Test Network Infrastructure Configuration (Ağ Altyapısı Yapılandırmasını Test Edin)

Summary (Özet)

Yüzlerce web uygulamasını içerebilen birbirine bağlı ve heterojen web sunucusu altyapısının içsel karmaşıklığı, yapılandırma yönetimini yapar ve her bir uygulamayı test etme ve dağıtmada temel bir adımdır. Tüm altyapının güvenliğini baltalamak için sadece tek bir güvenlik açığı gerekir ve küçük ve görünüşte önemsiz sorunlar bile aynı sunucudaki başka bir uygulama için ciddi risklere dönüşebilir. Bu sorunları çözmek için, tüm mimariyi haritaladıktan sonra yapılandırma ve bilinen güvenlik sorunlarının derinlemesine bir incelemesini yapmak son derece önemlidir.

Uygulamanın güvenliğini korumak için web sunucusu altyapısının doğru yapılandırma yönetimi çok önemlidir. Web sunucusu yazılımı, arka uç veritabanı sunucuları veya kimlik doğrulama sunucuları gibi öğeler uygun şekilde incelenmez ve güvenli değilse, istenmeyen riskler getirebilir veya uygulamanın kendisini tehlikeye atabilecek yeni güvenlik açıkları getirebilir.

Örneğin, uzaktan bir saldırganın uygulamanın kendisinin kaynak kodunu (hem web sunucularında veya uygulama sunucularında da birkaç kez ortaya çıkan bir güvenlik açığı) açıklamasını sağlayacak bir web sunucusu güvenlik açığı, anonim kullanıcılar uygulama veya kullanıcılarına karşı saldırılar düzenlemek için kaynak kodunda açıklanan bilgileri kullanabileceğinden, uygulamayı veya kullanıcılarına karşı saldırılardan yararlanmak için uygulamayı tehlikeye atabilir.

yapılandırma yönetimi altyapısını test etmek için aşağıdaki adımların atılması gerekir:

- Altyapıyı oluşturan farklı unsurların, bir web uygulamasıyla nasıl etkileşime girdiklerini ve güvenliğini nasıl etkilediklerini anlamak için belirlenmelidir.
- Altyapının tüm unsurlarının, bilinen herhangi bir güvenlik açığı içermediğinden emin olmak için gözden geçirilmesi gerekir.

- Tüm farklı unsurları korumak için kullanılan idari araçlardan bir inceleme yapılmalıdır.
- Kimlik doğrulama sistemlerinin, uygulamanın ihtiyaçlarına hizmet etmelerini ve erişimden yararlanmak için harici kullanıcılar tarafından manipüle edilemeyeceğinden emin olmak için gözden geçirilmesi gerekir.
- Uygulama için gerekli olan tanımlanmış portların bir listesi sürdürülmeli ve kontrol altında tutulmalıdır.

Altyapıyı oluşturan farklı unsurları haritaladıktan sonra (harita Ağı ve Uygulama Mimarisi'ne bakın) kurulduktan sonra, kurulan her bir öğenin yapılandırılmasını gözden geçirmek ve bilinen herhangi bir güvenlik açığı için test etmek mümkündür.

Test Objectives (Test Hedefleri)

- Uygulamaların ağ boyunca belirlenen yapılandırmalarını gözden geçirin ve savunmasız olmadıklarını doğrulayın.
- Kullanılan çerçevelerin ve sistemlerin güvenli olduğunu ve bakımsız yazılım veya varsayılan ayarları ve kimlik bilgileri nedeniyle bilinen güvenlik açıklarına duyarlı olmadığını doğrulayın.

How to Test (Nasıl Test Edilir)

Known Server Vulnerabilities (Bilinen Sunucu Güvenlik Açıkları)

Uygulama mimarisinin farklı alanlarında bulunan, web sunucusunda veya arka uç veritabanında bulunan güvenlik açıkları, uygulamanın kendisini ciddi şekilde tehlikeye atabilir. Örneğin, uzaktan, doğrulanmamış bir kullanıcının web sunucusuna dosya yüklemesine veya hatta dosyaları değiştirmesine izin veren bir sunucu güvenlik açığını düşünün. Bu güvenlik açığı, uygulamayı tehlikeye atabilir, çünkü sahte bir kullanıcı uygulamanın kendisini değiştirebilir veya arka uç sunucuları etkileyecek kod tanıtabilir, çünkü uygulama kodu diğer uygulamalar gibi çalıştırılır.

Testin kör bir penetrasyon testi ile yapılması gerekiyorsa sunucu güvenlik açıklarını gözden geçirmek zor olabilir. Bu durumlarda, güvenlik açıklarının, tipik olarak otomatik bir araç kullanılarak uzak bir siteden test edilmesi gerekir. Bununla birlikte, bazı güvenlik açıkları için test, web sunucusunda öngörülemeyen sonuçlar

doğurabilir ve diğerleri için test (hizmet saldırılarını reddetmekte doğrudan yer alanlar gibi) test etmek, testin başarılı olması durumunda hizmet kesintisi nedeniyle mümkün olmayabilir.

Bazı otomatik araçlar, alınan web sunucusu sürümüne dayalı güvenlik açıklarını işaretleyecektir. Bu hem yanlış pozitiflere hem de yanlış negatiflere yol açar. Bir yandan, web sunucusu sürümü yerel site yöneticisi tarafından kaldırılmış veya gizlenmişse, tarama aracı, sunucuyu savunmasız olarak işaretlemeyecektir. Öte yandan, yazılımı sağlayan satıcı, güvenlik açıkları sabit olduğunda web sunucusu sürümünü güncellemezse, tarama aracı mevcut olmayan güvenlik açıklarını işaretler. İkinci durum aslında bazı işletim sistemi satıcılarının, işletim sisteminde sağladıkları yazılıma güvenlik açıklarının liman yamalarını geri almasıyla çok yaygındır, ancak en son yazılım sürümüne tam bir yükleme yapmazlar. Bu, Debian, Red Hat veya SuSE gibi GNU/Linux dağıtımlarında olur. Çoğu durumda, bir uygulama mimarisinin güvenlik açığı taraması, yalnızca mimarinin “maruz kalan” unsurlarıyla (web sunucusu gibi) ilişkili güvenlik açıklarını bulacaktır ve genellikle kimlik doğrulaması arka uçları, arka uçlar veya kullanılan ters proxyler gibi doğrudan maruz kalmayan unsurlarla ilgili güvenlik açıklarını bulamaz.

Son olarak, tüm yazılım satıcıları güvenlik açıklarını kamuya açık bir şekilde açıklamaz ve bu nedenle bu zayıflıklar kamuya açık bir şekilde kaydedilmez [2]. Bu bilgiler yalnızca müşterilere açıklanır veya eşlik eden tavsiyeleri olmayan düzeltmeler yoluyla yayınlanır. Bu, güvenlik açığı tarama araçlarının kullanışlılığını azaltır. Tipik olarak, bu araçların güvenlik açığı kapsamı ortak ürünler için çok iyi olacaktır (Apa Apache web sunucusu, Microsoft'un İnternet Bilgi Sunucusu veya IBM'in Lotus Domino'su gibi) ancak daha az bilinen ürünlerden yoksun olacaktır.

Bu nedenle, test cihazına kullanılan yazılımların dahili bilgileri ve yazılıma uygulanan sürümler ve ürünler de dahil olmak üzere kullanılan yazılımların iç bilgileri sağlandığında açık açıkları gözden geçirmek en iyi şekilde yapılır. Bu bilgilerle, test cihazı satıcının kendisinden bilgileri alabilir ve mimaride hangi güvenlik açıklarının mevcut olabileceğini ve uygulamanın kendisini nasıl etkileyebileceğini analiz edebilir. Mümkün olduğunda, bu güvenlik açıkları gerçek etkilerini belirlemek ve başarılı sömürü olasılığını azaltabilecek veya reddedebilecek herhangi bir dış unsur (görmü izinsiz giriş algılama veya önleme sistemleri gibi) olup olmadığını tespit etmek için test edilebilir. Testçiler, bir yapılandırma incelemesi yoluyla, güvenlik açığının mevcut olmadığını bile belirleyebilir, çünkü kullanımda olmayan bir yazılım bileşenini etkiler.

Satıcıların bazen güvenlik açıklarını sessizce düzelteceklerini ve düzeltmeleri yeni yazılım sürümleriyle kullanıma sunacağını belirtmek de yararlıdır. Farklı satıcılar, eski sürümler için sağlayabilecekleri desteği belirleyen farklı serbest bırakma döngülerine sahip olacaktır. Mimari tarafından kullanılan yazılım sürümlerinin ayrıntılı bilgisine sahip bir test cihazı, kısa vadede desteklenemeyen veya zaten desteklenmeyen eski yazılım sürümlerinin kullanımıyla ilgili riski analiz edebilir. Bu kritik öneme sahiptir, çünkü artık desteklenmeyen eski bir yazılım sürümünde bir güvenlik açığı ortaya çıkarsa, sistem personeli bunun doğrudan farkında olmayabilir. Hiçbir yama mevcut olmayacak ve tavsiyeler bu sürümü artık desteklenmediği kadar savunmasız olarak listelemeyebilir. Güvenlik açığının mevcut olduğunun ve sistemin savunmasız olduğunun farkında olmaları durumunda bile, uygulama mimarisinde önemli bir kesinti süresi getirebilecek veya en son yazılım sürümüne uyumsuzluklar nedeniyle uygulamayı yeniden kodlanmaya zorlayabilecek yeni bir yazılım sürümüne tam bir yükseltme yapmaları gerekecektir.

Administrative Tools (İdari Araçlar)

Herhangi bir web sunucusu altyapısı, uygulama tarafından kullanılan bilgileri korumak ve güncellemek için idari araçların varlığını gerektirir. Bu bilgiler statik içeriği (web sayfaları, grafik dosyalar), uygulama kaynak kodu, kullanıcı kimlik doğrulama veritabanları vb. İdari araçlar, kullanılan siteye, teknolojiye veya yazılıma bağlı olarak değişecektir. Örneğin, bazı web sunucuları, kendileri olan ve web sunucuları (iPlanet web sunucusu gibi) olan idari arayüzler kullanılarak yönetilir veya düz metin yapılandırma dosyaları (Apa Apache durumunda [3]) tarafından yönetilir veya işletim sistemi GUI araçlarını (Microsoft'un IIS sunucusunu veya ASP.Net'i kullanırken) kullanır.

Çoğu durumda sunucu yapılandırması, FTP sunucuları, WebDAV, ağ dosya sistemleri (NFS, CIFS) veya diğer mekanizmalar aracılığıyla yönetilen web sunucusu tarafından kullanılan farklı dosya bakım araçları kullanılarak ele alınacaktır. Açıkçası, uygulama mimarisini oluşturan elemanların işletim sistemi diğer araçlar kullanılarak da yönetilecektir. Uygulamalar, uygulama verilerinin kendisini (kullanıcılar, içerik vb.) yönetmek için kullanılan onlara gömülü idari arayüzlere de sahip olabilir.

Mimarinin farklı kısımlarını yönetmek için kullanılan idari arayüzleri haritalandırdıktan sonra, onları gözden geçirmek önemlidir, çünkü bir saldırgan

herhangi birine erişirse, uygulama mimarisinden ödün verebilir veya zarar verebilir. Bunu yapmak için şu şekildedir:

- Bu arayüzlere erişimi kontrol eden mekanizmaları ve bunların ilişkili duyarlılıklarını belirleyin. Bu bilgiler online olarak kullanılabilir.
- Varsayılan kullanıcı adı ve şifreyi değiştirin.

Bazı şirketler web sunucu uygulamalarının tüm yönlerini yönetmemeyi seçer, ancak web uygulaması tarafından teslim edilen içeriği yöneten diğer taraflara sahip olabilir. Bu harici şirket ya yalnızca içeriğin bir kısmını (haber güncellemeleri veya promosyonlar) sağlayabilir veya web sunucusunu tamamen (içerik ve kod dahil) yönetebilir. Bu durumlarda İnternet'ten mevcut olan idari arayüzleri bulmak yaygındır, çünkü İnternet'i kullanmak, harici şirketi yalnızca yönetime yönelik bir arayüz aracılığıyla uygulama altyapısına bağlayacak özel bir hat sağlamaktan daha ucuzdur. Bu durumda, idari arayüzlerin saldırılara karşı savunmasız olup olmadığını test etmek çok önemlidir.

References (Referanslar)

- [1] Tivoli Kimlik Doğrulama Müdürü olarak da bilinen WebSEL, Tivoli çerçevesinin bir parçası olan IBM'den bir ters proxy'dir.
- [2] Symantec'in Bugtraq, ISS'nin X-Force veya NIST'in Ulusal Güvenlik Açığı Veritabanı (NVD) gibi.
- [3] Apache (NetLoon gibi) için bazı GUI tabanlı yönetim araçları var, ancak henüz yaygın olarak kullanılmıyorlar.