©2016 Published in 4th International Symposium on Innovative Technologies in Engineering and Science 3-5 November 2016 (ISITES2016 Alanya/Antalya - Turkey)

Central Authentication System

*[1]M. Akcay and [2]Mustafa Mercanlı

[*1] Faculty of Engineering, Department of Computer Engineering, Dumlupinar University, 43100 Kutahya, Turkey
[2]Faculty of Engineering, Department of Computer Engineering, Dumlupinar University, 43100 Kutahya, Turkey

Abstract

Central Authentication System is a Java Servlet application which runs on the application server. The designed system connects and queries an Active Directory according to some parameters that come from client side. It also can be checked if an Active Directory user is a member of an Active Directory Group directly or recursively. This makes the authentication flexible because of group authentication.

Key words: Authentication, Security, System

Merkezi Kimlik Doğrulama Sistemi

Özet

Merkezi kimlik doğrulama sistemi uygulama sunucusunda çalışabilen bir Java Servlet uygulamasıdır. İstemci tarafından gelen parametrelere göre tasarlanan sistemde bulunan bir Active Directory sunucusuna bağlanır ve yine istemciden aldığı parametrelerle ilgili kimlik bilgilerini Active Directory üzerinde doğrular. İstenildiği takdirde kullanıcının parametre ile belirtilen Active Directory grubuna üye olup olmadığı da sorgulanabilir. Bu sayede bir uygulama veya sisteme kullanıcı kimlik ve grup doğrulaması sonucuna göre çeşitli varyasyonlarla kullanıcı üye girişi yapılabilir.

Anahtar Kelimeler: Kimlik doğrulama, Güvenlik, Sistem

1. Giriş

Bir kurum/kuruluş bünyesinde birçok uygulama, ara yüz veya sisteme kullanıcı girişi yapmak için veri tabanı ile iletişime geçip daha önceden kayıtlı olan kullanıcı adı, parola ve bazen de yetki grubu bilgilerini doğrulaması gerekmektedir. Her uygulamanın veri tabanı da genelde ayrı olduğundan bu durum her uygulamanın kendine has bir kullanıcı doğrulama mekanizmasının olması sonucunu doğurmuştur. Bu durum bir süre sonra karışık bir hal alıp, hangi kullanıcı adı ve parola hangi uygulamaya ait gibi birtakım soru ve dolayısıyla sorunlar doğuracaktır. Bunun yanında bir de parola gizliliğini de göz önünde bulundurmak gerekmektedir. QR kodlar kullanılarak güvenlik uygulamaları geliştirilmiştir [1-3].

Kimlik doğrulamada parmak izinin görüntü verisi kullanılarak yapılan çalışmalar [4], güvenlik çalışmalarında RFID kullanımı [5], güvenlikte mobil bulut bilişim çalışmaları (Mobile Cloud Computing, MCC) kullanımları [6] giderek artmaktadır.

*Corresponding author: Address: Faculty of Engineering, Department of Computer Engineering Dumlupinar University, 43100, Kutahya TURKEY. E-mail address: makcay26@gmail.com, Phone: +902742652031

Kullanıcı kimlik doğrulama ve diğer cihazlarla iletişim kurulurken EEG cihazları kullanılmakta [7] ve kişilerin vücudundaki titreşimleri algılayarak güvenlikte kullanılan sistemler geliştirilmektedir [8].

Bu çalışmada kullanıcı giriş (`login`) işlemlerini merkezi bir şekilde yönetmek ve kimlik doğrulama için model geliştirmektir.

Bir sonraki bölümde kimlik doğrulama yöntemlerinden birkaçı özetlenecektir. Daha sonra Active Directory ve kullanım alanları açıklanacaktır. Merkezi kimlik doğrulamada kullanılan Java Servlet, uygulama sunucuları, http protokolü, uygulamanın çalışma prensibi, Java Naming Directory Interface (JNDI), geliştirilen çalışmanın uygulanmasına bir örnek sırasıyla detaylı olarak merkezi kimlik doğrulama sistemine katkıları açıklanmıştır. Sonuç ve öneriler bölümü ile yapılan çalışma özetlenmiştir.

2. Kimlik Doğrulama Yöntemleri

Bu bölümde kimlik doğrulama yöntemlerinden nümerik kişisel kimlik, kullanıcı kimlik, parmak izi ve tek kullanımlık şifre ile kimlik doğrulama yöntemleri açıklanacaktır.

2.1. Nümerik Kişisel Kimlik Doğrulama (PIN)

Kişisel kimlik numarası (PIN) bir tür güvenlik parolasıdır. Dört rakamlı numaradan oluşur. 0000-9999 arası 10.000 farklı parola oluşturulabilir. Orijinal açılımı Personal Identification Number'dır [9].

2.2. Kullanıcı Kimlik Doğrulaması

Kullanıcı, bir kişinin bilgisayar veya bilgisayar Ağın'da kullandığı bir hesabıdır. Bir kullanıcı sisteme giriş yapar ve takma isim veya gerçek isim alabilir. Bir kullanıcının sisteme giriş yaptığı zaman hemen bir kullanıcı hesabı açılır ve kullanılmaya başlar. Bir kullanıcının kullanıcı hesabının olması bir sisteme kimlik doğrulaması ve sağladığı veya bu sisteme bağlı kaynaklara erişmek için verilen yetki olmasını sağlar; Ancak, kimlik doğrulama yetkilendirme anlamına gelmez. Bir hesaba giriş yapmak için, bir kullanıcı genellikle muhasebe, güvenlik, günlük ve kaynak yönetimi amaçları için şifre veya diğer kimlik bilgileri ile kendini kimliğini doğrulamak için gereklidir. Çok kullanıcılı sistemlerde kullanıcılardan bir veya daha fazlasına bazı özel yetkiler tanınabilir veya yöneticilik yetkisi verilebilir.

2.3. Parmak İzi Doğrulaması

Parmak izi, parmakların son eklemi ve uç kısmındaki kıvrımların meydana getirdiği iz. Parmak izi insan vücudunun tabii halinden istifade edilerek bulunmuş ve bugün şahıs tespitinde çok fazla kullanılan bir yöntemdir. İnsan vücudunun dış derisinde bulunan her kıvrımda ter gözenekleri vardır. Bunların her biri iç deriye kadar uzanır. Her gözenek orada çiviye benzeyen ve Papila denen iki sıralı çıkıntılarla iç deriye sanki çivi atmış gibidir. Bu sebeple dış deri hasara uğrasa, hatta tamamıyla dökülse bile, bu Papilalar yine de parmak izinin tespiti için yeterlidirler. Yine, yeni

çıkan derilerdeki izler de eskisinin aynısı olurlar. Fakat iç deride bulunan Papila'lar tamamıyla kaybolursa o zaman parmak izini tespit etmek mümkün olmaz; zira bu durumda parmak içi kıvrımları tamamen kaybolmaktadır.

2.4. Tek Kullanımlık Şifre İle Kimlik Doğrulama

Tek kullanımlık Şifre'nin (OTP- One Time Password) amacı erişimi kısıtlanmış kaynaklara yetkisiz erişimi daha da zor hale getirmektir. Bilinen sabit şifreler yeterli deneme şansı ve zaman verildiği takdirde yetkisiz kişiler tarafından aşılabilir. Tek kullanımlık şifre uygulamasında şifre sürekli değiştiği için bu risk büyük oranda azaltılmış olur. Tek kullanımlık şifrelerin temelde üç türü vardır. Birincisi matematiksel bir algoritmayla önceki şifreden bir sonrakini üretir. İkincisi doğrulama sunucusu ile şifreyi temin eden istemci arasında eş zamanlama ilkesine dayanır. Üçüncüsü yine bir matematiksel algoritmaya dayanır ancak önceki şifreyi bir sonrakini türetmek için kullanmak yerine, doğrulama sunucusunun ürettiği bir değere karşılık, oluşturulacak ikinci bir değerin şifre olarak kullanılmasıdır.

3. Active Directory

Active Directory Windows Server ağlarındaki bir dizin hizmetidir. Dizin hizmeti, ağdaki kaynakların bilgisini tutan ve bu bilgiyi kullanıcılara ve uygulamalara sunan ağ hizmetidir. Dizin hizmeti ağ kaynaklarına ulaşmak, bu kaynakları isimlendirmek ve güvenli bir şekilde yönetmek için gereken ortamı sağlamak amacıyla oluşturulur.

Active Directory ortamdaki ağ altyapısına büyük ölçüde işlevsellik kazandırmaktadır. Özellikle kaynakların kontrolünün ve yönetiminin merkezileştirilmesi Active Directory organizasyonunun en önemli özelliğidir. Active Directory, fiziksel topoloji üzerine dayalı bir sistemin kullanıcıya daha basite indirgenmiş şekilde görünmesini ve kaynaklara erişim esnasında kullanıcının eriştiği kaynağın (örneğin yazıcı) ağın neresinde olduğunu veya kaynağın ağa nasıl bağlanmış olduğunu bilmeksizin bağlanmasını sağlar. Active Directory çok büyük işletmelerdeki yoğun bilgileri alt kümelere bölerek saklayabilir ve böylece verilerin büyümesi veya küçülmesi durumunda, yani şirketin büyümesi veya küçülmesi durumunda sisteme esneklik kazandırır.

Active Directory kurulumu bir Windows Server ağ çapında sistem ayarlarını, kullanıcı profillerini ve uygulama bilgilerini Active Directory veri tabanında saklar. Active Directory sistem yöneticilerinin, domain kapsamındaki tüm bilgisayarlarda masa üstü özelliklerini, ağ servislerini ve uygulamaları merkezi bir noktadan yönetebilmelerini sağlar. Active Directory ayrıca, kullanıcıların sisteme bir kez dâhil olmasını, yani `login` olmasını ve ardından ağdaki birçok kaynağa tek bir `login` ile güvenli bir şekilde erişebilmesi konusunda da sistem yöneticilerine merkezi erişim kontrolü sağlar.

4. Java Servlet

Java servlet, Java EE'de Java Servlet API'siyle uyumlu bir Java (programlama) sınıfı olup HTTP istemlerine cevap vermek için kullanılır. Belirli bir istemci-sunucu protokolüne bağlı olmamasına rağmen genelde bu protokolle kullanılır. Servlet kelimesi genelde HTTP servlet yerine bu yüzden

kullanılmaktadır Dolayısıyla bir yazılım uzmanı, Java platformu sayesinde servlet'i bir Web sunucusuna dinamik içerik sağlamak için kullanabilir. Üretilen kod genelde HTML olsa da bazen XML de olabilir. Servlet'ler, CGI ya da ASP.NET gibi Java-dışı Web içerik teknolojilerinin Java'daki karşıt üründür. Servlet'lerle HTTP çerezleri veya URL yeniden yazımı kullanılarak oturum değişkenlerinin sistem durumunu birçok sunucu hareketleri boyunca koruması sağlanmaktadır.

Java sarmal hiyerarşisi Java Servlet içinde bulunan servlet UPA'sı, bir Web container'iyle bir servlet'in beklenen etkileşimini tanımlar. Web taşıyıcısı, aslında Web sunucusunun servlet'lerle etkileşen kısmıdır. Web taşıyıcısı, servlet'lerin yaşam çevrimini yönetmekle görevlidir, bunun için her servlet'e bir URL eşler ve URL istemcisinin doğru erişim hakları olmasını sağlar.

Java Servlet istem alıp buna dayanarak bir yanıt üreten bir nesnedir. Temel servlet paketi, servlet istem ve yanıtını sunan Java nesnelerin yanında servlet'in düzenleşim parametrelerini ve işletme çevresini de tanımlar. Java Servlet paketi, Web sunucusu ve istemcisi arasında yollanan çoklu istem ve yanıtları izleyen oturum yönetimi nesnelerini de içine alan HTTP'ye özgü servlet elemanlarının alt sınıflarını tanımlar. Servlet'ler bir WAR dosyası içine paketlenebilirler.

5. Uygulama Sunucuları

J2EE kapsamında ele alınınca, uygulama sunucusu, J2EE teknolojilerini kullanarak geliştirilen uygulamaları, standartlara (J2EE belirtimine) uygun olarak çalıştıran yazılımlardır. Kullanıcı ara yüzü ile veri tabanı (veya bu görevi gören sistemler) arasında yer alırlar. Çok katlı bir mimari düşünülürse, uygulama sunucusu, mimarinin orta katlarını içinde bulundurur. Bu yüzden, uygulama sunucuları, orta kat yazılımı (Middleware) sayılırlar ve orta kat yazılım teknolojilerini kullanırlar. Genellikle, kullanıcı ara yüzüyle bilgiişlem servisleri arasında veya veri işlem katıyla veri tabanı arasında ya da dağıtık nesnelerle ölçeklenebilirlik servisleri arasında bu teknolojilerden yararlanırlar.

6. HTTP Protokolü

HTTP (Hyper-Text Transfer Protocol, Hiper-Metin Transfer Protokolü) bir kaynaktan dağıtılan ve ortak kullanıma açık olan hiperortam bilgi sistemleri için uygulama seviyesinde bir iletişim protokolüdür. Uygulamalı bilgi sistemleri, basit bir şekilde bilgi almaktan çok daha fazla uygulamaya ihtiyaç duyar. Bu uygulamalar arama, son kullanıcı ara yüzünün güncellenmesi ve etkileşimli olarak bilgi girişi gibi işlevleri de gerektirmektedir. HTTP, bir isteğin amacının ne olduğunu anlatan bir takım açık uçlu yöntemler ve üstbilgi kullanımına izin vermektedir. Bir tek biçimli kaynak tanımlayıcısı, yer belirleyici ya da kaynak ismi tarafından sağlanan kaynağa, bir yöntemin uygulanışını bildiren bir dizi kural üzerine kurulmuştur. Gönderiler, Çok Amaçlı İnternet Posta Uzantıları tarafından tanımlandığı ve İnternet postasında kullanılana benzer bir biçimde aktarılmaktadır. HTTP aynı zamanda, SMTP, NNTP, FTP, Gopher ve WAIS iletişim kurallarını destekleyen İnternet sistemleri ile kullanıcı istemcileri, vekil sunucular ve Geçitler arasında iletişim için özelleştirilmiş bir iletişim kuralı olarak da kullanılır. Bu haliyle HTTP, muhtelif uygulamalar tarafından sağlanan kaynaklara, basit hiperortam erişimine izin vermektedir. Günümüzde hayatın önemli bir parçası haline gelen İnternet, HTTP sayesinde her türlü bilgiye sorunsuz erişimi kolay

kılmaktadır. İstek-cevap bilgisayarların birbirleriyle konuşmaları için kullanılan temel metotlardan birisidir. İstek-Cevap kullanılırken, ilk bilgisayar bir istek gönderir ve ikinci bilgisayar da bu isteği yanıtlar.

7. Geliştirilen Model ve Uygulamanın Çalışma Prensibi

Bu çalışmada geliştirilen uygulama yukarıda bahsedilen HTTP protokolü ile iletişime geçebilen bir Java Servlet uygulamasıdır. Yani istemci tarafından istekler parametre olarak HTTP GET veya POST metodu ile sunucuya gönderildiğinde uygulama bu parametreleri çözümleyerek bir Active Directory sunucusunda doğrulama işlemi yapmaktadır. Active Directory den aldığı olumlu ya da olumsuz yanıtları da istemciye JSON veri olarak geri göndermektedir. Bu doğrulama işleminde Active Directory etki alanı kullanıcı adı ve parola doğrulaması olduğu gibi isteğe bağlı olarak kullanıcı parametre ile gönderilen gruba üye mi, değil mi kontrolü de rekürsif olarak yapılabiliyor. Yani kullanıcı gruba doğrudan üye olmasa dahi kullanıcının bulunduğu başka bir grup, ilgili gruba dâhil de olsa sistem bunu çözümleyebilmektedir**.**

8. JNDI

JNDI (Java Naming Directory Interface) Java'nın dizin hizmetlerine bağlanmasını sağlayan bir API'dir. JNDI sayesinde Active Directory'de bulunan dizin, bilgisayar, kullanıcı, grup vb. bütün içeriği Java Objelerine dönüştürebilmektedir. Uygulama JNDI a karşı iki farklı bakış açısını taşımaktadır. Birincisi tüm Active Directory objelerini tek seferde çekip bundan sonra yapılacak bütün sorgulamaları yerel olarak gerçekleştirmek, ikincisi ise sorgulama işlemlerini zamanı geldiğinde Active Directory'e session açarak gerçekleştirmek. İki metodun da yeri ve zamanına göre gerekli olduğu bir gerçek. Ancak bu uygulama JNDI dan sadece kimlik doğrulama ve grup bilgilerini sorgulama tarzı temel şeyleri yapmasını isteyeceğinden tüm Active Directory içeriğini çekmek gibi bit gerekliliğe ihtiyaç duyulmamıştır. Bunun için de sıklıkla bir standart üzerine oturtulmuş `LDAP Search Filter`'a başvuruldu. `LDAP Search Filter` a bir örnek olarak aşağıdaki cümleyi verebiliriz.

```
"(&(objectCategory=person)(objectClass=user)(!(cn=MustafaMercanlı
)))"
```

Bu cümle JNDI dan Active Directory sunucusundan bilgi çekerken ilgili arama cümleciğine göre bilgileri çekeceğini, yani adı `MustafaMercanlı` olan kullanıcıyı getireceğini söylemiştir.

9. Uygulamanın Çalışmasına Bir Örnek

Bu sistem yukarıda açıklandığı gibi bir orta kat (middleware) yazılımdır. Bir uygulama sunucusu üzerinde çalışan Java Web Archive (WAR) dosyasıdır ve bir Java Servlet uygulamasıdır. Client uygulamayı tetiklemek için uygulamaya bir HTTP GET veya POST isteği göndermelidir. Bu isteği gönderirken sistem belli başlı birkaç parametreye de ihtiyaç duymaktadır. Bu parametreler;

sam: Kullanıcının Microsoft Active Directory ya da Open LDAP kullanıcı hesabının adıdır.
pass: Kullanıcının Microsoft Active Directory ya da Open LDAP kullanıcı hesabının parolasıdır.

domain: Active Directory'nin üzerinde bulunduğu etki alanı adıdır. Örneğin "dpu.gov.tr" gibi.
dc: Active Directory etki alanına ait herhangi bir etki alanı denetleyicisi (Domain Controller) Ip numarası
group: Bu parametre isteğe bağlıdır. Eğer ki kullanıcının parametrede belirtilen gruba üyeliğinin olup olmadığı sorgulanır.

Bu bilgilerden sonra uygulama sunucumuz üzerinde çalışan uygulamamıza tarayıcı adres çubuğumuzdan aşağıdaki gibi bir HTTP GET isteği gönderelim. GET isteğini öncelikle herhangi bir parametre tanımlamadan gönderelim.

```
http://192.168.11.130:8086/LdapAuthServlet/
```

Bu isteğe karşı uygulama sunucusunda bulunan servlet 'in ne yanıt verdiğine bir bakalım

```
{Parametreler:{domain="Etki alani adi",dc="Etki alani denetleyicisi Ip si",sam="Login olacak kullanici adi",pass="Login olacak kullanici parolasi",group="Sorgulanacak grup(istege bagli)"}}
```

Görüldüğü gibi sistem bizi parametrelerle ilgili rehberlik yaptı. Hangi parametreler ile çalışacağını belirtti.

Daha sonra da kullanıcı adı ve parola doğru olarak ancak herhangi bir grup üyeliği kontrolü yapmadan gönderelim.

```
http://192.168.11.130:8086/LdapAuthServlet/?sam=ogr1&domain=proje.com&dc=192.168.11.10&pass=Proje123
```

Bu parametreleri alan uygulamamız hemen ilgili Active Directory ile iletişime geçip ilgili kullanıcının kimlik bilgileri kontrol etmiştir. Kullanıcı adı ve parola doğru olduğu için sonuç olumlu döndü.

```
{Login:true}
```

Daha sonra kullanıcının parolasını yanlış girerek tekrar istek gönderelim.

```
{Login:false}
```

Görüldüğü gibi kullanıcının parolasını girdiğimizde kimlik doğrulamanın gerçekleşmediğini belirten bir JSON veri karşımıza çıkmaktadır.

Şimdi de kullanıcının bir grup üyeliğini sorgulatalım.

```
http://192.168.11.130:8086/LdapAuthServlet/?sam=ogr1&domain=proje.com&dc=192.168.11.10&pass=Proje123&group=DPU
```

Bu isteğe gelen yanıt ise

```
{Login:true,isMember:true}
```

Yani bu kullanıcının veya onun üst grubunun DPU gurubuna bir üyeliği bulunmadığı ancak kullanıcı adının ve parolasının doğru olduğu bilgisi gelmiştir.

Sonuç olarak isteği gönderen taraf bu JSON formattaki yanıtı alıp bir nesneye çevirerek gelen cevaba göre istediği şekilde kullanıcıyı sayfalara yönlendirebilecektir.

4. Sonuçlar ve Öneriler

Bu çalışmadaki amaç ortamda bulunan kimlik denetleme mekanizmalarını her uygulama için ayrı ayrı veri tabanlarından doğrulamak yerine, bu işi merkezi tek bir kaynaktan doğrulamaktadır. Bu çalışmanın sonucunda beklentiler ve yapılmak istenilenler büyük oranda gerçekleştirilmiştir. Bütün kimlik doğrulama işlemlerini Active Directory sunucusuna sorarak yönetilebilirlik ve kısmen de güvenlik sağlandı.

Bu çalışma bir kimlik doğrulama işlemini Active Directory kullanarak yerel veri tabanında tutmaktan çok daha güvenli bir şekilde kimlik doğrulama gerçekleştirmektedir. Ancak uygulama sunucusu ile kurulan iletişim HTTP protokolü üzerinden gerçekleşmektedir. Bu olay ortamın network trafiğini dinleyen kötü niyetli kişiler için bir fırsat teşkil etmektedir. Bu iletişimi HTTPS iletişimi kurarak veya istemci tarafına verilecek bir şifreleme algoritması ile şifresini karıştırıp uygulama sunucusu tarafında aynı algoritma ile çözerek veri transferini daha güvenli bir hale getirilebilir.

Kaynaklar

[1] Chen WY, Wang JW. Nested image steganography scheme using QR-barcode technique. Optical Engineering; 48(5), May 2009.

[2] Zigomitros A, Patsakis C. Cross format embedding of metadata in images using QR codes. Intelligent Interactive Multimedia Systems and Services, vol. 11 of the series Smart Innovation, Systems and Technologies, pp. 113-121.

[3] Chung CH, Chen WY, Tu CM. Image hidden technique using QR-barcode. The Fifth International Conference on Intelligent Information Hiding and Multimedia Signal, 2009.

[4] Aykut M, Ekinci M. Developing a contactless palmprint authentication system by introducing a novel ROI extraction method. Image and Vision Computing; 40 (2015), pp. 65-74.

[5] Cheng L, et al. A secure and lightweight authentication protocol for RFID. Electronics Information and Emergency Communication (ICEIEC), 2015 5th International Conference on. IEEE, 2015.

[6] Zkik K, Tebaa M, El Hajji S. A New Secure Framework in MCC Using Homomorphic Signature: Application in Banking Data. Transactions on Engineering Technologies, Springer Singapore, 2016, pp. 413-427.

[7] Rodriguez RJ. Electroencephalogram (EEG) based authentication leveraging visual evoked potentials (VEP) resulting from exposure to emotionally significant images. Technologies for Homeland Security (HST), 2016 IEEE Symposium on. IEEE, 2016.

[8] Strachan S, Panëels S. ViSecure: A Haptic Gesture Authentication System. International Conference on Human Haptic Sensing and Touch Enabled Computer Applications, Springer International Publishing, 2016.

[9] Personal Identification Number,https://en.wikipedia.org/wiki/Personal_identification_number