

İTÜ
DERS KATALOG FORMU
(COURSE CATALOGUE FORM)

Dersin Adı				Course Name		
Güvenli Programlama				Secure Programming		
Kodu (Code)	Yarıyılı (Semester)	Kredisi (Local Credits)	AKTS Kredisi (ECTS Credits)	Ders Uygulaması, Saat/Hafta (Course Implementation, Hours/Week)		
				Ders (Theoretical)	Uygulama (Tutorial)	Laboratuvar (Laboratory)
BLG460E	8	2	4	2	-	-
Bölüm / Program (Department/Program)		Bilgisayar Mühendisliği Bölümü / Bilgisayar Mühendisliği Programı Department of Computer Engineering / Computer Engineering Program				
Dersin Türü (Course Type)		Seçmeli MT (Optional)		Dersin Dili (Course Language)		İngilizce (English)
Dersin Önkoşulları (Course Prerequisites)		BLG 252E Object Oriented Programming				
Dersin mesleki bileşene katkısı, % (Course Category by Content, %)		Temel Bilim (Basic Sciences)	Temel Mühendislik (Engineering Science)	Mühendislik Tasarım (Engineering Design)	İnsan ve Toplum Bilim (General Education)	
		0	20	80	0	
Dersin İçeriği (Course Description)		Yazılımda güvenliği etkileyen hataların belirlenmesi ve giderilmesi, Yığın taşıma saldırıları, Komut iliştirme saldırıları, Ters mühendislik ve kod perdeleme yöntemleri, İnternet yörelerine yapılan saldırılar, Programlama dillerinde izin ve yetkilerin kullanımı, Temel kriptolojik işlevler ve bunların bilgisayar haberleşmesinde uygulanması, Temel işletim sistemi görevleri ve yazılım güvenliğini etkileyen yanları				
		Determining and mitigating programming mistakes that may affect software security, Stack overflow attacks, Injection attacks, Reverse engineering and code obfuscation, Attacks that target web sites, Handling permissions and authorization in programming languages, Basic cryptologic functions and their usage in computer communications, Basic operating system duties and its effect on software security				
Dersin Amacı (Course Objectives)		1. Bir yazılımın gerçekleşmesi sırasında karşılaşılabilecek güvenlik sorunlarını öğretmek. 2. Bilinen saldırılardan etkilenmeyen yazılımlar geliştirmeyi öğretmek. 3. Uygun programlama alışkanlıkları kazandırılarak yeni saldırılardan olabildiğince az etkilenecek yazılımlar geliştirmeyi öğretmek. 4. Güvenli bir yazılım ortaya koymak için gerekli teknik becerileri ve düşünce yapısını kazandırmak.				
		1. Teaching possible security flaws that may be encountered during software implementation. 2. Students can produce software that does not be affected by known development-time vulnerabilities. 3. Students will gain defensive development style to be less affected by future development-time vulnerabilities. 4. Students will gain the technical abilities to produce secure software.				
Dersin Öğrenme Çıktıları (Course Learning Outcomes)		Bu dersi başarıyla tamamlayan öğrenciler; 1. Bellek sızmalarına dirençli program yazabilirler. 2. Beklenmedik girişlerin güvenliği bozucu etkilerine karşı yazdıkları programları koruyabilirler. 3. Standart dışılıkları düzenleyerek güvenlik sağlama yöntemlerini bilirler. 4. Yazdıkları kaynak kodları perdeleyebilirler. 5. İnternet'te sık karşılaşılan saldırılara karşı önlem alarak kod yazabilirler. 6. İzinlerin ve yetkilendirmenin nasıl kullanılacağını bilirler. 7. Güvenli programlama stili kazanarak olası sorunlara dirençli yazılımlar oluşturabilirler.				
		Students who pass the course will be able to: 1. Write programs who can resist memory overflows. 2. Protect the programs they write against the improbable effects of malicious user input. 3. Know the protection by sanitizing non-standard components. 4. Obfuscate their code. 5. Write programs against well-known Internet vulnerabilities. 6. Know how to use permissions and authorization. 7. Obtain defensive programming style to mitigate future vulnerabilities.				

Ders Kitabı (Textbook)	David A. Wheeler, "Secure Programming HOWTO", 2015.		
Diğer Kaynaklar (Other References)	1. Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, David Svoboda, "Java Coding Guidelines 75 Recommendations for Reliable and Secure Programs", Addison-Wesley, 2014. (ISBN: 978-0321933157) 2. Robert C. Seacord, "Secure Coding in C and C++", 2 nd ed., Addison-Wesley, 2013. (ISBN: 978-0321822130) 3. Jon Erickson, "Hacking: The Art of Exploitation", 2 nd ed., No Starch Press, 2008. (ISBN: 978-1593271442) 4. Michael Howard, David LeBlanc, "Writing Secure Code: Practical Strategies and Proven Techniques for Building Secure Applications in a Networked World", Microsoft Press, 2004. (ISBN: 978-0735617223)		
Ödevler ve Projeler (Homework & Projects)	Dönem içinde dört ödev verilecektir.		
	There will be four homework during the term.		
Laboratuvar Uygulamaları (Laboratory Work)			
Bilgisayar Kullanımı (Computer Use)			
Diğer Uygulamalar (Other Activities)			
Başarı Değerlendirme Sistemi (Assessment Criteria)	Faaliyetler (Activities)	Adedi (Quantity)	Değerlendirmedeki Katkısı, % (Effects on Grading, %)
	Yıl İçi Sınavları (Midterm Exams)	1	40
	Kısa Sınavlar (Quizzes)		
	Ödevler (Homework)	4	20
	Projeler (Projects)		
	Dönem Ödevi/Projesi (Term Paper/Project)		
	Laboratuvar Uygulaması (Laboratory Work)		
	Diğer Uygulamalar (Other Activities)		
	Final Sınavı (Final Exam)	1	40

DERS PLANI

Hafta	Konular	Dersin Çıktıları
1	Güvenliğin temel kavramları	7
2	Yığıcı taşıma saldırıları ve korunma yolları	1, 7
3	Yığıcı taşıma saldırıları ve korunma yolları	1, 7
4	Dinamik bellek yönetimi	1, 7
5	Standart biçimde yazım saldırıları ve korunma yolları	3, 7
6	Komut ilişirme saldırıları ve korunma yolları	2, 7
7	Gerçek saldırılardan örnekler ve çözümler	7
8	Komut ilişirme saldırıları ve korunma yolları	2, 7
9	Ters mühendislik ve kod perdeleme yöntemleri	4
10	Kriptolojinin temelleri, bilgisayar haberleşmesi ilkeleri	5
11	XSS, CSRF saldırıları ve korunma yolları	5
12	Senkronizasyon sorunları	5
13	Güncel programlama dillerinde izinlerin ve yetkilerin kullanımı	6
14	Test ve statik analiz araçları	7

COURSE PLAN

Weeks	Topics	Course Outcomes
1	Basic concepts of security	7
2	Buffer overflow attacks and defenses	1, 7
3	Buffer overflow attacks and defenses	1, 7
4	Dynamic memory management	1, 7
5	Canonicalization attacks and defenses	3, 7
6	Injection attacks and defenses	2, 7
7	Real-world attacks examples and their solutions	7
8	Injection attacks and defenses	2, 7
9	Reverse engineering and obfuscation methods	4
10	Basics of cryptology, principles of computer communication	5
11	XSS & CSRF attacks and defenses	5
12	Race conditions	5
13	Permission and authorization mechanisms in contemporary languages	6
14	Test and static analysis tools	7

Dersin Bilgisayar Mühendisliği Programıyla İlişkisi
(1: “az”, 2: “kısmi”, 3: “Tam”, Eğer cevabınız “Hiçbiri” ise boş bırakınız.)

Bilgisayar Mühendisliği Programı Çıktıları ve Performans Ölçütleri		Katkı Seviyesi		
		1	2	3
a	Matematik, temel bilimler ve mühendislik bilgilerini bilgisayar mühendisliği alanında uygulama becerisi		X	
b	Deney tasarlayıp yürütebilme ve verileri analiz edip yorumlama becerisi	X		
c	Bir sistemi, sistem bileşenini veya süreci; ekonomik, çevresel, sosyal, politik, etik, üretilebilirlik, sürdürülebilirlik, emniyet ve kaza önleme gibi istenilen gereksinimleri karşılayacak şekilde tasarlama becerisi			X
d	Mevcut bir yapıyı veya sistemi eleştirel yaklaşımla gözleme, irdeleme ve sonuçta düzeltme ve iyileştirme becerisi			X
e	Birden çok disiplinden oluşan bir takım çalışması yürütebilme becerisi			
f	Mühendislik problemlerini belirleme, formüle etme ve çözme becerisi		X	
g	Mesleki ve etik sorumlulukları kavrama			X
h	Etkin sözlü ve yazılı iletişim kurabilme becerisi			
i	Mühendislik çözümlerinin küresel, toplumsal ve çevresel boyutlarda etkisini kavramak için gereken geniş kapsamlı bir eğitime sahip olma			X
j	Yaşam boyu öğrenme gereğini algılamış ve kendi kendine öğrenme yeteneğini kazanmış olma		X	
k	Güncel/Çağdaş konulara ilişkin bilgi sahibi olma		X	
l	Mühendislik uygulamaları için gerekli teknikleri, yetenekleri ve modern mühendislik araç ve gereçlerini kullanabilme becerisi			
m	Değişen koşullara uyum sağlama yeteneği		X	

1: Az, 2. Orta, 3. Yüksek

Relationship between the Course and Computer Engineering Curriculum

(1: “Little”, 2: “Partial”, 3: “Full”, Leave blank if your answer is “None”)

Computer Engineering Department Program Outcomes and Performance Criteria		Level of Contribution		
		1	2	3
a	an ability to apply knowledge of mathematics, science, and engineering to the field of computer engineering		X	
b	an ability to design and conduct experiments, as well as to analyze and interpret data	X		
c	an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability			X
d	an ability to observe and examine an existing structure or system in a criticizing attitude and finally correct or enhance it			X
e	an ability to function on multi-disciplinary teams			
f	an ability to identify, formulate, and solve engineering problems		X	
g	an understanding of professional and ethical responsibility			X
h	an ability to communicate effectively			
i	the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental and societal context			X
j	a recognition of the need for, and an ability to engage in life-long learning		X	
k	a knowledge of contemporary issues		X	
l	an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice			
m	an ability to adapt to changing conditions		X	

1: Low, 2. Medium, 3. High

<u>Düzenleyen (Prepared by)</u>	<u>Tarih (Date)</u>	<u>İmza (Signature)</u>
	10 Ekim 2016 10 October 2016	