AnonV: uma arquitetura para verificação do grau de anonimização em coletas de tráfego de rede

Marco Aurélio Vilaça de Melo[1], **Dorgival Guedes**[1]

[1] Departamento de Ciência da Computação — Universidade Federal de Minas Gerais
Belo Horizonte, MG.

{vilaca,dorgival}@dcc.ufmg.br

Abstract. *Researchers and network administrators face a difficult dilemma when they work with traffic data files collected from the network: how to extract useful information for their work and yet to guarantee the privacy of users, whose information travel through the network, and prevent the leakage of sensitive information that may compromize network security?*
This work presents a study of aspects of privacy and safety in the use and sharing of network traffic log files, and proposes a methodology for the analysis of the file anonimization process.

Resumo. *Pesquisadores e administradores de rede encontram-se frente a um dilema ao trabalhar com arquivos de dados de tráfego coletado: como extrair informações úteis para seu trabalho, mas ainda garantir a privacidade dos usuários, cujas informações trafegam pela rede, e evitar o vazamento de informações sensíveis sobre a segurança da mesma?*
Este trabalho faz um estudo sobre aspectos de privacidade e segurança no uso e compartilhamento de arquivos de registro de tráfego de rede (traces) e propõe uma metodologia para análise do processo de anonimização desses arquivos.

1. Introdução

Na última década o mundo presenciou um grande crescimento no uso da Internet. Além disso, houve também uma grande diversificação nas aplicações disponíveis através dessa rede. Todo esse crescimento se traduz em tráfego, mensagens que circulam pelos canais da rede. Esse tráfego é de grande interesse para duas comunidades ligadas à área de redes de computadores: pesquisadores e administradores de sistema, que usualmente podem coletá-lo utilizando ferramentas como o `tcpdump` e armazená-lo em arquivos de *log*, ou *traces*.

Apesar de sua importância, a coleta de tráfego tem implicações complexas, por poder incluir inclusive os dados dos usuários durante sua interação com servidores e outros usuários da rede. As pessoas e as instituições têm se tornado mais conscientes desse fato e, por consequência, se tornam mais preocupadas com as suas informações que transitam na rede.

Diante desse quadro, várias técnicas e ferramentas para tornar anônimos os dados de rede têm sido propostas tentando garantir um nível adequado de anonimato aos dados distribuídos e, ao mesmo tempo, preservando as principais informações necessárias para a pesquisa ou documentação da rede [Koukis et al. 2006]. Para atender a diferentes demandas, essas ferramentas usualmente admitem diferentes configurações que determinam

o nível de anonimização dos dados coletados. Se torna responsabilidade do administrador da rede que realiza a coleta garantir que esse nível seja adequado em cada caso.

Considere-se por exemplo, um administrador que é abordado por um pesquisador que deseja uma amostra de tráfego da rede do primeiro a fim de avaliar uma hipótese de pesquisa. O pesquisador apresenta um coletor especialmente desenvolvido para coletar o tráfego de forma anonimizada, mas preservando a informação essencial à pesquisa. O administrador pode até ter interesse no tipo de resultado esperado, mas só pode fornecer os dados se tiver garantias de que a privacidade dos seus usuários não será violada em relação ao que exige a lei e a ética, o que poderia implicar em um análise detalhada da ferramenta oferecida.

Como outro exemplo, a prática judicial brasileira admite a coleta de tráfego para fins de investigação criminal, desde que os dados coletados se restrinjam ao objeto de um mandato judicial. Tal mandato pode determinar que apenas tráfego de um determinado usuário, conjunto de usuários, ou associados a um determinado servidor ou protocolo sejam coletados. O perito responsável pela investigação deve, então, definir a configuração da ferramenta que será usada para extrair essa informação e, possivelmente, anonimizá-la.

O problema, em ambos o casos, é que normalmente o administrador de rede que realiza a coleta do tráfego propriamente dita depende unicamente das garantias fornecidas pelos desenvolvedores da ferramenta e pelo pesquisador/investigador que a configura para determinar se os requisitos legais/éticos/administrativos de anonimato são atendidos.

Tendo isso em mente, torna-se necessária uma ferramenta independente para confirmar se os dados coletados e anonimizados dessa forma satisfazem as exigências de privacidade enquanto mantêm as informações úteis para cada fim. Este trabalho discute os aspectos principais que tal ferramenta deve considerar e propõe *AnonV*, uma arquitetura para esse tipo de verificação.

O restante deste trabalho está organizado da seguinte forma: na seção 2 discutimos os principais trabalhos relacionados; na seção 3 apresentamos o conceito de anonimato e as diversas formas em que tráfego de rede coletado pode violá-lo. Em seguida, a seção 4 apresenta a arquitetura proposta e o protótipo implementado e, finalmente, a seção 5 apresenta as conclusões e sugestões para trabalhos futuros.

2. Trabalhos relacionados

De forma geral, as pesquisas na área de anonimização de dados de rede se concentram em novas técnicas e ferramentas de anonimização [Luo et al. 2006] e técnicas para recuperar informações anonimizadas, ou seja, ataques contra as soluções propostas [King et al. 2009, Kohno et al. 2005, Ribeiro et al. 2008].

Uma das primeiras ferramentas de anonimização foi o `tcpdpriv`. Ele se preocupa apenas com os cabeçalhos dos pacotes IP, UDP e TCP, sendo capaz de gerar diversos níveis de anonimização, pois permite a escolha de vários campos do cabeçalho para serem anonimizados [Minshall 2005].

Recentemente, novas ferramentas oferecem uma maior flexibilidade, podendo ser amplamente estendidas e programadas pelo usuário. O exemplo mais significativo dessa linha é sem dúvida o `FLAIM` [Slagell et al. 2006], que tem uma linguagem de especificação dos campos a serem anonimizados, tornando a configuração muito flexível. Além

disso, disponibiliza várias técnicas de anonimização para cada um dos campos dos protocolos da arquitetura TCP/IP. Na identificação dos testes a serem feitos pela ferramenta proposta foram considerados todos os recursos de anonimização dessas ferramentas.

Pang [Pang et al. 2006] foi um dos primeiros a identificar a necessidade de uma ferramenta que análise os dados anonimizados para verificar se os mesmos estão realmente de acordo com determinada política de anonimização, dando uma maior confiabilidade e segurança ao se disponibilizar dados de redes. Também são encontrados alguns artigos que analisam a ética e os problemas jurídicos que o compartilhamento de dados pode gerar [Allman e Paxson 2007, Ohm et al. 2007].

3. Anonimização

No dicionário Aurélio a definição de anonimato é "sem o nome ou assinatura do autor; sem nome ou nomeada; obscuro". Portanto, podemos dizer que no contexto da informatização dos dados a informação anônima é aquela que não seja possível identificar a quem ela se refere.

Sendo assim, anonimização de dados de tráfego de rede é o processo de retirar as informações que possam levar à identificação dos usuários da conexão. De forma mais abrangente, essa anonimização engloba também o conteúdo da informação trocada e também as informações que interferem na segurança da rede de origem e destino dos dados.

Os pacotes que trafegam pelas redes de computadores contêm várias informações essenciais para a comunicação de rede. Além das informações contidas dentro dos dados transmitidos pela aplicação, diversos campos dos cabeçalhos da pilha TCP/IP podem conter informações que identificam algum usuário e/ou equipamento de rede e que afetam diretamente a sua privacidade e a segurança da rede.

Para entender os desafios da anonimização e da verificação da segurança de um arquivo anonimizado devemos primeiro discutir as principais técnicas de anonimização e como elas costumam ser aplicadas a protocolos da arquitetura TCP/IP.

3.1. Técnicas de anonimização de dados

Uma solução aparentemente óbvia para garantir a privacidade é simplesmente excluir dos dados as informações consideradas sensíveis do ponto de vista de privacidade e segurança, uma técnica denominada *truncation* [Burkhart et al. 2008]. Infelizmente, dessa forma pode-se destruir a qualidade dos dados para a pesquisa e auditoria pois eles, por exemplo, não poderiam ser separados em função de suas origens e destinos. Diante disso, pode ser necessário, ao invés de excluir as informações, substituí-las por outras que mantenham parte da informação, por exemplo, as características que separam os endereços IP em diferentes máquinas, apesar de não permitir sua identificação. Nesse caso, é necessário garantir que a partir desses identificadores não seja possível deduzir o valor original dos dados.

Para tentar anonimizar os dados garantindo que as informações sensíveis no que diz respeito à segurança e ao anonimato sejam eliminadas foram criadas várias técnicas de anonimização. Entretanto, essas técnicas trazem consigo uma relação de compromisso, pois quanto melhor é a anonimização (no sentido do alto grau de dificuldade para se revertê-la), pior é a qualidade desses dados para a pesquisa.

Substituição por *Black Marker* [Slagell et al. 2006]: Essa técnica é implementada pela maioria das ferramentas de anonimização e consiste simplesmente na substituição das informações relevantes por um valor constante. Dessa forma, equivale a *truncation*; entretanto, existem algumas variações dessa técnica como, por exemplo, usar o *black marker* apenas em partes de um campo. Em um endereço IP pode-se anonimizar com essa técnica apenas os dois últimos bytes do campo, por exemplo.

Substituição aleatória: Como o próprio nome diz, essa técnica faz uma substituição de todos os valores de um campo por valores aleatórios [Slagell et al. 2006]. Como o uso indiscriminado de valores aleatórios eliminaria toda a relação entre ocorrências de um mesmo valor, como no caso de *truncation*, normalmente mantém-se a relação entre valores anonimizados e valores originais. Isto é, cada valor encontrado pela primeira vez é substituído por um valor aleatório; novas ocorrências do mesmo valor original são então sempre substituídos pelo mesmo valor.

Criptografia: como a anterior, sua característica é substituir informações existentes por outras, mantendo um mesmo padrão de substituição. Diferente da substituição aleatória, os valores originais dos campos não são substituídos de forma aleatória e sim gerados através de um chave criptográfica [Luo et al. 2006]. Dessa forma, se a mesma chave for usada várias vezes o valor original sempre será anonimizado pelo mesmo valor, facilitando a correlação entre diversos *traces*.

Deslocamento: consiste em somar ao valor a ser anonimizado um valor fixo, que pode ser definido para cada arquivo (às vezes combinado a um pequeno desvio aleatório, para tornar o processo mais difícil de detectar) [Slagell et al. 2006]. Essa técnica nos lembra a criptografia, pois mantém o mesmo valor para os campos iguais, mas diferente daquela, os valores são sempre gerados a partir de um valor fixo somado ao valor do campo. Essa anonimização geralmente é utilizada em campos relativos a tempo.

Preservação de prefixos: consiste em usar uma substituição aleatória ou com criptografia, porém preservando as relações entre prefixos dos dados originais. Isto é, se dois endereços compartilham um prefixo de n bits, os resultados anonimizados dos mesmos endereços devem também compartilhar um prefixo de exatamente n bits [Xu et al. 2002]. Isso é de particular importância no caso de endereços IP, onde prefixos comuns podem identificar endereços em uma mesma rede.

Com exceção de *black marker*, as técnicas mencionadas são interessantes porque dificultam a identificação dos valores originais, mas ainda mantêm algumas características importantes para a análise, pois permitem que as distribuições de valores em cada campo se mantenham.

Entretanto, o compromisso existente entre a qualidade da anonimização e a qualidade da informação restante, faz com que essa maior permanência de informação nos *traces* implique em uma maior fragilidade na privacidade dos dados e na segurança da rede, pois os dados ficam mais suscetíveis a ataques. Por exemplo, com a separação das máquinas ataques como *fingerprinting* [Ribeiro et al. 2008], ataques de injeção de dados [Slagell e Yurcik 2004] e ataques que inferem as máquinas através do seu comportamento [Coull et al. 2007], se tornam mais visíveis em certos casos.

3.2. Aspectos de anonimização de protocolos da arquitetura TCP/IP

Para avaliar as demandas por anonimização e seu impacto sobre as informações contidas nos principais protocolos da Internet, podemos considerar as camadas da arquitetura.

Aplicação: A análise de protocolos de aplicação exige conhecimento específico sobre cada protocolo e aplicação em particular, o que não é tarefa simples. Entretanto, um atacante que tenha interesse em extrair informações do conteúdo dessas mensagens teria condições de fazê-lo caso essa informação estivesse disponível. Devido à dificuldade de se garantir o anonimato nesse caso, na maioria das vezes essa informação nem é coletada: administradores comumente configuram coletores como o `tcpdump` para coletar apenas os primeiros bytes de cada pacote, em número suficiente para cobrir os cabeçalhos dos protocolos até a camada de transporte. Ferramentas de anonimização também normalmente removem essa informação.

Nesse sentido, uma questão importante seria identificar se, no processo de coleta ou anonimização de uma certa amostra de tráfego, os dados foram realmente removidos de cada pacote. É possível que, se todos os pacotes de rede forem apenas truncados em um certo comprimento, alguns bytes de dados ainda estejam presentes e em certos casos, como pacotes em que os cabeçalhos mais dos protocolos de transporte/rede sejam mais curtos. Para alguns tipos de aplicação, apenas alguns bytes já podem ser suficientes para obter informações que deveriam ser omitidas.

Transporte (UDP e TCP): O protocolo UDP é conhecido por não ser orientado a conexões e não oferecer garantias de entrega. Por não ter funções complexas, ele possui um número reduzido de campos em seu cabeçalho. Esses campos são PORTA DE ORIGEM e PORTA DE DESTINO do pacote, COMPRIMENTO DO CABEÇALHO e SOMA DE VERIFICAÇÃO.

Os campos PORTA DE ORIGEM e PORTA DE DESTINO são os que identificam a terminação da comunicação nos protocolos UDP e TCP. Geralmente, aplicações padronizadas possuem uma porta padrão na qual o sistema operacional ficará aguardando uma conexão. Por esse motivo, portas podem trazer a identificação da aplicação. Normalmente de grande interesse para análise de tráfego, podem ser considerados um elemento de segurança em algumas redes: a descoberta, por um adversário, de que determinado servidor aceita conexão em uma certa porta pode ser considerada uma falha de segurança, pois essa informação indica qual serviço é executado em um certo servidor; com essa informação o adversário poderá explorar possíveis falhas de segurança existentes nesse serviço.

O campo SOMA DE VERIFICAÇÃO (*checksum*) é formado pelo resultado da soma de complemento de um do cabeçalho e dados do pacote TCP. Com essa soma de complemento de um, campos de dados de aplicação de até quatro bytes são passíveis de serem descobertos. Como os dados utilizados em alguns campos são dados considerados sensíveis ao anonimato e a segurança, esse campo deve ser anonimizado para evitar esses riscos. A forma de anonimização pode ser por *black marker*, ou utilizando-se algum tipo de codificação que indique apenas se o *checksum* do pacote original estava correto (ou não).

Por ser orientado a conexões, o protocolo TCP possui funcionalidades de ordenação e confirmação de recebimentos dos pacotes que exigem a combinação de diversos

campos de controle no seu cabeçalho. Os primeiros campos de interesse são PORTA DE ORIGEM e PORTA DE DESTINO, que têm a mesma função dos campos homônimos do protocolo UDP. Os campos NÚMERO DE SEQUÊNCIA, NÚMERO DE CONFIRMAÇÃO e ANÚNCIO DE JANELA são usados na numeração dos dados enviados e no controle de fluxo e, junto com alguns tipos de *flags* podem ser usados para identificação do sistema operacional das máquinas de uma rede (*OS fingerprinting* [Spangler 2003]). Por outro lado, eles podem ser do interesse de auditores e pesquisadores, pelas informações sobre sequenciamento de operações na rede.

Rede: a camada de rede é principalmente definida, na arquitetura TCP/IP, pelo protocolo IP (*Internet Protocol*). Além dele, outros protocolos importantes que merecem menção neste trabalho são ICMP e protocolos de roteamento como RIP, OSPF e BGP. Apesar de alguns campos do cabeçalho (tanto de IPv4 quanto IPv6) poderem ser usados para ataques do tipo *OS fingerprinting*, a informação principal nessa camada é, sem dúvida, aquela associada aos endereços IP encontrados em diversos campos dos protocolos mencionados. Esses endereços, se descobertos, permitiriam a identificação de máquinas de origem e destino de determinados pacotes (IPv4 e IPv6), a identificação de funcionalidades especiais da rede (ICMP) e da topologia da rede (protocolos de roteamento).

Além do problema de endereços, técnicas de anonimização que tenham ainda o objetivo de permitir análises de tráfego devem ainda observar a manutenção da informação de fragmentação de pacotes. Essa informação, contida nos campos identificação, flags e deslocamento de fragmento (*offset*) deve se mantida consistente, mas deve-se considerar questões de *OS fingerprinting* e identificação de *hosts* por associação de valores de identificação de pacote.

Rede Local: as mesmas preocupações com identificação de máquinas através de endereços IP também se aplica aos endereços da camada de rede local (endereços MAC). Nesse caso, técnicas que mantenham a associação de prefixos podem ser úteis para agrupar endereços de hardware de um mesmo fabricante, por exemplo. Além dos cabeçalhos de rede local, deve-se atentar para as mensagens do protocolo ARP, utilizado para fazer o mapeamento entre endereços MAC e endereços IP. Nesses pacotes os dois tipos de endereço podem ser encontrados e pode-se derivar deles a associação de endereços IP com endereços de placas de rede.

4. AnonV

O sistema proposto tem como objetivo auxiliar os administradores de rede na tarefa garantir que requisitos de anonimato sejam garantidos durante a atividade de coleta e disponibilização de dados de rede. Essa disponibilização pode ser solicitada pelos pesquisadores de uma empresa ou universidade ou até mesmo por uma ordem judicial. Esse pedido normalmente incluirá restrições sobre a política de anonimização que garantam que o arquivo resultante contenha a informação que auxiliará os solicitantes em alguma pesquisa ou processo.

Diante de tal pedido, que provavelmente pode vir acompanhado com uma ferramenta de anonimização a ser utilizada ou das exigências sobre quais dados podem ou não ser ocultados (e como isso pode ser feito), o administrador de rede provavelmente não terá a certeza que a sua expectativa de anonimização será atingida e, principalmente, não saberá se os dados disponibilizados constituirão uma ameaça à segurança da sua rede ou

se permitirão algum tipo de quebra da privacidade dos seus usuários.

Devido à grande quantidade de dados que esses *traces* de rede podem conter, analisar esses dados manualmente para conferir se os dados foram anonimizados de acordo com a política solicitada, se torna uma tarefa impossível. Por isso, propomos o desenvolvimento de uma ferramenta que compare o arquivo original com o arquivo anonimizado e faça uma análise de quais dados foram anonimizados e qual o método utilizado. A seguir discutimos as características dessa ferramenta.

4.1. Arquitetura

O sistema proposto vem suprir a necessidade de automatizar a tarefa de análise dos arquivos anonimizados. Conforme mostra a figura 1, a ferramenta desenvolvida tem como entrada o arquivo de *trace* original e o mesmo arquivo anonimizado pela ferramenta sugerida. Caso a ferramenta de anonimização tenha um arquivo de configuração associado, esse pode ser utilizado, através de um processo de tradução, para gerar um arquivo de configuração para a ferramenta de verificação, indicando como avaliar o resultado da anonimização.

Com base na informação fornecida, a ferramenta compara os pacotes encontrados nos dois arquivos. Essa comparação avalia se os campos considerados sensíveis do ponto de vista da quebra de segurança e/ou privacidade sofreram algum tipo de alteração. Se for detectada alguma mudança é provável que determinado campo sofreu algum tipo de anonimização, que precisa então ser qualificada.

Finalmente, um relatório deve ser produzido, descrevendo as conclusões da análise, identificando os campos que foram ou não anonimizados e as formas de anonimização utilizadas. Com base nessa informação o relatório deve também apresentar uma discussão do possível impacto de cada transformação aplicada (ou falta de tal transformação) para as políticas de privacidade e segurança da organização.

Idealmente, uma ferramenta que implemente essa arquitetura deve ser configurável e extensível em cada uma dessas etapas. Novos algoritmos que verifiquem padrões específicos ou que analisem um novo protocolo não previsto originalmente devem poder ser adicionados de forma simples. Arquivos de configuração poderiam determinar exatamente quais testes deveriam ser aplicados e até definir de forma completa uma política de anonimização desejada. Nesse caso, o relatório poderia ser simplificado para indicar apenas se o arquivo atende ou não às exigências da política proposta.

4.2. Fases da Metologia

A metodologia proposta para a ferramenta de análise de anonimização de *traces* propõe que o pacote seja analisado em todas as camadas da arquitetura TCP/IP.

4.2.1. Identificação dos pares dos pacotes

Primeiramente, os dois arquivos de entrada podem não conter exatamente os mesmos pacotes, pois filtros podem ser aplicados retirando completamente certos pacotes do arquivo original. Isso pode ser feito para restringir o foco, no arquivo a ser disponibilizado, a pacotes que atendam um certo critério (como “manter apenas tráfego HTTP”), ou

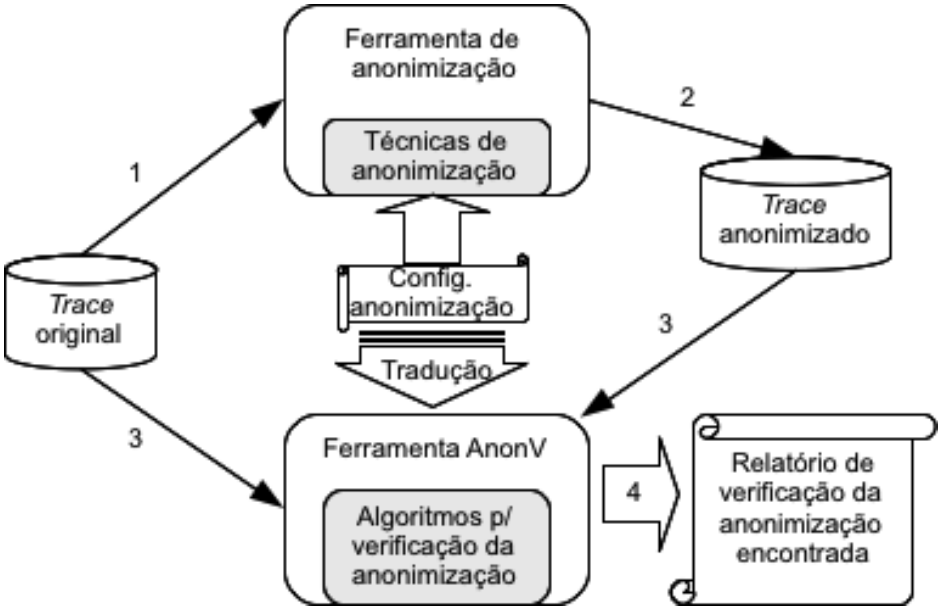

Figura 1. Funcionamento do sistema proposto

por que certos pacotes podem ser considerados sensíveis demais para serem distribuídos (como pacotes de protocolos de roteamento, em certos casos). Nesse caso, deve-se fazer uma comparação entre os pacotes de cada arquivo para identificar o par de pacote a ser analisado. Para isto, a primeira análise deverá ser feita no campo de marca de tempo (*timestamp*) que o `tcpdump` inclui no arquivo.

Não é suficiente, entretanto, realizar uma comparação direta dos valores desse campo nos dois arquivos, por dois motivos importantes: primeiro, em redes rápidas, devido à resolução limitada do campo de *timestamp* do `tcpdump`, diversos pacotes podem ter o mesmo tempo associado a eles; segundo, em certos casos, como discutido anteriormente, o próprio *timestamp* pode ser anonimizado, seja com um deslocamento simples (igual) de todos os tempos, ou por um deslocamento com um componente aleatório.

Para resolver o primeiro problema, da unicidade dos tempos, outros campos de pouco interesse para a segurança e o anonimato podem ser usados na diferenciação dos pacotes, como os identificadores dos protocolos de enlace (rede local), rede e transporte, que usualmente são mantidos. Para o problema do deslocamento dos *timestamps*, é necessário aplicar-se um algoritmo de casamento de padrões temporais, que busque identificar um casamento entre os pacotes que valide os intervalos de tempo entre eles. Isso é possível, desde que o deslocamento seja simples (onde basta encontrar o valor de deslocamento aplicado para que todos os pacotes se alinhem) ou que o valor aleatório adicionado seja pequeno em relação à maioria dos intervalos. (Esse é normalmente o caso, pois normalmente deseja-se apenas ocultar o momento exato em que o *trace* foi coletado.

4.2.2. Análise da anonimização na camada de rede local

Depois de identificar os pares de pacotes correspondentes, deve-se primeiro analisar o pacote no nível de rede local e verificar se o pacote é Ethernet (ou, basicamente, qual-

quer protocolo ISO 802.x); caso não seja, a ferramenta pode possuir regras para avaliação do protocolo específico indicado. Em um primeiro momento, deve ser feita apenas a contabilização desses pacotes, já que a maior parte do tráfego é hoje coletada em redes desse tipo. No caso de pacotes Ethernet, deve-se verificar os endereços de origem e destino dos quadros, verificando quantos pacotes foram anonimizados e o método utilizado para isso.

Além disso, caso o pacote seja ARP, deve-se fazer uma verificação se os endereços foram anonimizados de forma idêntica aos endereços de *hardware* do pacote desse protocolo, para indicar se um mapeamento preciso (mesmo que anonimizado) esteja disponível.

4.2.3. Análise da anonimização na camada de Rede

Na camada de rede deve-se testar o tipo de pacote, ARP, IP, ICMP ou algum protocolo de roteamento. Caso o pacote seja ARP, como mencionado na seção anterior, deve-se verificar se há consistência na anonimização dos endereços de *hardware*. Caso o pacote seja ICMP ou de algum protocolo de roteamento, a ferramenta deve alertar sobre os riscos da presença desse tipo de pacote no arquivo anonimizado.

Se o pacote for IPv4/IPv6, a ferramenta analisa se houve anonimização em diversos campos. Para os campos TIPO DE SERVIÇO, COMPRIMENTO TOTAL, IDENTIFICAÇÃO, bit de não fragmentação do campo FLAGS e o campo TEMPO DE VIDA (TTL), analisa-se a quantidade de pacotes que foram anonimizadas, pois esses campos são utilizados para descobrir o sistema operacional da máquina. Caso a destruição de ocorrências seja muito irregular, isso pode ser usado para identificar uma máquina por isolamento do seu padrão de tráfego. Isso deve ser reportado no relatório final.

Em seguida temos o campo *checksum*. Pode ser importante analisar se o pacote original tinha algum erro e se o *checksum* do pacote anonimizado foi alterado para mantê-lo correto (ou errado) como no pacote original, após a anonimização. Como discutido anteriormente, este campo é calculado com base em dados de outros campos do cabeçalho, o que pode em certos casos permitir a um atacante recompor dados originais que deveriam ter sido removidos.

Por último, deve-se analisar o endereço IP de origem e de destino. Pela complexidade dessa análise, ela será discutida separadamente na seção 4.2.6.

4.2.4. Análise da anonimização na camada de transporte

Na camada de transporte, primeiramente deve ser identificado se o pacote é TCP ou UDP (um outro protocolo exigiria regras de processamento particulares e deveria ser claramente identificado no relatório). Caso ele seja UDP, deve ser verificado se houve anonimização da soma de verificação (pela possibilidade de recuperação de informação sensível em alguns casos) e dos números de portas, estas últimas podem ser importantes em análise de tráfego por protocolo, mas até essa informação pode ser anonimizada em certos casos, pois a identificação de protocolos usados pode levar à identificação de servidores ativos que podem ser atacados, constituindo-se em uma ameaça de segurança em certos casos. Já no TCP, além dos mesmos testes vistos no UDP, deve-se testar também os campos

NÚMERO DE SEQÜÊNCIA, ANÚNCIO DE JANELA e OPÇÕES do TCP para verificar se houve anonimização, pois esses campos são utilizados pelos ataques de identificação de sistema operacional, entre outros.

4.2.5. Análise da anonimização na camada de aplicação

Na camada de aplicação deve-se testar se há algum *payload* no pacote anonimizado, pois os dados contidos nele podem revelar informações privadas dos usuários. Caso exista algum pacote com payload, mesmo que seja apenas uma fração dos dados originais, deve ser alertado do grande risco que essa informação pode trazer à privacidade das pessoas, caso o arquivo de *trace* seja disponibilizado para a análise. Uma análise mais detalhada de dados de aplicação é normalmente de difícil implementação, pela grande variedade de aplicações possíveis e da interpretação dos dados de cada uma em termos de anonimato.

4.2.6. Análise da anonimização de endereços

Como discutido anteriormente, para os diversos tipos de endereços encontrados na arquitetura TCP/IP (como endereços de rede local, IP, números de portas) a análise do padrão de anonimização adotado exige a coleta de informações sobre todos os endereços encontrados. Idealmente, deve-se montar um mapeamento entre endereços encontrados no arquivo original e os endereços a eles associados no arquivo anonimizado. A partir daí, diversas observações devem ser feitas.

- Se algum endereço for encontrado no arquivo anonimizado sem transformação, isso deve ser claramente indicado no relatório, pois pode constituir uma falha do processo de anonimização.
- Se as relações entre os dois mapeamentos forem de um para muitos, os dados de identificação de máquinas individuais provavelmente foram removidos do arquivo. É importante, entretanto, uma verificação cuidadosa para determinar se o mesmo padrão se aplica a todos os endereços e que não há endereços que recebem tratamento diferenciado.
- Se for observada uma relação 1:1 entre os dois conjuntos, a anonimização não utilizou a técnica de *black marker* nem substituição aleatória. Isso pode ser útil na análise de comportamento de máquinas, mas pode constituir uma ameaça às políticas em alguns casos. Para se verificar se foi utilizada uma técnica de preservação de prefixos, pode-se montar um *trie*binário para cada um dos dois conjuntos e verificar a equivalência da topologia de ambos (e da localização das chaves).
- É interessante verificar-se a distribuição estatística dos endereços encontrados nos dois conjuntos (por exemplo, através da CDF da frequência com que cada endereço aparece em cada conjunto.
- Caso se conheça o prefixo da rede da organização onde o tráfego foi coletado (normalmente disponível se a ferramenta for aplicada no momento da coleta) é interessante fazer o tratamento separado dos endereços da própria organização e dos endereços externos. Algumas ferramentas podem usar técnicas diferentes em cada caso; uma ferramenta maliciosa poderia mascarar alguns endereços e não ou-

tros, por exemplo, para tentar extrair informações que comprometam a segurança da rede.

- A identificação de técnicas de anonimização que preservem prefixos deve ser feita através da construção de dois *tries*, um com os endereços do arquivo original e outro com os endereços do arquivo anonimizado. A técnica utilizada na anonimização é do tipo que preserva prefixos se um *trie* pode ser transformado no outro quando um número finito de inversões de bits no seu caminhamento. Isso equivale a verificar se todos os caminhos de profundidade n em um *trie* possuem uma relação biunívoca com caminhos de mesma profundidade na outra árvore.

Diversas análise mais sofisticadas são ainda possíveis sobre os conjuntos de endereços coletados, como técnicas de avaliação da qualidade da informação disponível, técnicas de análise de correlação e similares.

4.3. Protótipo

Durante o trabalho foi desenvolvido um protótipo da ferramenta proposta, seguindo os passos básicos da metodologia. O objetivo nesse caso era verificar a viabilidade de certos tipos de processamento, identificar os pontos mais complexos do processamento e colocar em prática os conceitos envolvidos.

Para o desenvolvimento, foram analisadas diversas plataformas para manipulação de arquivos de *trace* de tráfego de redes considerando o formato `pcap` usado pelo programa `tcpdump`, hoje considerado um padrão para essa área. Existem diversas ferramentas que fazem a análise desses arquivos, mas todas com objetivos já bastante específicos, que não poderiam ser alteradas para os nossos objetivos. Procuramos então bibliotecas de programação que simplificassem o desenvolvimento de uma nova ferramenta. Apesar de haver bibliotecas até para a linguagem C para esse fim, a característica hierárquica, em camadas, da arquitetura TCP/IP, faz com que o processamento dos diversos protocolos encapsulados nos pacotes coletados seja mais simples em uma linguagem orientada a objetos.

Bibliotecas orientadas a objetos para processamento de arquivos no formato `pcap` se aproveitam do fato de que cada entrada do arquivo possui certos campos em comuns, presentes em todos os pacotes (os campos de controle criados durante a coleta e os campos do cabeçalho do nível de rede local). Uma classe básica descreve então apenas esses campos e permite seu acesso direto a partir das entradas do arquivo. Com base nas informações dos protocolos dos níveis inferiores pode-se identificar o tipo do protocolo de cada camada superior. Para se analisar então os campos do protocolo de um novo nível, basta que se utilize então uma classe derivada da classe original, porém mais especializada para identificar os campos específicos do protocolo. Dessa forma a cada protocolo processado identifica-se o tipo do protocolo superior e promove-se o objeto contendo o pacote extraído do arquivo para uma classe mais específica que detalha cada protocolo.

Bibliotecas com hierarquias de classes desse tipo existem para diversas linguagens, como Perl, Python, Ruby, C++ e Java, entre outras (e, muitas vezes, diversas bibliotecas diferentes para cada linguagem). Inicialmente experimentamos com bibliotecas para Perl e Ruby, mas a decisão final foi adotar Java, com a biblioteca `Jpcap`[1] para o desenvolvimento do protótipo. Essa combinação ofereceu o melhor compromisso entre

[1] `http://netresearch.ics.uci.edu/kfujii/jpcap/doc/`

aspectos como documentação, poder de expressão, possibilidade de expansão e simplicidade de utilização.

O sistema proposto segue a arquitetura proposta e tem como entrada o nome dos arquivos a serem analisados. O primeiro teste que é executado é o teste que identifica se os pacotes tratados nos dois arquivos são os mesmos; ele faz isso com base no campo de tempo do arquivo PCAP, considerando inclusive possíveis deslocamentos introduzidos. Além do tempo, o tipo de pacote e o número de sequência são verificados para validar a associação.

Em seguida, para cada camada da arquitetura TCP/IP, verifica-se se os campos identificados como passíveis de recuperação de dados privados ou que comprometam a segurança da rede, foram realmente anonimizados. Além disso, o protótipo analisa mais detalhadamente o endereço IP, que é o campo com maior risco de identificação de usuários devido, principalmente, aos endereços nele presentes. Os endereços encontrados em cada arquivo para cada par de pacotes identificados como correspondentes são inseridos em uma base de dados indexada tanto pelos endereços do arquivo original quanto pelos endereços do arquivo anonimizado. Com isso, pode-se verificar se a conversão é biunívoca ou se algum tipo de *black marker* foi utilizado. No primeiro caso, uma análise posterior (ainda não implementada) se faz necessária para determinar o tipo de anonimização utilizada (se criptográfica, aleatória ou por preservação de prefixos). A ferramenta relata se houve ou não anonimização por *black marker* e relata a quantidade de endereços que foram anonimizados utilizando essa técnica.

```
ARQUIVO ORIGINAL: trace1.pcap
ARQUIVO ANONIMIZADO: trace1-anon.pcap
Número total de pacotes:  107441 /  107441
Total de pacotes IP:      106870 /  106870
Total de pacotes ARP:        425 /     425
Outros protocolos:           116 /     116
. . .
Endereços de hardware (MAC address):
  No. de pacotes c/ end. MAC de origem não anonimizado: 0
  No. de pacotes c/ end. MAC de destino não anonimizado: 541
Endereços IP:
  No. de pacotes c/ end. IP de origem não anonimizado: 0
  No. de pacotes c/ end. IP de destino não anonimizado: 0
  No. de end. IP associados a um mesmo correspondente: 1 / 1
```

Figura 2. Exemplo de relatório produzido

Ao final é impresso um relatório com os campos que foram anonimizados e o número total de pacotes anonimizados para cada caso. A figura 2 apresenta um trecho de uma execução do protótipo. Nela podemos verificar que no caso considerado, nenhum pacote foi removido pelo processo de anonimização. Uma análise posterior indicou que os 541 pacotes que não tiveram o endereço MAC de destino removido na verdade representam pacotes ARP e de outros protocolos que utilizam *broadcast* Ethernet, enviados para o endereço padrão FF:FF:FF:FF:FF:FF, que não foi anonimizado (esse teste será incluído posteriormente no protótipo). Os endereços IP foram todos anonimizados (nenhum

endereço permaneceu inalterado entre os dois arquivos) e como a maior associação entre endereços foi 1:1, a anonimização é biunívoca (também pretendemos estender a ferramenta para determinar exatamente o tipo de anonimização empregada).

5. Conclusão e trabalhos futuros

O uso da Internet cresce a cada dia e cresce também a necessidade, por parte de auditores e pesquisadores, de usar os registros de tráfego de rede para propor novas soluções ou analisar situações que coloquem em risco a rede de uma empresa ou até mesmo o bom funcionamento da Internet. Por outro lado, cresce também a preocupação com a circulação de dados com informações privadas. Essa preocupação faz com que os administradores de redes enfrentem um dilema, onde a necessidade de uso e troca de dados de conexão se torna cada dia maior, mas as legislações limitam cada vez mais a divulgação de dados que contenham informações pessoais.

Diante disso, este trabalho apresentou uma análise das características técnicas do tráfego de rede IP sob a óptica da privacidade e segurança de rede. Com base nessa análise, foi proposta uma metologia e apresentado um protótipo de uma ferramenta que auxilie o profissional a identificar se os dados que pretende disponibilizar foram anonimizados da forma que se pretendia.

O protótipo desenvolvido é ainda apenas uma prova de conceito para ferramenta e a metodologia de verificação propostas. Uma linha clara de ação é o desenvolvimento de uma ferramenta completa, aproveitando melhor recursos de configuração e extensão dinâmicas para criar uma ferramenta que possa ser distribuída para uso pela comunidade. Além de extensões simples como as já mencionadas (aumento dos testes por tipos de anonimização de endereços, geração de mensagens detalhadas sobre cada tipo de vulnerabilidade identificada (ou não), outras possibilidades necessitam ainda de mais estudos, Em particular, seria interessante desenvolver-se um formato (linguagem) para a descrição do que seriam políticas aceitáveis de anonimização e divulgação de dados, de forma que a ferramenta, ao invés de gerar um relatório final com recomendações de pontos a serem considerados pelo administrador, gerasse um relatório simplificado, simplesmente indicando quais pontos da política estariam sendo observados/violados pela anonimização sendo considerada. Técnicas de teoria da informação podem também ser aplicadas para avaliar o volume teórico de informação contida no arquivo original e na versão anonimizada.

Finalmente, o princípio por trás da ferramenta e muitos dos algoritmos propostos não se limitam ao contexto de arquivos de *trace* de rede no modelo `pcap`. Essa metodologia pode ser estendida para avaliar a anonimização de arquivos de fluxos (Netflow) e *logs* de servidores HTTP, por exemplo.

Agradecimentos

Esta pesquisa foi parcialmente financiada pelo CNPq, Fapemig e pelo Instituto Nacional de Ciência e Tecnologia para a Web, InWeb (MCT/CNPq 573871/2008-6).

Referências

Allman, M. e Paxson, V. (2007). Issues and etiquette concerning use of share measurement data. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, págs. 135–140.

Burkhart, M., Brauckhoff, D., May, M. e Boschi, E. (2008). The risk-utility tradeoff for IP address truncation. In *Proceedings of the 1st ACM Workshop on Network Data Anonymization*, págs. 23–30.

Coull, S., Wright, C., Monrose, F., Collins, M. e Reiter, M. (2007). Inferring sensitive information from anonymized network traces. In *Proceedings of the 15th Annual Network & Distributed System Security Symposium (NDSS 07)*, págs. 35–47.

King, J., Lakkaraju, K. e Slagell, A. (2009). A taxonomy and adversarial model for attacks against network log anonymization. In *Proceedings of the Symposium on Applied Computing (SAC'09)*, págs. 1286–1293.

Kohno, T., Broido, A. e Claffy, K. C. (2005). Remote physical device fingerprinting. In *Proceedings of the IEEE Symposium on Security and Privacy*, págs. 211–225.

Koukis, D., Antonatos, S., Antoniades, D., Markatos, E. P. e Trimintzios, P. (2006). A generic anonymization framework for network traffic. In *Proceedings of the IEEE International Conference on Communication (ICC 06)*, Vol. 5, págs. 2302–2309.

Luo, K., Li, Y., Ermopoulos, C., Yurcik, W. e Slagell, A. (2006). SCRUB-PA: A multilevel multi-dimensional anonymization tool for process accounting. Technical Report cs.CR/0601079, ACM Computing Research Repository (CoRR).

Minshall, G. (2005). Tcpdpriv. http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html, acessado em 2010.

Ohm, P., Sicker, D. e Grunwald, D. (2007). Legal issues surrounding monitoring during network research. In ACM, editor, *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, págs. 141–148.

Pang, R., Allman, M., Paxson, V. e Lee, J. (2006). The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38.

Ribeiro, B., Chen, W., Miklau, G. e Towsley, D. (2008). Analyzing privacy in enterprise packet trace anonymization. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS 08)*.

Slagell, A., Lakkaraju, K. e Luo, K. (2006). FLAIM: a multi-level anonymization framework for computer and network logs. In *Proceedings of the 20th Large Installation System Administration Conference (LISA'06)*, págs. 68–77.

Slagell, A. e Yurcik, W. (2004). Sharing computer network logs for security and privacy: a motivation for new methodologies of anonymization. In *Proceedings of the Workshop on the Value of Security Through Collaboration (SECOVAL)*.

Spangler, R. (2003). Analysis of remote active operating system fingerprinting tools. Disponível em http://www.packetwatch.net/documents/papers/osdetection.pdf, acessado em dez 2009.

Xu, J., Fan, J., Ammar, M. e Moon, S. B. (2002). Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *Proceedings of the 10th IEEE International Conference on In Network Protocols*, págs. 280–289.