Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

14. Güvenlik

Yazılım Geliştirme Sürecinde Güvenlik

Güvenlik hedefi

Güvenlik amacı, bir bilgisayar sistemi, verileri ve kaynakları ile etkileşime giren aracıların (insanlar veya harici sistemler), sistem sahibinin bu tür etkileşimlere izin verdiği kişiler olduğundan emin olmaktır.

Güvenlikle ilgili dikkat edilmesi gereken noktaların tüm yazılım geliştirme sürecinin bir parçası olması gerekir. Seçilen sistem mimarisi üzerinde büyük bir etkiye sahip olabilirler.

Güvenlik İhtiyaçları ve Tehlikeleri

İhtiyaçlar

- Sır tutma (Secrecy): Bilgileri kimin okuyacağının kontrolü
- Bütünlük (Integrity): Bilgi değişikliklerinin veya kaynakların nasıl kullanıldığının kontrolü
- Kullanılabilirlik (Availability): Bilgi ve kaynaklara hızlı erişim sağlamak
- Hesap verebilirlik (Accountability): Kaynaklara kimin erişimi olduğunu bilmek

Tehlikeler

- Bilginin zarar görmesi — integrity
- Hizmetin aksaması (Hizmet reddi) — availability
- Para hırsızlığı — integrity
- Bilgi hırsızlığı — secrecy
- Gizlilik kaybı — secrecy

Butler W. Lampson, Computer Security in the Real World
IEEE Computer, June 2004

Güvenlik Ekonomisi

Sisteminiz ne kadar güvenli olmalıdır?

Güvenli sistemler oluşturma, yazılım geliştirmeye sürecinde maliyete ve zamana mal olur.

"Pratik güvenlik, koruma maliyetini ve kayıp riskini dengeler, bu da bir kayıptan kurtulmanın maliyeti ile olasılığının çarpımıdır... Risk, iyileşme maliyetinden az olduğunda bunu iş yapmanın bir maliyeti olarak kabul etmek daha iyidir… daha iyi güvenlik için ödeme yapmaktan daha iyidir."

"Uygulamada birçok şirket, insanların yetersiz güvenlikten şikayet etmelerine rağmen, aslında fazla para harcamayacaklarını, birçok özellikten ödün vermeyeceklerini veya çok fazla zorluğa katlanmayacaklarını öğrendi."

Butler W. Lampson, 2004

Kuruluş İçinde Güvenlik: Kişiler

Birçok güvenlik sorunu kuruluş içindeki kişilerden kaynaklanmaktadır.

- Büyük bir organizasyonda, bazı dürüst olmayan ve hoşnutsuz çalışanlar olacaktır.

  -> Sahtekâr (Örneğin, finansal sistemlerden çalmak)

  -> Kötü niyetli

- Güvenlik, güvenilir kişilere dayanır. Ya sahtekârlarsa?

İnsanlar doğası gereği güvensizdir.

- Dikkatsizlik (örneğin, bilgisayarları açık bırakırlar, şifreleri paylaşırlar)

Güvenlik için Tasarım: İnsanlar
Bazı genel öneriler…

- **Sorumlu** kişilerin sistemi kullanmasını kolaylaştırın (örneğin, güvenlik prosedürlerini basitleştirin).

  Örnek: Bir bankadaki karmaşık güvenlik prosedürleri.

- **Dürüst olmayan** veya **dikkatsiz** kişilerin işini zorlaştırın (örneğin, şifre yönetimi).
- İnsanları **sorumlu davranışlar** konusunda eğitin.
- Sistemin güvenliğini, özellikle değişikliklerden sonra iyice ve tekrar tekrar **test edin**.
- **İhlalleri gizlemeyin ve kamuoyu ile paylaşın.**

Dışarıdan Davetsiz Misafirler

Tüm ağ sistemleri, harici davetsiz misafirlerin güvenlik ihlallerine karşı savunmasızdır:

- Amaçları finansal,
- kötü niyetli
- sırlar
- Ve daha kötüleri olabilir.

Modern yazılım o kadar karmaşıktır ki, tüm güvenlik açıklarını ortadan kaldırdığınızı garanti etmek imkansızdır.

- Birçok yetenekli kişi ve kuruluş, sürekli olarak yeni güvenlik açıklarını keşfetmeye ve bunlardan yararlanmaya çalışmaktadır.

Dışarıdan Davetsiz Misafirler

Dış güvenlik açıklarına örnekler:

- Yetkisiz erişim — yazılımı değiştirin, dinleme cihazları kurulumu
- Arka kapılar — Kimlik doğrulamayı atlama
- Hizmet reddi — aşırı yük ve diğer engelleme biçimleri
- Dinleme
- Sızdırma
- kimlik avı vb.,
- vs.

Bu liste Wikipedia'dan oluşturulmuştur.

Harici Davetsiz Misafirler: Çevrimiçi Veriler

Çevrimiçi sistemler büyük miktarda özel veriyi yakalar ve kaydeder:

- Alışveriş siteleri: isimler, adresler, kredi kartları (ör., Amazon)
- Finansal sistemler: finansal veriler, sosyal güvenlik numaraları, kredi verileri
- Sosyal medya: arkadaşlar, arama terimleri, konum verileri (ör. Facebook)

Bu veriler saldırıya uğrarsa:

Kullanıcılara zararlar:

- Mali kayıplar (örneğin, kredi notları)
- Duygusal kayıplar (örneğin, tıbbi geçmiş)

Sistemleri çalıştıran şirketlerin zararları:

- Para cezaları (hatta cezai suçlamalar)
- Zararlar için yasal uzlaşmalar
- Müşteri kaybı
- Sorunları çözmenin teknik maliyetleri

Harici Davetsiz Misafirler: Riski En Aza İndirme

Harici davetsiz misafirlere karşı güvenliği garanti etmenin bir yolu yoktur, ancak dikkatli yazılım geliştirme büyük bir fark yaratabilir.

Riskler nasıl en aza indirilir?:

- Sistem tasarımı — güvenli protokoller, kimlik doğrulama, erişim engelleri
- Programlama — defansif programlama ve titiz testler
- İşletim prosedürleri — yedekleme, denetim, güvenlik açığı testi
- Personelin eğitimi ve izlenmesi

Riskleri En Aza İndirmek: Sistem Mimarisi

Sistem mimarisi riskleri çeşitli şekillerde en aza indirebilir:

- Güvenli protokoller, ör. HTTPS şifrelemesi.
- Kimlik doğrulama:
  - -> İletim sırasında ve saklandığında parolaların şifrelenmesi
  - -> İki faktörlü kimlik doğrulama (etkili ancak rahatsız edici olabilir)
- Engeller, ör. güvenlik duvarları, özel ağlar ve sanal özel ağlar.
- Veri güvenliği, ör. depolanan verilerin şifrelenmesi, yedekleme.

Riskleri En Aza İndirmek: Çevrimiçi Veriler

Veri toplama

- Yalnızca temel verileri toplayın.
- Kullanılmayan verileri silin.
- Hassas verileri şifreleyin.
- Düzenli olarak güvenlik denetimleri gerçekleştirin.

Belaya hazırlıklı olun

- Tüm verileri yedekle.
- Gizlilik yasalarını bilin.
- Bir afet planınız olsun:
  - kim lider olacak?
  - kullanıcılara nasıl bir mesaj?
  - halkla İlişkiler
  - teknik kurtarma prosedürleri

Güvenlik Teknikleri: Bariyerler

Karmaşık bir sistemin parçalarını ayıran engeller yerleştirin:

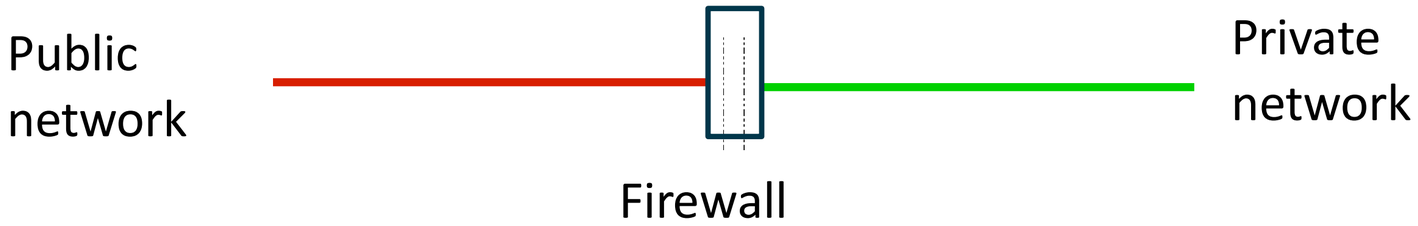
- Bileşenleri izole edin, örneğin bir bilgisayarı bir ağa bağlamayın
- Güvenlik duvarları
- Belirli sistemlere veya sistem bölümlerine erişmek için kimlik doğrulaması oluşturun

Her engel, sistemin izin verilen kullanımlarına kısıtlamalar getirir.

Engeller, sistem alt sistemlere bölünebildiğinde en etkilidir.

Örnek. Internet Explorer'ın Windows'a entegrasyonu

Bariyerler: Güvenlik Duvarı (firewall)



Güvenlik duvarı, iki ağ kesiminin birleştiği yerde bulunan bir bilgisayardır:

- Sınırı geçmeye çalışan her paketi inceler
- Belirli kriterleri karşılamayan herhangi bir paketi reddeder, örn.,

  -> TCP bağlantısı açmak için gelen istek

  -> Bilinmeyen bir paket türü

Güvenlik duvarları, esneklik kaybı, kullanıcılar için rahatsızlık ve ekstra sistem yönetimi ile daha fazla güvenlik sağlar.

Güvenlik Teknikleri: Kimlik Doğrulama (authentication) ve Yetkilendirme (authorization)

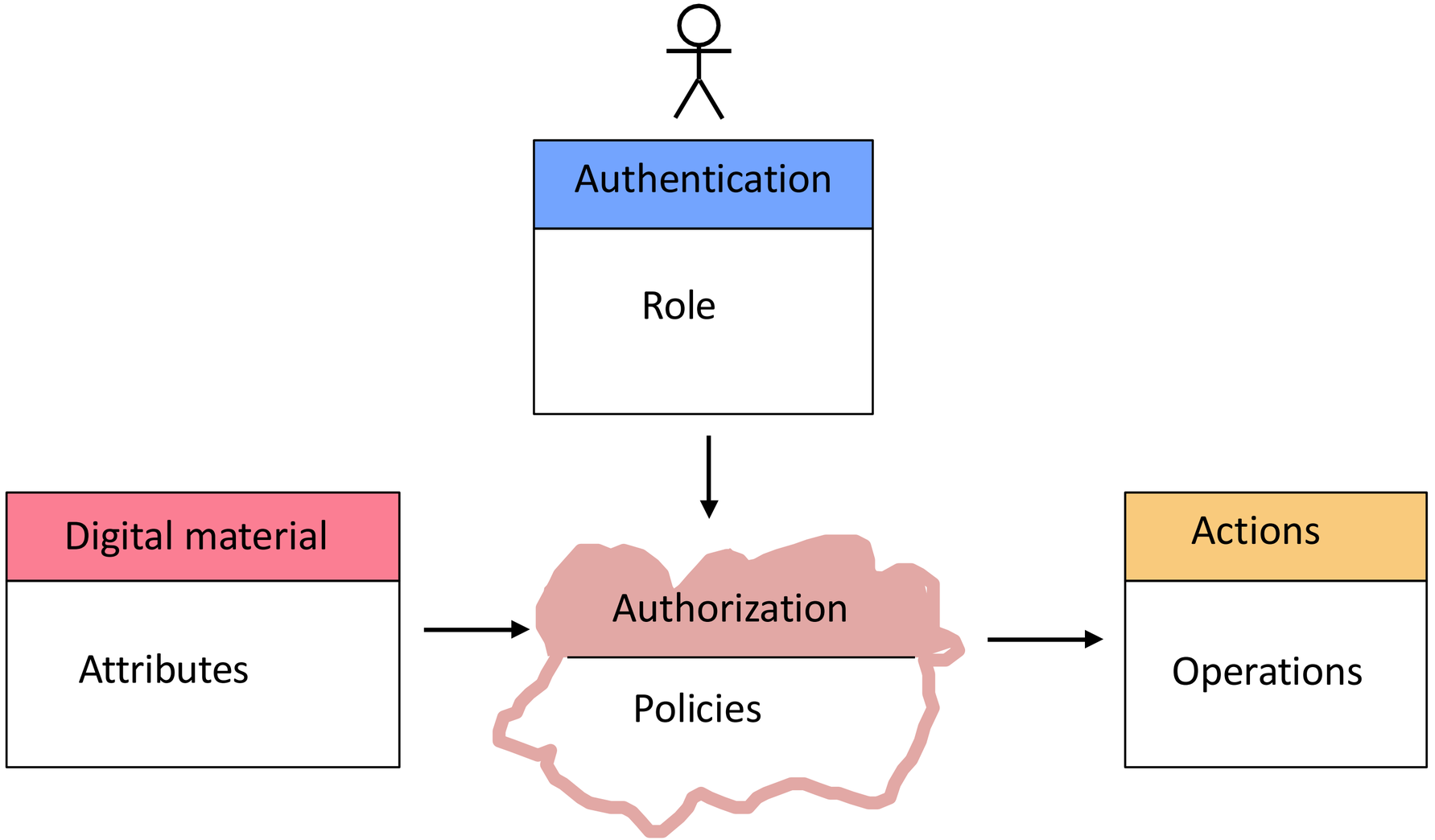
Kimlik doğrulaması, bir aracının kimliğini belirler:

- Kullanıcı ne biliyor (örneğin, şifre)?
- Kullanıcı sahip olduğu şey nedir (örneğin, akıllı kart)?
- Kullanıcının fiziksel erişimi nedir (örneğin, crt-alt-del)?
- Kullanıcının fiziksel özellikleri nelerdir (örn. parmak izi)?

Yetkilendirme, kimliği doğrulanmış bir aracının neler yapabileceğini belirler:

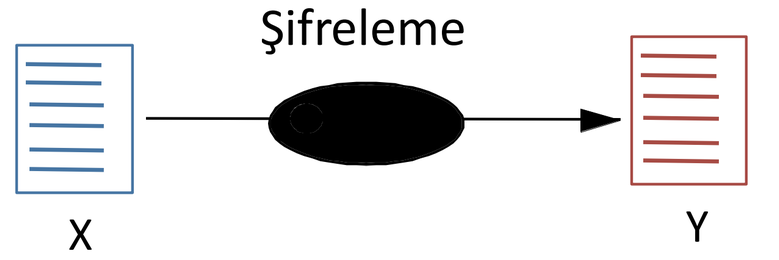
- Erişim kontrol listeleri (Access control lists, ACLs)
- Grup üyeliği

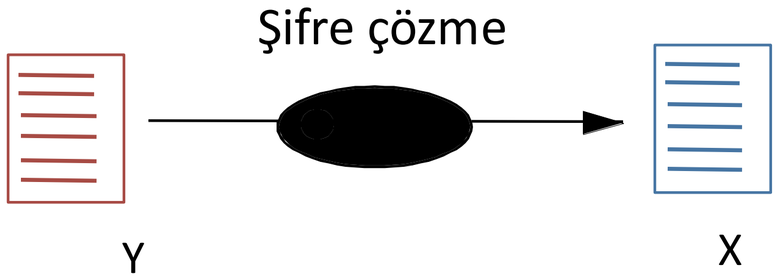
Örnek: Dijital İçerik için Erişim Mimarisi



Güvenlik Teknikleri: Şifreleme

Bitler yetkisiz ajanlar tarafından görüntülendiğinde ve algoritmalar bilindiğinde bile verilerin güvenli bir şekilde saklanmasına ve iletilmesine izin verir.





- Özel anahtar ve genel anahtar
- Dijital imzalar

Güvenli Yazılım Programlama

Dış dünyayla arayüz oluşturan programların (örneğin, web siteleri, posta sunucuları) izinsiz girişlere karşı koyacak şekilde yazılması gerekir.

En önemli 25 programlama hatası için, bakınız: *Common Weakness Evaluation: A Community-Developed Dictionary of Software Weakness Types*. http://cwe.mitre.org/top25/

- Bileşenler arasında güvensiz etkileşim
- Riskli kaynak yönetimi
- Geçirgen savunma sistemleri

Proje yönetimi ve test prosedürleri, programların bu hatalardan kaçındığını vurgulamalıdır.

MITRE

Rank	ID	Name	Score	KEV Count (CVEs)	Rank Change vs. 2021
1	CWE-787	Out-of-bounds Write	64.20	62	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.97	2	0
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	22.11	7	+3 ▲
4	CWE-20	Improper Input Validation	20.63	20	0
5	CWE-125	Out-of-bounds Read	17.67	1	-2 ▼
6	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	17.53	32	-1 ▼
7	CWE-416	Use After Free	15.50	28	0
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.08	19	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	11.53	1	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	9.56	6	0
11	CWE-476	NULL Pointer Dereference	7.15	0	+4 ▲
12	CWE-502	Deserialization of Untrusted Data	6.68	7	+1 ▲
13	CWE-190	Integer Overflow or Wraparound	6.53	2	-1 ▼
14	CWE-287	Improper Authentication	6.35	4	0
15	CWE-798	Use of Hard-coded Credentials	5.66	0	+1 ▲
16	CWE-862	Missing Authorization	5.53	1	+2 ▲
17	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	5.42	5	+8 ▲
18	CWE-306	Missing Authentication for Critical Function	5.15	6	-7 ▼
19	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	4.85	6	-2 ▼
20	CWE-276	Incorrect Default Permissions	4.84	0	-1 ▼
21	CWE-918	Server-Side Request Forgery (SSRF)	4.27	8	+3 ▲
22	CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	3.57	6	+11 ▲
23	CWE-400	Uncontrolled Resource Consumption	3.56	2	+4 ▲
24	CWE-611	Improper Restriction of XML External Entity Reference	3.38	0	-1 ▼
25	CWE-94	Improper Control of Generation of Code ('Code Injection')	3.32	4	+3 ▲

Riskleri En Aza İndirmek: Prosedürler

İşletim prosedürleri güvenlik sorunlarını öngörmelidir.

Kıdemli bir personel üyesinin güvenlikten sorumlu olması gerekir.

Ekipman

- Sistem yazılımını en son güvenlik yamalarıyla güncel tutun.
- Güncel virüs kontrol yazılımı çalıştırın.
- Parolalar, kişisel ekipman ve standart olmayan yazılımlarla ilgili kurallar açık olmalıdır.

Rutin kontroller

- Ağ güvenliği testlerini düzenli olarak çalıştırın.
- Parola denetleyicilerini çalıştırın.

Eğitim

- Personeli güvenlik konusunda bilgilendirin. Tavsiyelerini sorun.

Operasyonlar: Kurtarma

Er ya da geç her sistem donanım, yazılım, operasyonel veya güvenlik sorunları nedeniyle başarısız olur.

İşletim prosedürleri, her an meydana gelebilecek veri kaybını ve sistem hasarını öngörmelidir.

Yedekleme teknikleri

- Düzenli aralıklarla sistemi kontrol edin.
- Düzenli aralıklarla tüm verileri yedekleyin.
- Tüm önemli işlemlerin tam denetim kayıtlarını tutun

Kurtarma yazılımı

- Kurtarma yazılımı karmaşıktır. Gerçekçi durumlarda düzenli olarak test edilmesi gerekir.
- Tüm sistemi yeniden oluşturmak iyi bir uygulamadır, ancak bu, büyük veri tabanlarıyla mümkün olmayabilir.

Yazılım Geliştirme Sürecinde Güvenlik

Sonuç olarak,

Bir sistemin tamamen güvenli olduğunu asla garanti edemezsiniz, ancak riskleri en aza indirmek ve sorunlardan kurtulabilmek için çok şey yapabilirsiniz.

Erciyes Üniversitesi
Bilgisayar Mühendisliği Bölümü

BZ 313 Yazılım Mühendisliği

14. Güvenlik

Ders Sonu