim-a-fim, com transmissão *broadcast*, que elimina a possibilidade de um ataque baseado em IP ser eficaz contra o servidor de *log*. No contexto deste trabalho, o pacote RadNet é montado sobre o quadro *Ethernet* consumindo 28 dos 1500 *bytes* disponíveis no quadro, onde cada pacote deve ter ao menos um interesse associado a ele. Ademais, o próprio interesse pode ser cifrado, por ser tratado como um cabeçalho extra pelo protocolo, adaptando-se assim aos requerimentos de segurança das aplicações que utilizam o protocolo.

A Figura 1 ilustra um possível cenário que implementa a RadNet-S, uma aplicação de gerência de segurança, na qual são apresentados apenas os tráfegos referentes à propagação de *logs* na rede de gerência de segurança. Ainda na Figura 1, as setas tracejadas roxas representam o tráfego RadNet gerado diretamente dos equipamentos monitorados, as setas pontilhadas vermelhas representam *streamings* de *logs* propagados através do UDP/IP e a seta azul representa o mecanismo de tradução entre os protocolos UDP/IP e RadNet. O nó RadNet *Gateway* é necessário devido à ausência de suporte ao protocolo RadNet nos equipamentos legados, sendo tarefa do RadNet *Gateway* converter as mensagens vindas através da porta 514 UDP para mensagens RadNet. O cenário previamente descrito surgiu como uma aplicação do RadNet para o problema de proteção dos servidores de *logs*. Assim, o RadNet-S é voltado para ferramentas de *Security Information and Event Management* (SIEM) ou Gerenciadores de *logs*, garantindo a integridade do

histórico de eventos ocorridos na rede sob monitoramento.

Geralmente os *logs* são procurados apenas quando ocorre algum tipo de problema ou falha, seja de sistema ou de segurança. Contudo, quando o servidor no qual os *logs* são armazenados é comprometido, não existem mais provas íntegras e fiéis que possam ser utilizadas para a depuração dos eventos ocorridos no ambiente afetado. Indiferente da ferramenta de SIEM, ou sistemas afim, todas as soluções em operação dependem do endereçamento IP, o que fornece um meio pelo qual um usuário malicioso pode comprometer a integridade dos dados armazenados por tal ferramenta ou sistema. Como o Radnet-S independe de endereçamento fim-a-fim, ataques deste tipo são incapacitados de enviar respostas ao atacante sobre o sucesso ou falha de sua tentativa. A ausência de fluxo direcionado para um IP, ou outro protocolo que enderece unicamente um dispositivo na rede, inviabiliza o atacante de explorar o resultado da tentativa de ataque, ainda que no caso de um ataque bem sucedido, pois não há canal de retorno para as mensagens enviadas.

4. Avaliação Experimental

A Figura 2 apresenta o cenário experimental utilizado para aferir quantitativamente como o compartilhamento do meio *Ethernet* é impactado entre serviços de rede tradicionais, que executam sobre o IP, e o sistema de monitoramento RadNet-S previamente proposto. Dois *switches* são interligados por um único enlace e dois computadores são ligados em cada um dos *switches*. No *switch* superior foram ligados o servidor do RadNet-S e um servidor de *streaming* UDP utilizado para gerar carga na rede. No *switch* inferior estão ligados o agente gerador de *logs* e o cliente do sistema de *streaming*. Como visto na Seção 3, o protocolo de comunicação RadNet utiliza transmissão *broadcast* para propagar as mensagens entre os nós. Os impactos negativos das mensagens *broadcast* no protocolo *Ethernet* são conhecidos [Elmeleegy and Cox 2009], assim o cenário experimental foi projetado de modo a permitir quantificar o impacto que o uso da RadNet-S tem nas aplicações tradicionais. Dentro do escopo deste trabalho, a aplicação considerada

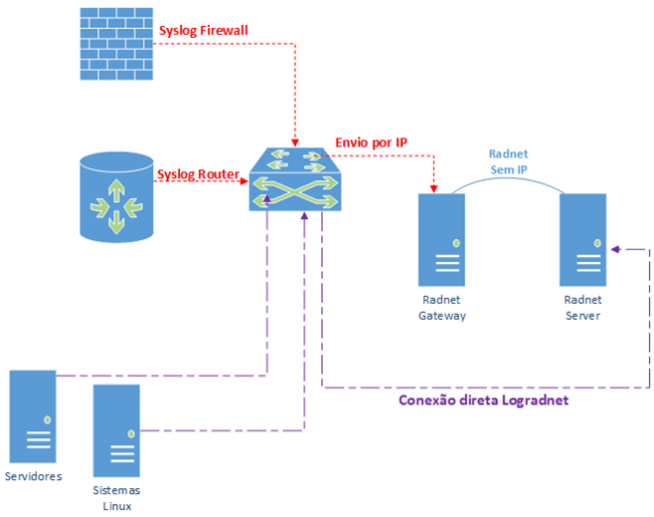


Figura 1. Possível cenário de utilização do protocolo RadNet-S

é um sistema de *streaming* UDP/IP utilizado para gerar tráfego no canal de comunicação.

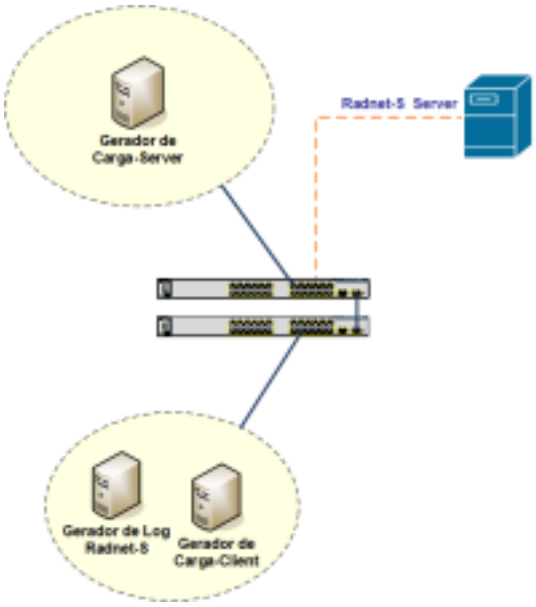


Figura 2. Cenário utilizado para avaliação do sistema RadNet-S

A Tabela 1 apresenta a configuração dos nós computacionais utilizados na avaliação experimental. Entre os recursos disponíveis vale ressaltar a presença de interfaces *Gigabit Ethernet*. A rede *Gigabit* proporcionou a avaliação de cenários instrumentais cujo tráfego de *streaming* na rede alcançou o patamar de $904\ Mbps$, o que significa uma ocupação do canal de comunicação superior à 90%, levando em consideração os *overheads* dos cabeçalhos *Ethernet*, UDP/IP e RadNet.

Tabela 1. Configuração dos nós computacionais

Recurso	Modelo
CPU	AMD Athlon™ II X3 445 3 Núcleos, $3,10\ GHz$
RAM	$8\ GBytes$
Rede	$1000\ Mbps$
S.O.	Ubuntu 16.04, $64\ Bits$

O *Streaming benchmark* utilizado é um sistema cliente/servidor que propaga um vídeo com duração de 120 segundos quebrados em segmentos de 1000 *Bytes* e uma sequência usada para validar a execução. A taxa de transmissão é controlada pelo cliente através da chamada de sistema *sleep()*, que também armazena o instante do recebimento de cada novo segmento. Esses valores são armazenados no final de cada execução do *benchmark*. Após isso, a aplicação verifica a ocorrência de perdas de mensagens e calcula a latência de cada segmento recebido, ou seja, o intervalo entre o recebimento de mensagens ignorando o intervalo de tempo do *sleep()*. Com base nas latências obtidas é possível aferir o comportamento da variação da latência, ou seja, o *jitter* ocorrido na transmissão do *streaming benchmark* usado nos experimentos. Os experimentos foram repetidos 20 vezes e coletados usando três perfis tráfego IP:

- $400\ Mbps$

- $800\ Mbps$
- $904\ Mbps$

Estes perfis de tráfego representam uma utilização efetiva do canal de comunicação de $41,84\%$ ou $418,4\ Mbps$ para o primeiro perfil, $83,68\%$ ou $836,8\ Mbps$ para o segundo e finalmente $94,56\%$, ou $945,584\ Mbps$ para o terceiro perfil de banda passante estudado. A RadNet-S utilizada recebe *logs* de um agente RadNet-S, descrito na Seção 3, onde cada *log* é uma *string* de 100 *Bytes*. Foi desenvolvida uma aplicação responsável por simular esses registros de eventos com uma taxa configurável. Esta aplicação também introduz na mensagem um contador para verificar a ocorrência de perdas durante os experimentos.

Observando o cenário experimental previamente descrito e a proposta do RadNet-S, é possível perceber que a avaliação experimental do sistema deve ser baseada em um ambiente onde podemos controlar a saturação do canal de dados, ou seja, a banda total utilizada pelas aplicações tradicionais. Neste trabalho, isso foi feito utilizando um *microbenchmark* cliente/servidor UDP que permite diferentes taxas de transmissão, cujos pacotes são numerados e marcados com o *timestamp* da recepção, viabilizando assim a aferição do *jitter* da rede quando na presença do tráfego RadNet-S.

4.1. Avaliação Qualitativa

O servidor de *logs*, juntamente com os agentes e o protocolo de comunicação, são considerados recursos fundamentais, tanto para o administrador da rede, quanto para um potencial atacante. Ter um servidor de *logs* íntegro e seguro significa ter a monitoração *online* das principais transações da rede, como também poder retroagir no histórico de transações para realizar alguma pesquisa mais detalhada. Um dos principais requisitos de segurança de um sistema de gravação de *logs* é a sua invisibilidade. Como um dos primeiros alvos estratégicos de um ataque mais elaborado, a invisibilidade do processo de gravação para o atacante muitas vezes impede seu ataque.

A invisibilidade do processo de gravação deve ser analisada de diferentes abordagens. Sob o ponto de vista de aplicação, ser invisível significa que o processo de gravação dos *logs* deve funcionar independentemente da aplicação. O processo de gravação de *logs* deve ser totalmente transparente para a aplicação, mesmo que seus próprios *logs* estejam sendo gravados. Se a aplicação estiver ciente sobre o processo de gravação ou mesmo puder interferir com este processo, um atacante pode utilizar esta aplicação para corromper o processo de *logs* e esconder suas ações. Analisando o sistema operacional, o processo de gravação de *logs* deve ser leve o bastante para não comprometer o funcionamento das demais aplicações executando no sistema. Um processo que consuma muitos recursos, além de não ser desejável do ponto de vista de desempenho, pode revelar sua existência e chamar a atenção do atacante.

Normalmente, o servidor de *logs* atua como um concentrador de conexões dentro da rede e portanto a análise qualitativa de segurança envolve diferentes aspectos. Uma primeira abordagem se refere a quantidade de recursos de rede que o processo de gravação requer para operar adequadamente. O servidor é tão mais invisível quanto menor a quantidade de recursos utilizados na rede. Um observador malicioso com acesso à rede certamente irá notar um fluxo que consuma muita banda passante, por exemplo. Uma segunda abordagem, considerando o mesmo observador malicioso na rede, está relacionada com a

caracterização dos fluxos observados. Sendo um protocolo orientado a conteúdo, o protocolo RadNet dispensa endereçamentos como IP ou MAC. Desta forma, a maioria das aplicações de *scanning* utilizadas por atacantes para levantamento de vulnerabilidades é inútil, mesmo sendo executada de dentro da rede em questão. Outro fator importante, que aumenta consideravelmente a invisibilidade do processo, é a completa ausência de relação entre as mensagens trocadas com o servidor de *logs*, que poderia ser utilizada pelo atacante para identificar conexões entre dois nós dentro da rede. Com isto, mesmo estando conectado à mesma rede dos agentes de comunicação de gravação de *logs*, um observador malicioso não consegue definir os agentes nem rastrear as mensagens de gravação. A Figura 3 abaixo ajuda a entender o cenário proposto.

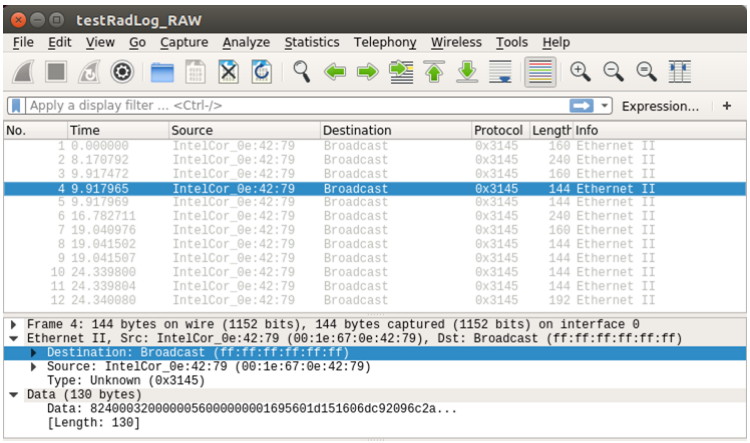


Figura 3. Exemplo de visualização dos pacotes RadNet através da ferramenta Wireshark

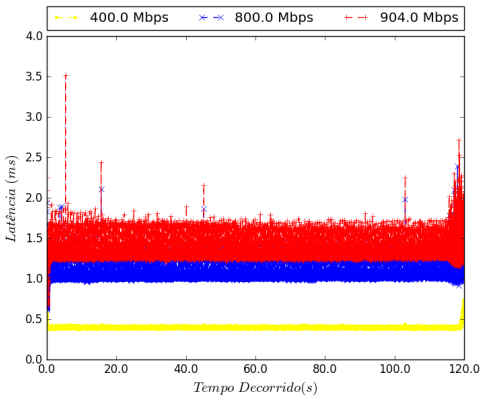
Analisando a Figura 3, é possível notar várias mensagens com a origem no MAC universalmente válido[1] 00:1e:67:0e:42:79 e destino ff:ff:ff:ff:ff:ff (*broadcast*). Desta forma, mesmo que a comunicação com o servidor de *logs* seja categorizada como *unicast*, as mensagens trocadas mostram-se de forma diferente, impossibilitando o estabelecimento de um padrão para encontrar o servidor de *logs*.

4.2. Avaliação Preliminar: sem tráfego RadNet-S

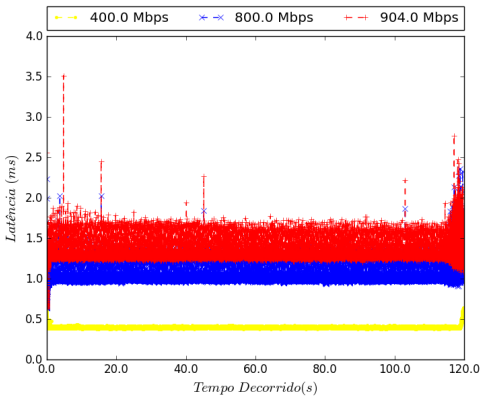
Em uma rede isolada tem-se ainda eventos assíncronos ocorrendo nos nós computacionais, com o *jitter* na rede não sendo nulo. Contudo, espera-se que o mesmo se aproxime de um valor constante quando as únicas fontes de tráfego são os clientes e o servidor de *streaming microbenchmark*. A Figura 4(a) apresenta os resultados obtidos quando apenas o tráfego do *microbenchmark* ocupa a rede. Nesta figura, o eixo das ordenadas representa a variação da latência média (*jitter* médio) e o eixo das abscissas o número do segmento utilizado no cálculo.

Nas aferições feitas com os menores consumos de banda passante, observa-se que o *jitter* médio tende a ser menor. Isso ocorre devido a maior parte do intervalo entre

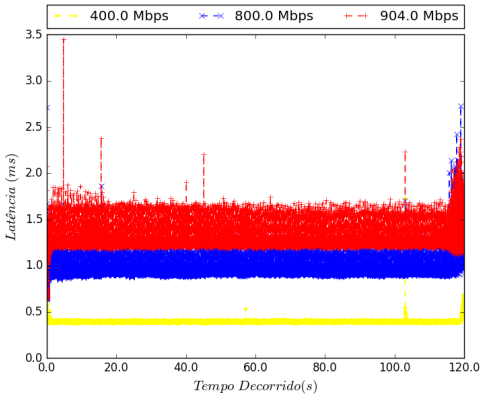
[1]O *bit* menos significativo deste octeto indica que o endereço é individual *unicast* (I/G - *Individual/Group*) e o segundo *bit* menos significativo indica que o endereço é universalmente válido (U/L - *Universally or Locally Administered*)



(a) Avaliação da rede sem tráfego RadNet-S



(b) Avaliação da rede com tráfego RadNet-S de 100 Eventos por segundo



(c) Avaliação da rede com tráfego RadNet-S de 1000 Eventos por segundo

Figura 4. Avaliação instrumental no ambiente de teste

pacotes ser dominada pelo *sleep()* do código, que tende a ter uma variação menor que os demais eventos assíncronos, logo produzem uma menor variabilidade como pode ser visto no gráfico. Ainda na Figura 4(a), são plotados os resultados da latência média das 20 execuções com perfil de $400\ Mbps$ em amarelo, de $800\ Mbps$ em azul e de $904\ Mbps$ em vermelho. Na figura percebe-se que o *jitter* aferido durante os 120 segundos cresce quando a saturação do canal de comunicação é aumentada. Um dos fatores que contribui para esse efeito é que a contribuição da latência no intervalo entre mensagens cresce quando a duração do *sleep()* é reduzida, em parte pelas variações inerentes a chamada de sistema serem mais significativas quando o intervalo requisitado é menor. Os resultados apresentados na Figura 4(a) indicam que a abordagem de avaliar o *jitter* na rede quando esta é saturada por uma aplicação IP, que simula o tráfego esperado em uma rede de produção, é promissora por conta dos impactos ínfimos que o sistema RadNet-S causa neste tipo de ambiente.

A Tabela 2 apresenta o sumário dos experimentos realizados no ambiente experimental sem a presença do tráfego RadNet. A primeira coluna contém a banda passante utilizada pelo tráfego IP, a segunda coluna contém a Latência Média (μ) em milissegundos, em seguida o Desvio Padrão (σ) e o Coeficiente de Variação (CdV). É possível

Tabela 2. Sumário do Experimento sem tráfego de Logs

Mbps	$\mu(ms)$	$\sigma(ms)$	$CdV(\%)$
400	$0,4024$	$0,03165$	$7,8666$
800	$1,0512$	$0,13803$	$13,1303$
904	$1,3047$	$0,15672$	$12,0116$

observar tanto na Figura 4(a) como através dos valores mostrados na tabela que o aumento da banda passante ocupada pelo tráfego IP induziu um aumento da latência média dos pacotes, assim como um aumento na variação da mesma, indicado pelo aumento no desvio padrão amostral do experimento. Um ponto interessante a ressaltar é o fato de o tráfego IP/UDP de $800\ Mbps$ obter uma variação na latência superior ao experimento com $904\ Mbps$ (ou uma ocupação de $94,56\%$ do canal de *Gigabit*), ambos utilizando os cabeçalhos *Ethernet* e IP. Esse efeito decorre da maior sensibilidade da aplicação de teste quando a mesma não está saturando o canal de comunicação, uma vez que qualquer retardo provocado por eventos assíncronos acaba por ter um impacto maior devido ao intervalo entre pacotes quando comparado ao experimento com $94,56\%$ de ocupação do canal.

4.3. Ambiente experimental com 100 Eventos por segundo

Usando os cenários instrumentais descritos no início desta seção, foi avaliado o impacto que o tráfego RadNet-S de 100 pacotes por segundo provoca na latência percebida pelo gerador de tráfego UDP durante 120 segundos. Como anteriormente, cada pacote representa um *log* de 100 *Bytes* propagado através do sistema RadNet-S, o que representa uma taxa de transferência média de $80\ Kbps$. Levando em consideração os cabeçalhos do protocolo, tem-se uma utilização do canal de $120\ Kbps$, com apenas um interesse associado por mensagem RadNet.

Na Figura 4(b) são apresentados os resultados obtidos usando uma carga de $100Eventos/s$. Como no cenário anterior, observa-se um aumento da latência quando o canal de comunicação começa a ser saturado, apresentando um efeito similar para os cenários com baixa ocupação do canal de comunicação. Esse efeito torna-se mais claro observando o sumário do experimento apresentados na Tabela 3, que novamente mostra um aumento do coeficiente de variação dos experimentos quando a ocupação do canal é aumentada, com um pico no cenário de $800\ Mbps$. As análises dos valores brutos do experimento para $800\ Mbps$ indicam que este pico ocorreu devido a maior interferência que os eventos assíncronos, como escalonamento e *daemons* de sistemas, provocam para intervalos maiores entre mensagens.

Tabela 3. Sumário do Experimento com tráfego de 100 pacotes por segundo (80 Kbps) no sistema RadNet-S

Mbps	$\mu(ms)$	$\sigma(ms)$	$CdV(\%)$
400	$0,4022$	$0,0308$	$7,6472$
800	$1,0209$	$0,1553$	$15,2171$
904	$1,9026$	$0,1522$	$12,5007$

Comparando apenas os coeficientes de variação apresentados na Tabela 2 com os apresentados na Tabela 3, percebe-se o efeito que a introdução do fluxo RadNet-S teve

sobre o comportamento do *jitter* nos cenários onde a ocupação da rede era maior. Neste sentido, é interessante perceber que a introdução do fluxo de $100\ Eventos/s$ apresentou uma maior variação para o cenário, na qual a ocupação do canal pelo fluxo UDP é de $83,68\%$. Ainda neste experimento, a Tabela 3 mostra que, apesar do tráfego RadNet-S ser propagado usando mensagens *broadcast*, o mesmo não degrada o desempenho das aplicações tradicionais, quando a soma dos tráfegos UDP e RadNet-S é inferior a capacidade do canal de comunicação.

4.4. Ambiente experimental com 1000 Eventos por segundo

Dando sequência à avaliação, foi aferido o impacto sobre o *jitter* da rede provocado por um tráfego de monitoramento da segurança de $1000\ Eventos/s$. Mantendo fixo o tamanho do *log* em 100 *Bytes*, tem-se uma taxa de transferência média de $800\ Kbps$. Levando em consideração os cabeçalhos, obteve-se uma utilização de $1,2\ Mbps$ do canal de comunicação *Ethernet*. Vale notar que, como nos casos anteriores, apenas um interesse por mensagem é utilizado. Além disso, este cenário representa uma rede sobre ataque cuja quantidade de eventos logados é de aproximadamente $3,927x$ maior que um ataque previamente ocorrido no Laboratório Nacional de Computação Científica (LNCC), onde foram coletados 22 milhões de *logs* em 24 horas, possibilitando assim averiguar o comportamento do RadNet-S em condições críticas, como durante um ataque de negação de serviço distribuído (DDoS).

A Figura 4(c), mostra a latência média durante os 120 segundos das 20 execuções. Observa-se que a latência média tende a aumentar quando a ocupação do canal é aumentada, sendo importante notar que, como anteriormente, a banda efetiva requerida pelo experimento $946,784\ Mbps$ foi inferior a banda passante disponível na rede *Gigabit*.

Tabela 4. Sumário do Experimento com tráfego de 1000 pacotes por segundo (800 Kbps) no sistema RadNet-S

Mbps	$\mu(ms)$	$\sigma(ms)$	$CdV(\%)$
400	$0,3977$	$0,0315$	$7,9166$
800	$0,9591$	$0,1633$	$17,0241$
904	$1,2577$	$0,1589$	$12,6329$

A Tabela 4 apresenta as diferentes vazões de tráfego UDP, a média, o desvio padrão amostral da latência e o coeficiente de variação. Observa-se que o aumento da ocupação do canal resulta em um aumento tanto da média amostral como do coeficiente de variação, com um pico em $800\ Mbps$ causado pela interferência de eventos assíncronos ao experimento. Sendo que comparando as linhas desta tabela com os resultados apresentados nas seções anteriores, é possível perceber com clareza que a introdução do tráfego RadNet-S degrada levemente o tráfego UDP, ou seja, aumenta o *jitter* na rede. Entretanto, apesar dessa leve degradação os resultados demonstram que, mesmo em cenários extremos, onde a rede encontra-se com perfil de tráfego RadNet-S similar ao esperado durante um ataque DDoS, o impacto provocado pelo sistema RadNet-S sobre o tráfego UDP é insuficiente para prejudicar a operação dos serviços tradicionais que executam sobre um ambiente de rede. Claramente, a variação aferida foi inferior a 1% e que variações superiores foram medidas durante o experimento como mostra a comparação entre $800\ Mbps$ e $904\ Mbps$.

5. Conclusão

Este artigo introduz a RadNet-S, um mecanismo de proteção para transmissão segura e ofuscada dos registros de eventos, *logs*, para servidores *Syslog* em redes locais usando o protocolo RadNet. O RadNet-S difere das propostas existentes para proteção da rede de gerência de segurança por optar pelo uso de um protocolo de comunicação sem endereçamento fim-a-fim orientado a interesse, que impossibilita ao atacante obter retorno das suas tentativas de comprometer o servidor *Syslog*. Com esta proposta, é possível oferecer proteção aos servidores de *log* contra atacantes que busquem acesso ao mesmo. Além disso, os resultados apresentados na Seção 4 mostram que essa melhoria não introduz ruídos significativos na rede como aferido pelo *streaming microbenchmark*.

Finalmente, foram realizados experimentos em que a banda efetiva sendo utilizada do enlace *Gigabit* chegou a $946,784\ Mbps$, ou seja, $94,68\%$ da capacidade do canal. Este cenário representou uma rede sobre ataque, no qual a taxa de geração de *logs* corresponde a 1000 pacotes por segundo – um cenário 4 vezes maior que um ataque real ocorrido previamente no Laboratório Nacional de Computação Científica (LNCC), no Rio de Janeiro. Com base nestes promissores resultados, as direções futuras deste trabalho será realizar uma avaliação dentro de uma rede de produção, como a do LNCC, e construir mecanismos que permitam a propagação dos *logs* através do RadNet-S para ambientes com suporte limitado a pacotes *broadcast*.

Referências

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.

Bauer, M. (2002). Paranoid penguin: stealthful sniffing, intrusion detection and logging. *Linux Journal*, 2002(102):17–23.

de Carvalho, C. A. B., Agoulmine, N., de Castro, M. F., and de Castro Andrade, R. M. (2017). How to improve monitoring and auditing security properties in cloud storage? *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

Depren, O., Topallar, M., Anarim, E., and Ciliz, M. K. (2005). An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. *Expert Systems with Applications*, 29(4):713 – 722.

Dutra, R. C., Moraes, H. F., and Amorim, C. L. (2012). Interest-centric mobile ad hoc networks. In *2012 IEEE 11th International Symposium on Network Computing and Applications*, pages 130–138.

Elmeleegy, K. and Cox, A. L. (2009). Etherproxy: Scaling ethernet by suppressing broadcast traffic. In *IEEE INFOCOM 2009*, pages 1584–1592.

François, J., Aib, I., and Boutaba, R. (2012). Firecol: A collaborative protection network for the detection of flooding ddos attacks. *IEEE/ACM Trans. Netw.*, 20(6):1828–1841.

Gerhards, R. (2009a). The Syslog Protocol. RFC 5424.

Gerhards, R. (2009b). The Syslog Protocol. RFC 5424 (Proposed Standard).

Gonçalves, F. B., França, F. M., and de Amorim, C. L. (2016). Interest-centric vehicular ad hoc network. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*, pages 1–10. IEEE.

Jamdagni, A., Tan, Z., He, X., Nanda, P., and Liu, R. P. (2013). Repids: A multi tier real-time payload-based intrusion detection system. *Computer Networks*, 57(3):811 – 824.

Koller, R., Rangaswami, R., Marrero, J., Hernandez, I., Smith, G., Barsilai, M., Necula, S., Sadjadi, S. M., Li, T., and Merrill, K. (2008). Anatomy of a real-time intrusion prevention system. In *2008 International Conference on Autonomic Computing*, pages 151–160.

Kruegel, C. and Vigna, G. (2003). Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, pages 251–261, New York, NY, USA. ACM.

Kurose, J. F. and Ross, K. W. (2012). *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6th edition.

Lunt, T. F. (1993). A survey of intrusion detection techniques. *Computers & Security*, 12(4):405 – 418.

Park, J., Noh, J., Kim, M., and Kang, B. B. (2017). Invi-server: Reducing the attack surfaces by making protected server invisible on networks. *Computers & Security*, 67:89 – 106.

Popa, R. A., Lorch, J. R., Molnar, D., Wang, H. J., and Zhuang, L. (2011). Enabling Security in Cloud Storage SLAs with CloudProof. In *Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference*, USENIXATC'11, pages 31–31, Berkeley, CA, USA. USENIX Association.

Ray, I., Belyaev, K., Strizhov, M., Mulamba, D., and Rajaram, M. (2013). Secure Logging as a Service – Delegating Log Management to the Cloud. *IEEE Systems Journal*, 7(2):323–334.

Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX Conference on System Administration*, LISA '99, pages 229–238, Berkeley, CA, USA. USENIX Association.

Silva, R. S. and Macedo, E. L. C. (2017). A cooperative approach for a global intrusion detection system for internet service providers. In *2017 1st Cyber Security in Networking Conference (CSNet)*, pages 1–8.

Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, Upper Saddle River, NJ, USA, 6th edition.

Stiawan, D., Abdullah, A. H., and Idris, M. Y. (2010). The trends of intrusion prevention system network. In *2010 2nd International Conference on Education Technology and Computer*, volume 4, pages V4–217–V4–221.