2019 졸업프로젝트 경진대회

DDoS 방어를 위한 Docker 기반 DROP-FAST*

Defense Against DDoS using DROP-FAST Implementation

이동욱
고려대학교 정보대학 컴퓨터학과
lee.dongwook94@gmail.com
지도교수: 이희조

요약

본 논문에서는 기존에 연구된 DROP-FAST의 고도화를 위한 연구를 진행하였으며, DROP-FAST의 트래픽 분산을 통해 호스트 서버의 가용성을 보장하고 Docker를 통한 Replica 서버의 생성 및 구동을 보다 효율적으로 관리하고자 한다. 실험을 통하여 DROP-FAST의 트래픽 분산 성능을 검증하였으며, Docker를 활용한 DROP-FAST는 기준에 개발된 DROP-FAST와 동일한 성능을 보였으며, Docker Swarm을 통한 replica 서버들 간 오케스트레이션의 편의성을 제공한다. 더 나아가, 기존의 연구에서 제시한 main 서버와 replica 서버간 동기화, IP 기반 공격 대응과 TTL 설정 문제를 효과적으로 해결할 수 있는 것으로 나타났다.

서 론

많은 전자기기들이 인터넷에 연결됨에 따라 이를 악용하는 사이버 위협들이 증가하고 있다. 특히 증가하는 IoT 기기가 DDoS 공격에 활용되며 DDoS 공격은 매년 최고 볼륨을 경신하고 있다. 따라서 DDoS 공격이 감지되면 클라우드 시스템 기반의 복제 서버를 생성하여 트래픽을 분산시켜 공격을 방어할 필요가 있다. 5G 네트워크로 도입으로 네트워크 가용성이 더욱 강조되는 현 시점에서 서버 및 통신 장비의 가용성을 보장하고, DDoS 공격으로부터 서버와 사용자를 보호하여 안전하고 신뢰도 높은 서비스를 제공할 필요가 있다.
본 논문은 DROP- FAST의 트래픽 분산 기법을 통해 호스트 서버의 가용성을 보장하고 Docker를 통한 replica 서버의 생성 및 구동을 보다 효율적으로 관리하고자 한다.

모 델

2-1. DROP-FAST

DROP-FAST는 DDoS 공격을 방어하기 위해, 호스트 서버의 트래픽 볼륨을 꾸준히 모니터링하며, 특정 임계치를 넘어서게 되면 Docker 서비스를 활용해 복제 서버를 생성, 트래픽을 분산시키는 방법을 사용한다.

2-2. DROP-FAST 구성 요소

DROP-FAST는 크게 4가지 세부 요소들로 구성된다: main 서버, 클라이언트, DROP-FAST 코어, Docker replica 서버.

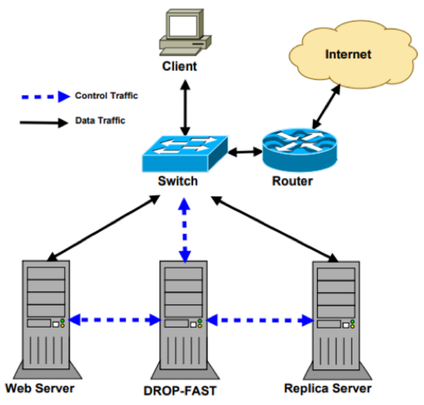


그림 1. DROP-FAST Architecture

(1) 메인 서버 및 클라이언트

DROP-FAST에서 메인 서버란, 가용성을 보장하여 DDoS 공격으로부터 보호하기 위한 타겟 서버이다. 클라이언트는 DROP-FAST에서 보호하는 메인 서버에 접근하고자 시도하는 유저들 혹은 시스템을 의미한다.

(2) DROP-FAST 코어

DROP-FAST 코어는 DROP-FAST의 핵심 기능들을 담고 있는 서버이다. 리소스 모니터링을 통해 공격 상황을 감지하는 IDS 기능과 공격이 감지되었을 때 공격자의 트래픽을 replica 서버로 분산하는 DNS 기능을 수행한다.

(3) Docker 레플리카 서버

DROP-FAST의 IDS기능이 메인 서버의 대한 DDoS 공격을 탐지했을 시, 휴식 중인 replica 서버로 트래픽이 분산되어 가용성을 보장하면서도 공격에 대응할 수 있도록 해준다. Docker replica 서버의 장점은 Swarm 기능을 활용하여 다수의 replica 서버 장비 간의 통신 및 Load balancing을 자동으로 수행한다.

2-3. DROP-FAST 동작 과정

DROP-FAST 코어는 메인 서버에 대한 IDS의 역할을 수행한다. 이러한 모니터링 도중, DDoS 공격이 감지된다면 Docker replica 서버를 구동한 후 메인 서버와 동기화한다. 그 후 DNS서버에 트래픽을 분산할 수 있도록 트래픽 분산을 요청한다. 이후 꾸준한 리소스 모니터링을 진행하며 트래픽 볼륨이 임계치 미만으로 내려가게 된다면 replica 서버의 구동을 해제한다.

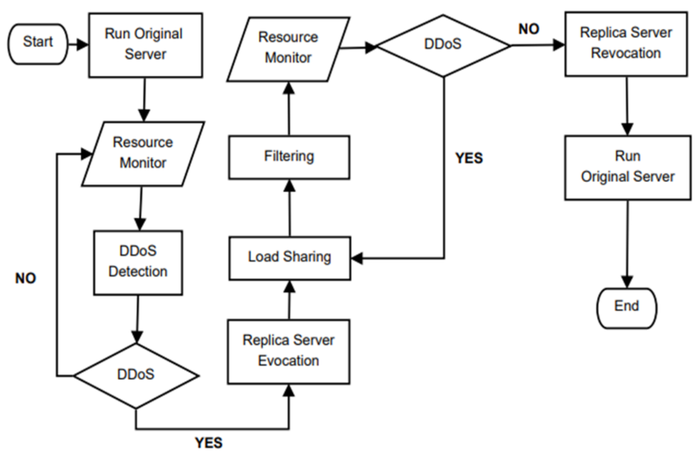


그림 2. Control flow of DROP-FAST

실험 및 토의

3-1 실험 환경

본 논문에서는 학습을 위해 다음과 같은 컴퓨터 환경에서 실험을 진행하였다.

운영체제: Ubuntu 18.04.3,

네트워크 모니터링: ntopng

DoS 공격 툴: LOIC

그림 3. LOIC

3-2. 실험 내용

본 논문에서는, 가용한 장비 부족으로 인해 Drop-Fast의 성능을 검증하기 위한 실험을 진행했으며, 위에서 설명한 In-Line 방식을 채택하여, DoS공격 간 main 서버와 replica 서버의 실시간 트래픽을 측정하였다.
DROP-FAST 코어에 포함되어 있는 방화벽 기능 중 하나인 iptables와 ipset을 활용하여 임의로 Blacklist IP를 선정해 놓고, 해당 Blacklist IP에서 main 서버에 대한 접근을 시도했을 때, replica 서버로 rerouting을 하도록 설정하였다.
서버를 공격하는데 사용하는 툴은 LOIC을 사용했으며, 해당 툴로 공격 대상의 IP를 입력하고, HTTP 공격타입을 선택하여 DoS 공격을 2분간 실행한다.

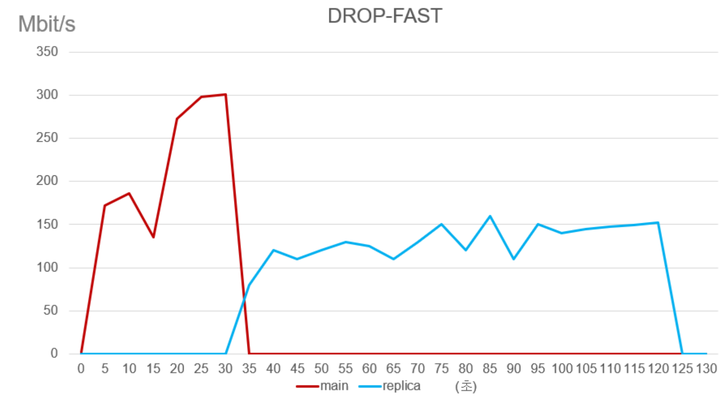
3-3. 실험 결과

(1) DROP-FAST main 서버의 트래픽

버를 보호하는 IDS(snort)이 보호 서버에 대한 공격을 탐지하고 Drop-Fast를 설정하였을 때의 main 서버와 replica 서버의 트래픽을 측정하였으며, 결과는 아래의 그래프와 같다.

(2) DROP-FAST replica 서버의 트래픽

버를 보호하는 IDS(snort)이 보호 서버에 대한 공격을 탐지하고 Drop-Fast를 설정하였을 때의 main 서버와 replica 서버의 트래픽을 측정하였다.



그래프 1. main, replica 서버 트래픽

결 론

위 실험들의 결과를 바탕으로 Docker를 활용한 DROP-FAST 기술은 기존의 클라우드 기반 DROP-FAST 보다 효과적으로 main 서버의 가용성을 확장할 수 있으며, replica 서버를 보다 효율적으로 관리할 수 있는 것을 알 수 있다.

기존의 논문에서 제시한 문제들의 해결방안 역시 아래와 같이 해결할 수 있다.

(1) replica 서버와 main 서버의 동기화 문제

Docker는 Volume이라는 폴더에 수정, 혹은 쓰기가 가능한 부분 및 정보들을 저장함으로 해당 폴더를 공유함으로써, main 서버와 replica 서버의 동기화 문제를 해결할 수 있다.

(2) IP 기반 공격에 대한 대응

해당 공격 방법에 대한 대책으로 본 연구의 실험에서 보였던 것과 동일한 In-Line방법을 제시한다. 메인 서버에 대한 공격을 IDS에서 탐지하고 replica 서버로 reroute해 줌으로써, main 서버의 가용성을 효율적으로 보장할 수 있다.

(3) TTL 문제

해당 문제는 DROP-FAST의 내부 정책에 따른 문제임으로, Docker를 활용한다 하여도, 해결책을 찾을 수 없는 문제이나, 아래의 표를 참고하여 정책을 수립하여 해당 문제를 최소화 할 수 있다.

Type	최대 TTL 시간(초)	최소 TTL 시간(초)	평균 TTL 시간(초)	TTL 시간 중앙값(초)
유명 도메인(500개)	129540	1	6468	300
악성도메인(276개)	5	1~4	4.55	5

표 1. 도메인별 TTL 시간