Web Uygulamaları Güvenliği

BIL342 Programlama Labaratuarı I
1. Deney

Danışman:
Kerem ERZURUMLU

Çağrı TIRAŞ

İçindekiler

Özet

Web Uygulamaları ve Güvenlik İhtiyacı

İnternet hızla gelişmiş, günümüzün bir parçası haline gelmiştir. İnternet uygulamaları, bazı işlemlerin internet üzerinden yapılmasına olanak veren programlardır. Günümüzde bankacılık işlemleri, sağlık hizmetleri(hasta randevu sistemleri, hasta takip sistemleri, vb.), öğrenci işleri, e-ticaret uygulamaları ve benzeri pek çok konuda web uygulamaları insanların hayatlarına girmekte, insan yaşamında vazgeçilmez bir yer edinmektedir. Kuruluşlar açısından bakıldığı zaman ise; internet uygulamaları, internet kullanıcı sayısı göz önüne alınırsa ucuz ve büyük kitlelere ulaşabilen bir tanıtım alanı, daha geniş bir hizmet ağı ve daha az iş gücü benzeri sebeplerle tercih edilmektedir.

İnternet uygulama alanlarının genişliği, hemen her kuruluşun daha büyük kitlelere hitap etmek amacıyla bir şube de internet üzerinde açmalarına olanak sağlamaktadır. İşlemlerin herkesin kullanımına açık olan internet üzerinden yapılması veya daha da geniş anlamda bir bilgisayar ağı üzerinden yapılıyor olması ve tüm kullanıcıların iyi niyetli olmaması, internet uygulamaları ve dolayısıyla kuruluşlar için tehdit bir oluşturmaktadır.

İnternet üzerinde pekçok saldırı programı serbestçe dağılmakta ve hemen her türlü bilgiye ulaşılabildiği bu ortamda pek çok saldırı yönteminin anlatıldığı kaynaklara ulaşmak da pek zor olmamaktadır.

İnternet uygulamalarının hızla artması, internet kullanıcı sayısının artması, bilgiye ulaşımın kolaylığı ve insan doğası(yapmış olma isteği, zarar verme isteği, merak) sebebi ile internet saldırganları ve dolayısıyla internet saldırıları hızla artmaktadır.

Yeterince korunmayan bir sistemin sebep olabileceği zararlar kullanım alanına göre göze alınamayacak büyüklüktedirler. Saldırılar sonucunda, kullanıcılara ait bilgilerin yeterince korunamaması, kaybedilmesi, hizmette aksaklıklar meydana gelmesi ve benzeri durumlar meydana gelebilmektedir. Zarar gören sistemlerin onarılması çok masraflı olabileceği gibi yeterince korunamayan bilgiler de kullanıcılar açısından zarara sebep olabilmektedir. Hizmeti veren kuruluş için kaybedilecek prestij, çoğu zaman aksaklıkları gidermek için harcayacağı paradan daha büyük bir kayıp olacaktır.

Web Uygulamaları

Web Uygulamalarının Genel Özellikleri

Web uygulamaları, kullanıcı ile etkileşimli çalışan programlardır. Kullanıcıya bir takım işlemleri yapma imkanı sunulur, kullanıcı web uygulaması ile kendisine sağlanan bilgi giriş alanlarını kullanarak çeşitli istemlerde bulunur, uygulamayı yönlendirir ve amacına ulaşır.

Web uygulamaları, istemci/sunucu mantığıyla çalışan programlardır. Sunucu sistemler, genellikle belli işlere atanmış güçlü iş istasyonlarıdır(web sunucu, kütük sunucu vb). Web uygulamaları, genel olarak işlemlerin sunucu tarafında yapıldığı uygulamalardır. İstemci

program, kullanıcı için etkileşim arayüzü sağlar. Kullanıcı, sunucu sistemin sağladığı olanakları istemci programı aracılığıyla kullanır.

Web uygulamaları için "platformdan bağımsızdır" ifadesi, genel olarak doğru bir ifadedir. Uygulama yazılımı gerçekleştirilirken kullanılan programlama dilleri yaygın diller olup hemen hemen tüm işletim sistemler için gerekli desteği sağlarlar. Asıl olarak web uygulamaları geliştirmeyi amaçlayan PHP, ASP, JSP gibi programlama dillerinin yanında genel olarak CGI(Common Gateway Interface) olarak adlandırılan C, Perl, Shel, Delphi benzeri programlama dillerinin kullanımı da mümkündür.

CGI (Common Gateway Interface), Web Servisleri ile bu servislerin dışındaki programlar arasında etkileşim (ortak çalışma) platformu oluşturmak için geliştirilmiş bir standarttır. CGI, aslında bir programdır. Web'in statik yapısına, HTML kodu içinden çağrılan CGI programları dinamik bir nitelik kazandırmaktadır.(Acar,2000)

Internet üzerindeki hemen bütün kullanıcı arayüzleri, Common Gateway Interface (CGI), kullanılarak hazırlanmıştır. CGI www kullanıcılarının www sunucusunun çalıştığı makine üzerinde belirlenen programları çalıştırmasını sağlayan bir sistemdir.

CGI'yı geliştirmedeki en büyük amaç Web sunucusu üzerinden sunucu tarafından programlar çalıştırabilmekti. Bu yeni teknolojiye Common Gateway denilmesinin sebeplerinden en önemli üçü şunlardır:

- CGI programları sunucudan bağımsız olmalıydı
- CGI programları hemen her dille yazılabilmeliydi
- Hemen her istemcide çalışabilmeliydi

Dolayısıyla, CGI bir programlama dili değildir. Piyasadaki, bir girdiyi işleyip, çıktı üretebilen her dil CGI programları geliştirmek için kullanılabilir.
CGI programlarında temel fikir, uygulamaların mantık katmanının, sunucu tarafında oluşturulmasıydı.

CGI ile hazırlanan kullanıcı arayüzlerinin başlıca avantajlarını aşağıdaki gibi sıralayabiliriz:

- Kolay ve hızlı hazırlanabilirler. CGI arayüzlerinde, diğer kullanıcı yüzlerindeki birçok çabaya gerek yoktur. Kullanıcıyla ilişkiyi www tarayıcısı (web browser) yaptığından, kullanıcı zaten web tarayıcısı için çoktan hazırlanmış karmaşık kullanıcı arayüzü işlemleriyle uğraşmak zorunda kalmaz.
- İstenilen herhangi bir programlama dili kullanılabilir.
- Kolay kullanılır, kullanıcının alışık olduğu arayüzler. CGI arayüzleri Internet kullanan herkesin alışık olduğu Netscape, Lynx, Mosaic gibi web tarayıcılarından yararlanır. Bu nedenle programda kullanıcının çoktan alışık olduğu bir arayüz kullanılmış olur.
- Değişik ortamlarda çalışabilir. Programınızın asıl çalıştığı sistem web sunucusunun çalıştığı sistem olmasına karşın, programınıza bilgisayar ağına bağlı herhangi bir bilgisayardan ulaşılabilir. Kullanılabilecek hemen hemen bütün işletim sistemleri ve ortamlar için web tarayıcısı bulunabileceği için, yazılan tek program birçok farklı ortamdan kullanılabilir.
- Dağıtık ortamlarda kullanılabilme olasılığı. Programın kullanıcı arayüzü (web tarayıcısı) başka bir ortamda, web html formları başka bir sistemde, CGI programları başka bir sistemde olabilir. (Stanek, 2000)

Bu avantajların yanında sayılabilecek bir kac dezavantaj,

- Dikkatsiz yazılmış CGI programları güvenlik açıklarına neden olabilirler.
- Kullanıcı arayüzleri çok kolay ve hızlı oluşturulmalarına karşın, web tarayıcısının yetenekleriyle sınırlıdır.(Stanek, 2000)

CGI programcıklarını (script) kullanarak okuyucu ile gerçek bir etkileşim içinde güçlü, kişisel ve profesyonel Web yayımları yaratılabilir. CGI programcıkları Web sunucusu ile diğer uygulamalar arasında bir ağgeçit gibi davranan dış kaynaklı programlardır. CGI programcıklarını okuyuculardan gelen girişleri işlemek için kullanabilir, böylelikle okuyucu ile çift yönlü bir iletişim yolu kurulabilir. Okuyucu girişleri doldurma formlarının verileri şeklinde olabildiği gibi veri tabanı sorgulamaları için anahtar sözcükler ya da okuyucunun inceleyicisinin ya da bağlantının tanımladığı değerler şeklinde de olabilir.
CGI programcıkları bu girişleri bir indekse veri olarak eklemek, veri tabanında sorgulama yapmak, kişisel dokümanlar yaratmak için kullanılabilir. CGI programcıklarının en mükemmel özelliği karmaşıklıklarını kullanıcıdan gizlemeleridir. Web üzerinde bir doldurma formunu ya da bir haritayı kullanırken ne olduğunu bilmeseniz de bir ağgeçit programcığı kullanıyor olabilirsiniz. Bu nedenle her şey otomatik yapılıyor gibi görünür. Veriyi girin, farenin düğmesine basın ve bir an sonra sonuç görüntülensin.

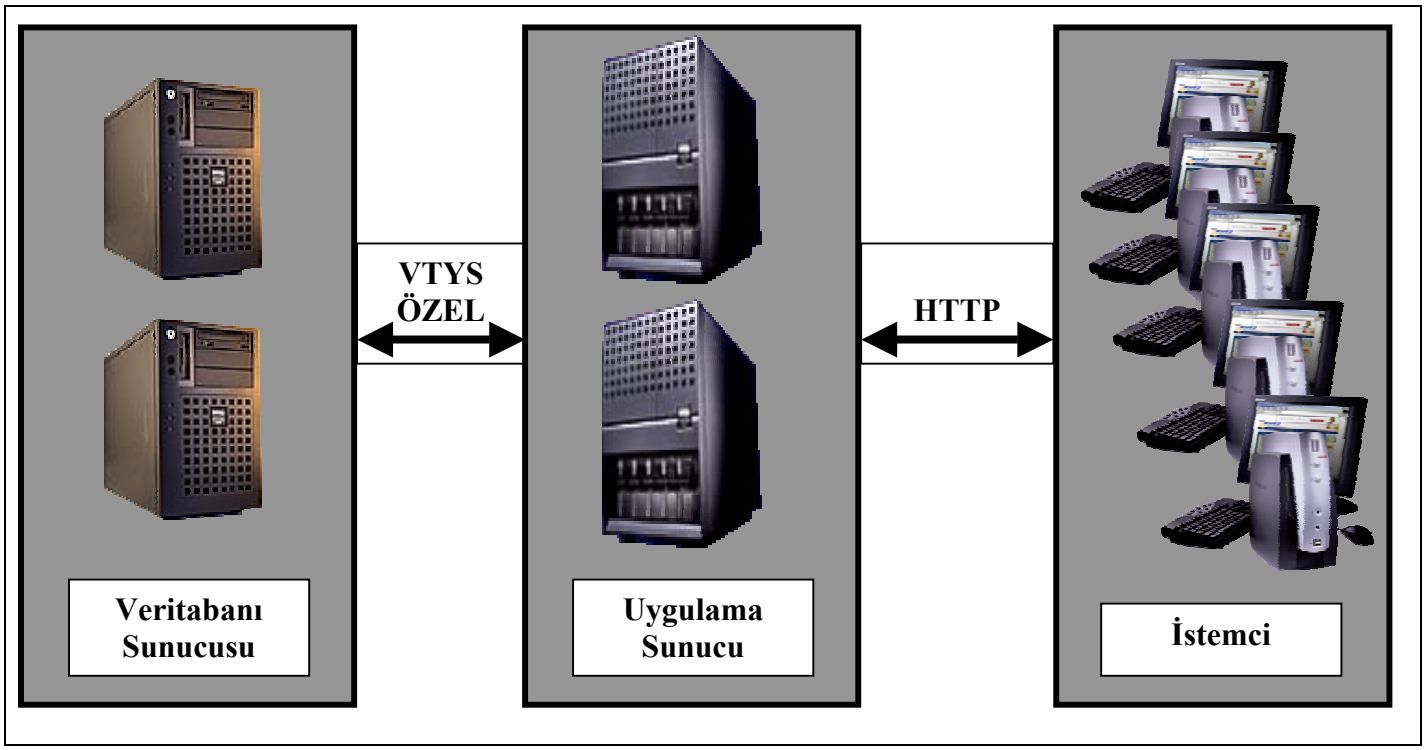
Web Uygulama Mimarisi

Web uygulamaları, üç katmanlı mimariler için iyi bir örnektir. Kullanıcı arayüzü(web tarayıcısı), web sunucu(web uygulaması), veritabanı sunucusu bileşenlerinden oluşur.

Veritabanı sunusu sadece veritabanı bilgilerine sahiptir. Uygulama mantığı ile ilgili detaylardan arındırılmıştır ve bu konuda VTYS dili kodlaması gereksizdir. Uygulama sunucu ile VTYS' ne özel bir protokol ile iletişim kurar.

Web sunucusu web uygulamasını taşıyan kesimdir. Kullanıcı arayüzü tarafından kendisine iletilen istekleri yorumlar, veritabanı sunucusu ile iletişim kurar ve kullanıcı isteğini karşılar. Uygulama mantığı, süreç denetimleri tamamen bu katmandadır ve uygulama mantığı bir servise dönüşür. Uygulama sunucu, uygulama mantığını tamamen kendi üzerinde taşıdığı için uygulama bakımı ve değişiklik yönetimi kolaydır. Uygulama sunucu üzerinde yapılan değişiklikler, istemci üzerinde bir değişiklik gerektirmez (istemci, uygulama sunucudaki değişikliklerden etkilenmez). Uygulama sunucu (web sunucu), kullanıcı arayüzü(web tarayıcısı) ile HTTP protokolünü kullanarak iletişim kurar.

İstemci sunucu arasında, gerekli durumlarda SSL (Secure Socket Layer) kullanılarak, güvenli iletişim sağlanabilir. SSL network üzerindeki bilgi transferi sırasında, güvenliğin ve gizliliğin sağlanması amacıyla Netscape tarafından geliştirilmiş bir güvenlik protokolüdür. SSL güvenlik alanında gönderdiğiniz bilgiler, çok sıradan görünen, fakat çözülmesi gerçekte imkansız olan bir kodlama sistemiyle şifrelenir. Girdiğiniz bilgiler önce şifrelenmiş ve diğer kişiler tarafından okunması imkansız bir bilgiye dönüştürülür. Bu bilgi ancak diğer uçtaki web sunucu tarafından çözülüp anlamlı bir bilgi haline getirilebilir.



Şekil 1 Üç Katmanlı Mimari

Web İstemcileri

Web uygulamalarında, üç katmanlı mimarinin son kesimi olan kullanıcı arayüzü için, istemci yeteneklerine göre sınıflandırma yapılabilir.

Alabildiğince hafif istemciler (ultra thin client), "herker bağlansın, herkes kullansın" mantığıyla çalışırlar. Kullanıcı arayüzü HTML kullanılarak hazırlanır. Kullanıcılara daha kullanışlı bir arayüz sağlayabilmek amacıyla JavaScript benzeri script dilleri kullanılarak işlevsellik kazandırılabilir. İstemci olarak, HTML desteği olan bir internet tarayıcısı (web browser) yeterlidir.

Microsoft istemcileri, alabildiğince hafif istemcilere göre daha gelişmiş bir kullanıcı arayüzü sağlarlar. HTML sayfaları içerisine, özel başlıklar (header) kullanılarak ActiveX nesnelerinin gömülmesi ile oluşturulurlar. ActiveX nesneleri kullanılarak, kullanıcı işletim sisteminin de (Microsoft Windows) bazı özelliklerinden faydalanılarak iş yükünün bir kısmı istemciye aktarılabilir. Kullanıcı için HTML ve ActiveX desteği olan bir istemci ve daha güçlü bir donanım gereklidir.

Java istemcileri, başlı başına bir Java uygulaması olabileceği gibi, HTML kodu ile taşınan Java appletleri de olabilir. Kullanıcı diğer seçeneklere göre daha güçlü bir donanım ihtiyacı duyar.

Web Uygulamaları Güvenliğinin Sağlanması

Web uygulamalarının güvenliği, yalnızca uygulama yazılımında alınacak tedbirlerle sağlanamamaktadır. Güvenliğin sağlanması için ağ güvenliği, sunucu işletim sistemi güvenliği, web sunucu yazılımı güvenliği ve uygulama güvenliği bir bütün olarak düşünülmelidir. Bu mekanizma bir pramit yapısı gibi düşünülebilir, herhangi birinde güvenlik sağlanamaz ise hiçbirinde sağlanamaz.

Ağ Güvenliğinin Sağlanması

Ağ güvenliğinin sağlanması için, ağın uygun bir şekilde bölümlendirilmesi ve güvenlik duvarı (firewall) ile ağın bölümleri arasında yalıtımın gerçekleştirilmesi gereklidir.

Sunucu bilgisayar sistemleri genellikle belirli bir işe adanmış sistemlerdir. Web sunucu bilgisayar sistemine HTTP dışındaki isteklerin ulaşması engellenmelidir.

Güncelleme için yalnızca kısıtlı sayıda güvenilen bilgisayar sistemine izin verilmelidir. Web sunucuya yönelik herhangi bir saldırı, yerel ağ dışından doğrudan saldıran bir saldırgan, kötü niyetli bir yerel ağ kullanıcısı veya yeterince korunamamış bir yerel ağ bilgisayarını basamak olarak kullanan bir saldırgan tarafından gerçekleştirilebilir.

Sunucu İşletim Sistemi Güvenliğinin Sağlanması

İşletim sistemleri sürekli olarak geliştirilmektedir. Uygulama sunucuları için seçilen işletim sisteminin ve kullanılan yazılımların güncel sürümleri kullanılmalıdır. Güncel sürümler, bazı hataları düzeltilmiş, güvenlik açıkları (pek çoğu) kapatılmış, daha güvenilir sistemlerdir.

Yazılım için bazı açıklar tespit edilmiş, fakat yeni bir sürüm çıkarılması düşünülmüyor veya karşılan açıklar, yeni bir sürüm çıkarılmasını bekleyemeyecek kadar önemli olabilirler. Bu gibi durumlarda ilgili açığı kapatmak amacıyla yama programları çıkartılır. İşletim sistemi güvenliği için mevcut yazılımların mevcut yamalarının uygulanması önemli bir etkendir.

Bilgisayar işletim sisteminin güvenliğini tehdit eden bir diğer unsur da verdiği servislerdir. Sunucu bilgisayar sistemi üzerinde verilen bir serviste meydana gelebilecek (bulunacak) bir açık sunucu işletim sistemi ve dolayısıyla verilen diğer servisleri de tehlikeye atacaktır. Bu sebeple çalışması şart olmayan ağ hizmetleri durdurulmalıdır.

Çok kullanıcılı sistemlerde, kullanıcılara görevlerini gerçekleştirebilmeleri amacıyla çeşitli yetkiler verilir. Kullanıcı parolaları bu yetkileri kullanmak, sisteme girmek için bir anahtardır. Bu sebeple parolaların güvenliği düzenli olarak denetlenmelidir.

Sunucu sistem üzerinde işlemler, bir sunucu yazılım veya bir sistem kullanıcısı tarafından gerçekleştirilir. Kötü niyetli bir kullanıcı, bir kullanıcı şifresi ele geçirmiş veya sunucu yazılımın bir açığından faydalanan kötü niyetli bir saldırgan, kullanıcı veya sunucu yazılımın yetkili olduğu her işi yapabilir. Bu sebeple kullanıcılara ve sunucu yazılımlara sadece yapması gereken işlemleri yapabileceği kadar yetki verilmelidir.

Sunucu yazılımlar, genellikle kendilerini çalıştıran kullanıcıların yetkilerine sahiptirler. Bu sebeple sistem üzerinde tam yetkili olması gerekmeyen sunucu yazılımları yetkili kullanıcı tarafından çalıştırılmak yerine, bu hizmet için tanımlanmış, yetkileri sınırlandırılmış, sıradan kullanıcılar tarafından çalıştırılmalıdır. Web sunucu için 'root' yerine 'nobody' veya veritabanı sunucusu için 'Administrator' yerine 'oracle' kullanılabilir.

Bazı sunucu yazılımları, öntanımlı kullanıcılara sahiptir ve bu kullanıcılar için yine öntanımlı parolalarla kurulurlar. Bu parolalar değiştirilmeli ve güvenliğinin denetlenmelidir. Örnek olarak veritabanı yönetim sistemi ORACLE için tam yetkili 'oracle' kullanıcısı ve 'system' parolası verilebilir.

Web Sunucu Yazılımı Güvenliği

Web sunucu yazılımı, web uygulamalarını çalıştıran, kullanıcı isteklerine cevap veren yazılımdır. Web sunucu yazılımları, kullanıcılara bir dizi örnek uygulamalar sunarlar. Bu uygulamalar, kullanıcı hizmetlerinin nasıl yapıldığı, sistemin nasıl kullanıldığına yönelik bilgiler içerirler. Web sunucu yazılımı ile beraber herkese sunulan bu örnek uygulamalar, kurulum sırasında yerleştirildikleri yerlerin bilinebiliyor olması, kaynak kodlarının incelenebilir olması benzeri sebeplerle web sunucu için risk oluştururlar. Web sunucu yazılımı ile birlikte gelen örnek uygulamaların kaldırılması gereklidir.

Web sunucu, belirli bir küme dosya türü dışındaki dosyaları işlememelidir. Web sunucu üzerinden sunulabilecek dosya türleri, ".html", ".asp", ".php", ".jsp", ".txt" benzeri web hizmetine yönelik dosya türleridir. Web sunucu yazılımının, çalıştırması gerekmeyen sistem dosyalarını çalıştırması, kullanıcı parola dosyaları ve benzeri şekillerde güvenliği tehdit edecek işlemler yapması engellenmelidir.

Web sunucu yazılımları, basit bir tanımla kullanıcıların istedikleri dosyaların içeriklerini onlara sunan yazılımlardır. Kullanıcı, doğrudan bir dosya istememiş ve bir dizin belirtmiş ise, web sunucu yazılımı, kullanıcıya bu dizin içeriğini listeler.Bu durum ise, web uygulamasının kullanıcı ile doğrudan bağlantıya izin vermediği, bir işlem sırasında parametre geçirip kendisine dönen değerleri kısıtlayarak kullanıcıya sunduğu veya bir takım kullanıcı tanımlama işlemleri sonucunda kullanımına izin verdiği alt uygulamaların sıradan kullanıcılara görüntülenmesine izin verir. Kullanıcıların yetkileri olmayan işlemlere ulaşmalarının engellenmesi, saldırganların saldırabileceği hedef sayısının azaltılması gibi sebeplerle, web sunucu yazılımının dizin listesi gösterimi iptal edilmelidir.

Uygulama Yazılımlarının Güvenliği

Web uygulama yazılımlarının güvenliği için "kullanıcıların tümü art-niyetlidir" yaklaşımı benimsenmelidir. Bu yaklaşım, kullanıcı arayüzünde HTML gibi sade bir dil kullanılan hafif istemci uygulamalarında, kullanıcıdan gelebilecek beklenmedik bir bilginin, uygulama çalışmasını aksatmasını maksimum seviyede engelleyecektir.

HTML kullanılarak tasarlanmış kullanıcı arayüzleri, kullanıcı tarafından kolaylıkla değiştirilebilen ve sunucuya değiştirilmiş halleriyle yönlendirilebilirler. JavaScript

programcıkları kullanılarak, kullanıcı tarafında yapılan form denetimleri, web tarayıcı JavaScript desteği kapatılarak tamamen engellenebilir. Kullanıcı tarafından girilen her türlü bilginin kullanıma uygunluk(uzunluk, tür, içerik) denetimi uygulama yazılımı içerisinde, sunucu tarafında yapılmalıdır.

Yoğun içerikli bir kullanıcı arayüzünde, kullanıcı hatasının web sunucusunda belirlenip, kullanıcıya hata mesajı bildirilmesi, kullanıcı açısından vakit kaybına yol açacaktır. JavaScript benzeri dillerle kullanıcı tarafında yapılan denetimler kullanıcının yaptığı bir hatayı anında farketmesi için kullanılmalıdır. Yine de bu kontrollerin kullanıcı tarafından iptal edilebileceği unutulmamalı, kullanıcıdan gelen tüm bilgiler uygulama içerisinde kontrol edilmelidir.

Web uygulamaları, HTTP protokolu gereği, kullanıcı tarafından girilen bilgileri, kullanıcı arayüzü – uygulama sunucu arasında metin olarak taşırlar. Uygulama kendisine metin olarak gelen parametreleri, kendi içinde kullanılan tanım gereği sayısal veya mantıksal bir ifade olarak kullanabileceği gibi yine bir metin değeri olarak da kullanabilir. Sayısal olmayan bir değerin, sayısal bir ifade gibi kullanılması elbette ki uygulama yazılımının çalışmasını olumsuz etkileyecektir.

Web uygulamalarında, işlemlerin yapılışı sırasında, istemci ve sunucu arasındaki bağlantı sürekli değildir. Kullanıcının bir istekte bulunması sonucu, web tarayıcısı, uygulama sunucu ile iletişime geçer ve kullanıcı isteğini iletir. Uygulama sunucu, bu istek karşılığı yapılması gereken işlemleri yapar ve web tarayıcısına bir cevap üreterek bağlantıyı keser. Birden çok adımdan oluşan bir işlemin gerçekleştirilmesi sırasında ise kullanıcıyı tanımlayan, kullanıcının hangi işlemin hangi adımında olduğunu ve dolayısıyla hangi işlemin yapılmasının gerektiğini gösteren bir takım bilgilerin uygulama tarafından bilinmesi gereklidir. Kullanıcı tanımlama ve işlem adımlarının denetlenmesi, oturum yönetimi olarak adlandırılır.

Kullanıcı, sisteme kendisini tanıtarak oturumu başlatır. Bu aşamadan itibaren, yapılan her işlem için bu kullanıcı sorumludur. Yapabileceği işlemler ve bu işlemlerin alt adımları belirlidir. Oturum yönetimi, kullanıcının kendisini tanıtmadan sisteme girmesini, yetkisi olmayan işlemleri gerçekleştirmesini veya çok adımlı bir işlemin bazı alt adımlarını atlamasını engelleyecek şekilde, dikkatli programlanmalıdır.

Oturum yönetimi, iş akışını doğru bir şekilde denetleyebilmek için kullanıcı ve yapılan işleme ait birtakım bilgiyi oturum süresince veya gerektiği kadar saklamak zorundadır. Bu bilgiler ya çerezler (cookie) vasıtasıyla kullanıcı tarafında veya sunucu tarafında tutulur. Kullanılması zorunluluk olmadığı durumlarda, çerezler kesinlikle kullanılmamalıdır.Kullanıcı tarafında tutulan çerezler, kullanıcı tarafından değiştirilebilir özelliktedirler. Kullanıcı bu çerezlerde yapabileceği değişikliklerle, başka bir kullanıcı kimliğine geçmek, bazı işlem adımlarının çevresinden dolaşmak gibi oturum yönetimini bozan eylemlerde bulunabilir. Bu sebeple, çerezler içerisinde yalnızca kullanıcıyı tanımlayacak bilgi girilmeli, diğer tüm bilgiler sunucu üzerinde tutulmalıdır. Kullanıcıyı tanımlayan bilginin de değiştirilmesi mümkün olduğu için bu bilginin “artıklığı” yüksek olmalı, tahmin yoluyla başka kullanıcıları tanımlayan bilgilere ulaşılamamalıdır.

Yığıt Taşırma (Buffer Overflow)

Bir uygulama yazılımında tanımlanmış küçük bir alana büyük bir veri yüklemeye çalışırsanız ne olur? Uygulama çakılır veya bazı durumlarda, uygulamanın 'istenmedik' komutları işletmesi sağlanabilir. Bu durum aşağıdaki örnekle açıklanabilir.

```
void dikkatsiz(char *param) {
    char hataKaynagi[16];
    strcpy(hataKaynagi,param);
    ...
}
```

Yukarıdaki dikkatsiz yordamı "/cgi-bin/swc" programı tarafından "dikkatsiz(crt)" şeklinde kullanılmaktadır. Fakat dikkatsiz yordamı, uzunluğunu kontrol etmediği "param" parametresini 16 karakter uzunluğundaki yerel değişkeni "hataKaynagi"na kopyalamaktadır.

Kullanıcı tarafından, bu uygulamanın aşağıdaki şekilde kullanımı bir yığıt taşırma saldırısıdır. Uygulama yazılımı içerisinde, kullanıcıdan alınan bilgilerin uzunluk denetimlerinin eksikliğinden faydalanılır.

"http://kurban.edu.tr/cgi-bin/swc?ctr=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

Uygulama çalıştırılırken, bir yordama sapılması durumunda, sistem tarafından bu yordama bir çalışma alanı ayrılır. Bu alan içerisinde, belleğin başından sonuna doğru sırasıyla, yordam yerel değişkenleri, amaç programı ve yığıt alanı bulunur. Yordam yerel değişkenleri ve amaç programı boyutları belirli bellek alanlarıdır. Yordam çağırma sırasında, yığıt alanına önce yordam parametreleri daha sonra da yordam bitiminde işlemin devam edeceği dönüş adresi yerleştirilir.

Belleğin Başı **Belleğin Sonu**

Yereller	Sçg	Dönüş @	Param

Yığıtın Üstü **Yığıtın Altı**

Şekil 2 Yığıt Düzeni

Yukarıdaki örnekte, "hataKaynagi" değişkenine, tanımı gereği 16 karakterlik yer ayrılmıştır. Fakat kullanıcı tarafından gönderilen büyük boyutlu parametre, bu değişkenin kendisine tanınan alanın dışına taşmasına sebep olacaktır. Kullanıcı bu parametrenin uzunluğunu sürekli olarak arttırarak uygulamanın çakılmasını sağlamayı amaçlamaktadır. Bu örnekte uygulamanın çakıldığı yer kullanıcı parametresinin "strcpy()" komutundan sonraki işletilecek komutu bozduğu nokta olacaktır.

Bu seviyeden sonra, saldırgan hedef değiştirecek, bu noktaya anlamsız ifadeler yazmak yerine, sunucu bilgisayar sistemi tarafından çalıştırılabilecek komutlar(sunucu sistem için

derlenmiş assembler kodlar) göndererek yapmak istediği işlemleri gerçekleştirmeye çalışacaktır.

Veri Tabanı ve Kabuk(shell) Sızmaları

Kullanıcıdan alınan bilgilerin, içerik kontrollerindeki eksikliklerden faydalanmaya dayalı bir saldırıdır.

“http://kurban.com/scripts/..%c0%9v..%c0%9v..%c0%9v..%c0%9v..%c0%9v../winnt/system32/cmd.exe?/c+dir+c:\”
“http://kurban.com.tr/cgi-bin/simple/view_page?mv_arg=|cat%20/etc/passwd|”

Kullanımı kabuk sızmasına güzel bir örnek olduğu gibi, aynı zamanda sisteme kurulan örnek uygulamaların nasıl tehlikeler oluşturabileceğine de bir örnektir.

“http://kurban.com/xsql/demo/adhocsql/query.xsql?sql=select%20username%20from%20ALL_USERS”

Kullanımı ise veri tabanı sızmasına güzel bir örnektir.

Güvenliğin İzlenmesi ve Saldırı Tespiti

Bilindik saldırıları tespit etmek ve kaydını tutmak üzere ağ temelli saldırı tespit sistemleri kullanılabilir. Bu sistemler, bilindik saldırılara ait bir veritabanına sahiptirler ve ağ trafiğini izleyerek bu saldırıları tespit ederler, engellerler ve kaydederler. Aşağıda bir saldırı tespit sistemi tarafından kaydedilen saldırı örneklenmektedir.

saldirgan1.saldiran.edu.tr - - [27/Nov/2001:18:13:28 +0200] "GET /search.dll?query=%00&logic=AND HTTP/1.1" 404 216 "-"

Saldırganların sisteme sızmayı başarması durumunda ilk yapacakları, sonraki girişleri için arka-kapı açmak olacaktır. Dosya bütünlük denetleyicileri ile dosya ve dizinlerdeki beklenmedik değişiklikler tespit edilebilir.

Saldırı tespitinde kullanılan saldırı tespit sistemleri, dosya bütünlük denetleyicileri yanında bir diğer saldırı tespit kaynağı da Web sunucu kayıtlarıdır. Web sunucu kayıtlarındaki anormallikler izlenmeli belirlenen olumsuzluklar giderilmelidir.

Sonuç

Web uygulamalarının güvenliğinin sağlanması için uygulanabilecek birkaç temel adım pek çok saldırıyı engelleyecektir.

- Ağ güvenliğinin, sistem güvenliğinin, sunucu yazılımı güvenliğinin sağlanması
- Uygulama geliştirme sürecinde güvenlik boyutunun düşünülmesi/değerlendirilmesi
- Sistem ve sunucu kayıtlarının düzenli incelenmesi ve problemlerin analizi