Framework Analítico para Detecção de Ataques de ECN Não-Responsivo em Arquiteturas de Rede L4S usando Aprendizado Supervisionado

Lucas de Sousa Correia[1], **Jaaziel Batista da Silva**[1], **Jose Silva Floriano**[1]

[1]Instituto Federal da Paraíba (IFPB)
CEP 58015-435 – João Pessoa – PB – Brazil

1. Definição do Tema

A arquitetura L4S (Low Latency, Low Loss, Scalable Throughput) é um serviço de rede emergente projetado para resolver os problemas de latência e jitter (variação de latência) causados pelo bufferbloat em redes tradicionais. O L4S o faz implementando um mecanismo de Fila Dupla Acoplada (Dual Queue Coupled AQM) no roteador. Este mecanismo separa o tráfego em duas filas:

Fila Clássica (C): Para tráfego legado (ex: TCP Reno, CUBIC).

Fila L4S (L): Uma fila prioritária e ultra-rasa para tráfego de baixa latência.

Para que um fluxo de dados seja classificado na Fila L prioritária, o remetente marca seus pacotes com o codepoint ECN ECT(1). Ao fazer isso, o remetente estabelece um ”contrato” com a rede: ele sinaliza que está usando um controle de congestionamento escalável (como o TCP Prague) e que irá reagir adequadamente aos sinais de Congestionamento Experiente (CE) enviados pelo roteador.

O Vetor de Ataque: O Ator Não-Responsivo

O ataque ECN não-responsivo ocorre quando um usuário malicioso explora essa arquitetura para ganho próprio, em detrimento dos usuários legítimos.O ataque é executado da seguinte forma:

Engano (Ludibriação): O atacante envia um fluxo de dados marcando seus pacotes com ECT(1). Isso engana o roteador L4S, fazendo-o classificar o tráfego malicioso na Fila L, que é prioritária e de baixa latência.

Não-Implementação do Controle L4S: O ponto crucial do ataque é que o host do atacante não implementa um protocolo de controle de congestionamento L4S compatível, como o TCP Prague. O TCP Prague é projetado para entender o feedback ECN preciso (AccECN) e reduzir drasticamente a taxa de envio ao primeiro sinal de CE.

Ação Não-Responsiva: Como o software do atacante ”desconhece” a semântica do L4S, ele ignora deliberadamente as flags de sinalização CE enviadas pelo roteador. O atacante não possui o mecanismo AccECN para processar esses sinais e, portanto, não realiza o ajuste e a redução na taxa de transferência de dados.

Impacto: O fluxo malicioso, não controlado, satura rapidamente a Fila L (que é intencionalmente rasa). Isso reintroduz o bufferbloat precisamente onde o L4S foi projetado para eliminá-lo, destruindo a garantia de baixa latência para todos os usuários legítimos (LL) que compartilham o mesmo roteador. O ataque torna as aplicações de tempo real, como jogos na nuvem e VR, inutilizáveis.

Dado que o ataque explora uma falha na resposta ao controle de congestionamento, ele não pode ser mitigado por firewalls tradicionais. Portanto, a proposta é implementar mecanismos de Machine Learning (ML) diretamente no gargalo da rede — o roteador central. Este roteador, fazendo uso de IA através de aprendizado de máquina supervisionado, irá realizar o sniffing (captura) em tempo real do tráfego de rede para detectar os padrões desse comportamento malicioso.

2. Objetivo

O objetivo principal desta pesquisa é projetar, implementar e validar um sistema de detecção de intrusão (IDS) leve e eficaz, baseado em aprendizado de máquina supervisionado, capaz de identificar e prever ataques ECN não-responsivos em redes L4S em tempo real.

O foco é colocado na predição de eventos de rede em tempo real. A detecção de anomalias tradicional (baseada em desvios de uma linha de base) pode ser lenta. A abordagem proposta visa usar um modelo preditivo que, ao invés de apenas reagir a um estado de rede já degradado (alta latência), aprenda a reconhecer os precursores e os padrões de comportamento que definem o ataque.

Para atingir este objetivo, serão implementados algoritmos de Árvore de Decisão (Decision Tree), o que oferece algumas características importantes como:

Interpretabilidade (White-Box): As Árvores de Decisão geram um modelo que é um conjunto de regras legíveis (ex: ”SE atraso-fila-L ¿ 1ms E throughput-fluxo-X ¿ 90Mbps ENTÃO Ataque”). Isso permite não apenas detectar o ataque, mas ”identificar métricas e eventos” Eficiência: São algoritmos rápidos para inferência, tornando-os viáveis para execução no roteador (o gargalo da rede) sem introduzir sobrecarga significativa.

O objetivo final é que o modelo de IA no roteador analise continuamente os fluxos de rede, extraia características (features) relevantes em tempo real e classifique o comportamento do tráfego. Ao detectar um comportamento anômalo que corresponda à assinatura de um ataque ECN não-responsivo, o sistema pode (em trabalhos futuros) acionar mecanismos de mitigação, protegendo assim a integridade do serviço L4S para usuários legítimos.

3. Ferramentas e Dataset

Para executar esta pesquisa, será construído um testbed de rede virtualizado, controlado e reprodutível, seguido de um pipeline de análise de dados.

3.1. Ferramentas de Virtualização e Provisionamento

O ambiente experimental será criado usando Infraestrutura como Código (IaC):

VirtualBox: O hipervisor que fornecerá a infraestrutura de máquina virtual (VM) subjacente.

Vagrant: A ferramenta de orquestração para definir a topologia de rede

Ansible: A ferramenta de gerenciamento de configuração (provisionamento). O Ansible será usado para configurar automaticamente o software em cada VM.

3.2. Ferramentas de Análise e Machine Learning

O agente de detecção no roteador e a análise de dados offline serão implementados em Python, usando as seguintes bibliotecas:

Python: A linguagem de programação principal para o script de captura e para o modelo de ML.

scikit-learn: A principal biblioteca de Machine Learning. Ela será usada para implementar o algoritmo DecisionTreeClassifier e para realizar a divisão e validação dos dados.

Pandas: Usado para coletar, estruturar, limpar e processar os dados de log da rede. Os logs de rede brutos (sejam capturas de pacotes ou estatísticas de fila) serão agregados em janelas de tempo e organizados em DataFrames do Pandas para extração de features.

Matplotlib: Usado para a visualização dos dados e dos resultados do modelo.

3.3. Dataset

O dataset para o aprendizado supervisionado não existe publicamente, pois o ataque é específico da arquitetura L4S. Portanto, ele será gerado usando o testbed virtualizado. O processo consistirá em executar simulações controladas :

Coleta de Tráfego Benigno: Execução de tráfego L4S (LL) e Clássico (C) legítimos. Os logs deste período serão rotulados como BENIGNO.

Coleta de Tráfego de Ataque: Execução dos fluxos benignos em conjunto com o fluxo ECN não-responsivo malicioso. Os logs deste período serão rotulados como MALICIOSO.

Estes logs rotulados (logs próprios de testes e experimentação) formarão o dataset de treinamento e teste.

4. Plano de avaliação

O plano de avaliação será focado em validar a eficácia do modelo de Árvore de Decisão em detectar com precisão os fluxos maliciosos, conforme especificado na solicitação.

A avaliação será brevemente dividida em duas fases:

1. Definição de Características: O primeiro passo é definir quais métricas de rede serão usadas como features (preditores) para o modelo. Será usado um conjunto de métricas que se espera que mudem durante um ataque como jitter, rtt, latência, throughput e queue-delay. A hipótese é que um ataque bem-sucedido causará um aumento mensurável no queue-delay da Fila L, o que, por sua vez, aumentará o rtt e o jitter dos fluxos benignos, enquanto o throughput do atacante permanecerá alto. 2. Avaliação do Modelo: Após treinar o modelo de Árvore de Decisão no dataset rotulado, sua performance será avaliada usando um conjunto de dados de teste (dados que o modelo nunca viu antes). Será verificado a acurácia na detecção de fluxos maliciosos, mas também métricas mais robustas como Precisão (Precision) e Recall (Sensibilidade) para garantir que o modelo não esteja apenas acertando os casos benignos (que são fáceis), mas que seja realmente eficaz em ”capturar” o ataque quando ele ocorre.