Reverse Engineering Nedir? Kodun Gizli Dünyasına Yolculuk

“Zararlı yazılım mı? Korkmayın! Size kötü niyetli bir hacker olmayı değil ama en azından ‘Burada ne dönüyor?’ diye anlamayı vadediyorum. Unutma: İnsan bilmediğinin düşmanıdır.”

Reverse engineering (tersine mühendislik), çalıştırılabilir bir dosyanın mesela bir .exe’nin arkasındaki **mantığı ve işleyişi çözme** sürecidir. Kaynak kodu yok, elimizde sadece derlenmiş bir dosya var. İşte biz de o dosyayı alıp, **geriye doğru bir mühendislik** yolculuğuna çıkıyoruz. Bu; yazılıma “Sen burada ne yapıyorsun?” diye sormak gibi.

Yazılım geliştiriyorsan, sistemlere ilgin varsa, güvenlikçi olmasan bile kesinlikle bu gizli dünyaya bir göz atmalısın.

İki Ana Analiz Yöntemi

1) Statik Analiz

Programı çalıştırmadan, onun yapısını analiz etme işidir.

Kullanılan Araçlar:

- **IDA Pro** – Hem statik hem dinamik analizde kullanılır. Ben ona “IDA kadınım” diyorum. Tüm sırrı o çözüyor.
- **Ghidra** – NSA destekli açık kaynaklı güçlü bir analiz aracı.
- **Binary Ninja** – Sezgisel arayüz, pratik kullanım.
- **PE-bear, Detect It Easy (DIE)** – Binary detayları öğrenmek için birebir.
- **HxD, 010 Editor** – Hex editörleri, dosyanın en ham haline ulaşmak için.

Avantajlar:

- Kod çalıştırılmadan incelendiği için **güvenlidir**.
- İçindeki fonksiyonlar, şifreleme algoritmaları vs. çözülebilir.
- Sisteme zarar vermez.

Dezavantajlar:

- **Obfuscation** (kod karmaşıklaştırma) varsa analiz zorlaşır.
- Programın davranışları tam olarak gözlemlenemez.
- Anti-disassembly teknikleri ters mühendisliği engelleyebilir.

2) Dinamik Analiz

Programı **çalıştırarak** onun sistem üzerindeki etkilerini gözlemleriz. Nereye bağlanıyor, ne yazıyor, hangi işlemleri yapıyor?

Kullanılan Araçlar:

- **x64dbg / x32dbg**, **OllyDbg** – Canlı debugging.
- **Process Monitor**, **Process Explorer** – Arka planda ne yapıyor, her şeyi gözlemler.
- **Wireshark** – Ağ trafiğini dinler, dışarı veri gönderiyor mu anlarız.
- **RegShot** – Kayıt defterindeki değişiklikleri izler.

Avantajlar:

- **Gerçek zamanlı** zararlı davranışlar ortaya çıkar.
- Obfuscation olsa bile, **hareketler izlenebilir**.
- Dosya oluşturma, ağ bağlantıları gibi net davranışlar görünür.

Dezavantajlar:

- Yazılım çalıştırıldığı için **risklidir**.
- Sanal makine fark edilirse analiz başarısız olabilir.
- Bazı davranışlar tetiklenmedikçe gözükmeyebilir.

Kodun Gizli Dünyası: Assembly ile Tanışma

Şimdi biraz daha derine inelim. Bilgisayar mühendisliği öğrencileri, özellikle **Mikroişlemciler** dersinde şu tür kodları görünce dumur olur:

```
MOV AX, 5
MOV BX, 3
ADD AX, BX
```

“İki sayıyı toplamak bu kadar mı zahmetli?”
Evet. Çünkü senin yazdığın Python, C, C++ kodları... hepsi arka planda bu hale dönüştürülüyor.
Gelin, bir C kodu üzerinden süreci anlatalım:

Yüksek Seviyeden Makineye Giden Yol:

```
int a = 5;
int b = 3;
int c = a + b;
```

Ama bilgisayar bunu şu şekilde işler:

1. **Yüksek seviyeli dil** (C, Python vs.)
2. **Derleyici / yorumlayıcı**, kodu **Assembly diline çevirir**
3. **Assembly**, makineye yakın ama okunabilir ara katmandır.
4. Son olarak, bu da **makine dili**ne (0'lar ve 1'ler) çevrilir.

Ve işte biz Reverse Engineering yaparken bu yolu **tersten** gideriz! Elimizdeki derlenmiş dosyayı alırız, Assembly seviyesinde kodları çözer, programın ne yaptığını anlamaya çalışırız.

Reverse Engineering'de Ne Öğreniriz?

- Fonksiyonlar nasıl işliyor?
- Bellekte neler oluyor?
- Şifreleme algoritmaları nasıl uygulanıyor?
- Ağ bağlantısı yapılıyor mu?
- Kullanıcı verileri dışarı çıkıyor mu?

Bunları çözmek biraz sabır, biraz da pratik ister. Ama inanın, her çözülen detay insanı mutlu ediyor.

Reverse Engineering'de Kullanabileceğiniz Diğer Araçlar

1) IDA Pro:

Reverse dünyasının kraliçesi. Statik ve dinamik analizde kullanılabilir. Ben ona *"IDA kadınım"* diyorum.

2) x64dbg / x32dbg

Adım adım debug yapmaya yarar.

3) Process Monitor

Gerçek zamanlı dosya, kayıt defteri ve işlem etkinliklerini izler.

4) Wireshark

Ağ analiz aracı, dışa veri gönderimi tespiti için kullanılır.

5) Detect It Easy (DIE)

Binary'leri sınıflandırır, packer'ları tespit eder. Dosyanın pack edilmiş olup olmadığını analiz eder.

Sonuç: Sabır, Merak ve Biraz Dedektiflik

Reverse engineering bir yolculuktur. Bazen anlamakta zorlanırsın, bazen saatlerce takılırsın. Ama küçük bir fonksiyonun ne işe yaradığını çözdüğünde yaşadığın mutluluk anı, her şeye değer.

"Reverse öğrenmek hacker'lık değil, bir merak işidir.
Bilgiye giden yol bazen tersten yürümektir."

İster yazılımcı, ister güvenlikçi ol, bu dünyaya bir adım at. Belki de içinde bir kod dedektifi yatıyordur.