Storm Chaser: O-RAN에서 Signaling Storm DoS를 차단하는 eBPF/XDP 기반 보안 프레임워크

김현문[1], 이승수[2]

*인천대학교 (학부생[1], 교수[2])

Storm Chaser: eBPF/XDP-based Security Framework to Defeat Signaling Storm DoS in O-RAN

Hyeonmoon Kim[1], Seongsoo Lee[2]

*Incheon National University

(Undergraduate Student[1], Professor[2])

요 약

차세대 RAN 구조로 떠오르고 있는 Open RAN의 표준화를 위해 등장한 O-RAN은 개방형, 지능형 무선 접속망 개발을 촉진하고 있는 동시에, 공격 표면의 확장으로 새로운 보안 위협을 만들어내고 있다. 본 논문은 E2 Interface를 통해 통신하는 E2 Termination과 E2 Node 간에서 발생할 수 있는 Signaling Storm DoS Attack에 대해 소개하고 이를 막기 위해 eBPF/XDP를 활용한 보안 프레임워크인 Storm Chaser를 제안한다. 실험을 통해 Storm Chaser가 악성 RIC Indication Message를 효과적으로 차단하는 것을 보여준다.

I. Introduction

Open RAN의 표준화를 촉진하기 위해 설립된 O-RAN Alliance에서 제안한 Open RAN 구조는 기존에 폐쇄된 RAN 구조에서 개방형 인터페이스, 가상화, 지능형 RAN의 특징을 가지며 차세대 RAN 구조로 각광받고 있다. 그러나 O-RAN에서 제시한 새로운 구성 요소의 등장과 더불어 공격 표면이 확장되고, 데이터 도난과 RAN access control에 대한 보안 문제 등 새로운 위협도 증가하고 있다.

이러한 보안 위협을 해결하기 위해, O-RAN Alliance Working Group 11(WG 11)에서는 안전한 개방형 RAN을 운영할 수 있도록 O-RAN 보안 사양을 담당한다. 그중 E2 Interface를 통한 정보의 기밀성 및 무결성을 보장하기 위해 IPSec의 사용이 요구되는데, OSC Near-Real Time RIC (Near-RT RIC)에서는 E2 통신에 대한 보안 기능을 구현할 수 있는 API를 제공하지 않아 해당 오픈소스를 이용할 경우 보안성이 보장되지 않는 문제점을 가지고 있다.

본 논문에서는 Near-RT RIC와 E2 Node 간의 E2 Inteface를 통한 통신에서 발생할 수 있는 DoS 공격 중 Signally Storm Attack에 대해 소개하고, 이를 막기 위해 eBPF/XDP 기술을 활용한 Storm Chaser 프레임워크를 제안한다.

II. Background and Motivation

2.1 O-RAN

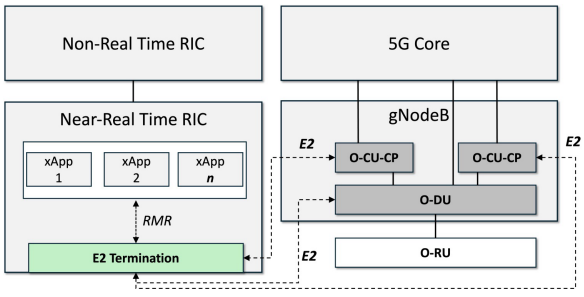

Fig 1. O-RAN architecture

Fig 1은 전반적인 O-RAN의 아키텍처로 RAN Intelligent Controller(RIC)와 Central

Unit(CU), Distributed Unit(DU)로 이루어진 gNodeB, Radio Unit(RU) 그리고 5G Core로 구성된다. 특히, Near-Real Time RIC은 사용자 지정 로직을 수행하는 애플리케이션인 xApp과 이를 지원하는 데 필요한 서비스로 구성된 E2 Node로부터 실시간 Key Performance Indicator (KPI) 데이터가 담긴 RIC Indication Message를 SCTP 패킷으로 전달받는다.

E2 Node는 E2 Interface를 통해 RIC와 연결되는 O-CU, O-DU, O-eNB 등의 구성 요소들을 지칭한다. E2 Termination은 E2 Interface로 연결된 E2 Node와 rmr로 연결된 xApp 사이에서 네트워크 데이터를 담은 E2 Application Protocol(E2AP) 메시지와 네트워크 최적화를 위한 제어 메시지를 SCTP 프로토콜로 주고받으며 E2 Node를 제어한다.

2.2 Signaling Storm DoS Attack

Signaling Strom DoS Attack은 Virginia Tech로부터 발견된 E2 Node와 Near-RT RIC 간에 발견된 취약점 중 하나로[1], E2 Termination이 악의적인 E2 Node로부터 길이가 긴 페이로드를 포함한 RIC Indication Message를 SCTP 패킷으로 대량 받게 되면 과부하로 인해 DoS를 일으키는 공격이다. 이를 통해, 서비스 지연이 발생하고, 해당 xApp과 통신 중인 다른 E2 Node와의 시간에 민감한 시그널링도 영향을 받게 되어 최종적으로는 모든 연결된 E2 Node에서 DoS 상태가 발생할 수 있다.

III. Storm Chaser Design

3.1 Storm Chaser 아키텍처

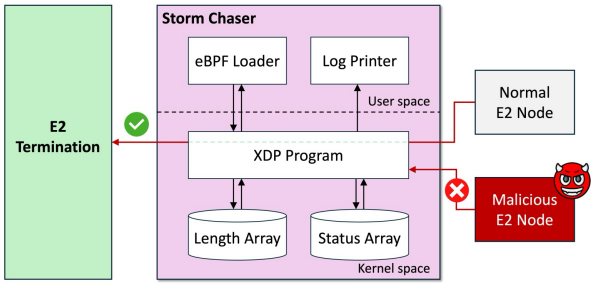

Fig 2. The architecture of Storm Chaser

Storm Chaser는 두 가지 요구사항을 충족하도록 설계되었다. 첫째, 정상 RIC Indication Message가 포함된 SCTP 패킷 사이즈의 기준을 설정할 수 있어야 한다. 이를 위해, 최근 수신한 패킷에 대하여 평균값을 측정한 뒤 사용자 지정 임곗값의 연산을 통해 패킷의 정상 여부를 결정한다. 둘째, 악성 패킷 필터링으로 인한 오버헤드를 최소화해야 한다. 이러한 오버헤드를 최소화하기 위해 eBPF/XDP를 사용하여 커널 공간에서 패킷을 필터링하고 차단 패킷에 대해서만 사용자 공간으로 데이터를 전달한다.

Storm Chaser의 동작 흐름은 Fig 2와 같다. 먼저 eBPF loader 모듈은 E2 Termination이 E2 Node로부터 트래픽을 전달받는 네트워크 인터페이스에 대하여 XDP 프로그램과 eBPF 맵을 커널에 로드한다. 이후 E2 Node로부터 트래픽이 전달되어 XDP Hook을 트리거하면, XDP 프로그램의 콜백 함수가 실행되어 트래픽을 검사한다. 이때 패킷의 사이즈를 필터링 알고리즘으로 검사한 후 정상으로 판단되면 XDP_PASS를 반환하여 패킷을 통과시키고, 비정상으로 판단되면 지정한 기간 동안 해당 송신 IP/Port로부터 송신되는 비정상 패킷의 수를 측정한다. 측정한 비정상 패킷의 수가 특정 값을 넘어가면 XDP_DROP으로 즉시 차단한 뒤, 해당 패킷의 출처 IP 주소와 패킷의 사이즈를 사용자 공간의 Log Printer로 전달하여 출력한다. 이는 정상 패킷의 경우에도 간헐적으로 사이즈가 큰 패킷을 보내는 경우가 있기 때문이다.

3.2 eBPF/XDP 필터링 알고리즘

E2 Termination의 Ingress에 설치된 eBPF/XDP 프로그램은 Fig 3과 같이 작동한다. (1) 수신된 패킷이 SCTP 프로토콜인지 확인한다. 이를 위해 콜백함수의 인자 값으로 받는 ctx를 이용하여 IP 헤더를 구한 뒤, 프로토콜을 확인하여 SCTP 프로토콜이 아니라면 XDP_PASS를 반환한다. (2) 패킷이 블랙리스트에 속해있는지 송신 IP/Port를 key 값으로 blacklist_arr 배열을 조회하여 확인한다. 패킷이 블랙리스트에 속해있다면, 해당 패킷이 수신된 횟수를 조회하여 일정 횟수 이상 수신되었다면 XDP_DROP을 반환한다. (3) length_arr 배열이

Algorithm 1: Pseudo-code for traffic filtering

```
Input: xdp_md *ctx
Output: XDP_PASS or XDP_DROP
#1 Check 4-layer protocol
ip_header ← ctx.data + sizeof(ETH_header);
if ip_header.protocol ≠ sctp then
    return XDP_PASS;
#2 Check blacklist
if ip, port in blacklist_arr.address then
    if blacklist_arr.count > 5 then
        return XDP_DROP
#3 Check array is full
is_full ← status_arr[2];
if !is_full then
    go to #4;
#4 Check packet size
packet_length ← ctx.data_end - ctx.data;
avg ← status_arr[1];
if packet_length > avg + THRESHOLD then
    update blacklist_arr.count;
    return XDP_PASS;
#5 Update array
idx ← status_arr[0];
length_array[idx] ← packet_length;
avg ← new_avg;
status_arr ← idx, avg, is_full;
update status_arr & length_arr;
return XDP_PASS;
```

Fig 3. Malicious SCTP packet filtering

가득 찼는지 확인하기 위해 status_arr를 조회하여 is_full 변수에 저장한다. 만일 length_arr 배열에 값이 가득 차지 않았다면, 5번째 단계로 넘어간다. (4) ctx를 이용하여 패킷의 길이를 구한다. Storm DoS의 공격 패킷은 패킷의 사이즈가 비정상적으로 크기 때문에, 패킷의 사이즈가 length_arr 배열의 평균값보다 일정 수치 이상 크다면 패킷의 송신 IP/Port를 blacklist_arr에 저장하고 패킷의 수신 횟수를 갱신한 뒤, XDP_PASS를 반환한다. (5) status_arr를 조회하여 length_arr가 가장 오래된 값을 새로운 패킷 사이즈로 갱신한 뒤, 새로운 평균 패킷 사이즈와 가장 오래된 값의 인덱스, length_arr가 가득 찼는지를 계산하여 status_arr 배열에 갱신한다. 이후 갱신한 배열들을 eBPF Map에 업데이트한 뒤 XDP_PASS를 반환한다.

IV. Evaluation

4.1 Test Environment

본 실험 환경은 Ubuntu 20.04 가상머신 3대로 쿠버네티스 클러스터를 구축하고 Master node에는 O-RAN SC Near-RT RIC를, gNB1와 gNB2에는 정상 E2 Node와 악성 E2 Node, OAI-5G gNB, OAI-5G Core를 배포하였다[2].

4.2 Use Case

정상 E2 Node가 E2 Termination과 통신하며 주고받는 SCTP 패킷의 크기를 학습한다. 이후 악성 E2 Node가 E2 Termination으로 보내는 실제 RIC Indication Message 내용에 더미 페이로드를 더해 큰 사이즈의 SCTP 패킷을 끊임없이 송신한다.

```
01:02:11.612091 IP (tos 0x2,ECT(0),  proto SCTP (132), length 452)
01:02:12.612119 IP (tos 0x2,ECT(0),  proto SCTP (132), length 452)
01:02:22.629122 IP (tos 0x2,ECT(0),  proto SCTP (132), length 452)
01:02:48.612093 IP (tos 0x2,ECT(0),  proto SCTP (132), length 452)
01:02:48.612468 IP (tos 0x2,ECT(0),  proto SCTP (132), length 452)
```

Fig 4. Packet loss due to Storm DoS attacks

```
2024/11/05 04:19:51 Waiting for events..
2024/11/05 04:20:22 [INFO] Malicious Address Banned!
        Size : 1374
        Source : 163.0.17.172:46758
```

Fig 5. The results of blocking Storm DoS attacks

Fig 4는 Storm Chaser를 적용하기 전 Storm DoS Attack이 성공하여 정상 E2 Node의 통신을 방해하는 것을 보여준다. 반면에 Fig 5는 Storm Chaser가 임계치보다 큰 크기의 패킷이 감지되면 해당 패킷을 차단하는 것을 보여준다.

V. Conclusion and Future Work

본 연구가 제안하는 Storm Chaser를 통해 O-RAN 환경에서도 eBPF/XDP 프로그램으로 보안성을 높일 수 있음을 확인했다. 향후 연구로 공격 표현을 조금 더 확장하여 악성 xApp에서 E2 Termination과 E2 Node의 취약점에 대해 조사하고 이에 대한 보안 프레임워크를 개발하는 연구를 진행할 예정이다.

[참고문헌]

[1] Radhakrishnan, V. K. (2023, July 3). Detection of Denial of Service Attacks on the Open Radio Access Network Intelligent Controller through the E2 Interface. https://vtechworks.lib.vt.edu/items/2d8b6b57-144d-4ae6-abcb-046722145- 821

[2] https://github.com/5GSEC/5G-Spector