ANÁLISE COMPARATIVA DE FERRAMENTAS DE MONITORAMENTO DE TRÁFEGO PARA O HONEYPOT MP-H

Ana Paula Carneiro Athayde, Rafael Rodrigues Obelheiro

INTRODUÇÃO

Ataques distribuídos de negação de serviço por reflexão (Distributed Reflection Denial of Service, DRDoS) constituem uma ameaça persistente na Internet (Hiesgen et al., 2024). Nesse contexto, honeypots que se comportam como refletores, sem contribuir para o tráfego de ataque, podem ser usados para observar esse tipo de atividade. O MP-H é um exemplo desse tipo de honeypot, oferecendo suporte a nove protocolos de aplicação empregados em ataques por reflexão e sendo desenvolvido e utilizado desde 2018 (Heinrich et al., 2021). Entretanto, sua operação não fornece informações abrangentes sobre o tráfego recebido que não esteja relacionado aos protocolos suportados. Essa visibilidade pode contribuir tanto para o monitoramento da segurança operacional quanto para a identificação de protocolos com potencial para futura incorporação ao MP-H. A captura completa de pacotes, realizada por ferramentas como tcpdump, fornece informações detalhadas, mas pode apresentar maior custo de processamento e armazenamento. Ferramentas baseadas em fluxos, por outro lado, representam as comunicações de forma agregada, podendo reduzir esse custo (Bejtlich, 2013). Assim, este trabalho investiga e compara ferramentas de captura de pacotes e de monitoramento baseado em fluxos ou análise de estado, buscando avaliar a visibilidade do tráfego recebido pelo MP-H e o impacto dessas ferramentas sobre CPU, memória e armazenamento.

DESENVOLVIMENTO

A metodologia foi dividida em preparação da infraestrutura, execução dos experimentos e consolidação das métricas. Foram avaliadas sete soluções de monitoramento: tcpdump, Wireshark/dumpcap, ntopng, Argus, Nfdump/Softflowd e GoFlow2/Softflowd. Um gerador de tráfego em Python produziu requisições para diferentes protocolos e serviços, incluindo DNS, NTP, SSDP, Memcached, CLDAP, CoAP, QOTD, HTTP, SSH e ICMP. Em todas as soluções foi aplicado um filtro padronizado para selecionar o tráfego entre os hosts de teste. Cada experimento teve duração de 30 minutos, com monitoramento do consumo de CPU, memória residente (RSS) e memória virtual (VSZ) por meio do pidstat, com intervalo de 5 segundos. Os dados foram processados em Python para comparação entre as soluções.

RESULTADOS

A avaliação demonstrou diferenças no consumo de recursos computacionais entre as ferramentas. Os resultados do experimento de 30 minutos são apresentados na Tabela 1, que reúne os valores médios de utilização de CPU e memória residente (RSS) obtidos durante o teste. Quanto ao consumo médio de CPU, a maioria das ferramentas apresentou valores inferiores a 0,2%. O ntopng apresentou o maior consumo, com média de 4,02% e picos de até 21,2%. Entre as demais ferramentas, o Argus apresentou média de 0,13%, enquanto GoFlow2, Wireshark/dumpcap, tcpdump, Nfdump e Softflowd permaneceram abaixo de 0,02%.

O mesmo comportamento foi observado no consumo de memória residente (RSS), com o ntopng apresentando o maior valor, de aproximadamente 234 MB. O GoFlow2 utilizou cerca de 15 MB, seguido pelo Argus, com aproximadamente 12 MB, e pelo Wireshark/dumpcap, com cerca de 9,1 MB. Os menores consumos foram observados no tcpdump e na combinação de Nfdump com Softflowd, com aproximadamente 3 MB e 3,5 MB de RSS, respectivamente.

Quanto ao volume de dados armazenados em disco, o tcpdump gerou 46,72 KiB e o Wireshark/dumpcap, 57,40 KiB, enquanto o Argus e o Nfdump geraram 24,39 KiB e 28,24 KiB, respectivamente. No caso do Nfdump, os dados foram distribuídos em 33 arquivos. Assim, no experimento, Argus e Nfdump utilizaram menos espaço de armazenamento que tcpdump e Wireshark/dumpcap.

Além do consumo de recursos, foram observadas diferenças na visibilidade dos protocolos e na forma de representação dos fluxos, conforme apresentado na Tabela 2. O Wireshark/tcpdump, utilizado como referência, registrou 985 pacotes distribuídos entre TCP, UDP, ICMP e ARP, preservando os metadados de cada pacote capturado. O Argus registrou os mesmos protocolos observados na captura de referência (TCP, UDP, ICMP e ARP), consolidando-os em 425 fluxos. Em contraste, o Nfdump com Softflowd, operando com NetFlow v5, registrou apenas tráfego TCP (21,8% dos pacotes) e UDP (78,2%), sem registros de ICMP e ARP. O GoFlow2 com Softflowd apresentou o mesmo comportamento, por utilizar a mesma sonda de exportação.

O ntopng, por realizar inspeção direta por meio de libpcap e nDPI, também identificou tráfego ICMP durante os 30 minutos, além de registrar alertas comportamentais gerados pelo mecanismo de inspeção, como fluxos UDP unidirecionais e sessões TCP sem troca de dados. Também foram observadas diferenças quanto à direcionalidade dos fluxos, conforme apresentado na Tabela 2. O Argus representou nativamente as comunicações bidirecionais em um único registro, mantendo métricas separadas para cada sentido, como pacotes e bytes de origem para destino e de destino para origem, sem necessidade de pós-processamento. Já o Nfdump com Softflowd e o GoFlow2 com Softflowd produziram registros estritamente unidirecionais, gerando dois registros independentes por sessão, um para cada sentido da comunicação, sem correlação explícita entre eles no dado bruto.

CONSIDERAÇÕES FINAIS

A análise comparativa mostrou que a escolha da ferramenta deve considerar tanto o nível de detalhamento necessário quanto os recursos disponíveis no ambiente. O ntopng apresentou a maior utilização de CPU e memória entre as ferramentas avaliadas, com média de 4,02% de CPU e aproximadamente 234 MB de RSS, valores que correspondem, respectivamente, a cerca de 31 vezes e 20 vezes o consumo do Argus. Apesar do maior custo operacional observado, o ntopng oferece inspeção de tráfego por meio do nDPI e detecção de comportamentos durante a análise. Por sua vez, tcpdump e dumpcap apresentaram baixo consumo de CPU e memória e permitem a análise detalhada dos pacotes capturados, embora resultem em maior volume de dados para armazenamento e possam apresentar limitações de escalabilidade em ambientes com elevado volume de tráfego.

Entre as ferramentas baseadas em fluxos, o Argus apresentou cobertura dos protocolos observados na captura de referência, incluindo ICMP e ARP, ausentes nas demais ferramentas de fluxo avaliadas. Além disso, representou nativamente as comunicações de forma bidirecional, mantendo métricas separadas para cada sentido do tráfego. Considerando conjuntamente a cobertura dos protocolos observados, a representação bidirecional dos fluxos e o baixo consumo de recursos, o Argus foi a solução baseada em fluxos que, no cenário experimental, apresentou maior aderência aos requisitos de monitoramento contínuo analisados para o MP-H. Já o Nfdump com Softflowd e o GoFlow2 com Softflowd apresentaram consumo de recursos ainda menor, mas registraram apenas tráfego TCP e UDP e produziram registros unidirecionais, o que dificulta a análise do tráfego. O GoFlow2 apresenta como principal vantagem a serialização dos registros em formato JSON, o que pode facilitar sua integração com pipelines externos de análise, algo que hoje não é usado no MP-H.

Palavras-chave: monitoramento de tráfego de rede; fluxos de rede; honeypot MP-H; avaliação de desempenho.

ANEXOS

Tabela 1. *Comparativo de consumo médio de recursos entre as ferramentas (experimento de 30 minutos).*

Ferramenta	Tipo de abordagem	CPU média (%)	Memória residente - RSS (KB)
Tcpdump	Captura de pacotes	0	2.996,00
Wireshark	Captura de pacotes	0,02	9.136,00
Softflowd + Nfdump (nfcapd)	Exportador de fluxo + Coletor NetFlow	0	3.536,00
Softflowd + GoFlow2	Exportador de fluxo + Receptor NetFlow/sFlow	0,02	14.913,92
Argus	Registro de fluxo	0,13	11.769,30
ntopng	Análise de estado / Web	4,02	234.242,33

Tabela 2. *Comparativo de representação de direcionalidade e suporte a protocolos por ferramenta.*

Ferramenta	Direcionalidade do Registro	Suporte a Protocolos (Camadas L2/L3)
tcpdump / Wireshark	Pacote a pacote (sem agregação de fluxo nativa)	Completo (captura bruta de quadros e pacotes)
Softflowd + Nfdump	Unidirecional (dois registros independentes por sessão)	Parcial (Apenas TCP/UDP, omite ARP por ser Camada 2 e limita ICMP por ausência de portas L4 no NetFlow v5)
Softflowd + GoFlow2	Unidirecional (dois registros independentes por sessão)	Parcial (Apenas TCP/UDP, restrições herdadas da sonda Softflowd / NetFlow v5)
Argus	Bidirecional nativo (um único registro por sessão com métricas de ida e volta)	Completo (TCP, UDP, ICMP e ARP)
ntopng	Inspeção de estado / Stateful (visão de sessão integrada na aplicação/web)	Completo (via libpcap e motor nDPI)

REFERÊNCIAS BIBLIOGRÁFICAS

BEJTLICH, Richard. **The practice of network security monitoring: understanding incident detection and response.** San Francisco: No Starch Press, 2013.

HEINRICH, T. et al. New Kids on the DRDoS Block: Characterizing Multiprotocol and Carpet Bombing Attacks. In: HOHLFELD, O.; LUTU, A.; LEVIN, D. (ed.). **Passive and Active Measurement.** Cham: Springer International Publishing, 2021. v. 12671, p. 269-283. DOI: 10.1007/978-3-030-72582-2_16.

HIESGEN, R. et al. The Age of DDoScovery: An Empirical Comparison of Industry and Academic DDoS Assessments. In: ACM INTERNET MEASUREMENT CONFERENCE, 2024. **Proceedings** [...]. New York: ACM, 2024. p. 259-279. DOI: 10.1145/3646547.3688451.

TCPDUMP.ORG. **Tcpdump & Libpcap**. [S. l.], [s. d.]. Disponível em: https://www.tcpdump.org/. Acesso em: 14 set. 2026.