한국컴퓨터정보학회 하계학술대회 논문집 제23권 제2호 (2015. 7)

미러링 환경에서의 페이지 리다이렉션 기술

정승일°, 박재경*
*°한국과학기술원 사이버보안연구센터
e-mail : sijung@kaist.ac.kr*, wildcur@kaist.ac.kr*

Page Redirection Techniques of Mirroring Environment

Seungil Jung°, Jae-Kyung Park*
*°Dept. of Cyber Security Research Center, KAIST

Abstract

본 논문에서는 미러링 환경에서 동작하는 악성링크 차단 및 URL 필터링 시스템에서 URL 접속을 차단할 수 있는 페이지 리다이렉션 기술을 소개한다. 미러링(Mirroring: Out of Path) 환경에서 URL 패킷을 차단하기 위해서 많이 사용하고 있는 방법으로는 RST (Reset) 패킷을 전송하여 세션을 종료하는 방법이다. 이 방법은 요청 서버에 RST(Reset) 패킷을 보내 강제로 종료하는 방식이기 때문에 사용자에게 접근 차단과 관련된 상태 등의 정보를 알려줄 수 없다. 현재 인라인(In-line) 방식에서 사용되고 있는 페이지 리다이렉션 기술을 미러링 환경에서 구현하여 사용자에게 차단 정보를 보여줄 수 있으며 다양한 장비 개발 환경에서 유용하게 사용할 수 있는 기술이라고 판단한다.

키워드: 미러링(Mirroring), 페이지 리다이렉션(Page Redirection), 접속차단, RST, 세션차단

I. Introduction

악성링크나 피싱사이트를 분석하여 수집이 되면 해당 사이트의 접근을 차단하는 것이 필요하다. 많은 보안 장비가 인라인(In-line)모드와 미러링(Mirroring)모드로 구성되어지는데 보안 장비의 경우 많은 자원을 소비하기 때문에 미러링 방식을 선호한다. 그러나 미러링 환경의 경우 분석을 위한 구성으로는 용이하나 해당 패킷을 컨트롤 하기에는 제약 사항이 많이 따른다. 예를 들어 인라인모드의 경우 특정 패킷을 차단하고자 할 경우 패킷을 지연시키면 되지만 미러링 환경에서는 이미 패킷이 통과된 후이기 때문에 조치할 방법이 없다.

따라서 자원소비가 많이 들고 전체 네트워크에 영향을 주지만 보안 장비의 최종 목적인 시스템 보호를 위해서 인라인모드로 구성하는 경우가 많다.

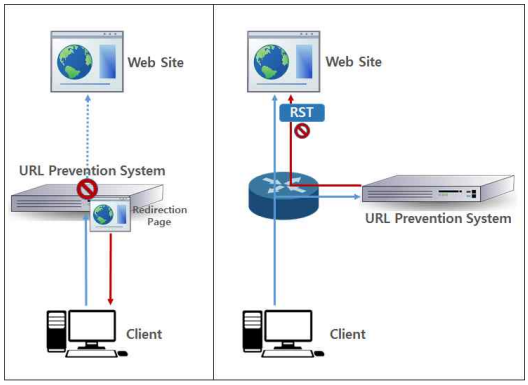
본 논문에서는 미러링 환경에서 패킷을 효과적으로 차단하고 차단 정보를 제공할 수 있는 페이지 리다이렉션 기술을 소개하고자 한다. 2장에서는 현재 사용되고 있는 URL 차단 기술을 알아보고, 3장에서는 미러링 환경의 페이지 리다이렉션 기술을 소개하고, 4장에서는 테스트 결과에 대해서 살펴본다.

II. Preliminaries

1. Related works

1.1 인라인(In-line) 모드 URL 차단

인라인 모드에서 URL을 차단하는 방법은 구조상으로 간단하다. 시스템 안에서 패킷을 컨트롤 할 수 있기 때문에 패킷을 Drop 하여 URL을 차단 할 수 있다. 패킷을 Drop 하면서 추가적으로 페이지 리다이렉션을 하여 원하는 페이지를 보여줄 수 있다. 아래 그림은 인라인 모드에서 페이지 리다이렉션하는 과정과 미러링 모드에서 RST(Rest) 패킷을 통해 URL을 차단하는 과정이다.



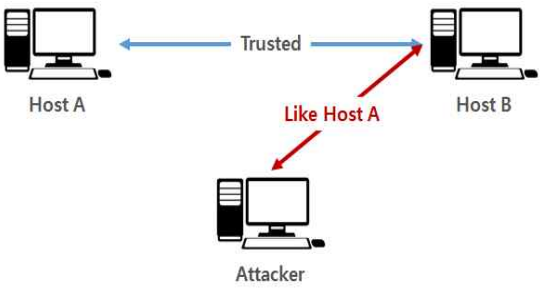
In-line Mode VS Mirroring Mode

1.2 미러링(Mirroring)모드 URL 차단

미러링 모드에서 URL을 차단하기 위해 주로 사용하는 방법은 TCP 프로토콜의 RST(Reset) 패킷을 이용한 세션 강제 종료 방법이 있다[1]. TCP 프로토콜 방식을 이용하여 RST 패킷을 보내 클라이언트에서 서버에 세션 종료 요청을 한다는 의미이다. 이 방법은 차단 시스템에서 '차단'을 한다는 의미보다는 서버에 세션 종료를 '요청'한다는 의미로 봐야한다.

1.3 IP Spoofing

Spoofing의 사전적 의미는 '속이다'이다. 따라서 IP Spoofing이란 'IP를 속이다'라는 의미로 자신의 IP를 속여서 다른 IP로 접속하는 것이다[2]. 신뢰관계를 형성하고 있는 통신 연결 중 제 3자가 신뢰관계의 호스트로 위장하여 접속할 수 있는 기술이다[3].



IP Spoofing

위 그림과 같이 공격자는 자신이 'Host A'인 것처럼 IP를 위장하여 'Host B'에 접속한다. 'Host B'는 이미 신뢰관계를 맺은 'Host A'로 알고 정보를 전달한다.

1.4 Packet Hooking

Packet Hooking은 패킷을 바꾸거나 가로채는 기술이나 행위를 말한다. Hooking의 사용 의미나 범위는 용도에 따라 다양하게 정의되고 있지만 여기서 말하는 Packet Hooking은 패킷을 가로채는 기술을 말한다. Packet Hooking을 통해 다양한 Hooking 방법 중 네트워크를 가로채서 변경하기 위한 'Netfilter Hooking'[4] 방법이 있다. 리눅스 커널에 탑재되어 있는 Netfilter 모듈을 이용하여 패킷을 Hooking 하여 트래픽을 변경 할 수 있다.

```c
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/skbuff.h>

#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/in.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>

/* Port we want to drop packets on */
static const uint16_t port = 25;

/* This is the hook function itself */
static unsigned int hook_func(unsigned int hooknum,
                              struct sk_buff **pskb,
                              const struct net_device *in,
                              const struct net_device *out,
                              int (*okfn)(struct sk_buff *))
{
        struct iphdr *iph = ip_hdr(*pskb);
        struct tcphdr *tcph, tcpbuf;

        if (iph->protocol != IPPROTO_TCP)
                return NF_ACCEPT;

        tcph = skb_header_pointer(*pskb, ip_hdrlen(*pskb), sizeof(*tcph), &tcpbuf);
        if (tcph == NULL)
                return NF_ACCEPT;

        return (tcph->dest == port) ? NF_DROP : NF_ACCEPT;
}

/* Used to register our hook function */
static struct nf_hook_ops nfho = {
        .hook     = hook_func,
        .hooknum  = NF_IP_PRE_ROUTING,
        .pf       = NFPROTO_IPV4,
        .priority = NF_IP_PRI_FIRST,
};

static __init int my_init(void)
{
        return nf_register_hook(&nfho);
}

static __exit void my_exit(void)
{
    nf_unregister_hook(&nfho);
}

module_init(my_init);
module_exit(my_exit);
```

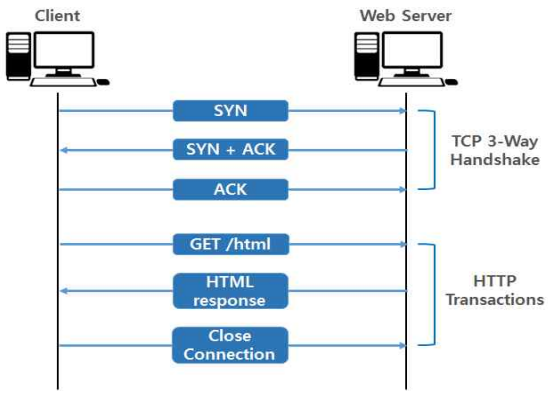
Netfilter Hooking Module Example(Wikipedia)

III. The Proposed Scheme

1. 페이지 리다이렉션(Page Redirection)

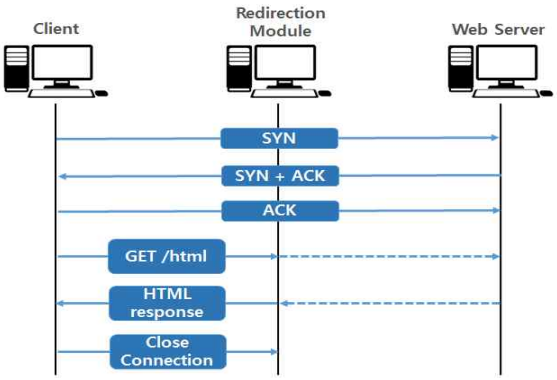
1.1 Execution Architecture

페이지 리다이렉션의 구현을 위해서 HTTP 프로토콜을 이용한다. 아래 그림은 정상적인 HTTP 통신이다.



Normal HTTP Transactions

TCP 3-Way Handshake를 통해 세션을 생성한 후에 GET 메소드를 전송하여 웹 페이지에 대한 응답을 받아 요청을 처리한다[5].

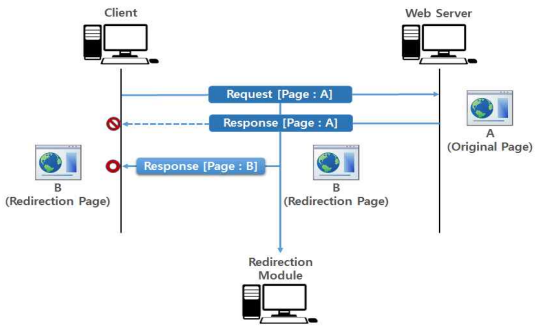


Page Redirection HTTP Transactions

위 그림은 페이지 리다이렉션 되는 경우의 실행 구조이다. Redirection 모듈을 이용하여 클라이언트의 GET 메소드를 Hooking 하여 리다이렉션 페이지를 응답하여 원하는 페이지를 처리하도록 한다. 클라이언트는 정상적으로 통신을 처리하였다고 판단한다[6].

1.2 Principle of Operation

미러링 모드에서 페이지 리다이렉션을 구현하기 위한 핵심 패킷의 처리 속도다. 인라인 모드의 경우 페이지를 요청하는 패킷을 Hooking 하여 원하는 작업을 처리하면 된다. 그러나 미러링 모드에서는 페이지를 요청하는 패킷을 미러링 받아 처리하기 때문에 오리지널 메시지는 해당 서버에 요청 패킷을 보내 서버에서 응답을 받게 된다. 그렇게 되면 서버에서 받은 응답을 클라이언트에서 처리하게 되어 해당 페이지가 로딩되기 때문에 원하는 페이지로 리다이렉션을 할 수 없다. 따라서 미러링 모드에서 페이지 리다이렉션이 동작하기 위해서는 오리지널 메시지가 받은 응답 메시지보다 먼저 클라이언트에 리다이렉션 페이지를 위한 패킷을 보내주어야 한다. 이와 같이 빠른 패킷을 처리하기 위해 커널 레벨에서 Hooking 모듈을 구현하였다.



Page Redirection Execution

위 그림은 리다이렉션 모듈에 의해 페이지가 리다이렉션되어 실행되는 동작을 보여준다. 클라이언트는 [웹페이지 A]로 접속을 시도하였다. 미러링 환경에서 이를 모니터링하고 있던 리다이렉션 모듈이 [웹페이지 A]의 응답 보다 모듈에서 생성한 [웹페이지 B]를 먼저 응답하여 클라이언트에 [웹페이지 B]가 출력되도록 하는 실행 과정이다.

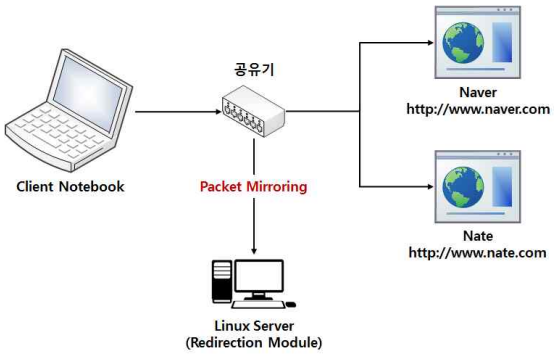
2. Testing

2.1 Test Environment

Item	Version
Linux Kernel	2.6.32
Module	Netfilter Hooking Module
Hooking Point	Netfilter INET_PRE_ROUTING

System Environment

위 표는 리다이렉션을 위한 모듈 구현 환경이다. 리눅스 커널에 Netfilter 후킹 모듈을 이용하여 구현하였다.

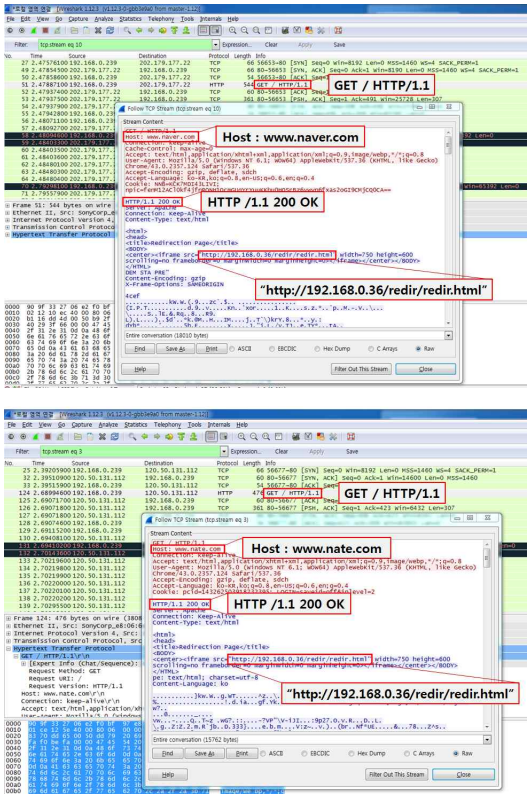


Test Environment Configuration

위 그림은 테스트 환경 구성도 이다. 네이버(www.naver.com)와 네이트(www.nate.com) 에 접속을 시도하면 리다이렉션 모듈이 패킷을 미러링하고 있다가 리다이렉션 페이지를 응답하는 구조이다.

2.2 패킷 분석

아래 그림은 와이어샤크(Wireshark)를 이용한 패킷 덤프 화면이다. 패킷을 확인해 보면 GET 요청에 대한 Response 패킷(HTTP/1.1 200 OK)을 확인 할 수 있다. 구현한 리다이렉션 모듈에서 Response 패킷을 정상적인 Response 패킷보다 먼저 보냈기 때문에 위와 같은 결과가 가능하다. 미러링 환경에서의 동작이기 때문에 정상적인 Response 패킷이 먼저 도착한다면 클라이언트는 먼저 온 패킷을 처리한다. 미러링 환경에서 페이지를 리다이렉션하기 위해 처리 속도가 중요한 이유가 이것이다.



Packet Analysis

((Top) www.naver.com, (Bottom) www.nate.com)

2.3 실행 화면

아래 그림은 테스트 결과 화면이다. 네이버와 네이트 포털 사이트에 접속을 시도하였는데 리다이렉션 페이지가 출력되었다.

Page Redirection Execution

((Top) www.naver.com, (Bottom) www.nate.com)

위 결과와 같이 미러링 환경에서 페이지가 리다이렉션 되었음을 볼 수 있다.

IV. Conclusion

미러링 환경에서 트래픽을 컨트롤 하여 페이지를 리다이렉션하는 모듈을 구현하였다. 이를 통해 미러링 환경에서도 사용자에게 URL이 차단되었을 경우 차단에 대한 정보 제공이 가능하였다.

다양한 보안 장비들이 인라인 모드와 미러링 모드로 나눠져 있는데 각 구성 환경에 따라 장단점이 있으나 미러링 모드의 장점 중 하나는 운영 중인 내부 네트워크에 영향을 주지 않고 보안 시스템 운영이 가능 하다는 것이다. 그러나 미러링 모드는 트래픽을 스니핑하는 수준이기 때문에 트래픽을 직접적으로 컨트롤 하는 데에 어려움이 많다. 보안 장비는 트래픽을 제어하여 시스템을 안전하게 보호하는데 목적이 있기 때문에 내부 네크워크에 영향을 주더라도 인라인 모드를 구성하는 경우가 많다. 미러링 환경에서 이러한 트래픽 컨트롤 문제를 해결할 수 있는 다양한 기술들이 개발이 된다면 미러링 모드의 장점은 더욱 부각될 것이며 다양한 구성 환경을 통해 더욱 효율적이고 강력한 보안 시스템 활용이 가능할 것이다.

References

[1] N. Weaver, R. Sommer, and V. Paxson, "Detecting forged TCP reset packets," in Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA. The Internet Society, 2009.

[2] J. S Jeon, Y. S Jeong and W. Y Soh, "Design of Packet Generator for TCP/UDP Protocols Using Packet Sniffing and IP Spoofing". Journal of Korean Institute of Information Scientists and Engineers, Vol. 32, No. 2, pp. 649-651, November 2005.

[3] C. Hofer and R. Wampfler, “IP Spoofing”, rvs.unibe.ch/teaching/cn%20applets/IP_Spoofing/IP%20Spoofing.pdf

[4] Wikipedia, “Netfilter Hooking,” https://en.wikipedia.org/wiki/Hooking#Netfilter_hook

[5] Chappell, Laura. "Inside the TCP Handshake." NetWare Connection (2000)

[6] S. M Jung, J. G Song, S. S Kim and G. C Park, "A Study on Efficient Monitoring System using Packet Sniffing". Proceedings of KIIT Summer Conference, pp. 587-590, June 2009.