PARDED
Uma ferramenta de detecção passiva de malwares com foco em rootkits que utilizam técnicas de ofuscação de tráfego

Maickel J. Trinks[1], **Mateus Terra**[3], **João Gondim**[2]

[1]Centro de Pesquisa e Desenvolvimento para a Segurança das Comunicações (CEPESC), Brasília-DF, Brasil

[2]Departamento de Engenharia Elétrica - Universidade de Brasília (UnB), Brasília-DF, Brasil

[3]Programa de Pós-Graduação Profissional em Engenharia Elétrica - PPEE, Departamento de Engenharia Elétrica - Universidade de Brasília (UnB), Brasília-DF, Brasil

{maickeljosue@yahoo.com.br, gondim@unb.br, mateus.b.s.terra@gmail.com}

Abstract. *PARDED is a passive malware detection tool, focusing on rootkits that use traffic obfuscation techniques. The system detects malicious behavior through a multi-agent system, installed in the analyzed terminals and in the network infrastructure, verifying suspicious data flows and enriching information with local and remote databases, in addition to having an intuitive visualization interface and generation of traffic blocking rules and cyber threat intelligence, enabling integration with existing conventional defense systems, without affecting the connection performance of the terminals.*

Resumo. *PARDED é uma ferramenta de detecção passiva de malwares, com foco em rootkits que utilizam técnicas de ofuscação de tráfego. O sistema detecta comportamento malicioso através de um sistema multiagente, instalado nos terminais analisados e na infraestrutura de rede, verificando fluxos de dados suspeitos, enriquecendo informações com bases de dados locais e remotas, além de possuir interface de visualização intuitiva e geração de regas de bloqueio de tráfego e inteligência de ameaças cibernéticas, permitindo integração com sistemas de defesa convencionais existentes, sem afetar o desempenho de conexão dos terminais.*

1. Introdução

Passive Rootkit Detector With Enriched Data (PARDED) é uma arquitetura multiagente, composta por elementos que trabalham em pontos distintos de uma infraestrutura de rede de comunicação, onde cada componente integrante é autônomo, como em [Julian and Botti 2019]. Sua função é detectar técnicas de ofuscação de tráfego em terminal infectado por software malicioso. Tais *malwares* impedem que análises locais, tanto pela verificação da tabela de conexões quanto por captura de tráfego local, detectem a transmissão de dados maliciosos.

Foram encontradas apenas duas publicações que utilizam a abordagem de detecção baseada em rede, nas quais foram apresentadas as arquiteturas MADEX

[Marques et al. 2021] e NERD [Terra and Gondim 2021]. O PARDED visa aprimorar a performance dessas arquiteturas e melhorar a efetividade da detecção ao eliminar gargalos na forma como o agente auditor atua para detecção e o bloqueio da comunicação maliciosa, de forma passiva [Trinks et al. 2023], a fim de evitar degradação na taxa de transmissão dos equipamentos de rede, além de utilizar dados enriquecidos para decisões de bloqueio de tráfego e prover integração com outros sistemas de defesa.

O PARDED possui como principal característica o aperfeiçoamento das técnicas de bloqueio em infraestruturas de redes de comunicação sem degradar a taxa de transmissão de dados nos equipamentos de rede, além de prover uma interface para integração com outros sistemas de defesa previamente existentes, como *firewalls* e sistemas de detecção de intrusão (IDS). A ferramenta permite ainda a integração com sistemas de inteligência de ameaças, com viés cibernético (CTI), conforme definições apresentadas em [Tounsi and Rais 2018].

Outra característica importante adicionada é a possibilidade de enriquecimento das informações acerca dos destinos maliciosos utilizados pelos *rootkits*, notadamente para sistemas de comando e controle (C2), auxiliando o analista de segurança com dados importantes para análise comportamental, outros comprometimentos registrados para o IP detectado, informações existentes acerca do domínio malicioso, entre outros. O enriquecimento é realizado através de bases de informação locais ou externas, como nomes de domínio consultados pelo *malware*, geolocalização, lista de IPs comprometidos ou análises de empresas de segurança cibernética.

O conceito da arquitetura PARDED foi apresentado no *IV Congreso de Ciência de la Computacion, Eletronica e Industrial* - CSEI 2022 [Trinks et al. 2023], com o título *Multi-agent Architecture for Passive Rootkit Detection with Data Enrichment*, onde foram descritas as melhorias arquiteturais que o PARDED introduz ao NERD [Terra and Gondim 2021] e ao MADEX [Marques et al. 2021]. Neste artigo apresenta-se a implementação do PARDED, agregando à detecção um arcabouço para enriquecimento dos alertas e integração com ferramentas de segurança, levando a um *inicial report*. Assim, elevamos o grau de maturidade de detecção [Bromander et al. 2016] para uma caracterização que pode correlacionar alertas de TTP (Táticas, Técnicas e Procedimentos) e até possíveis artefatos e agentes maliciosos.

O artigo está organizado da seguinte forma: a seção 2 apresenta a arquitetura e as funcionalidades da ferramenta; a seção 3 detalha o desempenho em diversos cenários de teste; a seção 4 descreve a demonstração realizada com a ferramenta; a seção 5 explica com mais detalhes onde encontrar o código do sistema e instalação a solução; por fim, a seção 6 apresenta as conclusões.

2. Arquitetura e Funcionalidades

A arquitetura PARDED foi desenvolvida nas linguagens Nim, C e Python 3, aproveitando as principais características das diferentes linguagens de programação em cada um dos elementos e subsistemas, como velocidade de processamento, facilidade de uso de APIs existentes e possibilidade de criação de interfaces de visualização interativas. O sistema é composto por quatro elementos principais, de forma integrada, conforme Figura 1. Cada um dos elementos é detalhado nas subseções abaixo.

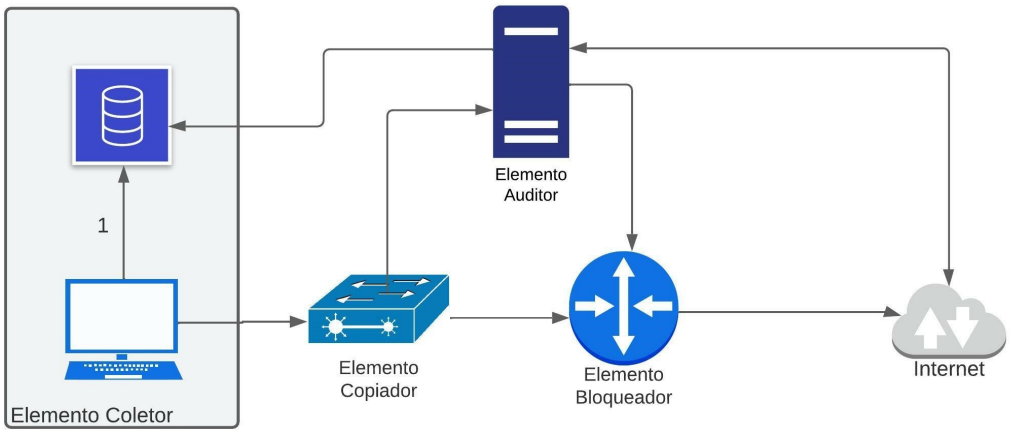

Figura 1. Arquitetura PARDED proposta (Figura do Autor)

2.1. Elemento Coletor (ECol)

O Elemento Coletor é inserido no terminal a ser analisado, verificando o tráfego através da captura de tráfego pelo próprio terminal, com auxílio da biblioteca `libpcap`. Em virtude dos bons resultados apresentados, optou-se por utilizar a abordagem prevista na arquitetura NERD [Terra and Gondim 2021]. Os pacotes de dados, ao serem transmitidos pelo terminal analisado, são detectados pelo ECol e armazenados em uma base intitulada Base de Fluxos. Os pacotes gerados por *rootkits* que ofuscam o tráfego de rede não são identificados e, portanto, não serão adicionados à Base de Fluxos. Dessa forma, é possível a um elemento exterior diferenciar tráfegos legítimos de ofuscados.

2.2. Elemento Auditor (EAud)

O elemento auditor possui a função de comparar o tráfego capturado na rede com o percebido pelo Coletor. Assim, fluxos que foram ofuscados e não foram detectados pelo ECol, ao trafegarem na rede, serão detectados pelo EAud e, ao ser comparado com a tabela de fluxos do ECol, classificados como suspeitos. o EAud não executa internamente a função de bloqueio, permitindo que trabalhe *out-of-band*. Dessa forma, análises mais complexas podem ser realizadas antes do bloqueio e sem impacto de desempenho na comunicação do terminal, ou seja, sem afetar performance de transmissão ou tráfego em tempo real.

O EAud foi separado em sistemas (módulos), conforme Figura 2, facilitando alterações nos métodos de captura e análise, inserção de novas bases de enriquecimento e execução das etapas de processamento de forma assíncrona.

O Sistema de Análise Inicial foi desenvolvido em linguagem C, em virtude de ser o módulo mais crítico e que precisa ser performático. Ele utiliza uma lista de bibliotecas disponíveis para linguagem C, em especial a utilização da biblioteca `etcdlib.h` para comunicação com o coletor, `libpq-fe.h` para comunicação com o banco de dados relacional, padrão SQL (PostgreSQL), e `pcap.h` para criação de um *handle* de captura de pacotes. A cada pacote, o sistema verifica se o fluxo é conhecido, através da Base Temporária (em memória RAM). Caso não seja conhecido, é feita consulta ao ECol. Se for considerado legítimo, é adicionado na Base Temporária; caso contrário, na Base de Conexões (fluxos suspeitos). Pacotes DNS Response também são tratados e a relação de IPs e domínios é adicionada na Base DNS.

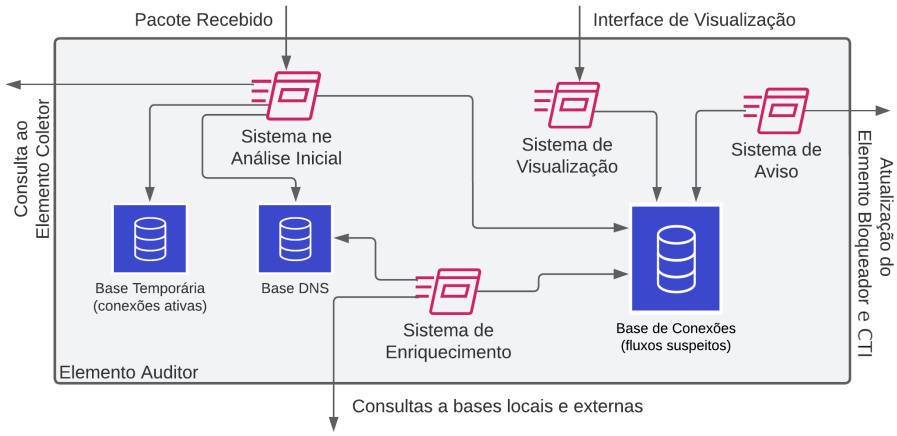

Figura 2. Elemento Auditor da estrutura PARDED (Figura do Autor)

O Sistema de Enriquecimento verifica, através da Base de Conexões e de forma assíncrona, quais dados ainda não foram enriquecidos e consulta as bases (locais e externas) configuradas. O Sistema de Enriquecimento atualiza então a Base de Conexões com as novas informações obtidas. Foi implementado em Python 3.

O Sistema de Aviso verifica, através da Base de Conexões e também de forma assíncrona, quais destinos atingiram os limiares configurados para que um fluxo suspeito seja considerado malicioso. Quando um limiar é atingido, o sistema cria arquivos de aviso ou bloqueio e repassa a regra ao Elemento Bloqueante. Essas regras devem ser customizadas de acordo com a API disponibilizada pelo Elemento Bloqueante, seja ele um firewall ou IPS. Na implementação apresentada, os arquivos de bloqueio estão no formato *Snort*. São criados, ainda, arquivos de CTI no formato STIX 2.1, permitindo integração com sistemas de inteligência de ameaças. Também foi implementado em Python 3.

O Sistema de Visualização permite que o analista de segurança verifique os fluxos assinalados como suspeitos e suas características, além de um panorama de funcionamento da própria ferramenta, facilitando a utilização em ambientes de monitoramento e integração com demais sistemas de segurança. Implementado em linguagem Python 3, com auxílio do *framework* de visualização de dados *Dash*, facilitando a criação de *dashboards* interativos e atualizações em tempo real.

2.3. Elemento Copiador

A função do Elemento Copiador é permitir o funcionamento passivo do Elemento Auditor, espelhando o tráfego que passa por ele. Assim, o tráfego original é encaminhado para o próximo elemento de rede e sua cópia é enviada para análise do EAud, evitando atrasos na transmissão de dados ao destino.

A arquitetura PARDED não define como deve ser implementado o Elemento Copiador, pois é possível a utilização de qualquer equipamento existente na infraestrutura que realize cópias dos pacotes trafegados, como comutadores, roteadores, hubs ou firewalls, facilitando a integração em redes complexas.

2.4. Elemento Bloqueante (EBlk)

O Elemento Bloqueante (EBlk) é um equipamento *inline* que verifica se o pacote de rede recebido deve ou não ser bloqueado, baseado no aviso recebido pelo EAud. Esse elemento pode ser qualquer dispositivo que permita receber atualizações de regras de alerta ou bloqueio de forma remota, como roteadores e firewalls.

3. Desempenho

O desempenho da arquitetura foi testado através da geração de tráfego e utilização de um *rootkit* em ambiente controlado. Para tal, foram utilizados os softwares `wget` e `iperf` para criar tráfego legítimo e o Rootkit Linux Nuk3Gh0st [Schällibaum 2019] para o tráfego malicioso (entretanto, qualquer rootkit que utiliza técnicas de ofuscação de tráfego pode ser utilizado). Tanto o Elemento Coletor quanto o Auditor foram executados em máquinas virtuais com o sistema operacional Linux Debian 11 (*bullseye*) (x86_64, *release* 5.10.0-16, *version* 5.10.127-1).

O primeiro teste foi executado sem atuação de ofuscação de tráfego, ou seja, apenas com fluxos de dados legítimos. Em seguida, foram realizados testes com transmissão apenas de fluxos maliciosos. Por fim, um terceiro cenário foi gerado para testes de detecção de fluxos legítimos e suspeitos trafegando concomitantemente.

3.1. Fluxos Legítimos

Nesse cenário, o *rootkit* não estava ativo e tráfego legítimo foi gerado a partir do ECol, com espelhamento de tráfego ao EAud. O resultado esperado era a detecção de todos os fluxos como legítimos, com inclusão dos fluxos conhecidos na Base Temporária (EAud) e das respostas de consultas DNS inseridas na Base DNS (EAud). O teste, contendo de 50 a 60 mil pacotes, foi executado 30 vezes. O desempenho obtido está demonstrado na Tabela 1, com a média de pacotes por fluxo de cada execução e o tempo de processamento por pacote (com e sem carga de 50Mbps).

Tabela 1. Tempo de resposta do sistema (sem atuação do *rootkit*).

Tipo de Processamento	Característica do Fluxo	Média de Pacotes	Tempo de processamento (ms)	Tempo de proces. com carga de 50Mbps(ms)
Sem consulta ao Coletor	Base Temporária	55.393	0,00110	0,00100
	Resposta DNS	2030	0,00407	0,00734
Com consulta ao Coletor	Fluxos Legítimos	983	516,78	551,05

Todos os pacotes recebidos no EAud foram percebidos corretamente como legítimos. Os pacotes que não estavam presentes na Base Temporária do EAud (o que requer consulta ao ECol) foram, em média, processados em 0,5 segundos. Os demais pacotes do fluxo (já inseridos na Base Temporária), 95% do total gerado, foram processados em menos de 0,0012 milissegundos cada pelo Sistema de Análise Inicial. Os pacotes "DNS response", tratados localmente para detecção dos possíveis domínios utilizados pelos *rootkits*, foram processados em aproximadamente 0,0041 milissegundos.

Tabela 2. Tempo de resposta do sistema pra fluxos exclusivamente suspeitos.

Tipo de Processamento	Característica do Fluxo	Média de Pacotes	Taxa (pacotes por segundo)	Tempo de processamento (ms)
Com consulta ao Coletor	Fluxos Suspeitos	100	1 pps	1.742,77
		100	10 pps	1.674,18
		500	10 pps	1.838,37

3.2. Fluxos Exclusivamente Maliciosos

Nesse cenário, o desempenho foi medido através da geração de pacotes maliciosos (não detectados pelo ECol). Foram utilizados apenas pacotes TCP SYN com porta de origem diferente para cada pacote, garantindo que cada um dos pacotes de dados transmitidos fosse detectado pelo EAud como um novo fluxo. Essa abordagem permitiu verificar o desempenho do sistema no pior cenário, onde cada pacote gera uma consulta ao ECol e gera também a necessidade de atualização da Base de Conexões. Os resultados estão presentes na Tabela 2. O teste foi executado 30 vezes para cada um dos 3 cenários apresentados (100 pacotes a 1pps, 100 pacotes a 10pps e 500 pacotes a 10 pps).

Todos os pacotes recebidos e tratados no EAud foram percebidos corretamente como suspeitos, ou seja, não houve falsos-positivos. O tempo de resposta foi, em média, de 1,71 segundos. A elevação do tempo de processamento deve-se ao fato do tratamento e atualização da base de conexões suspeitas, comprovando a diferença entre utilização de API do banco de dados relacional PostgreSQL e consultas diretas em memória volátil.

3.3. Fluxos Legítimos e Maliciosos

Para detecção de fluxos suspeitos em uma situação próxima a uma rede real, foram transmitidos ao mesmo tempo pacotes legítimos e maliciosos. Nesse cenário, o *rootkit* estava ativo e foi realizada transmissão de tráfego ofuscado entre os fluxos legítimos, correspondente a aproximadamente 2% do tráfego sem carga (100 fluxos de 10 pacotes cada), com intuito de verificar se os pacotes maliciosos seriam corretamente detectados e os dados inseridos na Base de Conexões. O teste, contendo de 50 a 60 mil pacotes, foi repetido 30 vezes. Os resultados estão demonstrados na tabela 3, com a média de pacotes por fluxo de cada execução e o tempo de processamento por pacote (com e sem carga).

Tabela 3. Tempo de resposta do sistema (com atuação do *rootkit*).

Tipo de Processamento	Característica do Fluxo	Média de pacotes	Tempo de processamento (ms)	Tempo de proces. com carga de 50Mbps(ms)
Sem consulta ao Coletor	Base Temporária	54.058	0,00114	0,00100
	Resposta DNS	1.329	0,0370	0,00730
Com consulta ao Coletor	Fluxos Legítimos	684	536,15	561,26
	Fluxos Suspeitos	100	1.642,64	1.709,62

Não houve alteração significativa no tempo de processamento de pacotes suspeitos, de *DNS Response* ou de pacotes presentes na Base Temporária. Todos os pacotes foram detectados corretamente, sem falsos-positivos ou falsos-negativos.

Tabela 4. Ações do Sistema de Enriquecimento.

Tipo de Processamento	Descrição da Base de Dados	Número de Consultas	Resposta (ms)
Consulta à base local	Nó pertencente a rede TOR	100	42,18
Consulta à base remota	Plataforma VirusTotal	30	1.241,14

3.4. Enriquecimento

Os testes de enriquecimento foram realizados com divisão em dois tipos de consulta: local, realizado com auxílio de base de dados de nós da rede TOR [Dingledine et al. 2004], obtida através do website *dan.me.uk* [Dan 2022]; e remoto, realizados em tempo real através de consultas na API da plataforma Virustotal [Chronicle Security 2023]. A ferramenta possui ainda geolocalização dos endereços de destino, obtida através da base GeoLite2-City, disponibilizada pelo empresa Maxmind Inc. [Maxmind 2023].

As consultas do Sistema de Enriquecimento feitas utilizando a consulta local (base de dados de nós TOR) foram processadas em aproximadamente 42 milissegundos cada. Em média, foram necessários cerca de 1,2 segundos para consultas do Sistema de Enriquecimento na base remota. Os resultados estão apresentados na Tabela 4.

3.5. Detecção de Limiares

Para execução dos testes de desempenho do Sistema de Aviso, foram definidos dois cenários possíveis: o primeiro, quando a consulta não encontra fluxos que atingiram os limiares de detecção; o segundo, quando um limiar de detecção é atingido e ações posteriores são executadas (criação de regra de bloqueio no formato Snort e CTI no formato STIX). Os resultados estão demonstrados na Tabela 5.

Tabela 5. Ações do Sistema de Aviso.

Tipo de Processamento	Número de Consultas	Resposta (ms)
Sem bloqueio (limiares não atingidos)	50	9,31
Com bloqueio (limiares atingidos)	50	89,60

4. Demonstração

A demonstração a ser apresentada, compilada nos vídeos Auditor (https://youtu.be/oCjbl4vpyO8) e Coletor (https://youtu.be/MBCC6ZzzPZk), descreve as principais funcionalidades da ferramenta, executando a geração de tráfego legítimo e malicioso, permitindo a visualização das técnicas de análise do tráfego, de enriquecimento dos dados e de atingimento de limiares de detecção (com geração dos arquivos de inteligência de ameaças cibernéticas e regras de bloqueio), além da interface visualização de fluxos suspeitos.

5. Código e manuais

O código fonte encontra-se no endereço https://github.com/maickelj/parded. O usuário e a senha para acesso, caso necessário, é “parded2023sbseg”. O arquivo README contém

as instruções necessárias para execução do código, requisitos de instalação e compilação. Os vídeos “auditor.mp4” e ”coletor.mp4”contêm, assim como o conteúdo dos links apresentados na seção 4, a demonstração do funcionamento da solução.

6. Conclusão

Esse trabalho apresentou uma arquitetura capaz de detectar *rootkits* que utilizam técnicas de ofuscação de comunicação no terminal infectado, de forma passiva, escalável, generalizável e adaptável, sem impactos significativos no desempenho da infraestrutura de rede, com enriquecimento de dados de múltiplas fontes e incorporação das informações oriundas de múltiplos terminais, permitindo integração com outros sistemas de defesa previamente existentes, como IDS e firewalls.

Portanto, o PARDED permite novos estudos de técnicas de detecção e bloqueio de *malwares* baseados em rede, enriquecimento de informações maliciosas e geração de *insights* para analistas de segurança cibernética.

Referências

Bromander, S., Jøsang, A., and Eian, M. (2016). Semantic cyberthreat modelling. *STIDS*, pages 74–78.

Chronicle Security (2023). Virustotal. http://www.virustotal.com. Acessado em 15 jul 2023.

Dan (2022). Tor Node List. https://www.dan.me.uk. Acessado em 12 jul 2023.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC.

Julian, V. and Botti, V. (2019). Multi-Agent Systems. *Applied Sciences (Switzerland)*, 9(7).

Marques, R. S., Epiphaniou, G., Al-Khateeb, H., Maple, C., Hammoudeh, M., De Castro, P. A. L., Dehghantanha, A., and Choo, K. K. R. (2021). A Flow-based Multi-Agent Data Exfiltration Detection Architecture for Ultra-low Latency Networks. *ACM Transactions on Internet Technology*, 21(4).

Maxmind (2023). GeoLite2 Free Geolocation Data. https://dev.maxmind.com/geoip/geolite2-free-geolocation-data?lang=en. Acessado em 24 Fev 2023.

Schällibaum, J. A. (2019). Nuk3 gh0st. https://github.com/juanschallibaum/Nuk3Gh0st. Acessado em 10 jul 2023.

Terra, M. B. and Gondim, J. J. (2021). NERD: A Network Exfiltration Rootkit Detector based on a Multi-agent Artificial Immune System. *2021 Workshop on Communication Networks and Power Systems, WCNPS 2021*.

Tounsi, W. and Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security*, 72:212–233.

Trinks, M., Gondim, J., and Albuquerque, R. (2023). Multi-agent architecture for passive rootkit detection with data enrichment. In *CSEI*, pages 29–41, Cham. Springer Nature Switzerland.