Network Forensics
Trabalho Prático 2
Redes de Computadores
PL53

Gonçalo Soares[a93286], Mariana Rodrigues[a93229], and Rita Teixeira[a89494]

Universidade do Minho

Abstract. Atualmente, vivemos num mundo em que a tecnologia e a *internet* encontram-se cada vez mais presentes no nosso dia a dia. Existindo cada vez mais, uma grande necessidade dos dispositivos encontrarem-se conectados à rede global. Devido a isso, progressivamente mais, vamos sendo alvos de ataques (*cyber-attacks*, entre outros), existindo uma tremenda necessidade de sermos capazes de detetar intrusos que tentem entrar dentro da rede.
Este é um dos temas mais preocupantes da atualidade, ***Network Forensics***.
Network Forensic é um ramo de perícia digital que focaliza a monitorização e análise do tráfego de rede. Por fim, podemos dizer que este tópico envolve a deteção de tráfego anómalo e de intrusos, bem como a sua identificação.

Keywords: Internet · *cyber-related crimes* · tráfego de rede · *NFAT*.

1 Introdução e contextualização

A evolução das redes computacionais e da *internet* potencializou a criação de diversas oportunidades para *cyber-related crimes*. Grande parte das tecnologias, nos dias de hoje, encontram-se conectadas à *internet* por toda a parte do mundo. Assim, podemos inferir que, atualmente, a *internet*, para além de ser um serviço bastante importante e fundamental, pode ser alvo de um número elevado de ataques.

Forensics é o uso das evidencias deixadas durante um ataque, de modo, a tentar perceber o que o *hacker* fez exatamente. De modo análogo, *Digital Forensics* é a ciência que se preocupa, essencialmente, pela recuperação e investigação dos recursos armazenados eletronicamente. Normalmente, esta ciência oferece uma ajuda bastante crucial em investigações criminais.

2 Ferramentas de Network Forensic

Ferramentas de *Network Forensic*, também conhecidas como **NFAT (Network Forensic Analysis Tool)**, são cruciais na ajuda à investigação de atividades

"suspeitas" e na deteção de intrusos. Estas ferramentas servem para analisar certas características do tráfego de rede, como por exemplo, a origem e destino dos pacotes e o seu tipo de atividade.

As **NFAT**s são projetadas para serem compatíveis com os dispositivos de hardware de rede, tais como *firewalls*, tornado possível a recolha e a preservação da rede de tráfego.

Existem diferentes tipos de **NFAT**s, enquanto que umas focam-se mais na análise de tráfego de rede ou incorporam mecanismos de análise, tais como o *Wireshark*, *tcpdump*, *NetworkMiner*, entre muitas outras. Já outras ferramentas, como *SilentRunner* e o *NetIntercept*, focam-se mais na monitorização da rede ou em avaliar ameaças internas, como por exemplo *malware*.

Agora iremos entram mais em pormenor sobre algumas das ferramentas enumeradas anteriormente:

- **Wireshark:** é um programa (*open-source*) designado para capturar, filtrar e analisar tráfego de rede, em tempo real. Esta ferramenta torna possível o controlo do tráfego de uma dada rede, conseguindo com isto monitorizar a entrada e saída de dados do computador, em diferentes protocolos, ou independentemente da rede à qual o computador encontra-se ligado.
  Um exemplo mais prático é a capacidade de um dado usuário, através do *Wireshark*, conseguir capturar todas as transmissões de um outro usuário, através de um *hub* ou *switch*, que se encontre na mesma rede local.
- **Tcpdump:** é uma ferramenta de linha de comando disponível para capturar e analisar o tráfego de rede principalmente em sistemas baseados em *Unix*.
- **SilentRunner:** foca-se na monitorização de todos os pacotes que passam por uma respectiva rede, concentrando-se na deteção de ameaças internas e dando o alerta no caso da deteção de uma anormalidade.

3 Técnicas de Network Forensics

Como já foi mencionado anteriormente, o propósito de *Network Forensics* é o de monitorizar o tráfego duma *network* com a finalidade de prevenir um ataque e de reunir informação e provas de maneira a identificar a origem de um ataque. Assim sendo, são necessárias técnicas que permitem levar a cabo o objetivo deste conceito.

3.1 IP Traceback

A primeira técnica a ser descrita é denominada de *IP Traceback*, que determina com segurança a origem de um *packet* na Internet, por exemplo para ajudar um investigador forense a identificar as fontes do *packets* IP de ataque. Esta técnica permite que a vítima possa identificar os caminhos de rede percorridos pelo tráfego de ataque sem exigir suporte operacional interativo de provedores de serviço de Internet.

A técnica apresentada é maioritariamente utilizada em "ataques de máscara". Estes são definidos como um ataque no qual se usa uma identidade falsa, como

uma identidade de rede, para obter acesso não autorizado às informações dum computador pessoal. Quando um ataque acontece, existe a possibilidade de um caminho de conexão entre o atacante e a vítima, sendo este apresentado por

$$h_1 \rightarrow h_2 \rightarrow h_3 \rightarrow ... \rightarrow h_n$$

sendo assim possível um rastreamento deste caminho.

Algumas funções que são usadas quando é mencionada a técnica *IP Traceback* são as seguintes:

Link State Testing Para esta função, o ataque tem que se encontrar em andamento e consiste num procedimento de rastreamento do *router* mais próximo da vítima e determinando o *upstream link* que foi usado para transportar o tráfego de ataque.

Input Debugging Uma vez que reconhece que está a ser atacada, a vítima desenvolve um ataque único que descreve a caraterística comum contida em todos os *packets* de ataque. A partir daqui, a vítima comunica este ataque que criou para se defender e, assim, o *upstream router* consegue implementar planos de defesa eficazes contra os ataques realizados.

Controlled Flooding A inundação é usada no algoritmo de roteamento de redes de computadores em que cada pacote de entrada é enviado por meio de cada link de saída, exceto aquele pelo qual chegou. *Controlled flooding* é um algoritmo no qual cada pacote de entrada é enviado para todas as linhas de saída, exceto para aquela que chegou.

ICMP Traceback Cada *router* irá gerar uma amostra de um dos pacotes que se encontra a encaminhar e irá copiar o conteúdo numa mensagem de rastremaento ICMP, que irá ter informações sobre *routers* adjacentes e a mensagem será enviada ao destino. Probabilidade dita que está técnica é aplicada para ataques que se originam de fontes do tipo inundação (*flooding*) de maneira ao receptor receber pacotes suficientes para reconstruir o caminho do ataque.

Packet Marking Techniques A ideia por trás das técnicas de marcação de pacotes consiste em amostrar o caminho um nó de cada vez, em vez de registar o caminho inteiro. Um campo dum "nó", grande o suficiente para conter um único endereço dum *router*, é reservado no cabeçalho do *packet*. Para IPv4, este seria um campo de 32 bits na parte Opções do cabeçalho IP. Ao receber um pacote, o *router* escolhe escrever o seu próprio endereço no campo do nó com uma probabilidade p. Dado que pacotes suficientes poderiam ser enviados e que a rota permanece estável, a vítima receberia pelo menos uma amostra para cada *router* no *path* do ataque.

Como os *routers* são ordenados em série, a probabilidade de um pacote ser marcado por um *router* e não pelos *routers* downstream é uma função estritamente decrescente da distância até a vítima. Assumindo que a probabilidade de marcar p é a mesma para todos os *routers*, a probabilidade de receber um pacote marcado de um *router* d salta de distância e não marcado por nenhum outro *router* desde aí é $p(1-p)^{d-1}$. A Figura abaixo ilustra a probabilidade de receber um pacote marcado de um *router* a 1, 2, 3, 4, 5 e 6 saltos de distância e não marcado por nenhum outro *router* num caminho de 6 saltos para diferentes valores da probabilidade individual de marcação p. Uma característica interessante a ser observada é que à medida que o número de *routers* intermediários aumenta, a chance de pelo menos um *router* no caminho marcar o pacote também aumenta.

A probabilidade limite (*threshold probability*) pode ser definida como o valor mínimo de probabilidade a ser atribuído a cada *router* no caminho para garantir que pelo menos um *router* no caminho marque o pacote. Isso diminui à medida que o número de saltos aumenta.

O tempo de convergência é definido como o número limite mínimo de pacotes necessários para determinar a sequência de *routers* que formam o caminho do ataque. Para determinar a ordem dos *routers* no caminho do ataque, cada *router* deve ter marcado um número diferente de vezes nos pacotes. O *router* mais próximo da vítima terá o maior número de marcas e o *router* mais próximo do invasor terá o número mínimo de marcas. Em geral, para determinar um caminho de ataque *n-hop*

$$Atacante \rightarrow R_n \rightarrow R_n - 1 \rightarrow ... \rightarrow R_i \rightarrow R_i - 1... \rightarrow R_2 \rightarrow R_1 \rightarrow Vítima$$

as duas condições a seguir devem ser satisfeitas:

- a vítima deve receber pacotes de forma que cada *router* no caminho de ataque tenha marcado pelo menos um pacote;
- o número de pacotes marcados pelo *Router Ri* deve ser estritamente maior que Ri-1, para qualquer $2 \leq i \leq n$.

Source Path Isolation Engine O SPIE é uma técnica baseada em *hash* que gera testes para tráfego dentro de uma rede. Esta cria resumos de *hash* de pacotes com base no cabeçalho do pacote e num fragmento de carga útil. Estes são então armazenados num *Filtro Bloom* e usados para rastrear a origem de qualquer pacote único entregue pela rede no passado recente.

3.2 Honeypots and Honeynets

Uma maneira muito eficiente de descobrir se o nosso programa tem erros ou vulnerabilidades é o de nos adiantarmos aos atacantes e tentarmos nós corromper o dito programa. Assim sendo, uma *Honeynet* é uma rede projetada especificamente com o objetivo de ser comprometida. Uma *Honeynet* comprometida pode ser usada para observar as atividades e o comportamento do intruso de maneira a ser possível realizar uma análise detalhada das ferramentas utilizadas

pelos invasores e identificar as vulnerabilidades exploradas pelos atacantes para comprometer os *Honeypots*. De um modo geral, um *Honeypot* pode variar dum simples programa com um *socket* à escuta numa porta, a um sistema de produção completo que pode ser emulado em vários sistemas operacionais. Por vezes, um *Honeypot* atrai tráfego fingindo ser um sistema de chamariz (*decoy system*), colocando-se perante a *Internet* como um sistema legítimo que oferece serviços. A ideia essencial é que qualquer tráfego direcionado ao *Honeypot* seja considerado um ataque ou intrusão. Qualquer conexão iniciada de entrada para o *Honeypot* provavelmente será uma sondagem, varredura ou ataque. Qualquer conexão de saída de um *Honeypot* implica que alguém comprometeu o mesmo e iniciou a atividade de saída. Deste modo, a análise forense das atividades de um *Honeypot* é menos provável de levar a falsos negativos e falsos positivos quando comparado aos sistemas de deteção de intrusão de rede mais evasivos e dependentes de assinatura. *Honeypots* podem-nos ajudar a detectar vulnerabilidades que ainda não foram compreendidas ou detetadas.

Honeypots podem ser classificados em dois tipos, dependendo dos serviços configurados disponíveis para um atacante comprometer ou sondar o sistema: baixa interação e alta interação. Um *Honeypot* de alta interação pode ser completamente comprometido, permitindo assim um atacante obter acesso a todos os aspetos do sistema operacional e lançar novos ataques à rede. Quanto maior o número de *Honeypots* implementados, mais provável é que se possa coletar informações sobre ataques ou sondagens de rede. Por exemplo, a maioria das solicitações de conexão TCP são geradas e enviadas para endereços IP selecionados aleatoriamente. É possível identificar que o pedido de conexão é intencionado a para tráfego malicioso apenas depois dum *handshake TCP* ter terminado e da carga contendo o tal conteúdo malicioso ter sido recebido. A probabilidade disso acontecer será maior à medida que o número de *Honeypots* aumenta.

Um *Honeynet* normalmente emprega um *Honeywall*, que atua como um *firewall* de maneira a proteger o mundo exterior de ataques que nascem de dentro da *Honeynet*. Para proteger sistemas não-*Honeypot* com ataques originados num *Honeypot* comprometido, um *Honeywall* pode ser configurado com vários recursos de controlo e captura de dados. Um Honeywall também pode modificar dinamicamente pacotes de dados maliciosos que visam especificamente vulnerabilidades noutros sistemas e torná-los benignos.

Um *Honeywall* pode capturar e monitorizar todo o tráfego de dados que se encontra entrando, saindo ou dentro da *Honeynet*. Os dados capturados podem ser usados para analisar as etapas que um invasor realizou para comprometer um *Honeypot* e a maneira como esta mesma se encontra a ser utilizada. Para evitar que um *Honeywall* intercepte as comunicações, um atacante por vezes pode instalar um *software* de criptografia para cifrar todas as comunicações entre a máquina do atacante e o *Honeypot*. Em tais situações, um *Honeywall* pode não ser capaz de descriptografar as comunicações. De maneira a resolver esse problema, um *software* de registro chamado *Sebek* foi recentemente desenvolvido. *Sebek* é uma ferramenta de software executada em cada *Honeypot* que faz parte do sistema operacional da máquina. Este apenas interceta os dados depois do

software de criptografia do invasor os descriptografar. As informações decifradas são enviadas para um servidor remoto para posterior análise.

4 Conclusão

Nos dias de hoje é fundamental proteger os nossos dispositivos de diversos ataques. Uma forma de prever os mais variados ataques e, acima de tudo, percebe-los é através da análise forense.

Essa investigação da *Network forensic* é um processo crucial na ajuda da investigação da *cyber-forensics* na obtenção, análise, avaliação, categorização e identificação de evidências cruciais. Possibilitando a apreensão de um *cyber-criminal* ou de uma dada pessoa suspeita de cometer um *cyber-crime*.

Com isto é fundamental a utilização de um sistema eficiente e robusto de metodologias de investigação que ajudem, melhorem e facilitem esse mesmo processo. Sendo crucial a utilização de ferramentas como *Wireshark*, *TCPDump*, entre outras.

Lista de Siglas e Acrónimos

NFTA *Network Foren-sic Analysis Tool*

References

1. NETWORK FORENSICS: A SURVEY (2013)
2. Network Forensics: An Analysis of Techniques, Tools, and Trends by Ray Hunt, Sherali Zeadally
3. Network Forensics: Notions and Challenges, by Ahmad Almulhem
4. TOOLS AND TECHNIQUES FOR NETWORK FORENSICS, Natarajan Meghanathan, Sumanth Reddy Allam and Loretta A. Moore
5. https://study.com/academy/lesson/network-forensic-analysis-definition-purpose.html