DNS spoofing

→ redirecting user to a target website
→ Kali comes with its own webserver to spoof dns.
→ do:

```
# service apache2 start
```

→ go to kali's ip:
do: ifconfig & go to eth0 ip.
→ open that ip & you'll see a default apache page
→ you can change the contents of the page by going to:

/var/www/html directory on kali & changing htmls.

Subject :

PAGE NO.:
DATE:

★ Run bettercap using our default caplet:
→ run dns spoof:
★ dns.spoof.address ← address where user is redirected to. (by default set to ip of interface)
★ dns.spoof.all = true (any dns request ack.)
★ dns.spoof.domains = targets that need to be spoofed.
★ do: # set dns.spoof.domains ~~tiitb.learn.iitb~~ < site url >
eg. learn.iiitb.net

→ Start: # dns.spoof on

★ This won't work against websites using hsts.

★ We can also do html changes while a website loads.
→ HTML is responsible for buttons, texts etc.
→ We can do code injection.

★ Java HTML code injection

→ Inject javascript code in loaded pages
→ Code is executed by target browser
→ replace links, images
→ insert html elements
→ first you need a js code.

★ Say: alert("javascript test");
save this as alert.js
★ Go to hstshijack.cap (usr/local/share/bettercap/caplets)
↳ modify payloads : add, ★: /root/alert.js

Subject : _____

PAGE NO.: _____
DATE: _____

→ go to terminal & run bettercap caplet
→ run :
→ hstshijack|hstshijack
→ now any time the target loads a webpage, the code will be injected.

★ Now we'll use everything via a graphical interface.
→ A web-interface is user friendly but requires more resources.

do: → # bettercap -iface eth0
→ # http-ui
• This runs the ui on a local url.
• username : user
• password : pass

★ In the events page, all events log are stored. You can mute any event.
★ In the LAN page, you'll see all devices connected to the network
★ Click on play button to start net.probe & recon.
★ You can easily click on any target ip to add to spoof targets.
★ You can run caplets via caplet page.
★ Run sniffers etc from advanced page.