BACKGROUND

Chris — Custom Attack Tools and Project Management

Matt — Interface Design and Framework Development
Basic ARP Poison
Heavy Network Traffic
Periods of MITM Loss
Python Tool With Scapy
ARP MITM
Anatomy of the Attack
ANATOMY OF THE ATTACK

HTTPS Downgrade Attack
Use as a Web Proxy
Customizations for Subterfuge
VECTOR TRANSPARENCY

TRADITIONAL SSL ATTACKS
CERTIFICATE ERRORS
EXPLOITING THE HUMAN

def index(request):
    if request.is_ajax():
        # Check ARP Table status
        command = 'ps -A l | sed -e/d;\s/pts.*// |'
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
            status = "on"
        else:
            status = "off"

        # Relay Template Variables
        return render_to_response("includes/credtable.inc", {
            "credential" : creds,
            "status" : status,
        })
    else:
        # Check Arpscan status
        command = 'ps -A l | sed -e /arpmitm/id;sed -e/d/s/ //;a/ pts.*// |
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
            status = "on"
        else:
            status = "off"

        # Read in subterfuge.conf
        with open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqrstuvwxyz") + "subterfuge.conf", 'r') as file:
            conf = file.readlines()

        # Relay Template Variables
        return render_to_response("home.ext", {
            "status" : status,
            "conf" : str(conf[20]).rstrip("\n"),
        })

def plugins(request):
    if request.is_ajax():
        print "AJAX REQUEST!"
    else:
        # Read in Config File
        f = open(str(os.path.dirname(__file__)).rstrip("abcdefghijklmnopqrstuvwxyz") + "subterfuge.conf", 'r')
A New MITM Tool –

- Intuitive Interface
- Easy to use
- Silent and Stealthy
- Open Source
Credential Harvesting
HTTP Code Injection
Denial of Service
Network View
def index(request):
    if request.is_ajax():
        # Get Crds from DB
        cred = query_db()
        print(cred)
        return render_to_response("home.html", { "crds": cred }, status=200)
    else:
        # Check Arpspoof status
        command = "ps -A l | sed -e '/arpmitm/id;/sed -e/d;s/ //;s/ pts.*///""
        a = os.popen(command)
        reply = a.read()
        if(len(reply)>1):
            status = "on"
        else:
            status = "off"
        # Relay Template Variables
        return render_to_response("home.html", { "status": status, "conf": str(conf[20]).rstrip("\n"),
            })

def plugins(request):
    if request.is_ajax():
        print "AJAX REQUEST!"
    else:
        # Read in Config File
        f = open(str(os.path.dirname(__file__)).rstrip("\n").rstrip("\n")
            "subterfuge.conf", 'r')