Click Modular Router를 이용한 보안 게이트웨이용 패킷처리 구조*

김혜진, 이재국°, 김형식
충남대학교 대학원 컴퓨터공학과
angella@csalab.cnu.ac.kr, {empire, hskim}@cs.cnu.ac.kr

A Packet Processing Architecture for Security Gateway Using the Click Modular Router

Hye-Jin Kim, Jae-Kook Lee°, Hyong-Shik Kim
Dept. of Computer Engineering, Chungnam National University

요 약

네트워크 인프라가 확대되면서 보안에 대한 중요성도 더불어 커지고 있으며, 또한 보안 게이트웨이에 대한 관심도 증가하고 있다. 본 논문에서는 소프트웨어 라우터인 Click Modular Router를 이용하여 게이트웨이에서 비정상 트래픽을 제거하는 필터링 기능과 내부 네트워크 정보를 은닉하는 기능을 제공하기 위한 트래픽 처리 구조를 제안한다.

1. 서론

우리나라는 이미 세계 최고 수준의 초고속 인터넷 망을 보유하고 있을 뿐 아니라, 1000만이 넘는 인터넷 사용자, 120만 가구의 초고속 건물 등 인터넷 강국의 대열에 속해있다[1]. 그러나 네트워크 인프라가 확대되면서 네트워크상의 불필요한 트래픽들로 인한 피해와 손실이 크게 늘어나고 있는 현실이다. 이러한 문제를 해결하기 위하여 라우터 단에서 보안기능을 담당하는 연구가 진행되고 있다. 단순히 목적지를 안내하는 것이 아닌 패킷분류, 필터링, 정보 은닉 등의 보안 기능이 추가되고 있다.

본 논문에서는 사용자에게 유연하고(flexible) 용이한(configurable) Click Modular Router를 이용하여 게이트웨이에서의 트래픽 처리 즉, 비정상 트래픽 제거를 위한 필터링 기능과 침입을 원천적으로 감소시킬 수 있도록 내부 정보 은닉을 위한 기능을 설계, 구현함으로써 안전성이 향상된 네트워크 환경을 제안한다.

2절에서는 간단하게 Click Modular Router에 대해서 알아보고 3절에서는 이를 이용한 게이트웨이용 패킷 처리 구조를 제안한다. 4절에서는 제안한 패킷처리 구조를 구현하고 5절에서는 간단한 테스트를 통해 성능분석을 한다. 끝으로 6절에서 결론을 맺는다.

2. Click Modular Router

Click Modular Router(이하 Click)는 MIT LCS's Parallel and Distributed Operating System group과 Mazu Networks, ICSI Center, UCLA가 공동으로 개발한 모듈화 된 소프트웨어 라우터로, 유연하고 구성이 용이하며 확장성이 뛰어나다[2].

Click은 엘리먼트(element)라고 불리는 간단한 패킷처리 모듈들의 조합으로 구성되는데 각각의 엘리먼트는 패킷분류나 큐잉, 스케줄링과 같은 단순한 기능을 한다. 엘리먼트 사이의 연결은 'push connection' 또는 'pull connection'을 통하여 이루어지며, 이를 따라 패킷이 흐르게 된다. 사용자는 목적에 부합하는 엘리먼트들을 선택하고 조합함으로써 다양한 종류의 라우터를 쉽게 구성할 수 있다. 또한, Click에서는 사용자가 직접 새로운 엘리먼트를 생성하여 추가할 수 있으며, 좀 더 구체적인 기능을 할 수 있도록 둘 이상의 엘리먼트를 조합하여 하나의 복합 엘리먼트(compound element)를 만들 수 있다.

Click은 이와 같은 패킷처리를 위한 엘리먼트와 함께 라우터에 필요한 여러 가지 정보를 설정하는 엘리먼트를 제공한다. 사용자는 이 엘리먼트를 사용하여 네트워크 정보나 스케줄 정보 등을 직접 지정한다.

3. 패킷 처리 구조 설계

본 절에서는 Click을 이용하여 비정상적인 트래픽을 제거하는 필터링 기능과 위장 IP(fake IP)와 위장 포트(fake port) 리스트에 기반 한 내부 정보 은닉 기능을 수행하는 패킷처리 구조를 설계한다.

우선 패킷 포워딩이 가능한 게이트웨이 기능을 설계한 후, 그림 1과 같이 내부 네트워크를 보호하기 위한 필터링 기능과 정보은닉 기능을 추가한다. 그림 1의 'SniffGatewayDevice'는 그림 2와 같이 구성된 복합 엘리먼트로, 'FromDevice', 'ToDevice', 'Queue' 등의 엘리먼트를 사용하여 input, output, queue를 통합한 입출력 기능을 하도록 처리한다.

'Classifier' 엘리먼트에서는 패킷을 ARP 응답, ARP 요청, 그리고 IP 패킷으로 분류하여 다음 엘리먼트로 넘긴다.

*본 연구는 "대학 IT 연구센터 육성지원사업"의 지원을 받아 수행한 연구 결과임.

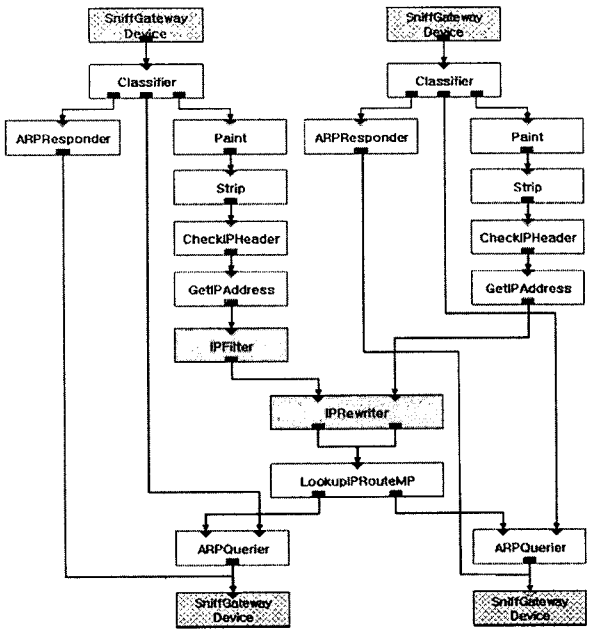

그림 1. Click을 이용한 게이트웨이 설계

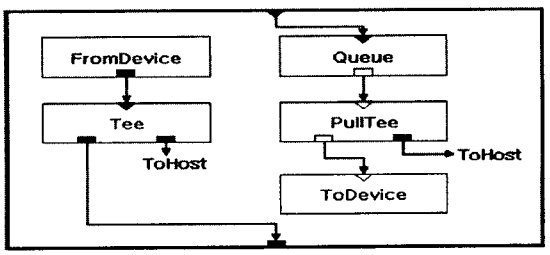

그림 2. 복합 엘리먼트 SniffGatewayDevice

ARP 패킷은 응용 게이트웨이 기능을 구현하기 위한 목적으로 IP 패킷과 별도로 처리한다. ARP 요청 패킷에 대해서는 자신의 이더넷(Ethernet) 주소로 응답을 만들어 내보내고(ARPResponder element), ARP 응답 패킷에 대해서는 패킷 정보를 리스트로 저장해 두었다가 IP 패킷을 내보낼 때 목적지의 이더넷 주소를 얻는데 이용한다(ARPQuerier element).

'IPFilter'는 IP와 포트를 기반으로 한 필터링 기능을, 'IPRewriter'는 정보은닉 기능을 위하여 개선한 엘리먼트를 나타낸다. IP 패킷 필터링은 외부로부터의 트래픽에 대해서만 동작하도록 패킷 분류 다음에 위치한다. 필터링 규칙은 서브넷, 호스트, 포트 등의 조합으로 유연하게 설정한다.

Click은 'IPRewriter'라는 엘리먼트를 이용하여 리맵핑 기능을 지원하지만 이 엘리먼트는 위장 포트에 대한 범위지정은 되지만 위장 IP를 다수로 지정하지 못한다. 본 논문에서는 내부 네트워크를 보호하기 위한 목적으로 패킷의 실제 근원지 IP와 포트를 대신하여 임의의 위장 IP와 특정범위의 위장 포트를 갖도록 'IPRewriter' 엘리먼트를 재설계한다. 이때 'IPRewriter' 엘리먼트는 변경된 정보에 대해 리스트를 유지하고 있다가 변경된 패킷에 대한 응답이 오면 원래의 IP와 포트를 찾아 다시 변경한다.

최종적으로 필터링과 내부 정보 은닉 처리를 한 패킷은 'LookupIPRouteMP' 엘리먼트를 통해 목적지를 찾아간다.

4. 필터링 기능과 은닉 기능의 구현

4.1 필터링 기능

패킷 필터링을 하는 'IPFilter' 엘리먼트는 지정해준 필터링 규칙에 따라 외부에서 오는 패킷에 대해 필터링을 수행한다. Click에서 제공하는 필터링 규칙은 그림 3과 같은 형태로 구성된다.

```
(1) allow src 168.188.129.23 && dst 168.188.1.1 &&
udp && src port 53 && dst port 53,
(2) deny dst 168.188.129.23 && tcp && src port 23
&& dst port > 1023 && ack,
```

그림 3. 필터링 룰의 예

각각은 dns 요청 패킷을 허가(포트 53)하고, telnet 응답 패킷을 거부(포트 23)하라는 필터링 규칙이다. allow 와 deny로 패킷에 대한 허가와 거부를 결정하며, 근원지와 목적지를 서브넷, 호스트로 유연하게 설정한다. && 기호를 사용하여 복합적인 규칙을 기술할 수 있다.

4.2 은닉 기능

```
#define POOLSIZE 5
String s_pool[POOLSIZE] = { "192.168.0.3",
                            "192.168.0.4",
                            "192.168.0.5",
                            "192.168.0.6",
                            "192.168.0.7" };
```

그림 4. 위장 IP 리스트

은닉 기능을 제공하기 위해서는 위장 IP의 리스트가 필요하다. 그림 4는 5개의 IP 주소로 구성된 위장 IP 리스트를 정의하기 위한 예이다.

또한 은닉에 사용될 위장 IP 리스트에서 새로운 매핑 요청이 있을 때 임의로 리스트에서 위장 IP가 선택되도록 하고, 위장 포트는 60000~65535의 범위 중에서 임의로 선택되도록 하기 위하여 그림 5와 같이 'IPRewriter' 엘리먼트를 수정하였다.

```
IPAddress pool[POOLSIZE];
for( int i = 0; i < POOLSIZE; i++)
    cp_ip_address( s_pool[i], &pool[i]);

int r_index = random() % POOLSIZE;

if (_variation_top) {
uint32_t val = (_sequential ? _next_variation :
uint32_t step = ( sequential ? 1 : random() | 1)
uint32_t base_p = ntohs(_sport);
uint32_t base_a = ntohl( saddr.addr());

    if (_is_napt)
    {
        lookup.set_dport(htons(base_p + val));
        lookup.set_daddr(pool[r_index].addr());
    }
    else
        lookup.set_daddr(htonl(base_a + val));
    if (!rev_map.findp(lookup)) {
        if (_is_napt)
        {
            out.set_sport(lookup.dport());
            out.set_saddr(lookup.daddr());
        }
        else
            out.set_saddr(lookup.daddr());
        _next_variation = val + 1;
        goto found;
```

그림 5. 위장 IP와 위장 포트의 임의 선택

5. 시험 및 성능분석

패킷처리 구조의 성능 평가를 위하여 간단한 테스트를 실행하였다. 그림 6과 같이 시스템을 구성하고, 게이트웨이에 트래픽처리 모듈을 적재한 후 트래픽을 발생시켜 성능을 측정한다.

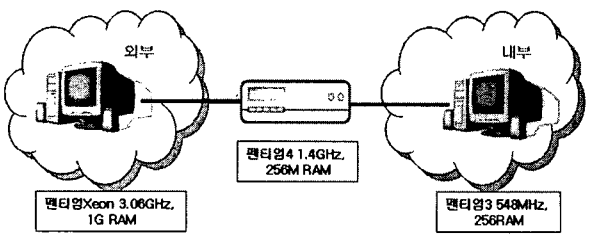

그림 6. 시험 환경

5.1 필터링 기능의 확장성 시험

먼저, 악의 있는 사용자 및 공격이 많아짐에 따라 차단해야 할 호스트와 서비스 역시 증가하므로, 필터링 규칙의 수가 미치는 성능을 테스트하였다.

그림 7은 보내는 속도를 달리하여 패킷을 전송했을 때 필터링 규칙의 수와 관련한 패킷 처리율을 보인다. 필터링 규칙의 수가 증가함에 따라 패킷 처리율에 약간의 차이를 두면서 감소한다. 실제 적용 과정에서 필터링 규칙의 수가 늘어남에 따라 다소 오버헤드가 있을 것으로 예상되지만 심각한 성능 저하는 회피할 수 있을 것으로 기대된다.

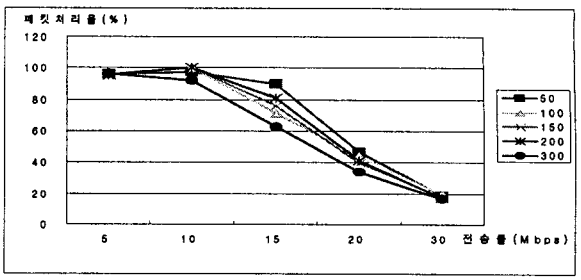

그림 7. 필터링 규칙 수의 증가에 따른 패킷 처리율

5.2 정보은닉 용량 시험

그림 8은 커넥션을 일정시간 동안 일정 수만큼 생성한 후, 보내는 속도를 달리하여 패킷을 전송했을 때의 패킷 처리율을 나타낸 그래프이다. 은닉 가능한 커넥션 수는 위장 IP수와 위장 포트 수의 곱과 같으므로, 내외부의 커넥션 수치가 높더라도 극복 가능하다. 그림 8에서와 같이 동시에 1000개의 커넥션이 생성되더라도 패킷 처리율에 큰 변동이 없음을 알 수 있다.

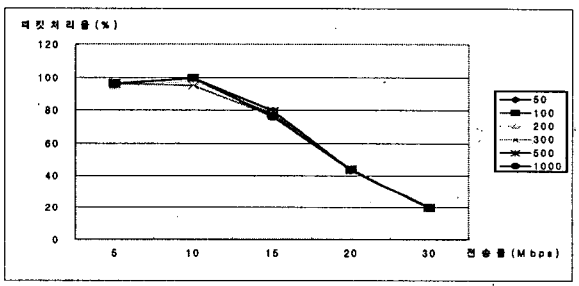

그림 8. 커넥션 수의 증가에 따른 패킷 처리율

6. 결론

Click은 개방적이고 확장성 높으며 구성이 용이한 라우터 프레임워크이다. 본 논문에서는 이를 이용하여 게이트웨이에서 내부 시스템을 보호하기 위하여 패킷 필터링 기능과 정보은닉 기능을 갖는 패킷처리 구조를 설계하고 구현하였다. 그리고 간단한 테스트를 통하여 패킷 필터링 기능과 정보은닉 기능의 동작성을 확인하였다. 필터링 규칙의 수에 대해서는 증가에 따른 약간의 오버헤드가 있으나, 커넥션의 수에 따른 패킷 처리율은 거의 차이가 없었으므로 실제 적용할 때에도 좋은 성능을 보일 것으로 예상된다. 그러나 패킷을 보내는 속도가 증가할수록 패킷 처리율은 감소하는 결과를 보였기 때문에 향후 패킷 처리율 향상을 위한 패킷처리 구조의 개선과 성능 최적화 방안이 필요하다.

참고문헌

[1] 이순구, "보안 어플라이언스를 위한 트래픽 처리 구조 설계," 충남대학교 컴퓨터과학과 석사논문, 2005년 2월.

[2] Click Modular Router (http://pdos.csail.mit.edu/click/)