CEN429 GÃ¼venli Programlama Hafta-1

GÃ¼venli Programlamaya GiriÅŸ ve Bilgisayar VirÃ¼sleri

Yazar: Dr. Ã–ÄŸr. Üyesi UÄŸur CORUH

Contents

1 CEN429 GÃ¼venli Programlama **1**
1.1 Hafta-1 . . . 1
1.1.1 Outline . . . 2
1.2 **Uygulama Koruma PlanÄ± (Application Protection Plan)** . . . 2
1.2.1 1. **Kod BÃ¶lme (Split)** . . . 2
1.2.2 2. **Kod DoÄŸrulama (Measure)** . . . 2
1.2.3 3. **Zamanlama (Time)** . . . 2
1.2.4 4. **Protokol Ä°zleme (Monitor)** . . . 3
1.3 **Bilgisayar VirÃ¼sleri** . . . 3
1.3.1 1. **VirÃ¼slerin Ã–zellikleri** . . . 3
1.3.2 2. **VirÃ¼s TÃ¼rleri** . . . 3
1.3.3 3. **VirÃ¼s KarÅŸÄ± Ã–nlemleri** . . . 3
1.4 **GÃ¼venlik Modelleri ve SaldÄ±rÄ± AÄŸaÃ§larÄ± (Attack Trees)** . . . 3
1.4.1 1. **SaldÄ±rÄ± AÄŸacÄ± Nedir?** . . . 3
1.4.2 2. **Maliyet Modelleme** . . . 4
1.5 **SaldÄ±rÄ± YÃ¶ntemleri (Attack Methods)** . . . 4
1.5.1 1. **Dinamik Analiz (Dynamic Analysis)** . . . 4
1.5.2 2. **Statik Analiz (Static Analysis)** . . . 4
1.5.3 3. **Program DÃ¼zenleme (Editing Phase)** . . . 4
1.6 **GÃ¼venli Ä°letiÅŸim Hedefleri** . . . 4
1.7 **HaftanÄ±n Ã–zeti ve Gelecek Hafta** . . . 4
1.7.1 Bu Hafta: . . . 4
1.7.2 Gelecek Hafta: . . . 5

List of Figures

List of Tables

1 CEN429 GÃ¼venli Programlama

1.1 Hafta-1

1.1.0.1 Ders PlanÄ± ve Ä°letiÅŸim, GÃ¼venli Programlama ve Bilgisayar VirÃ¼sleri

Download

- PDF[1]
- DOC[2]
- SLIDE[3]

[1] pandoc_cen429-week-1.pdf
[2] pandoc_cen429-week-1.docx
[3] cen429-week-1.pdf

- PPTX[4]

1.1.1 Outline

- GÃ¼venli Programlama ve Bilgisayar VirÃ¼sleri
- Uygulama Koruma PlanÄ±
  - Kod BÃ¶lme
  - Kod DoÄŸrulama
  - Zamanlama
  - Protokol Ä°zleme
- Bilgisayar VirÃ¼sleri
  - VirÃ¼slerin Ã–zellikleri
  - VirÃ¼s TÃ¼rleri
  - VirÃ¼s KarÅŸÄ± Ã–nlemleri
- SaldÄ±rÄ± AÄŸaÃ§larÄ± ve GÃ¼venlik Modelleri
- SaldÄ±rÄ± YÃ¶ntemleri
- GÃ¼venli Ä°letiÅŸim Hedefleri

1.2 Uygulama Koruma PlanÄ± (Application Protection Plan)

1.2.1 1. Kod BÃ¶lme (Split)

1.2.1.1 Teorik AÃ§Ä±klama: Kod bÃ¶lme, gÃ¼venilmeyen ortamda yÃ¼rÃ¼tÃ¼len iÅŸlemleri gÃ¼venilir bir ortama taÅŸÄ±ma yÃ¶ntemidir. Bu sayede gÃ¼venlik aÃ§Ä±klarÄ± minimize edilir.

1.2.1.2 Uygulama:

- **Uygulama:** Bir istemci-sunucu modelinde Åifreleme iÅŸlemlerini istemci yerine sunucuda gerÃ§ekleÅŸtiren bir sistem kurun. Bu, kritik iÅŸlemleri gÃ¼venli ortamda yÃ¼rÃ¼tmek iÃ§in kullanÄ±lÄ±r.

1.2.2 2. Kod DoÄŸrulama (Measure)

1.2.2.1 Teorik AÃ§Ä±klama: GÃ¼venilmeyen bir siteye ya da cihaza "DoÄŸru kodu mu Ã§alÄ±ÅŸtÄ±rÄ±yorsun?" Åeklinde sorular yÃ¶nelterek, sistemin beklenen davranÄ±ÅŸlarÄ± sergilediÄŸini kontrol ederiz.

1.2.2.2 Uygulama:

- **Uygulama:** Bir uygulamanÄ±n Ã§alÄ±ÅŸma sÄ±rasÄ±nda belirli matematiksel problemlere doÄŸru ve hÄ±zlÄ± yanÄ±t verip vermediÄŸini kontrol eden bir sistem geliÅŸtirin. Bu sistem, doÄŸruluÄŸu kanÄ±tlayamazsa iÅŸlem yapmaz.

1.2.3 3. Zamanlama (Time)

1.2.3.1 Teorik AÃ§Ä±klama: GÃ¼venilmeyen bir sistemde, iÅŸlem yapÄ±lmasÄ± gereken bir zorluk hesaplatÄ±lÄ±r ve belirli bir zaman dilimi iÃ§erisinde cevap beklenir. Bu teknik, saldÄ±rganlarÄ±n analiz iÃ§in yeterli zamanÄ± bulmasÄ±nÄ± engeller.

1.2.3.2 Uygulama:

- **Uygulama:** Bir "Zaman Temelli Soru-Cevap" uygulamasÄ± oluÅŸturun. Belirli bir sÃ¼re iÃ§inde cevap alÄ±nmazsa oturum sonlandÄ±rÄ±lsÄ±n.

[4] cen429-week-1.pptx

1.2.4 4. Protokol Ä°zleme (Monitor)

1.2.4.1 Teorik AÃ§Ä±klama: Veri transferi sÄ±rasÄ±nda protokol akÄ±ÅŸÄ±nÄ± izleyerek, olasÄ± gÃ¼venlik aÃ§Ä±klarÄ±nÄ± veya kÃ¶tÃ¼ niyetli iÅŸlemleri tespit ederiz.

1.2.4.2 Uygulama:

- **Uygulama:** Bir web sunucusunda yapÄ±lan HTTP isteklerini izleyen bir log sistemi oluÅŸturun. ÅžÃ¼pheli istekler algÄ±landÄ±ÄŸÄ±nda kullanÄ±cÄ±yÄ± engelleyin.

1.3 Bilgisayar VirÃ¼sleri

1.3.1 1. VirÃ¼slerin Ã–zellikleri

- **Uyuma Durumu (Dormant):** VirÃ¼s bir sÃ¼re sessiz kalabilir, algÄ±lanmaktan kaÃ§Ä±nÄ±r.
- **YayÄ±lma (Propagation):** Yeni dosyalara veya sistemlere bulaÅŸÄ±r.
- **Tetikleme (Triggering):** VirÃ¼sÃ¼n harekete geÃ§eceÄŸi zamanÄ± belirleyen olay.
- **Eylem (Action):** ZararlÄ± iÅŸlem yapÄ±lÄ±r, bu genellikle "payload" denir.

1.3.1.1 Uygulama:

- **Uygulama:** Bir simÃ¼lasyon oluÅŸturun. VirÃ¼s uyuma durumunda beklesin, belirli bir tarihte etkinleÅŸip bir dosya silme iÅŸlemi yapsÄ±n.

1.3.2 2. VirÃ¼s TÃ¼rleri

- **Program/Dosya VirÃ¼sÃ¼:** Program dosyalarÄ±na bulaÅŸÄ±r.
- **Makro VirÃ¼sÃ¼:** Word/Excel belgelerine bulaÅŸÄ±r ve belge aÃ§Ä±ldÄ±ÄŸÄ±nda Ã§alÄ±ÅŸÄ±r.
- **Boot SektÃ¶rÃ¼ VirÃ¼sÃ¼:** Sabit diskin Ã¶nyÃ¼kleme sektÃ¶rÃ¼ne bulaÅŸÄ±r, bilgisayar baÅŸlatÄ±ldÄ±ÄŸÄ±nda Ã§alÄ±ÅŸÄ±r.

1.3.2.1 Uygulama:

- **Uygulama:** FarklÄ± virÃ¼s tÃ¼rlerinin nasÄ±l Ã§alÄ±ÅŸtÄ±ÄŸÄ±nÄ± gÃ¶steren bir simÃ¼lasyon oluÅŸturun. Her virÃ¼s tÃ¼rÃ¼ farklÄ± tetikleyicilerle harekete geÃ§sin.

1.3.3 3. VirÃ¼s KarÅŸÄ± Ã–nlemleri

- **Ä°mza TabanlÄ± Tespit (Signatures):** VirÃ¼sÃ¼n bilinen kod parÃ§alarÄ±na dayalÄ± tespit yÃ¶ntemidir.
- **Åžifreleme:** VirÃ¼slerin kodlarÄ±nÄ±n ÅŸifrelenmesi, imza tespitine karÅŸÄ± koruma saÄŸlar.

1.3.3.1 Uygulama:

- **Uygulama:** ÅžifrelenmiÅŸ bir virÃ¼s simÃ¼lasyonu oluÅŸturun. VirÃ¼s kodu her Ã§alÄ±ÅŸtÄ±rÄ±ldÄ±ÄŸÄ±n farklÄ± bir anahtar ile ÅŸifrelenmiÅŸ olsun.

1.4 GÃ¼venlik Modelleri ve SaldÄ±rÄ± AÄŸaÃ§larÄ± (Attack Trees)

1.4.1 1. SaldÄ±rÄ± AÄŸacÄ± Nedir?

SaldÄ±rÄ± aÄŸacÄ±, bir saldÄ±rganÄ±n bir hedefe ulaÅŸma stratejilerini anlamamÄ±zÄ± saÄŸlayan bir yapÄ±dÄ±r. Bu model, gÃ¼venlik aÃ§Ä±klarÄ±nÄ± gÃ¶rselleÅŸtirerek saldÄ±rÄ±lara karÅŸÄ± etkili savunmalar geliÅŸtirilmesine yardÄ±mcÄ± olur.

1.4.1.1 Uygulama:

- **Uygulama:** Basit bir saldÄ±rÄ± aÄŸacÄ± oluÅŸturun. Ã–rneÄŸin, bir web uygulamasÄ±nda SQL enjeksiyonundan baÅŸlayarak, veritabanÄ±na eriÅŸime kadar olan adÄ±mlarÄ± modelleyin.

1.4.2 2. Maliyet Modelleme

Her saldÄ±rÄ± adÄ±mÄ±nÄ±n bir maliyeti vardÄ±r. Bu maliyetler saldÄ±rganÄ±n hedefe ulaÅŸmasÄ±nÄ± zorlaÅŸtÄ±rmak iÃ§in hesaplanabilir. Bir saldÄ±rÄ± aÄŸacÄ±nda, maliyetler her bir dÃ¼ÄŸÃ¼me atanÄ±r ve en az maliyetli yol hesaplanÄ±r.

1.4.2.1 Uygulama:

- **Uygulama:** Bir saldÄ±rÄ± aÄŸacÄ±nda her adÄ±mÄ±n maliyetini hesaplayan bir simÃ¼lasyon geliÅŸtirin. En dÃ¼ÅŸÃ¼k maliyetle hedefe ulaÅŸmayÄ± simÃ¼le edin.

1.5 SaldÄ±rÄ± YÃ¶ntemleri (Attack Methods)

1.5.1 1. Dinamik Analiz (Dynamic Analysis)

Bir programÄ±n Ã§alÄ±ÅŸÄ±rken hangi bÃ¶lÃ¼mlerinin tetiklendiÄŸini ve hangi girdilerle nasÄ±l davranÄ±ÅŸlar sergilediÄŸini anlamaya yarar.

1.5.1.1 Uygulama:

- **Uygulama:** Bir yazÄ±lÄ±mÄ±n Ã§alÄ±ÅŸma zamanÄ±nda hangi iÅŸlevlerin Ã§aÄŸrÄ±ldÄ±ÄŸÄ±nÄ± izleyen ve bu iÅŸlevlerin hangi girdilerle tetiklendiÄŸini gÃ¶steren bir izleyici oluÅŸturun.

1.5.2 2. Statik Analiz (Static Analysis)

Bir programÄ±n kaynak kodu veya derlenmiÅŸ halinin analiz edilmesi iÅŸlemidir. Bu analiz ile potansiyel gÃ¼venlik aÃ§Ä±klarÄ± belirlenir.

1.5.2.1 Uygulama:

- **Uygulama:** Bir disassembler kullanarak, basit bir programÄ±n derlenmiÅŸ kodunu analiz edin ve zayÄ±f noktalarÄ± tespit edin.

1.5.3 3. Program DÃ¼zenleme (Editing Phase)

Bir saldÄ±rgan, yazÄ±lÄ±mÄ±n iÃ§ iÅŸleyiÅŸini anladÄ±ktan sonra, lisans denetimlerini devre dÄ±ÅŸÄ± bÄ±rakmak veya kÄ±sÄ±tlamalarÄ± kaldÄ±rmak iÃ§in programÄ± dÃ¼zenleyebilir.

1.5.3.1 Uygulama:

- **Uygulama:** Lisans denetimini atlamak iÃ§in bir programÄ±n ikili dosyasÄ±nÄ± dÃ¼zenleyin. Hangi kÄ±sÄ±tlamalarÄ±n kaldÄ±rÄ±ldÄ±ÄŸÄ±nÄ± izleyin.

1.6 GÃ¼venli Ä°letiÅŸim Hedefleri

- **KarÅŸÄ±lÄ±klÄ± Kimlik DoÄŸrulama:** Ä°letiÅŸime giren iki tarafÄ±n birbirini doÄŸrulamasÄ±.
- **Anahtar Ä°ptali:** GeÃ§ersiz anahtarlarÄ±n iptal edilmesi.
- **YÃ¼ksek Performans:** GÃ¼venli iletiÅŸimde hÄ±z ve dÃ¼ÅŸÃ¼k gecikme sÃ¼resi esastÄ±r.

1.6.0.1 Uygulama:

- **Uygulama:** Ä°ki tarafÄ±n karÅŸÄ±lÄ±klÄ± olarak birbirini doÄŸrulamasÄ±nÄ± saÄŸlayan basit bir kimlik doÄŸrulama protokolÃ¼ oluÅŸturun.

1.7 HaftanÄ±n Ã–zeti ve Gelecek Hafta

1.7.1 Bu Hafta:

- **Uygulama Koruma PlanÄ±**
- **Bilgisayar VirÃ¼sleri ve TÃ¼rleri**

- **SaldÄ±rÄ± AÄŸaÃ§larÄ± ve GÃ¼venlik Modelleri**
- **SaldÄ±rÄ± YÃ¶ntemleri ve GÃ¼venli Ä°letiÅŸim Hedefleri**

1.7.2 Gelecek Hafta:

- **Veri GÃ¼venliÄŸi**
- **Kriptografik Teknikler**
- **UygulamalÄ± Åžifreleme**

$1. Hafta - Sonu$