Design Document

for

DNS Spoofing

<Assignment 3>

Version 1.0

Prepared by Wesley Yue

November 5, 2012

Table of Contents

1. Introduction

1.1 Abstract

This assignment puts the basic design and framework of the DNS protocol, and ARP poisoning together to implement a complete and working DNS traffic manipulation. For this assignment, Ruby was used to put these pieces together. The PacketFu package is the main component to get the DNS traffic manipulation to work. However, there are a couple components that the DNS spoof could fail.

Firstly, if the real DNS response was received first, then the target's browser will be directed to the real IP address instead of the spoofed IP address. Therefore, the target's machine needs to be able to receive the spoofed DNS response first, or else the DNS spoof attempt would fail. The other component that makes the DNS spoof attempt fail is when the DNS searches the IP address for that DNS query from the DNS cache instead of generating a DNS query packet.

Other than that, all the testing results are in the WY_Testing_Assignment3 document.

2. Design

2.1 Pseudo-Code

arpSpoof.rb pseudo-code

`echo 1 > /proc/sys/net/ipv4/ip_forward`

def initialize (config, targetsAddr, targetsMac, iface) method

Define the variables that are needed to DNS spoof

- config
- targetsAddr
- targetsMac
- gateway
- interface

end

```
def start method
        Construct the target's packet
                arp_packet_target = variables
                …
        Construct the router's packet
                arp_packet_router = variables
                …
        while(@arping)
                sleep 5
                arp_packet_target.to_w(@iface)
                arp_packet_router.to_w(@iface)
        end
end

def stop method
        @arping = false
end
```

sniffDNS.rb pseudo-code

```
Set arguments from TrollOP
        opt :iface, "NIC Device", :default => "wlan0"
        opt :targetsAddr, "Target's IP Address", :default => "192.168.0.33"
        opt :spoofAddr, "Spoof IP Address", :default => "24.86.113.108"

def initialize method
        Define the variables that are needed to DNS spoof
```

- config
- targetsAddr
- targetsMac
- gateway

```
        Set instance variables
end
```

```
def start method
        Start thread from ArpSpoof.new(@config, @targetsAddr, @targetsMac, @iface)
        Capture packet with filter
        For each captured packet do
                Parse it into a structure
                If the captured packet is DNS query
                        Define headerCount = 12
                        While headerCount < 100
                                If packet receives hex value of 00
                                        break
                                Else
                                        counter = payload[headerCount].to_i(16)
                                        if headerCount is not 12
                                                Add "." in @domainName
                                        end
                                        for i in 1 ..counter
                                                headerCount+=1
                                                Store hex char value into @domainName
                                        end
                                        headerCount+=1
                                end
                        end
                end
                Define type variable to get payload number of Query Type
                If Query Type is A
                        Get transaction ID
                        sendDNSResponse method
                end
        end
end

def sendDnsResponse method
        Get the @domainName
        Get the transaction ID
        Construct the DNS response packet
end
```

```
begin
        sniff = SniffDNS.new
        sniff.start
rescue
        Thread.kill($thread)
        `echo 0 > /proc/sys/net/ipv4/ip_forward`
        exit 0
end
```

2.2 Command-Line Parameters

sniffDNS Command Line Parameters

--iface <interface>	- The interface card to sniff packets from. (Default: wlan0)
--targetsAddr <ipaddr>	- The target's IP address. (Default: 192.168.0.33)
--spoofAddr <ipaddr>	- The spoof IP address. (Default: 24.86.113.108)

3. Summary

The arpSpoof.rb and sniffDNS.rb has put the basic design and framework of the DNS protocol, and ARP poisoning together to implement a complete and working DNS traffic manipulation. However, there are couple components that the DNS spoof could fail. One component is when the real DNS response is received first than the spoofed DNS response. The other component is when the DNS searches the IP address of the DNS query from the DNS cache.

The accompanying DVD will include the complete source code, and pcaps files for confirmation of design and evidence of test results. Moreover, there are extra pcaps files to ensure that the DNS spoof have worked, but it is not in the testing document. Also, the DVD will include the design and testing document.

4. Appendix

/root directory:	
README.txt	(instructions for the DNS spoof program)
/documents/ directory:	
WY_DesignDocument_Assignment3.docx	(docx version of the design document)
WY_DesignDocument_Assignment3.pdf	(pdf version of the design document)
WY_Testing_Assignment3.docx	(docx version of the testing document for the DNS spoof)
WY_Testing_Assignment3.pdf	(pdf version of the testing document for the DNS spoof)
/pcaps/ directory:	
spoofedGoogle.pcap	(DNS spoofed Google packet capture file)
spoofedBCIT.pcap	(DNS spoofed BCIT packet capture file)
spoofedYahoo.pcap	(DNS spoofed Yahoo packet capture file)
spoofedMilliways.pcap	(DNS spoofed Milliways packet capture file)
/src/ directory:	
arpSpoof.rb	(Ruby ARP poisoning's source file for this assignment)
sniffDNS.rb	(Ruby DNS spoof's source file for this assignment)