IDENTIFICAÇÃO DE TÚNEIS DNS EM NUVEM COMPUTACIONAL USANDO DETECÇÃO DE ANOMALIAS

Lorena de Souza Bezerra Borges, Robson de Oliveira Albuquerque,
Fábio Lúcio Lopes de Mendonça, Georges Daniel Amvame-Nze, Edna Dias Canedo
e Rafael Timóteo de Sousa Jr
Programa de Pós-gradução em Engenharia Elétrica – PPEE, Departamento de Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília – UnB, Brasília, Brasil, Zip Code 70910-900

RESUMO

A técnica de tunelamento DNS utiliza recursos do protocolo DNS para estabelecer canais de comando e controle entre máquinas cliente e servidores remoto, podendo ser explorada para acesso não-autorizado e exfiltração de dados privados. Atualmente, os ataques de tunelamento DNS afetam sistemas multiplataforma, englobando recursos computacionais local e em nuvem. Este artigo propõe um modelo operacional de identificação de túneis DNS usando detecção de anomalias em um ambiente Amazon Web Service (AWS). Ferramentas de tunelamento DNS foram testadas para produzir requisições anômalas e construir uma base de dados integrada à pilha ELK, processando métodos de aprendizado de máquina (machine learning) não-supervisionados, para análise e detecção de atividades maliciosas. O modelo proposto resultou em altos níveis de precisão e constituiu uma evolução no contexto de segurança em nuvem para as arquiteturas corporativas.

PALAVRAS-CHAVE

Túnel DNS, Tunelamento DNS, Cibersegurança, AWS, Nuvem Computacional, ELK

1. INTRODUÇÃO

Nos últimos anos vem aumentando o número de ataques baseados em *malwares* que funcionam como módulos agentes, infectando máquinas para estabelecimento de túneis de comunicação no intuito de extrair informações sensíveis e sigilosas das organizações. A exfiltração de dados não-autorizados provoca prejuízos, tanto financeiros quanto relacionados à confiabilidade de instituições, representando um relevante desafio para a segurança cibernética. Um dos mecanismos mais utilizados no vazamento de dados é o tunelamento DNS, onde códigos maliciosos estabelecem canais de comando e controle (C2) e encapsulam informações em pacotes DNS. Através dos túneis C2 é possível a transferência de comandos remotos entre o atacante e a máquina afetada, além da manipulação de dados privados de forma indevida. (Ishikura *et al.*, 2021).

Pesquisadores da Akamai reportaram a identificação de aproximadamente 13 milhões de novos domínios maliciosos por mês, apenas na primeira metade do ano de 2022, representando 20.1% de todos os domínios criados no mesmo período (Zurier, 2022). Recentemente, analistas de segurança da SentinelLabs identificaram um grupo responsável por espionagem e roubo de dados direcionado a países asiáticos, que utilizavam intensamente técnicas de tunelamento DNS, de forma silenciosa e há 10 anos, para transferência de informações após comprometimento do alvo (Chen, 2022).

A técnica de tunelamento DNS como ataque cibernético representa uma preocupação atual, principalmente pela eficiência dos incidentes e inerente dificuldade de detecção. O protocolo DNS é amplamente utilizado na Internet para tradução de nomes de domínios em endereços IP, possuindo características hierárquicas e recursivas no fluxo de comunicações entre servidores recursivos e autoritativos (Mockapetris, 1987). Devido ao essencial propósito do protocolo para navegação na Internet, a porta UDP/53, além de ser liberada de bloqueios de segurança, é indevidamente monitorada por ferramentas de controle de perímetro de rede.

Diante do exposto e como diferencial, a análise de tráfego DNS tunelado precisa ser multiplataforma, combinando recursos na nuvem e de rede local (*on-premise*), de maneira dinâmica, independente e escalável.

Nesse sentido, o objetivo deste estudo é criar mecanismos efetivos capazes de detectar tunelamento DNS, de maneira prática e funcional, como ferramenta para sistemas de controle de segurança organizacionais. Para tanto, propomos uma arquitetura modular, com altos índices de acurácia, composta por processos de coletas de registros DNS na nuvem Amazon Web Service (AWS) e flexibilidade de armazenamento de *dataset*. Por fim, a arquitetura possui análise e seleção de parâmetros baseados em estudos correlatos e modelagem dos dados com algoritmo de ML não-supervisionado para identificação de anomalias.

2. TRABALHOS RELACIONADOS

Muitos estudos vêm sendo desenvolvidos para detecção de tunelamento DNS na última década, incluindo análises de dados para extração de parâmetros e utilização de machine learning (ML) para criar inferências e detecção de comportamentos anômalos na rede. Wang et al. (2021) abordou várias técnicas de detecção de tunelamento DNS entre 2006 e 2020, classificando parâmetros entre pacotes e fluxos DNS, assim como diferenciando os métodos de detecção entre regras, assinaturas e baseados em ML.

Bai *et al*. (2021) e Ishikura *et al*. (2021) focaram na combinação eficiente entre seleção e extração de parâmetros identificadores de tunelamento DNS e algoritmos com o maior nível de acurácia possível. Em Chen *et al.* (2021), houve a definição de duas categorias de parâmetros: pacotes DNS, processados em tempo real e em sessões DNS, que dispensam análise de carga útil dos pacotes, porém com alta complexidade computacional. Em D'Angelo *et al.* (2022), foram extraídos dados do tráfego DNS para formação de imagens bidimensionais com objetivo de realizar classificação por algoritmos de redes neurais.

Apesar de inúmeros estudos na área, organizações vêm enfrentando desafios práticos e operacionais na implementação de modelos de detecção de anomalias, pela diversidade de fontes de dados em ambientes híbridos, no gerenciamento de grandes volumes de bases de dados (*dataset*) e na aplicação de algoritmos de ML isolados do restante da arquitetura (Liberty *et al*. 2020). Em complemento às análises citadas, este artigo sugere uma arquitetura operacional, com ênfase na coleta de dados de serviços na nuvem e seleção de parâmetros, com foco na performance dos algoritmos integrados ao sistema de monitoramento, para identificação efetiva de tunelamento DNS na rede.

3. METODOLOGIA

3.1 Arquitetura

A solução proposta descreve uma arquitetura modular composta de processos de coleta de consultas DNS em ambiente de nuvem computacional e elaboração de um *dataset* a partir de testes com ferramentas de tunelamento DNS amplamente utilizadas. A arquitetura ainda contempla a extração de parâmetros indicativos de tráfego malicioso e detecção de anomalias através do algoritmo de ML não-supervisionado Population, da solução Elastic Stack (ELK). O perfil dos serviços utilizados na nuvem foi na modalidade IaaS (*Infrastructure as a Service*), com gerenciamento independente, dimensionamento dinâmico de armazenamento e autonomia de softwares e aplicações. A solução ELK proporcionou módulos integradores para comunicação com serviços na AWS, manipulação e indexação de dados, visualização consolidada através de painéis Kibana e execução de algoritmos para inferência de padrões nos dados.

3.2 Topologia

Em uma determinada Availability Zone (AZ) foi configurada uma rede privada AWS com escopo de endereçamento interno segmentado em duas subredes: uma para máquinas clientes (instâncias EC2 Ubuntu 20.0 e Windows 10 Desktop) e outra para servidores DNS Bind9 e ELK. Uma máquina Kalilinux atacante foi configurada em uma AZ distinta, acionando módulos de ferramentas de tunelamento DNS e escutando requisições na porta UDP/53, representando o servidor atacante (Figura 1).

Instâncias EC2 possuem serviço de resolução de nomes nativo na AWS chamado Route 53 Resolver, porém como ponto de coleta de requisições redundante, cada máquina cliente foi configurada para utilizar primariamente o servidor Bind9. A escolha de um resolvedor DNS adicional tem por objetivo permitir diferentes tipos de registros para indexação, enfatizando a flexibilidade do processo de coleta da arquitetura.

O domínio *lsbb.link* foi registrado no AWS Route53 e o subdomínio *t1ns.lsbb.link* aponta para a máquina remota Kalilinux. Dessa forma, os módulos clientes das ferramentas de tunelamento alcançam recursivamente a máquina atacante através do domínio malicioso.

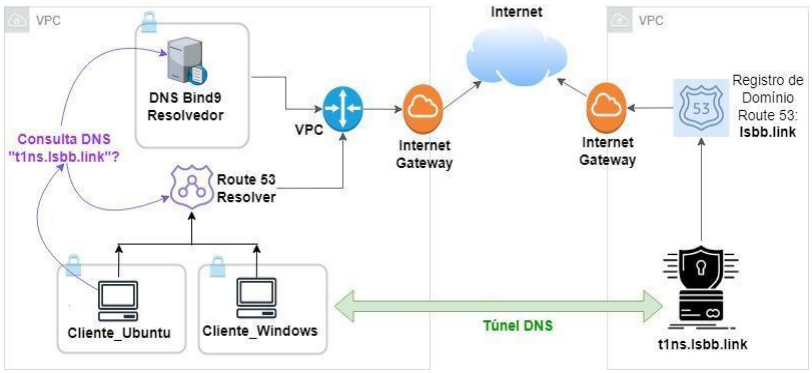


Figura 1. Topologia laboratório para tunelamento DNS

3.3 Coleta de Dados

Este estudo propõe dois métodos de coleta: pacotes DNS e fluxos DNS. Os pacotes DNS fornecem dados sobre domínios consultados, resource records (RR), tamanho do pacote e outros sinalizadores descritivos disponíveis em tempo real (*stateless*), sem a necessidade de uma coleta subsequente de pacotes. A perspectiva de fluxos DNS inspeciona padrões na comunicação ponta a ponta (*statefull*), porém aumentando o tempo de resposta da análise pela espera no recebimento de todos os pacotes para um mesmo evento (Mahdavifar, 2021).

O servidor resolvedor Bind9 representa um dos pontos de acesso para coleta (*endpoint*) e o módulo Packetbeat da solução ELK (Packetbeat, 2022) representa o agente integrador para captura de registros (*logs*) de consultas DNS e envia ao servidor ELK, de forma automatizada e sequencial, para indexação dos dados. Packetbeat também é capaz de coletar registros de fluxo DNS para um evento de requisição, representado pelo **sensor 1**, enquanto os registros de pacotes DNS representam o **sensor 2**, conforme a Figura 2.

O Virtual Private Cloud (VPC) e o Route 53 Resolver, ambos serviços nativos da AWS, também foram configurados como *endpoints* para coleta de dados. Um serviço de *logs* de fluxo de VPC foi configurado para coletar *logs* de tráfego da rede interna do laboratório, a cada 5 minutos, e enviá-los para um bucket S3. Da mesma forma, um serviço de envio de *logs* de requisições DNS para o Route 53 Resolver foi configurado e os dados enviados para outro bucket S3. O módulo integrador Filebeat (Filebeat, 2022) é responsável por verificar os registros recém-gravados nos buckets S3 e transferi-los para o servidor ELK (**sensor 3**).

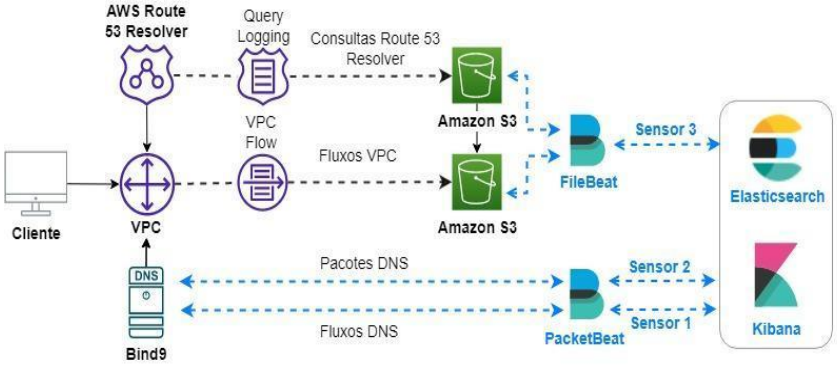


Figura 2. Processo de coleta de registros DNS

Os dados são previamente processados e indexados pelos módulos integradores para então serem enviados para a pilha ELK para futuras manipulações e buscas. Os índices resultantes são visualizados no Kibana após classificação, identificação de valores nulos, limpeza de dados e criação de novos índices a partir da combinação de dados brutos. A pilha ELK estrutura os dados em uma linha do tempo, possibilitando extrair recursos e alimentar métodos de ML para detecção de anomalias no tráfego DNS (Collier, 2019).

3.4 Seleção de Parâmetros

As principais características afetadas pelo tunelamento DNS representam indícios na construção de variáveis e dimensões para uma análise sistemática. A combinação de métricas estatísticas resulta em um processo de extração de parâmetros para o sistema de detecção proposto, detalhado na Subseção 3.5.

Número de solicitações de eTLD+1: para transmitir dados encapsulados em consultas DNS, há um aumento anormal no número de solicitações para o mesmo eTLD+1 (*Effective Top Level Domain plus one*), que representa o TLD (*.link*) mais uma camada de subdomínio (*lsbb.link*).

Tamanho dos pacotes DNS em bytes: Os pacotes UDP DNS sofrem aumento de tamanho ao transportar dados através dos subdomínios ou campos RR. Assim, as taxas de transferência em bytes, para um mesmo evento DNS, apresentam valores anômalos elevados.

Resource Records Types: de acordo com Herrmann et al. (2013), os tipos de RRs mais comuns são A (IPV4), AAAA (IPV6) e PTR (*reverse lookup pointers*), representando 99,4% das solicitações padrão. As ferramentas de encapsulamento tendem a alternar ou modificar os tipos de RRs utilizados nas consultas DNS (CNAME, TXT, MX, etc.) para aumentar a largura de banda de dados transferidos.

Quantidade de dados transmitidos: em um evento de tunelamento DNS, há um aumento tanto na duração do evento quanto na quantidade de dados recebidos e enviados para o mesmo par de IPs na comunicação.

Time-to-live TTL: é importante que o tráfego DNS encapsulado tenha o menor TTL possível para que as solicitações ao domínio malicioso não sejam armazenadas em cache no resolvedor local, forçando altas taxas de cache miss (falha em encontrar o subdomínio nos registros do servidor resolvedor), Ishikura *et al*. 2021.

3.5 Métodos de Detecção

O modelo ML Population detecta atividades incomuns de acordo com o comportamento anterior da população em análise (PopulationElastic, 2022). O modelo de detecção de anomalias usa séries temporais e analisa os dados em incrementos de tempo. Nossa abordagem de método utiliza aprendizado não-supervisionado e sem assistência prévia para treinamento do modelo. No entanto, a precisão do modelo de detecção de anomalias depende da atualização contínua de dados para definição de padrões (Veasey e Dodson, 2014).

O método Population divide parâmetros em dimensões e usa distribuição de probabilidades, como Poisson, Gaussian, log-normal ou combinações de modelos, para definir uma linha de base comportamental para cada variável. A análise é baseada na linguagem do teste de hipóteses, que descreve a maioria das observações como a hipótese nula (*null hypothesis*). O valor p (*p-value*) de uma estatística de teste é usado para rejeitar a hipótese nula e, em detecção de anomalias, identificar um dado como desvio ou *outlier* (Veasey e Dodson, 2014).

Os níveis de eventos anômalos são determinados empiricamente pelo *p-value* entre 0 (sem possibilidade) e 1 (certeza absoluta) (Collier, 2019). Quanto menor a probabilidade mais distante da distribuição normal e os alertas são gerados com uma pontuação normalizada, entre o intervalo de 0 a 100 (crítico acima de 75).

As características identificadas na Subseção 3.4 foram divididas em dimensões e combinadas com métricas estatísticas como valores únicos, máximos, médios ou inferiores, para isolar situações anômalas e criar métodos de detecção (*jobs*). Os métodos de detecção de túneis DNS foram classificados como *jobs* de fluxos DNS, Tabela 3 e *jobs* de pacotes DNS, Tabela 4. Esta divisão visa verificar a influência dos parâmetros DNS na detecção de anomalias pelo impacto do processamento em tempo real.

Tabela 3. Fluxo DNS por IP de destino e por eTLD+1

Fluxo DNS por IP de destino		**Fluxo DNS por eTLD+1**	
Parâmetro	**Detector**	**Parâmetro**	**Detector**
F1	high_count	F7	max("event.duration")
F2	max ("network.bytes")	F8	max ("bytes_in")
F3	max ("source.bytes")	F9	max ("bytes_out")
F4	distinct_count ("source.port")		
F5	distinct_count ("source.ip")		
F6	distinct_count ("destination.port")		

Tabela 4. Pacotes DNS por eTLD+1 e por TLD

Pacote DNS por eTLD+1		Pacote DNS por TLD	
Parâmetro	**Detector**	**Parâmetro**	**Detector**
F10	high_count	F19	high_count
F11	distinct_count ("dns.question.name")	F20	distinct_count ("dns.question.name")
F12	distinct_count ("dns.question.subdomain")	F21	distinct_count ("dns.question.subdomain")
F13	distinct_count ("dns.id")	F22	distinct_count ("dns.id")
F14	high_mean ("dns.answers_count")	F23	high_mean ("dns.answers_count")
F15	low_mean ("dns.answers.ttl")	F24	low_mean ("dns.answers.ttl")
F16	distinct_count("dns.answers.name")	F25	distinct_count("dns.answers.name")
F17	distinct_count("dns.answers.type")	F26	distinct_count("dns.answers.type")
F18	high_mean ("dns.opt.udp_size")	F27	high_mean ("dns.opt.udp_size")
		F28	distinct_count ("dns.question.type")
		F29	high_mean ("dns.answers.data")

4. AVALIAÇÃO

4.1 Dataset

O *dataset* consistiu em dados de solicitações usuais e aqueles gerados por ferramentas de tunelamento DNS, em uma janela de três semanas de testes. Para dados benignos, foram feitas consultas a sites legítimos, bem como navegação aleatória na Web, como e-mail, streaming de vídeo, localização geográfica, notícias etc. Para popular o *dataset*, *scripts* de consulta sequenciais foram executados nos 10.000 domínios mais consultados (TopDomains10k, 2022) e 1.000.000 principais domínios de acesso na Internet (TopDomains1M, 2022).

As ferramentas de tunelamento DNS testadas foram: Iodine para testes C2, Dnscat2 e DNSExfiltrator para transferência de arquivos. Ainda foi testado o DNSStager para infiltração de arquivos, no sentido servidor remoto para máquina atacada, usando o mesmo domínio malicioso *lsbb.link*. Flightsim, utilitário desenhado para gerar tráfego de rede malicioso e auxiliar equipes de segurança cibernética a avaliar controles, foi testado para um domínio diferente (*alphasoc.xyz)*, enviando comandos do tipo *heartbeat* e validando nosso modelo de detecção para uma estrutura de túnel previamente desconhecida para o algoritmo.

4.2 Resultados Experimentais

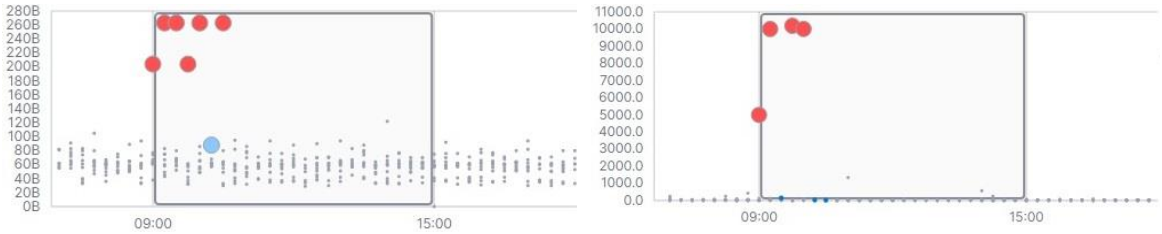
Métodos de detecção para fluxo DNS: Para o Iodine, os eventos DNS foram considerados anômalos e receberam pontuações altas, entre 92 e 94, devido ao aumento no número de dados transferidos da mesma origem e IP de destino, alta contagem de portas UDP abertas, bem como fluxos DNS na rede com mesmos índices identificadores. Tal comportamento indicou um fluxo contínuo de comunicação na porta UDP/53.

Para o Dnscat2, prevaleceram as mesmas características influenciadoras para Iodine, porém as transferências de dados aumentaram o nível da anomalia pela quantidade de bytes em trânsito, tanto de entrada quanto de saída, resultando em eventos decisivos na identificação de ameaças (pontuação 96). A perspectiva de fluxo não influenciou para o Flightsim, apesar do teste com dois disparos curtos (50 consultas para cada disparo). Valores reduzidos de consultas são silenciosos por serem semelhantes a comportamentos normais.

Para a ferramenta DNSStager houve alto índice de criticidade pela identificação de transferência de dados, com tempo anormal de sessão de eventos, além das subsequentes consultas DNS entre os mesmos recursos relacionados (pontuação 92). Finalmente, para a ferramenta DNSExfiltrator, a perspectiva de fluxo DNS foi a única que identificou anomalias pelo fato da ferramenta não realizar requisições ao domínio malicioso, não fornecendo assim informações a serem coletadas no DNS resolvedor, ou seja, a nível de pacotes DNS.

Por exemplo, as Figuras 3.a e 3.b indicam, respectivamente, os valores máximos de bytes recebidos e duração de eventos, durante um tunelamento por Dnscat2. Nas duas situações, as sombras em cinza representam dados considerados dentro do padrão comportamental da variável. Para os dados em vermelho, temos a identificação de anomalias pela probabilidade do *p-value* ser muito baixa, cerca de 5,56e-36 para bytes recebidos e 4,07e-9 para duração do evento, demonstrando assim desvios comportamentais nas amostras.

Métodos de detecção para pacote DNS: Apesar da ferramenta Iodine ter transmitido apenas comandos entre o servidor e a máquina afetada, ou seja, comunicação do tipo C2 (dados leves), o número de requisições para o mesmo eTLD+1 malicioso é relativamente alto, resultando em índices eficientes para detecção de anomalias com pontuações entre 96 e 99.

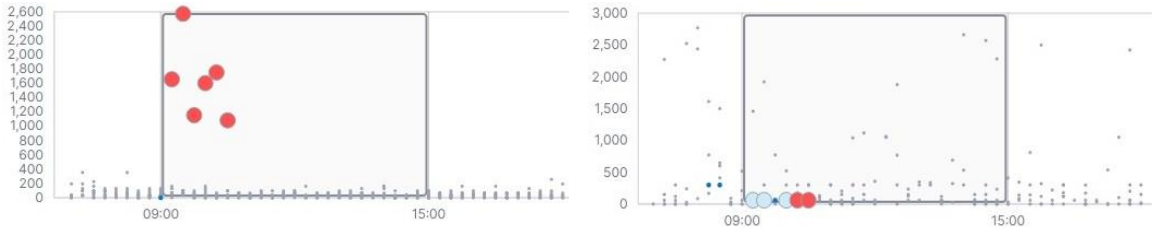


a) Bytes recebidos b) Duração de eventos

Figura 3. Representação de parâmetros de fluxo durante tunelamento DNS por Dnscat2

As detecções para Dnscat2 e DNSStager obtiveram alta acurácia, pontuando 99 e 95, respectivamente. Como DNSStager infiltrou dados na máquina atacada, é possível perceber que os métodos de detecção escolhidos foram eficientes, inclusive quando o sentido da comunicação é em *down streaming*. Os arquivos transferidos nos testes tinham em torno de 100KB, mostrando que os parâmetros de pacotes DNS selecionados neste estudo foram os mais eficazes quando usados para ferramentas baseadas na administração de domínios maliciosos na Internet.

O utilitário Flightsim, com o objetivo de gerar um número reduzido de consultas para um domínio diferente, processando requisições para verificação de *status* do canal, ainda recebeu uma pontuação crítica, mostrando que mesmo com apenas 1 disparo (pontuação 92) e 2 disparos (pontuação 93), o modelo proposto neste estudo identificou a anomalia de forma eficiente, em situações adversas ou mais semelhantes ao padrão.



a) Tamanho em bytes para respostas recebidas b) Tempo de TTL das respostas

Figura 4. Representação de parâmetros de pacote durante tunelamento DNS por Dnscat2

A Figura 4.a indica um número elevado de dados em bytes nas respostas às consultas, característica comum das ferramentas de tunelamento pela inserção de dados nos pacotes DNS. Para transferir toda a informação, é necessário dividir os dados em várias requisições, aumentando a contagem de subdomínios diferentes para um mesmo TLD. A Figura 4.b destaca a anomalia para valores de TTL inferiores ao padrão em pacotes DNS ou curto prazo de validade das respostas, não sendo encontradas em cache nos servidores resolvedores.

4.3 Considerações

Os métodos de detecção de tunelamento DNS, quando sobrepostos, resultaram em altos níveis de detecção de comportamentos anômalos. A Figura 5 compara as pontuações sumarizadas para cada ferramenta testada, em uma linha do tempo. Domínios que geraram falsos positivos como *akam.net*, *amazonaws.com*, *cloudflare.com*, são classificados como conteúdo de armazenamento e serviços hospedados em nuvem, CDN's (*Content Delivery Network*) e proxies DNS. Embora o método tenha pontuado eventos legítimos, os domínios maliciosos receberam as maiores pontuações e estão no topo da criticidade.

Os recursos de fluxo DNS não foram eficazes na detecção de tunelamento DNS com ferramentas que geraram números reduzidos e limitados de consultas, portanto, não foram identificados valores discrepantes. No entanto, para métodos que utilizam os pacotes DNS, a avaliação de dados específicos oferece a identificação de anomalias de forma eficiente sem necessitar de uma janela subsequente de eventos. A detecção foi eficaz para ataques com larguras de banda menores ou em ocorrências esparsas.

Uma lista de domínios confiáveis poderia ter sido aplicada, melhorando a identificação da anomalia, porém, por se tratar de uma manipulação de dados estática, que precisaria ser constantemente atualizada, optou-se por avaliar os resultados com as amostras originais. Também é importante observar que alguns ataques do tipo *ransomware* registram novos domínios especificamente para os incidentes (contornando *blacklists*).

O domínio malicioso *t1ns.lsbb.link* teve as maiores pontuações para anomalia em todos os eventos de teste. O tráfego encapsulado foi classificado como crítico, entre 95 e 99, resultando em pontuações elevadas principalmente para ferramentas que registram domínio próprio e sem utilização de camadas extras de criptografia. A detecção do domínio *alphasoc.xyz* demonstra que o método não-supervisionado foi capaz de identificar um evento de tunelamento DNS desconhecido em tempo real e as pontuações levemente inferiores, 92 e 93, foram resultados do reduzido número de consultas geradas nos testes (situação adversa).

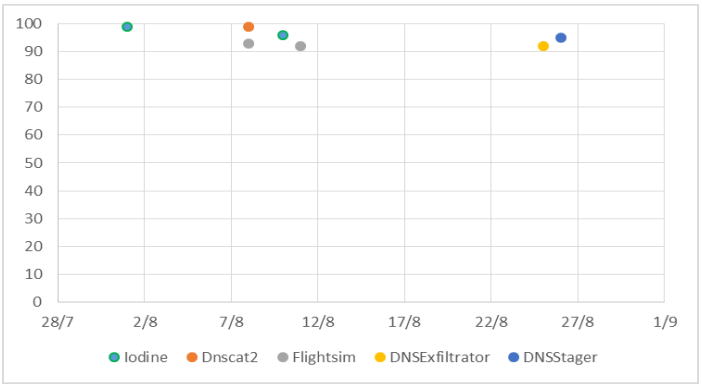


Figura 5. Eficiência da Metodologia de Detecção de túnel DNS

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho propôs um modelo de detecção de tráfego tunelado DNS que foi eficaz para as ferramentas Iodine, Dnscat2, DNSExfiltrator, DNSStager e Flightsim, para eventos de C2, exfiltração, infiltração de dados e comunicações do tipo *heartbeat*. Houve detecção de anomalias com altos níveis de criticidade, mesmo em transferências de arquivos leves ou poucas requisições no canal. Os eventos maliciosos atingiram níveis críticos (variando de 92 a 99) comprovando que o modelo de detecção de anomalias foi adequado para essa classe de ameaças, em termos de precisão e tempo de execução.

A integração de recursos na AWS, coleta e envio de dados para a solução ELK mostrou-se modular, flexível e operacional, podendo ser adaptada para compor soluções de defesa cibernética nas organizações. As camadas de coletas podem ser expandidas para abranger serviços *on-premises* e de outras plataformas em nuvem, de forma integrada e concentrando todos os registros na solução Elastic. Além do modelo de ML Population da ELK, outros algoritmos podem ser testados, comprovando a modularidade da arquitetura proposta.

Para trabalhos futuros, inúmeras ferramentas de tunelamento DNS ou *malwares* podem ser testados, além do treinamento de outros modelos de ML usando o *dataset* construído, adicionando novas perspectivas e inferências. Com a arquitetura proposta e a devida adaptação de parâmetros, é possível desenvolver estudos para tráfego DNS criptografado, DoT (DNS sobre TLS) e DoH (DNS sobre HTTPS), que representam desafios adicionais para análise e detecção de tráfego tunelado.

AGRADECIMENTOS

Este trabalho contou com suporte do CNPq - Conselho Nacional de Pesquisa (Outorgas 312180/2019-5 PQ-2 e 465741/2014-2 INCT em Cibersegurança), do Conselho Administrativo de Defesa Econômica (Outorga CADE 08700.000047/2019-14), da Advocacia Geral da União (Outorga AGU 697.935/2019), do Departamento Nacional de Auditoria do SUS (Outorga DENASUS 23106.118410/2020-85), da Procuradoria Geral da Fazenda Nacional (Outorga PGFN 23106.148934/2019-67), da Agência Brasileira de Inteligência (Outorga ABIN 08/2019) e da Universidade de Brasília (Outorga FUB/COPEI 7129).

REFERÊNCIAS

Askar. "*What Is DNSStager?*" GitHub, (28 Aug. 2022), github.com/mhaskar/DNSStager. Accessed 29 Aug. 2022.

Bai, H., Liu, W., Liu, G., Dai, Y. and Huang, S. (2021). *Application Behavior Identification in DNS Tunnels Based on Spatial-Temporal Information*. IEEE Access, 9, pp.80639-80653.

Bowes, R. (2022). GitHub - iagox86/dnscat2. [online] GitHub. Available at: <https://github.com/iagox86/dnscat2> [Accessed 4 August 2022].

Chen, J. (n.d.). Aoqin Dragon | *Newly-Discovered Chinese-linked APT Has Been Quietly Spying on Organizations For 10 Years*. [online] SentinelOne. Available at: https://www.sentinelone.com/labs/aoqin-dragon-newly-discovered-chinese-linked-apt-has-been-quietly-spying-on-organizations-for-10-years/ [Accessed 6 Oct. 2022].

Chen, S. et al. (2021) *DNS Covert Channel Detection Method using the LSTM model*, Computers & Security. Elsevier Advanced Technology. Available at: https://www.sciencedirect.com/science/article/pii/S0167404820303680 (Accessed: October 23, 2022).

Collier, R. and Azarmi, B. (2019). *Machine learning with the Elastic Stack*. Birmingham, UK: Packt Publishing.

D'Angelo, G., Castiglione, A. and Palmieri, F. (2022) *DNS tunnels detection via DNS-images, Information Processing & Management*. Pergamon. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0306457322000528 (Accessed: October 23, 2022).

Ekman, E. and Andersson, B. (2022). kryo.se: iodine (IP-over-DNS, IPv4 over DNS tunnel). [online] Code.kryo.se. Available at: <https://code.kryo.se/iodine> [Accessed 3 August 2022].

Elastic. (2022). Packetbeat: *Network Analytics Using Elasticsearch* | Elastic. [online] Available at: <https://www.elastic.co/pt/beats/packetbeat> [Accessed 9 October 2022].

Elastic. Filebeat (2022). *Lightweight Log Analysis & Elasticsearch*. [online] Available at: <https://www.elastic.co/pt/beats/filebeat> [Accessed 9 October 2022].

Elastic.co. (2022). *Performing population analysis | Machine Learning in the Elastic Stack [8.4]* | Elastic.[online] Available at: <https://www.elastic.co/guide/en/machine-learning/current/ml-configuring-populations.html> [Accessed 9 October 2022].

GitHub. (2022). GitHub - Arno0x/DNSExfiltrator: *Data exfiltration over DNS request covert channel*. [online] Available at: <https://github.com/Arno0x/DNSExfiltrator> [Accessed 4 August 2022].

Herrmann, D., Banse, C. and Federrath, H. (2013). *Behavior-based tracking: Exploiting characteristic patterns in DNS traffic*. Computers & Security, 39, pp.17-33.

Ishikura, N., Kondo, D., Vassiliades, V., Iordanov, I. and Tode, H. (2021). *DNS tunneling detection by cache-property-aware features*. IEEE Transactions on Network and Service Management, 18(2), pp.1203-1217.

Liberty, E., Karnin, Z., Xiang, B., Rouesnel, L., Coskun, B., Nallapati, R., Delgado, J., Sadoughi, A., Astashonok, Y., Das, P. and Balioglu, C. (2020, June). *Elastic machine learning algorithms in amazon sagemaker*. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (pp. 731-737).

Mahdavifar, S., Hanafy Salem, A., Victor, P., Razavi, A.H., Garzon, M., Hellberg, N. and Lashkari, A.H. (2021, December) . *Lightweight Hybrid Detection of Data Exfiltration using DNS based on Machine Learning*. In 2021 the 11th International Conference on Communication and Network Security (pp. 80-86).

Mockapetris, P. (1987). RFC 1035 - *Domain names - implementation and specification*. [online] Tools.ietf.org. Available at: <https://tools.ietf.org/html/rfc1035> [Accessed 3 August 2022].

Network Flight Simulator. GitHub (24 Aug. 2022), github.com/alphasoc/flightsim/blob/master/README.md. [Accessed 29 Aug. 2022.]

Nguyen, T. and Park, M. (2022). *DoH Tunneling Detection System for Enterprise Network Using Deep Learning Technique*. Applied Sciences, 12(5), p.2416.

OpenDNS Top Domains List. GitHub, (17 Aug. 2022), github.com/opendns/public-domain-lists. [Accessed 29 Aug. 2022].

Top-1000000-Domains. GitHub, (10 Aug. 2022), github.com/zer0h/top-1000000-domains/blob/master/top-1000000-domains. [Accessed 29 Aug. 2022].

Veasey, T. and Dodson, S. (2014). *Anomaly Detection in Application Performance Monitoring Data*. International Journal of Machine Learning and Computing, 4(2), pp.120-126.

Wang, Y., Zhou, A., Liao, S., Zheng, R., Hu, R. and Zhang, L. (2021). *A comprehensive survey on DNS tunnel detection.* Computer Networks, 197, p.108322.

Zurier, S. (2022). *Akamai finds 13 million malicious newly observed domains a month.* [online] SC Media. Available at: https://www.scmagazine.com/analysis/malware/akamai-finds-13-million-malicious-newly-observed-domains-a-month [Accessed 6 Oct. 2022].