Tersine Mühendislik Metodolojisi

“Tersine mühendislik nedir ve nasıl yapılır?” sorusunun pratik cevaplarına teorik bir yaklaşım

TTMO Meetup #1

07/09/2019

whoami

- Profesyonel öğrenci @İTÜ
- İki satır kodu bir araya zor getiririm *Segmentation Fault (core dumped)*
- Cuma akşamları doğal yaşam alanım olan CTF'lerde bulunurum
- Bir iki blog yazısı yazmışlığım vardır @erfur.github.io

CTFTIME @locexum

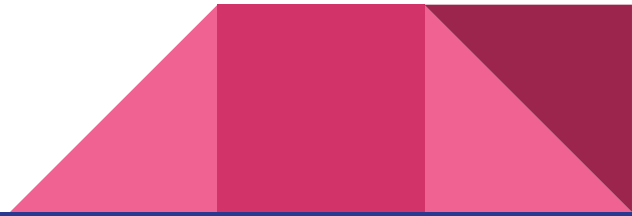
@ihavelotsofspac

@erfur

@erfur

Ne zorum vardı?

- Pratik ve çok dağınık bir alan
- Soyut temellere oturtmanın avantajları:
  - Uygulanabilirlik
  - Doğrulanabilirlik
  - Ölçeklenebilirlik



Tersine Mühendislik nedir?

“To reverse engineer a system is to infer how its underlying mechanism works.”

Lee, NY Louis, and P. N. Johnson-Laird. "A theory of reverse engineering and its application to Boolean systems." *Journal of Cognitive Psychology* 25.4 (2013): 365-389

“Tersine mühendislik bir sistemin altında yatan mekanizmanın nasıl çalıştığını anlamaktır.”

???

“Tersine mühendislik bir sistemin altında yatan mekanizmanın nasıl çalıştığını anlamaktır.”

Sistem

Sistem, birbiriyle etkileşen veya ilişkili olan, bir bütün oluşturan cisim veya varlıkların bileşkesidir. Bu varlıklar soyut veya somut olabilirler. Vikipedi

???

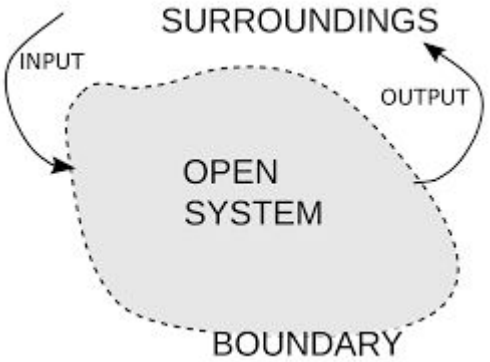
“Tersine mühendislik bir sistemin altında yatan mekanizmanın nasıl çalıştığını anlamaktır.”

Sistem

Sistem, birbiriyle etkileşen veya ilişkili olan, bir bütün oluşturan cisim veya varlıkların bileşkesidir. Bu varlıklar soyut veya somut olabilirler. Vikipedi

???

“Tersine mühendislik bir sistemin altında yatan mekanizmanın nasıl çalıştığını anlamaktır.”



Neden tersine mühendislik?

- Var olan sistemi geliştirmek, hatalarını bulmak/gidermek için
  - Hata ayıklama
  - İmplementasyon sonucu oluşan yan etkileri inceleme
  - Binary Exploitation
- Başka bir üreticinin (rakip/düşman/bilinmeyen) sistemini kopyalamak ya da bu sistemle entegrasyon yapabilmek için
  - Araba üreticilerinin diğer markaların modellerini incelemesi
  - 2. Dünya Savaşı'nda İngilizlerin Almanların Enigma kodunu çözmeleri
  - Dökümantasyonu olmayan API'ler
- Kayıp dizayn bilgisini, soyut modeli elde edebilmek için
  - Y2K olayı sonucu kaybolan kaynak kodları
  - Zararlı yazılım analizi

~~Tersine~~ Düzüne(Forward) mühendislik nedir?

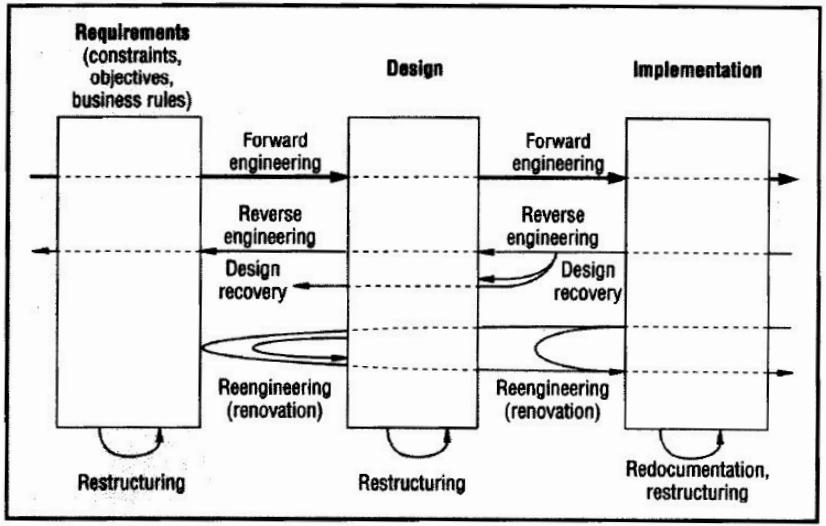


Figure 1. Relationship between terms. Reverse engineering and related processes are transformations between or within abstraction levels, represented here in terms of life-cycle phases.

Chikofsky, Elliot J., and James H. Cross. "Reverse engineering and design recovery: A taxonomy." *IEEE software* 7.1 (1990): 13-17.

Gereksinimler → Dizayn → Uygulama → Sistem

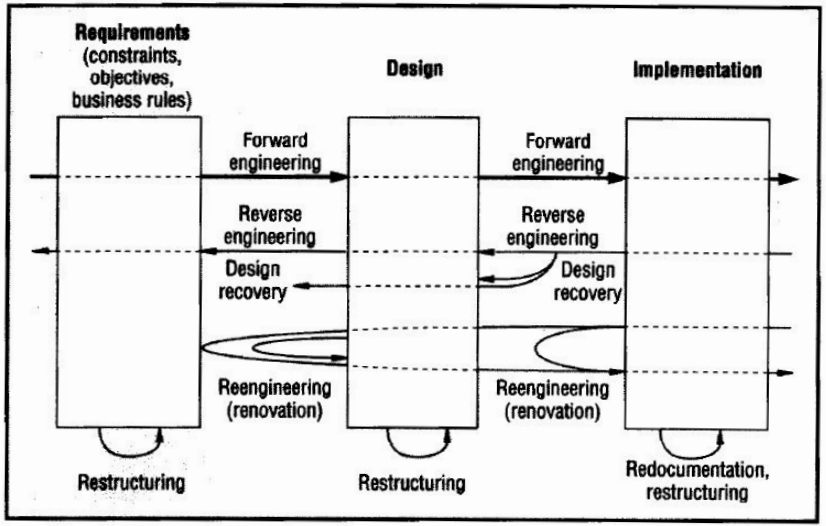
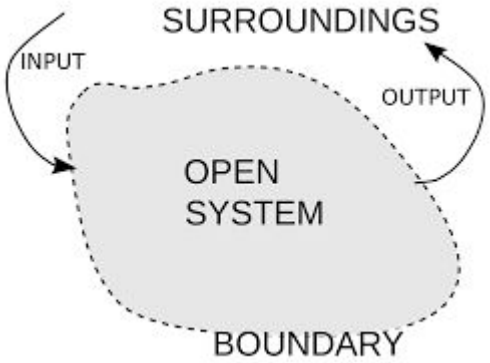


Figure 1. Relationship between terms. Reverse engineering and related processes are transformations between or within abstraction levels, represented here in terms of life-cycle phases.



Chikofsky, Elliot J., and James H. Cross. "Reverse engineering and design recovery: A taxonomy." *IEEE software* 7.1 (1990): 13-17.

Gereksinimler → Dizayn → Uygulama → Sistem

Girdi:
Sayılardan oluşan bir liste

Çıktı:
Sayıların sıralanmış halinden oluşan bir liste

Obje sayısı fazla, bu sebeple çalışma zamanı: $O(n^2)$'den az olsun

Gereksinimler → Dizayn → Uygulama → Sistem

Girdi: Sayılardan oluşan bir liste Çıktı: Sayıların sıralanmış halinden oluşan bir liste Obje sayısı fazla, bu sebeple çalışma zamanı: $O(n^2)$'den az olsun	Stdin'den listeyi oku Merge sort kullanarak listeyi sırala Sıralı listeyi stdout'a yaz.

Gereksinimler → Dizayn → Uygulama → Sistem

Gereksinimler	Dizayn	Uygulama

Girdi:
Sayılardan oluşan bir liste
 	
Çıktı:
Sayıların sıralanmış halinden oluşan bir liste
 	
Obje sayısı fazla, bu sebeple çalışma zamanı: $O(n^2)$'den az olsun
	Stdin'den listeyi oku
	
	Merge sort kullanarak listeyi sırala
	
	Sıralı listeyi stdout'a yaz.
		Rosettacode'dan merge sort kodunu kopyala
		
		Scanf ile listeyi oku
		Printf ile listeyi yazdır
		
		`gcc -O3 sort.c -o a.out`
		`strip --strip-all a.out`

Gereksinimler → Dizayn → Uygulama → Sistem

Gereksinimler	Dizayn	Uygulama	Sistem

Girdi:
Sayılardan oluşan bir liste
 	
Çıktı:
Sayıların sıralanmış halinden oluşan bir liste
 	
Obje sayısı fazla, bu sebeple çalışma zamanı: $O(n^2)$'den az olsun
	Stdin'den listeyi oku
	
	Merge sort kullanarak listeyi sırala
	
	Sıralı listeyi stdout'a yaz.
		Rosettacode'dan merge sort kodunu kopyala
		
		Scanf ile listeyi oku
		Printf ile listeyi yazdır
		
		`gcc -O3 sort.c -o a.out`
		`strip --strip-all a.out`
			
			a.out

a.out

Hedef: Anlamlandırma

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

a.out

- Bu bir dosya
- Dosyalar başlık kısımlarına sahiplerdir

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

a.out

- Bu bir dosya
- Dosyalar başlık kısımlarına sahiplerdir

- İlk byteları okuyup dosyanın türü hakkında bilgi alabiliriz.

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

a.out

- Bu bir dosya
- Dosyalar başlık kısımlarına sahiplerdir

- İlk byteları okuyup dosyanın türü hakkında bilgi alabiliriz.

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

```
[erfur@battlestation rev-dialtone]$ file a.out
a.out: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked,
 interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=9cc75
9438a1edd4207f7d6b9b623985415589928, not stripped
```

Girdi

a.out
Çalıştırılabilir ELF dosyası

Çıktı

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

Girdi

a.out
Çalıştırılabilir ELF dosyası

Çıktı

strings çıktısından:

“Invalid input! Please only enter numbers (with spaces in between) to sort them.”

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

Girdi

a.out
Çalıştırılabilir ELF dosyası

Çıktı

strings çıktısından:

“Invalid input! Please only enter numbers (with spaces in between) to sort them.”

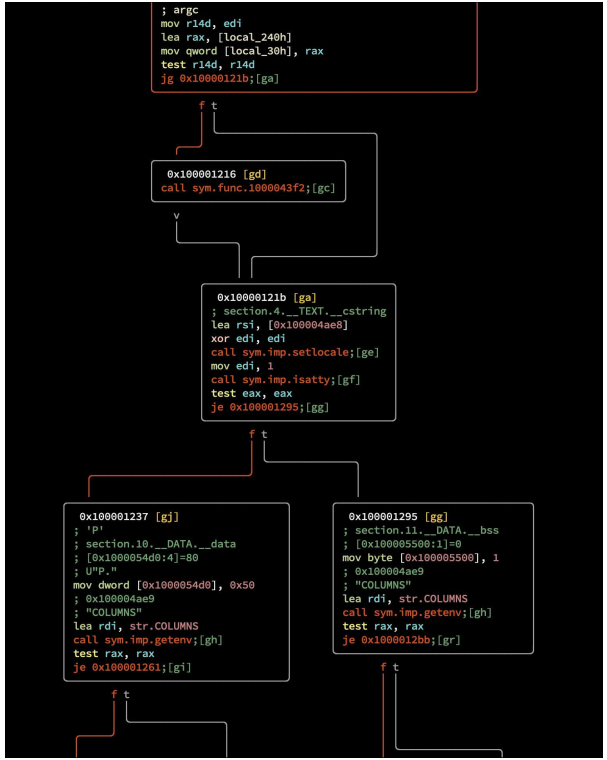
Girdi olarak verilen sayı listesini sıralayan bir program.

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

Girdi

a.out
Çalıştırılabilir ELF dosyası

Çıktı



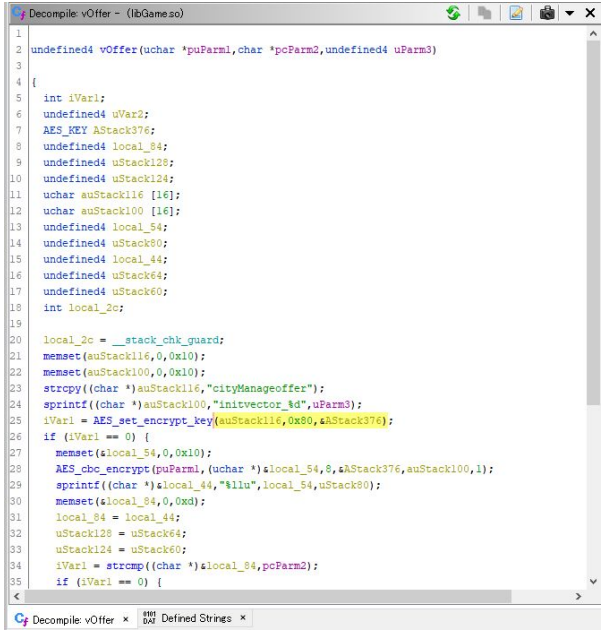
radare2

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

Girdi

a.out
Çalıştırılabilir ELF dosyası

Çıktı



GHIDRA Decompiler

1. Ön bilgiler
2. Spesifikasyon
3. Veri toplama
4. Veriyi işleme
5. Anlamlandırma

Sonuç

- Yapılan iş temelde bilinenleri kullanarak (bilgi+teknik) bilinmeyeni ortaya çıkararak daha soyut temsillere ulaşmak.
- Ön bilgi ve teknik ne kadar çoksa yapılacak iş o kadar az.
- Araçların önemi çok büyük, veriyi soyutlayıp küçülterek tersine mühendisin anlamlandırmasını kolaylaştırıyorlar.
- Mümkün olduğunca çok araç ve bilgiye sahip olmak önemli.
- Daha çok soyutlama yapabilmek için tekniklerin geliştirilmesi gerekiyor.

EOF