İNTERNET ORTAMINDA LİNUX

TABANLI GÜVENLİ DATA TRANSFERİ

Hazırlayan

Şevket Keser

Danışman

Öğrt. Gör. Ecir Uğur Küçüksille

T.C
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
TEKNİK EĞİTİM FAKÜLTESİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ BÖLÜMÜ

BİTİRME TEZİ

İNTERNET ORTAMINDA LİNUX İŞLETİM SİSTEMİ TABANLI GÜVENLİ DATA TRANSFERİ

Hazırlayan
Şevket Keser

Danışman
Öğrt. Gör. Uğur Ecir Küçüksille
ISPARTA – 2003 ©

İÇİNDEKİLER **SAYFA**

ŞEKİL LİSTESİ **SAYFA**

ÖNSÖZ

İletişim teknolojilerinin her geçen gün gelişmesi ve ilerlemesi doğrultusunda, günlük yaşama katkısı da o denli artmaktadır.Bunların günümüze kadarki olanlarından en hızlı ilerleyeni kuşkusuz internet , bilişim ve haberleşme teknolojileridir.Internetin günlük yaşamın her alanına girmesiyle birlikte, bu ortam üzerinde kişisel ve ticari bilgilerde taşınmaya başlanmıştır.İşte tam bu aşamada, internet ortamındaki ticari ve kişisel bilgilerin güvenliği önemli bir problem olarak karşımıza çıkmaktadır.Bu proje, internet ortamında taşınan her türlü verinin "güvenli" bir şekilde taşımasınmasını amaçlamaktadır.

Proje Danışmanlığımı yapan ve değerli fikirlerini benden esirgemeyen **Öğr. Gör. Ecir Küçüksille**'ye,

Bölümde Open Source nitelikli bu ilk çalışmayı destekleyen ve yapılması için gerekli ortamı oluşturan **Yard. Doç Dr. Akif Kutlu'** ya

Tez yazımı konusunda değerli fikirlenden yararlandığım **Öğr. Gör Abdulkadir Çakır** 'a,

Projenin tasarımı, hazırlaması, karşılaştığım problemlerde yol gostericiliğimi yapan ve bende her türlü desteklerini esirgemeyen, S. Demirel Üniversitesi Bilgi İşlem Merkezi Sistem Yöneticileri **Volkan Sönmez** ve **İsmail Kavak** Beylere,

Ayrıca maddi ve manevi destekleriyle her zaman yanımızda olan aileleme ve arkadaşlarıma *Sonsuz Teşekkürlerimle…*

Şevket Keser

Haziran – 2003

ÖZET

Internet yaşama nüfuz ettikçe, internet üzerindeki uygulamaların sayısı ve çeşidi de hızla artmaktadır. Bir haberleşme ortamı olmasının yanı sıra, internet, bir elektronik pazar yeri de olmaktadır. Internet'in önemi artıkça yapısından kaynaklanan, çözülmesi gereken problemler ortaya çıkmaktadır. Güvenlik bunların başında yer almaktadır; korunması gereken kişisel bilgilerin internet üzerinden aktarılacağı durumlarda, bankacılık işlemleri ve her türlü dosya transferi gibi, yeterli güvenliğin sağlanması elzemdir.

Bu proje intenet ortamında, kullanıcılar arasında transfer edilen her türlü veri için güvenli ortamlar oluşturmayı amaçlamaktadır.Proje gereklilikleri ve amaçları doğrultusunda geliştirilmiş ve hedeflenen sonuçlara erişilmiştir.

ABSTRACT

The usage of the internet is increasing, the count and kind of the applications increase very fast on the internet too. A communication environment however it is an electronic commerce. When the importance of the internet is increasing, some problems needed to analyse which are appears result from structure of the internet. Security is to take part of at first. when the personal informations which necessary to protected, must be supply enough security to transfer on the internet.

Example of this banking operations. this project, purposes to forming safe environments which transfers all of the databetween users on the internet. This project was developed direction of its necessitiesand aims and reached targetting results.

1.GİRİŞ

Internetin kullanılmaya başlandığı ilk yıllardan itibaren, günümüz insanına kattığı “değerler” sürekli artış trendi içerisindedir. Günümüzde bireyler, e-ticaret, e-devlet gibi birçok uygulama ile, alış-veriş, vergi ve fatura ödemeleri gibi hayatlarından zaman çalan bir çok işlemi, bilgisayarları aracılığı ile saniyeler içerisinde yapabilmektedirler. Bu işlemlerin gerçekleştirilebilmesi için Internet ortamında IP paketleri aracılığıyla taşınan verinin yoğunluğu gün geçtikçe artmaktadır. Taşınan verinin niteliği ise değişkendir. IP paketleri bazen bir e-mail mesajı taşırken, kimi zaman da bir kredi kartı numara bilgisi taşıyabilmektedir. Kredi kartı bilgisi, banka hesap bilgisi gibi ticari değer içeren verilerin taşınmasında güvenlik önemli bir sorun oluşturmaktadır. Özellikle bu tip verilerin üçüncü şahıslar tarafından hiçbir şekilde öğrenilmemesi gerekmektedir.*Bu proje; Internet ortamında üçüncü şahıslar tarafından görülmesi istenmeyen verilerin güvenli şekilde transferini amaçlamaktadır.*

Tez kitapçığı üç ana bölümden oluşmaktadır.Birinci bölümde, projenin niteliği gereği “Açık Kod (Open Source)” felsefesi, veri transferi ve Internet ortamında güvenlik kavramları üzerinde durulmuştur.İkinci bölümde ise “Proje Bileşenleri” yer almaktadır.Son bölümde ise tez konusu olan Internet Ortamında Linux Işletim Sistemi Tabanlı Güvenli Data Transferi (Open Source Security Data Transfer – OSSDT) konu alınmaktadır.

2.SERBEST YAZILIM - AÇIK KOD (OPEN SOURCE) FELSEFESİ

"Serbest Yazılım – Açık Kod", temelinde bir kullanıcıya bir yazılımı çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme özgürlüklerini veren akımın adıdır. Serbest yazılımlar, genel kanının aksine ücretsiz dağıtılmak zorunda değillerdir. Tek ve gerekli önkoşul, yazılımın kaynak kodu ile birlikte dağıtılmasıdır. "Serbest Yazılım" konusunun daha iyi anlaşılabilmesi için kullanıcıya tanınan özgürlükleri aşağıda sıralanmıştır.

- Kullanıcı, yazılımı her türlü amaç için çalıştırma özgürlüğüne sahiptir.
- Yazılımın nasıl çalıştığını inceleme ve gereksinimleri doğrultusunda değişiklik yapma hakkına sahiptir.
- Yeniden dağıtma ve toplum ile paylaşma özgürlüğüne sahiptir.
- Yazılımı geliştirme ve yeni hali ile toplum ile paylaşma özgürlüğüne sahiptir.

Bir yazılım, bütün kullanıcıları yukarıdaki bütün özgürlüklere sahip ise "Serbest Yazılım" olarak adlandırılabilir. Yani yazılımı kopyalama, değiştirme, aynen veya değiştirerek para ile satma, herkese ve her yerde dağıtma ve bedava verme özgürlüklerini tanımalıdır.

2.1 Serbest Yazılımın Tarihi

"Serbest Yazılım" kavramı her ne kadar 1980'lerde ortaya atılmışsa da, 1970'lerde bilgisayar kullanıcıları ürettikleri yazılımları "Serbest Yazılım" ilkelerine uygun olarak birbirleriyle paylaşırlardı. 1980’lerin başlarında yazılım üreten bazı firmaların kullanıcıları çok zorlayan lisans anlaşmalarını öne sürmeleri üzerine o sıralarda MIT'de çalışmakta olan Richard Stallman "Serbest Yazılım" fikrini ortaya atmış ve bütünüyle "Serbest Yazılım" olacak bir işletim sistemi yazmak için çalışmalara başlamıştır. Zamanla Stallman'a diger geliştiricilerin de katılması ile "Free Software Foundation (FSF)" kurulmuş ve ilerleyen zamanlarda bir taraftan Unix benzeri bir işletim sistemi için gerekli yazılımlar üretilirken, bir taraftan da "GNU General Public

Licence (GLP)" ve "copyleft" kavramları tanımlanmıştır. İsminin çağrıştırdığı gibi "copyleft" kavramı "copyright" yasaları düşünülerek geliştirilmiş ve yazılımlar ile ilgili özelleştirme ya da kontroller yerine özgürlükleri temel alan bir düzenlemeyi hedeflemiştir.

GNU sisteminin en büyük eksiği olan çekirdek (GNU Hurd) geliştirilirken, 1991 yılında Linus Torvalds, Linux ismini verdiği çekirdeği duyurmuş ve 1992 yılında GNU yazılımlarıda kullanılarak bütünüyle bir Serbest Yazılım olan GNU/Linux ortaya çıkmıştır. 1992 yılından günümüze kadar GNU/Linux işletim sisteminin yadsınamaz katkısı ile "Serbest Yazılım" üretimi ve kullanımı hızla artmıştır. Örneğin, Internet üzerinden ulaşılabilen web sayfalarının % 56.4 ü bir Serbest Yazılım olan "Apache" web sunucusu tarafından sunulmaktadır. GNU/Linux dağıtımlarının sayısı 100' ü geçmiştir.[1]

2.2 Neden Serbest Yazılım?

Kullanıcılarının "Serbest Yazılım"ı tercih etmelerinin bir çok nedeni olabilir. Bu kullanıcının beklentilerine göre çeşitlilik gösterebilir. İlk akla gelen nedenleri:

- Genellikle ücretsiz dağıtılır.
- Hiçbir kişinin veya kuruluşun tekelinde değildir.
- Kesinlikle daha güvenlidir.
- Kaynak kodu açıktır ve kaynak kodu ile birlikte dağıtılır.
- Öğrenmeyi teşvik eder.

Yukarıda sayılanlar, özellikle üniversitelerin ve diğer kuruluşların, hatta dünya üzerinde bazı devletlerin "Serbest Yazılım (Açık Kod) "a yönelirken göz önünde bulundurdukları başlıca nedenlerdir.

Son kullanıcı için de aynı şeylerin geçerli olduğunu söylemek yerinde olur. Donanım fiyatı ile başa baş giden lisans ücretleri, kararlı çalışmayan işletim sistemleri ve üzerinde çalışan uygulamalardan kaçınmak isteyen kullanıcılar "Serbest Yazılım"

örneklerine yönelmekte, öğrenmenin ve paylaşmanın esas olduğu bir evrene adım atmaktadır. Özellikle Üniversite/Akademi ruhunun içerdiği öğrenme, geliştirme ve üretme eylemlerine platform sağlaması nedeniyle "Serbest Yazılım"lar tercih edilmektedir.

3.INTERNET VE VERİ TRANSFERİ

Gelişen teknoloji sürecinde her gün dunyayı saran fiber omurgalarca taşınan veri her gecen gün artmaktadır. İnternet ortamında gerçekleştirilen her türlü (ses, görüntü vs..) iletişim IP paketleri aracılığıyla taşınan datalar ile gerçekleştirilir.Bu sistem TCP/IP temelinde çalışır.Kullanıcılar birbirleriyle değişik veri tiplerinde iletişim kurarlar.İletilmek istenen bilgi çeşitli formatlarda olabilir.Taşınmak istenen bilgi TCP/IP iletişim protokolü gereğince IP paketlerine bölünür ve dijital formatta gideceği adrese ulaştırılır.Internette yapılan temel işlem; kullanıcılar arasında çeşitli formatlarda veri transferini (dolayısıyla iletişimi) gerçekleştirmektir.Internetten önce bu işlem posta şirketleri aracılığıyla sağlanmıştır.Günümüzde ise internet teknolojisi yardımıyla elektronik ortama aktarılan her türlü veri istendiği yere ulaştırılabilmektedir.

4. “E-GÜVENLİK” KAVRAMI VE NEDEN GÜVENLİK ?

Hergün birçok şirket teknolojinin ne kadar ilerlediğini görüp bir şekilde bundan faydalanmak ve iş akış süreçlerini hızlandırmak yada revize etmek ister. Bir işletmenin konusu , büyüklüğü ne olursa olsun bugün en az 2-3 bilgisayarı yada farklı bölgelerdeki şubeleriyle haberleşme sistemleri vardır. Şirketler optimum kaynaklarla maksimum verim istediklerine göre de bu durum bir süre sonra kaçınılmaz olur. Bugün şehirlerarası yada uluslararası bir şirket olma kaygısı gütmeden ayakta kalmak mümkün değildir, bu yüzden olsa gerek , şirketler şube veya merkezlerini bir netvrork ile bileştirmeyi ve telefon, fax, e-posta sistemleri kurarak düşük maliyet ile hız kazanmayı istemektedirler. Tabiki bu durum bazı risklerde içerir. Örneğin telefon maliyetimizi düşürmek için 2 bölge arasında kurduğunuz ağ üzerinden bu işlemi gerçekleştirmeniz durumunda

görüşmelerinizi bir çalışanınızın sniffer kullanarak yakalaması ve Cisco her ne kadar yasalarla engellemeye çalışsada sesli görüşmelerinizi dinlemesi, staratejik kararlar içeren e-postalarmızın internette elden ele gezmesi, piyasaya çıktığında size milyon dolarlar kazandıracak ürününüzün planlarmın bir bilgisayar korsanı tarafından ele geçirilmesi sonrada açıklanması hatta rakibinize yüksek meblağlara satılması gibi riskler kullanıcıları beklemektedir.[2]

5.PROJE BİLEŞENLERİ (PLATFORM)

Bu bolumde proejnin teknik temelleri,ihtiyaçları ve gereksinimleri anlatılmaktadır.

5.1.. LİNUX İŞLETİM SİSTEMİ

Linux, serbestçe dağıtılabilen, GPL (GNU Public Licance) lisanslı,açık kodlu,çokgörevli, çok kullanıcılı UNIX işletim sistemi türevidir. Linux, İnternet üzerinde ilgili ve meraklı birçok kişi tarafından ortak olarak geliştirilmekte olan ve başta IBM-PC uyumlu kişisel bilgisayarlar olmak üzere birçok platformda çalışabilen ve herhangi bir maliyeti olmayan bir işletim sistemidir.

Linux, temel olarak Finlandiya Üniversitesinde öğrenci olan Linus Torvalds'ın ve İnternet üzerinde meraklı bir çok yazılımcının katkıları ile geliştirilmiştir. Linux gelişimi açık bir şekilde yapılmaktadır. Bunun anlamı, işletim sisteminin her aşaması açık olarak İnternet üzerinde yayınlanmakta, dünyanın dört bir yanında kullanıcılar tarafından test edilmekte, hataları ve eksiklikleri tesbit edilerek düzeltilmekte ve geliştirilmektedir. Zaman zaman bu deneme aşamaları belirli bir noktada durdurulur ve güvenilir bir işletim sistemi sunulup, geliştirme için ayrı bir seriye devam edilir. Geliştirmede yer alan bu açıklık Linux'un en büyük avantajlarından biridir. Gelişimi evrimseldir, hatalar anında kullanıcılar tarafından tesbit edilip rapor edilmekte ve birçok kişinin katkısıyla düzeltilmektedir. Bazı işletim sistemi sürümleri saatler içerisinde güncellenebilmektedir.

Linux, Andy Tannenbaum tarafından geliştirilmiş olan Minix işletim sistemine dayanmaktadır. Linus Torvalds boş zamanlarında Minix'ten daha iyi bir Minix işletim sistemi yaratmak düşüncesiyle 1991 Ağustos sonlarında ilk çalışan Linux çekirdeğini oluşturdu. 5 Ekim 1991 tarihinde 0.02 sürümü Linux ilk defa tanıtıldı. Linus, comp.os.minix haber grubuna gönderdiği yazıda yeni bir işletim sistemi geliştirmekte olduğunu ve ilgilenen herkesin yardımını beklediğini yazmıştı. İşletim sisteminin çekirdeği için verilen numaralar kısa sürede bir standart kazandı. a.x.y seklinde belirtilen çekirdek türevlerinde y bulunulan seviyeyi, x gelişim aşamasını göstermektedir. Tek sayılı x'ler geliştirme aşamalarını çift sayılı x' ler ise güvenilir Linux çekirdeklerini göstermektedirler. a ise değişik Linux sürümlerini belirtir. Bu yazının hazırlandığı Ocak 2003 içerisinde en son güvenilir (kararlı) Linux çekirdeği 2.4.18’dir

Linux gerçekten son yıllarda hızlı bir gelişme göstermiş, çesitli ülkelerden birçok kullanıcıya erişmiş ve yazılım desteği günden güne artmıştır. Değişik kuruluşlar Linux sistemi ve uygulama yazılımlarını biraraya getirerek dağıtımlar oluşturmuşlar ve kullanımını yaygınlaştırmışlardır.[3]

5.1.1 Neden Linux ?

Bu sorunun cevabı aşağıdaki maddelerde toplanmıştır.

- Açık kaynak kodludur.Böylelikle sistemde istenilen noktaya erişilebilir ve geliştirmye uygundur.
- Güvenlidir.Açık kodlu olması sebebiyle güvenlik açıkları kolay tespit edilenbilir ve gerekli yamalar rakiplerine oranla daha hızlı çıkartılır.
- Sağlamdır.Açık Kod yasılımlarının genel niteliği olan bu özellik Linux da da kendini göstermiştir.Özellikle çekirdek ve kritik uygulamlar herkese açık bir gelişim sureci ve test aşamaları izler.
- Ekonomiktir.Internetten kodlarıyla birlikte indirip bilgisayarınza kurabilirsiniz.

Linux işletim sistemin ilk akla gelebilecek tercih nitelikleri yukarıda sıralanmıştır.Linux ilk duyurulduğu günden itibaren sürekli yükseliş trendinde olan bir sistemdir.[5]

5.2.. APACHE

Linux işletim sistemi beraberinde gelen web sunucusudur.Proje Apache web server üzerinde çalışmaktdır.Apache web sunucusu güçlü ve karalı bir platform sunmaktadır.Özellikle internetteki web sunucularında %50 yi aşan oranlarda apache'nin kullanılması bunun başka bir kanıtıdır.

5.2.1 Neden Apache ?

Yüm bu yukarıda anlatılan niteliklerden ve projenin açık kod niteliğinde olmasından dolayı, yine açık kod bir yazılım olan ve Linux işletim sistemi beraberinde gelen "Apache" web sunucusu seçilmiştir.

5.3.. PHP

"PHP" OSSDT'in yazıldığı temel script dilidir.

5.3.1 PHP Nedir ?

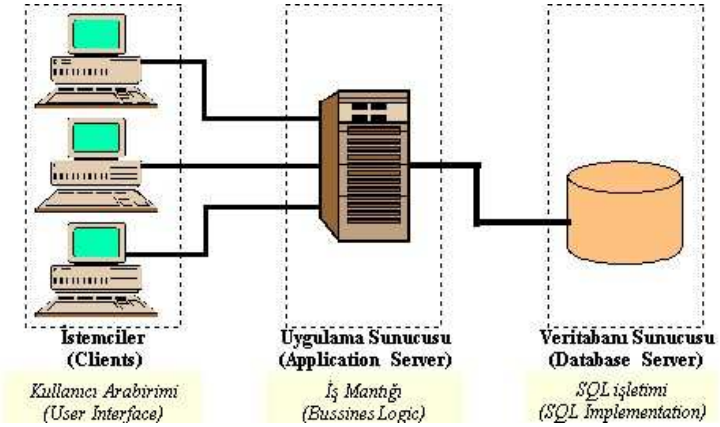
PHP şu şekilde tanımlanabilir: Sunucu tabanlı (server sided), HTML ile bir arada kullanılabilen (HTML embedded), açık kodlu bir script dilidir.[4]

5.3.2 Çalışma Modeli

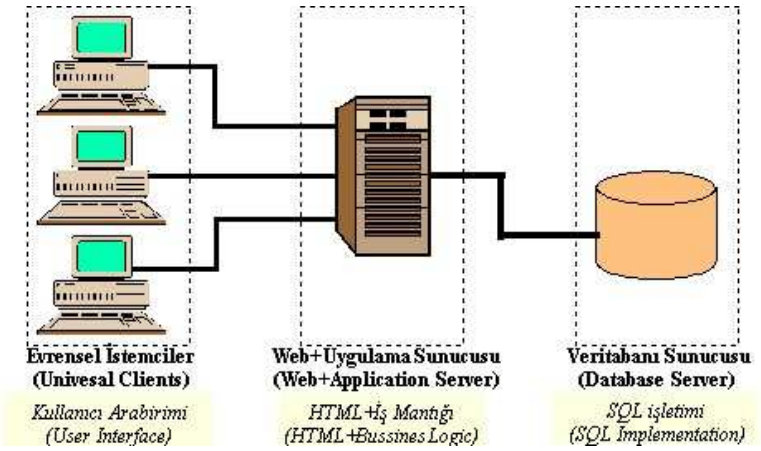
Veri tabanı sistemi ile evrensel ve hafif istemciler arasında yer alır.

Hem HTML üretimi, hem de uygulama mantığı PHP motoru üzerinde gerçekleşir.

Genel olarak Web tarafı ağır basan üç katmanlı yapıyı esas alır.



Şekil I PHP Çalışma Modeli, Üç Katmanlı Yapı



Şekil II PHP Çalışma Modeli, Web Tabanlı Mimari

5.3.3 Neden PHP?

PHP aşağıdaki kriterleri sağlmasıyla proje için diğer dillerin rasından seçilmiştir.

- Verimlilik
- Bakım kolaylığı
- Taşınabilirlik
- Uyumluluk
- Performans
- Hızlı Gelişim ve Yaygınlık
- Ekonomik

5.3.3.1 Verimlilik

Web uygulamaları hızlı geliştirme gerektiren türde uygulamalardır. Bu nedenle bu alanda script dilleri ön plana çıkmışlardır.Kod boyu kısalmalı. PHP web uygulamaları için yazılabilecek kod sayısını en dengeli düzeye çekmiştir. PHP, yapmanız gereken işi en kısa yoldan yaparken işlevselliği de sınırlamıyor. İşlevselliği çok iyi olmasına karşın Java servletleri yazılması gereken kod boyunu aşırı artırıyor.

Hazır araçlar. PHP'nin fonksiyon kütüphanesi oldukça geniş bir yelpazeye sahiptir. 4.0 versiyonunda 1729 fonksiyon tanımlı. Mail, FTP, LDAP ve HTTP gibi protokollerini kullanabilmek için RFC okumaya gerek yoktur.. Veritabanlarına erişim fonksiyonları işi kolaylaştırmış.Allaire firmasının Cold Fusion ürününde kullanılan CFML (Cold Fusion Markup Language) script dili kısa ve okunurluğu iyi olmasına karşın sadece 60 tag ve 200 fonksiyon ile sınırlanmıştır. Aynı şekilde ASP ile hazır gelen araçlar da oldukça sınırlıdır.Yeniden kullanılabilirlik ; PHP, nesneye yönelik programlama için desteklemektedir. Bu yönüyle PHP kullanıcılarının hazırlamış olduğu kütüphaneler kullanılabileceği gibi kendi kütüphanelerinizi oluşturup kullanmanız da izin vermektedir.

5.3.3.2 Bakım Kolaylığı

Web uygulamaları çok sık güncellenirler. Bazı durumlarda sıfırdan kod yazılması gereken durumlar ortaya çıkmaktadır.Bu açıdan kod bakımı ve değiştirilebilmesi kolay ve hızlı yapılabilmelidir.PHP bu yönüyle benzerlerinden üstündür.

5.3.3.3 Taşınabilirlik

Platformdan bağımsızlık sadece Java'nın sloganı değildir. PHP ile geliştirilen uygulamaların değişik işletim sistemleri ve değişik web sunucuları üzerinde çalışabiliyor olmasını önemli bir özellikltir.Bu konuda PHP'nin, java servletleri, JSP ve Perl dışında rakibi yoktur.

5.3.3.4 Uyumluluk

PHP, çok sayıda açık sistem ve protokol ile birlikte çalışabildiği gibi hem de pek çok özelleşmiş sistem için yerleşik (native) arabirim desteği sunmaktadır.

Destekleninen açık sistem protokolleri aşağıda verilmiştir.

HTTP/HTTPS	SMTP	SNMP
FTP	IMAP, POP3	Socket Arabirimi (TCP/IP)
LDAP	NNTP	

Projede PHP tarafından desteklenen HTTPS protokulü kullanılarak verilerin internet ortamında güvenliği sağlanmıştır.

Destekleninen Yerleşik Veri Tabanı Arabirimleri

MySQL	BerkeleyDB	Oracle
mSQL	MS-SQL	Sybase

PostgresSQL Interbase DB/2

Projede PHP tarafından desteklenen MySQL protokulü kullanılarak kullanıcı kimlik denetim bilgileri tutulmuştur.

Açık Sistem Teknolojiler

ODBC XML

OpenSSL / X509PDF Sertifikaları

Projede PHP tarafından desteklenen OpenSSL protokulü kullanılarak HTTPS protokolünün sertifikaları üretlmiştir.

5.3.3.5 Performans

8 saniye kuralı. Web'de ziyaretçiler genellikle yüklenmesi 8 saniyeyi geçen sayfalardan vazgeçerler. Buna karşın çoğu kez bant genişliğiniz sınırlıdır. Toplam 35KB büyüklüğünde olan bir sayfa 56Kbps'lik bir modemle 7-8 saniyede çekilir. Bu süreye sunucu cevap süresi eklenecektir.

Cevap süresi (response time). Normalde web sunucusu dinamik bir sayfada script işletimi bitene kadar cevap göndermez. Bu sürenin de 1-2 saniyeyi aşmaması zorunludur.

CGI kullanan uygulamalar işletim sistemi tarafından görevlerin paralel olarak hafızaya yüklenmesi ve bunlar arasında geçişler gerektirdiği için CPU, RAM, zaman harcaması fazladır ve tutarsız davranırlar. Her seferinde yeniden veri tabanına bağlantı gerektirirler.

Server API'si kullanan modüller ise CGI'ya göre 15-20 kat hızlı çalışabilirler. PHP'nin SAPI kullanan modül versiyonu da bu avantajdan yararlanır.

Zend optimizer ile ara kod iyileştirmesi kullanılarak buna ek olarak %40 - %100 hızlanma elde edilebilir.

Yerleşik (native) veritabanı desteği verdiği için ODBC kullanan sistemlere göre de hızlıdır.Sürekli veritabanı bağlantıları (persistent connections). Java servletlerindeki connection pooling mantığını programcıya saydam tutarak kullanır.

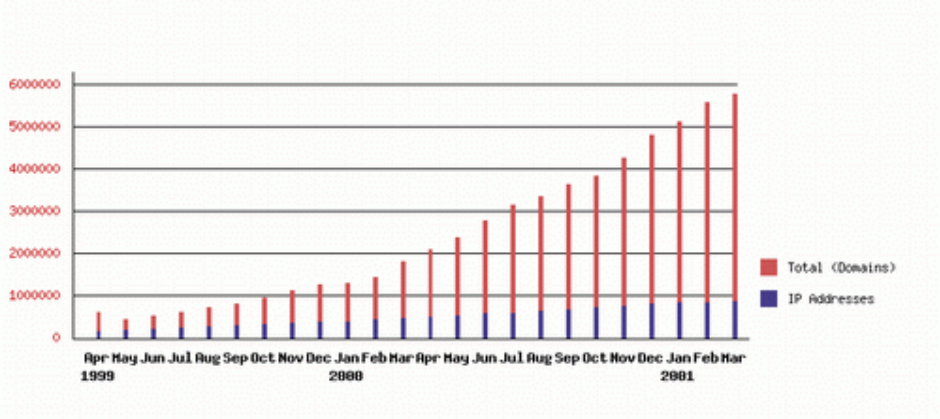
PHP standart görevler için ASP'de olduğu gibi COM nesneleriyle iletişim kurmak zorunda kalmaz. Bu tür işlevler sunucunun kendi hafıza alanı içinden çağrılır.

Önbellekleme. Zend Cache ve benzeri yazılımlar ile scriptler derlenmiş olarak hafızada tutulabilir.

5.3.3.6 Hızlı Gelişim ve Yaygınlık

Açık kaynak kodlu. Bir çok ülkeden gönüllü geliştiricilere sahip. Hatalar çok çabuk bulunup düzeltiliyor. Pek çok web sunucusu ve işletim sistemine port edilmiş durumda. Kendi sisteminiz üzerinde PHP'den kaynaklanan bir sorun olursa en kötü durumda PHP koduna müdahale etme şansınız var.

Dünya üzerinde PHP kullanan domain sayısı 5.747.237, IP adresi sayısı ise 882.439



Şekil III PHP Kullanan Domain Sayısı

5.3.3.6 Şifreleme ve OpenSSL

PHP ile tek yönlü şifreleme ve özet üretme fonksiyonları kullanılabilir (MD5, Crypt, ...) PHP tarafından simetrik çalışan pek çok şifreleme algoritmalarını da desteklenir(DES, DES3, Blowfish, IDEA, ...)

PHP 4.0 versiyonunda OpenSSL'i de dahil ederek asimetrik şifreleme (RSA, DSA) metodlarını da destekler hale gelmiştir.Bu bağlamda X509 sertifikaları da kullanılabilir.Böylece uzak noktalar arasında güvenli veri aktarımı gerçekleştirilebilir.

Tüm bu teknik özelliklerin yanında PHP, ekonomik, tüm eklentileri ile birlikte ücretsiz ve diğer gelişmiş özellikleri nedeniyle projenin yazılacağı script dili olarak seçilmiştir.

5.3.. FTP

FTP (Fıle Transfer Protocol) internete bağlı bir bir bilgisayardan diğerine (her iki yönde de)dosya aktarımı yapmak için geliştirilen bir internet protokolü ve bu işi yapan uygulama programlarına verilen genel addır. FTP ilk geliştirilen internet protokollerinden biridir.[6]

FTP protokolü ile ;

1) Bir başka bilgisayardan bir başka bilgisayara dosya aktarımı yapılırken, o bilgisayar ile etkileşimi aynı anda bağlantı kurulur

2) Protokol ile sağlanan bir dizi komutlar yardımıyla iki bilgisayar arasında dosya alma/gönderme işlemleri yapılır.

Projede FTP ile php aracılığla dosya upload ve download işlemleri gerçekleştirilmektedir.

5.4.. SSL (Secure Sockets Layer, Güvenli Soket Katmanı)

Günümüzde WEB veya WWW *(World Wide Web),* internet üzerindeki en bilinen hizmet türüdür ve internet üzerindeki uygulamalardan tamamına yakını bu ortam üzerinde çalışmaktadır. WEB, TCP tabanlı bir hizmettir. Bu tür uygulamalarda güvenliği ağlamanın en yaygın yolu SSL *(Secure Sockets Layer, Güvenli Soket Katmanı)'Air.* IETF ayrıca SSL ile geriye dönük olarak uyumlu, genel bir standart tanımlamak için oluşturduğu TLS *(Transport Layer Securiy, Taşıma Katmanı Güvenliği)* isimli bir çalışma grubu da bulunmaktadır. SSL, Netscape tarafından öncülüğü gerçekleştirilmiş ve SSL v3.0 bir internet taslak dokümanı olarak sunulmuştur. TLS, SSL v3.1 olarak düşünülebilir. SSL, her ne kadar TCP ve uygulama katmanı arasında saydam bir güvenlik sistemi olarak WEB dışındaki protokollere de uygulanabilir olsa da, pratik uygulamaları WEB ile sınırlı kalmıştır.

WEB sayfalarında, veya html dokümanlarında bağlantılar A *(Anchor)* etiketleri *(Tag)* ile belirtilirler. Tipik olarak bir html dokümanından başka bir dokümana veya siteye link <a href="http://sevketkeser.net/project/ossdt/"> şeklinde verilir. Bu tür bir bağlantıya kliklendiğinde tarayıcı *(internet Explorer veya Netscape Navigator)* söz konusu sunucuya 80 numaralı TCP portu üzerinden bir veya birkaç bağlantı açar. A etiketlerinde 80 numaralı port üzerinde çalışan http protokolü dışında başka protokoller de kullanılabilir. Bu durumda tarayıcı program söz konusu protokole uygun yazılım parçasını veya müstakil programı çalıştıracaktır. SSL bağlantıları, A etiketinde, https ile başlayan URL'ler kullanılarak belirtilir. SSL, TCP/443 üzerinde çalışır. IE ve NN'm SSL'e dahili desteği mevcuttur. Bir SSL bağlantısı yapılacağı zaman genellikle tarayıcı program bir uyarıda bulunur, bağlantı gerçekleştirildiğinde, örneğin lE'de bu (Sürüm 5.x) sağ alt köşede kapalı bir asma kilit sembolü ile belirtilir. Asma kilit sembolü üzerine kliklenerek detaylı bilgi alınabilir. SSL bağlantısı kurulurken, kullanıcı ve sunucu arasında bir dizi uzlaşı işlemi gerçekleştirilir; uzlaşı başarısız olursa SSL oturumu kurulmaz, hiçbir bağlantı yapılmaz ve veri iletimi de gerçekleşmez.[7]

5.4.1 SSL Mimarisi

SSL, TCP üzerinde uçtan uca güvenli bir bağlantı hizmeti sunulması için tasarlanmıştır ve katmanlı bir yapıya sahiptir. SSL Kayıt Protokolü çeşitli üst seviye protokollerine temel güvenlik hizmetleri sunar. WEB sunucusu ve kullanıcı arasında etkileşimi sağlayan http protokolü SSL üzerinde çalışabilir. SSL parçaları olarak tanımlanmış olan üç üst seviye protokol bulunmaktadır: Uzlaşı Protokolü, CipherSpec Değişim Protokolü ve Uyarım Protokolü.

SSL Oturumu ve SSL bağlantısı, iki önemli SSL kavramıdır.

Bağlantı: Sunucu/Kullanıcı arasında uygun bir türde hizmeti sağlayan mantıksal birbağlantıdır *(Bir web sayfasının indirilmesi gibi).* SSL için bunlar uçtan uca ilişkilerdir.Bağlantılar geçicidir. Her bağlantı bir oturuma ilişkilendirilmiştir.

Oturum: Sunucu ve kullanıcı arasında bir ilişkilendirmedir. Oturumlar Uzlaşı Protokolü ilekurulurlar. Oturumlar, birden fazla bağlantı arasında kullanılabilen kriptografik güvenlik parametrelerini tanımlarlar. Oturumlar her bir yeni bağlantı için yeni güvenlik parametre uzlaşı işlemlerinin tekrarlanmasını engellerler.

İki taraf arasında *(Sunucu ve kullanıcı üzerinde http gibi uygulamalar gibi...)* birden fazla güvenli bağlantı kurulabilir. Teorik olarak ayrıca birden fazla oturumun da kurulması mümkündür ancak pratikte kullanılan bir özellik değildir.

Her oturumla ilişkilendirilmiş çeşitli durumlar bulunmaktadır. Bir oturum kurulduğunda anlık olarak okuma ve yazma *(Alış-Veriş)* için işletim durumları vardır. Ayrıca Uzlaşı Protokolü etkin iken pending okuma ve yazma durumları oluşturulur. Uzlaşı Protokolünün başarılı bir şekilde işletilmesinden sonra bekleyen okuma ve yazma durumları, güncel durumlar haline gelirler. Bir oturum durumu aşağıdaki parametrelerle tanımlar:

Oturum Belirteci: Sunucu tarafından seçilen, etkin veya kaldığı yerden devam edilebilen bir oturum durumunu belirtmek için kullanılan rastgele bir Byte dizisi.

Eşlik Sertifikası: SSL oturumundaki uçlardan birine ait bir X.509.v3 sertifikası. Bu durum birimi hükümsüz olabilir.

Sıkıştırma Yöntemi: Kriptolama işleminden önce veriyi sıkıştırmak için kullanılacak algoritma.

CipherSpec: Kullanılacak kriptolama *(DEŞ gibi)* ve karıştırma algoritmasını *(MD5 veya SHA-1 gibi)* tanımlar. Ayrıca karıştırma boyu gibi parametreleri de belirtir.

Ana Şifre: Sunucu ve kullanıcı arasında paylaşılan 48 Byte uzunluğunda bir şifredir.

Sürdürülebilirlik: Oturumun yeni bağlantıları başlatmak için kullanılıp kullanılamayacağını belirtmek için kullanılan bir sahadır.

Bir bağlantı durumu aşağıdaki parametrelerle tanımlanır:

Sunucu ve kullanıcı rastgele sayısı: Her bir bağlantı için sunucu ve kullanıcı tarafındanrastgele seçilen Byte dizileri. Sunucu yazma MAC şifresi: Sunucu tarafından gönderilen veri üzerindeki MAC işlemlerinde kullanılan gizli anahtar.

Kullanıcı yazma MAC şifresi: Kullanıcı tarafından gönderilen veri üzerindeki MAC işlemlerinde kullanılan gizli anahtar.

Sunucu yazma anahtarı: Sunucu tarafından kriptolanan ve kullanıcı tarafından çözülen veri için kullanılan konvansiyonel gizli anahtar.

Kullanıcı yazma anahtanrı: Kullanıcı tarafından kriptolanan ve sunucu tarafından çözülen veri için kullanılan konvansiyonel gizli anahtar.

Başlatım vektörleri: CBC çalışma şeklinde bir blok şifre kullanılırsa, her anahtar için bir Başlatım Vektörü *(Initialization Vector, IV)* tutulur. Daha sonra nihayi şifre metin bloğu bir sonraki kayıt için IV olarak kullanılmak üzere saklanır.

Sıra numaraları: Bağlantının her iki ucundaki taraflar, her bağlantıda alınıp verilen mesajlar için ayrı sıra numaraları tutarlar. Bir uç ChipSpec Değişim mesajı aldığında, uygun sıra numarası sıfıra ayarlanır.

Tüm bunlarla beraber, SSL iletişimi bir dizi hiyerarşi ve protokolce sağlanır.Bunun sonucunda 128 bitlik güvenli iletişim client ve server arasonda krulmuş olur.

5.5 OpenGPG

Bu bölümde, projenin temel bileşenleriden birisi olan “OpenGPG” ve teknik özellikleri üzerinde durulmuştur. OpenGPG keyleri projede önemli bir görev üstlenmektedir.Transferi yapılan datalar serverda OpenGPG keyleri aracılığı ile şifreli olarak saklanır.[8]

5.5.1 Şifreleme ve temel kavramlar

5.5.1.1 Açık Anahtarla Şifreleme

Geleneksel şifrelemede tek bir anahtar kullanılır. Bilgiyi gönderen kişi bir anahtarla gönderiyi şifreler alıcı ise aynı anahtar ile onu açar. Ancak böyle bir haberleşmede şifre üçüncü şahıslara geçtiğinde haberleşme de açığa çıkmış olur. Internet ortamında böyle bir sistemle anahtarı yollamak ise iletişim güvenliğini en baştan tehlikeye atmak olur.

Ancak açık anahtar denilen kavramla tek anahtarla şifreleme ve açmanın yukarıda anlatılan açıkları kapatılmıştır. Açık anahtarda bir anahtar çifti vardır. Biri sadece ve sadece gönderende kalır/kalmalı(gizli anahtar) diğeri ise herkese verilen bir anahtardır(açık anahtar).

Buradaki kavram esasında çok basittir, kişinin bir anahtar çiftine sahip olması lazımdır. Bir dosya şifreli bir şekilde yollamak istenirse yollamak istenilen kişinin açık anahtarı ile o dosyayı şifrelemek gerekmektedir. Alıcı ise bu şifreli dosyayı ancak kendi gizli anahtarı ile açabilir başka hiçbir yöntem ile şimdilik bu şifreli dosya açılamaz.(Belki NSA (www.nsa.gov) bunu açabilecek bir algoritma geliştirmiştir ancak o da hala bilinmiyor).

En önemli nokta bu anahtarların kesinlikle kendi özel bilgisayarınızda veya sunucuda üretilmesidir, private key iyi bir şekilde korunmalıdır.Kendi özel

bilgisayarınızda da güvenlik açığı bulunmamalıdır.Bunun yanında projede, anahtarlar serverda üretildiği ve tutulduğu için server güvenliği gerekli şekilde sağlanmıştır.

Açık anahtarı herhangi bir ortamda, diskette, internette bir web sayfasında, e-postanıza bağlayarak ya da internet üstündeki bütün anahtarların saklandığı bir sunucuya koyarak herkese verilebir.

5.5.1.2 Sayısal imza

Bir mesajın zannedilen kişi tarafından yollanıp yollanmadığı sayısal imza ile doğrulanır. Burada mesaj açık/şifrelenmemiş olarak yollanır ancak sonda bir sayısal imza bulunur. Bu sayısal imza yazılan mesajdan ve göndericinin kişisel anahtarından oluşturulur. Alıcı ise mesajın arada hiç kimse tarafından değiştirilmediğinin sağlamasını gönderenin açık anahtarı ile yapar.

5.5.1.3 Güvenlik ağı

Çift anahtarlı sistemin açığı da açık anahtarın gerçekten mesajı yolladığı sanılan kişi tarafından yollanıp yollanmadığıdır. Eğer açık anahtar birisi tarafından değiştirilirse 3. şahıslar size yollanan şifreli mesajları bu kişi alıp kendisinde bulunan gizli anahtarla açıp okuyabilir.

Yukarıda anlatılan güvenlik açığı PGP ve aynı şekilde GnuPG'nin çözümünde daha önceden güvendiğiniz kişilerin/kuruluşların, yeni kişilerin şifrelerini imzalayarak size gönderilmesiyle çözülmektedir. Yani yeni bir açık anahtar size daha önceden güvendiğiniz birisi tarafında imzalı bir şekilde yollanmalıdır. Bir çeşit referans verilmektedir, "Bu açık anahtarın sahibi yakinimdir" denmektedir.

Bu konuda daha ayrıntılı bilgi için :Using trust to validate keys|||||||| http://www.gnupg.org/gph/en/manual.html#AEN385 adresine bakılmalıdır.

5.5.1.4 Güvenlik açıkları

PGP'nin şimdiye kadar bir açığı bulunmamıştır.. Ancak Şubat 2002'de bulunan bir Truva atı sistemdeki gizli PGP anahtarlarını alıp bir ftp sitesine yolladığı görülmüştür. Eğer gizli anahtarınızı koruyan şifreniz kuvvetli olmazsa bu anahtarınız da açığa çıkabilir.

Yapılması epey zor da olsa bir başka güvenlik açığı klavyeye yapılan vuruşların belirlenmesi ya da VNC veya PCAnywhere türü bir programla ekrandaki görüntünün ve dosyaların olduğu gibi başka bir bilgisayara aktarılmasıdır.

5.5.2 GPG Ve Temel Bileşenler Kavram Anahtar Create Etme

Public Key(genel anahtar) Sistemi: Public key sistemi, asimetriktir ve bir anahtar zinciri kullanarak şifreleme yapar. Bu anahtar zincirinde 2 tür anahtar vardır; *public key* ve *secret key*(özel anahtar). PGP de "Keyrings" dizininde *pubring.ptr* dosyasında genel anahtar bilgisi,*secring.skr* dosyasında da secret key bilgileri bulunur.

Secret Key: Kişi kendisine gönderilen şifrelenmiş mesajı açmak için secret keyini kullanır.Bu kişiye özel olmalıdır,saklanmalıdır.

Public Key: Public key otomatik olarak secret keyden türetilir.Secret keyin tersine, adından anlaşıldığı gibi,geneldir. Dünyanın heryerindeki kişi ve anahtar sunucularına güvenle gönderilebilir. Mesajların sifrelenmesinde alıcının public keyi kullanılır.Örneğin herhangi biri size şifreli mesaj göndermek istediğinde,sizin public keyinize ihtiyacı vardır.Bu durumda Public keyinizi,anahtar sunuculardan (Eger eklemişseniz),yada direk sizden elde edebilir.Mesaj, public keyiniz ile beraber özel bir oturum anahtarı ile kodlanarak internet üzerinde size güvenli bir şekilde gönderilir.Tabi bu mesajı açabilecek tek kişi sizsiniz.Bunun için secret key dosyanız bilgisayarınızda bulunmalı ve bu anahtarın şifresini bilmelidir..

PGP de public key sistemiyle,geleneksel şifrelemenin tersine secret key değil, herkese açık olan public key gönderilir,bu da PGP nin geleneksel şifrelemeye göre daha güvenli olduğuna dair nedenlerden biridir.

Digital Signature: Digital signature ile ,gönderilen mesajın kime ait olduğu doğrulanmış olur. Yani sözkonusu digital signature,gündelik hayatta kullandiğımız

imzadan çok farklı değildir. Digital signature,mesajın imza sahibi tarafından yazıldığına dair kanıt olusturur. PGP ,Digital signature gönderirken,kişiye ait bilgileri(isim,email adres gibi) bir algoritmayla işlemleyerek,bir *hash* oluşturur.Daha sonra,bu *hash* kişinin secret keyi ile şifrelenir. Böylece signature, kişiye özelleştirilmiş olur.Yani imza sadece mesajın sahibi tarafından atılabilir ve imzanın doğruluğu,kişinin public keyi ve digital signature ın işlemlendiği bir algoritmayla alıcı tarafından kontrol edilir.

5.5.3 Gnupg Uygulaması ve Anahtar Yönetimi

Şimdi Bütün anlatılanları,Unix tabanlı sistemlerde kullanılan "gnupg" programıyla örnekleyelim.

5.5.3.1 Anahtar Çifti (Keypair) Oluşturulması:

Yeni bir anahtar çifti oluşturmak için *--gen-key* parametresi kullanılır.Bu komut ilk kullanıldığında Gnupg, home dizininizde ".gnupg" dizini ni oluşturacaktır. gpg public key,secret key dosyalarını ve diger gpg ile ilgili dosyaları ".gnupg" dizininde oluşturacaktır.
NOT: *"No such file" gibi bir hata verildigi durumda,gpg ".gnupg" dizini oluşturmamıştır,siz kendiniz oluşturmalısınız (mkdir .gnupg).*
Örneği inceleyelim:

swordfish~root# gpg --gen-key

gpg (GnuPG) 0.9.4; Copyright (C) 1999 Free Software Foundation, Inc.This program comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under certain conditions. See the file COPYING for details.

Please select what kind of key you want:

(1) DSA and ElGamal (default)

(2) DSA (sign only)

(3) ElGamal (sign and encrypt)

Your selection? 1

Burada (1) seçeneği öntanımlıdır ve gnupg tarafından önerilen seçenektir... (1) seçeneğinin seçilmesiyle 2 keypair oluşturulur.DSA keypairi birincil/master keypair ve signature (imza) oluşturmak için kullanılacaktır.ElGamal ikincil/sub keypairi ise aynı zamanda şifreleme ve çözme işlemleri için kullanılır.Diğer bölümleri (1) seçeneğini seçtiğinizi varsayarak anlatılacaktır.

About to generate a new ELG-E keypair.

minimum keysize is 768 bits

default keysize is 1024 bits

highest suggested keysize is 2048 bits

What keysize do you want? (1024)

Anahtar uzunluğu seçeceksiniz..DSA key uzunluğu 512-1024 bit arasında olmalıdır, elgamal için ise uzunluk farketmez..Key uzunluğunu 1024 bitten fazla seçtiğiniz durumda DSA key uzunluğu 1024 bit olurken,ElGamal key de seçtiğiniz uzunlukta olacaktır.Öntanımlı olarak key uzunluğu 1024 bittir fakat ,günümüz bilgisayarları hızı düşünüldüğünde 2048 bit i öneririm.Ayrıca gnupg anahtar uzunluğunun 768 bitten az olmamasını önerir..

Please specify how long the key should be valid.

0 = key does not expire

= key expires in n days

w = key expires in n weeks

m = key expires in n months

y = key expires in n years

Key is valid for? (0)

Burada da anahtarların expiration(sonlanma) zamanı seçilecek.Bu seçimde dikkatli olmalı.(1) seçeneği için seçilen sonlanma zamanı DSA ve ElGamal için de geçerlidir.Gnupg (0) seçeneğini önerir.

(0) : key herzaman geçerli

: key n gün sonra sonlanacak

w : key n hafta sonra sonlanacak

m : key n ay sonra sonlanacak

y : key n yıl sonra sonlanacak

You need a User-ID to identify your key; the software constructs the user id from Real Name, Comment and Email Address in this form: "(heinrichh@duesseldorf.de)Heinrich Heine (Der Dichter) "

Real name: Sevket Keser

isminizi girebilirsiniz...

You need a Passphrase to protect your private key.

Enter passphrase:

Seçeceğiniz passphrase sizin screet keyinizdir ve kesinlikle unutulmamalı.Seçimde dikkatli olmalısınız,herhangi bir uzunlukta olabilir. Unutmayacağiniz ama tahmin edilmesi zor olan bir şifre seçimi yapın..Adınız ,numaranız gibi seçimler saf seçimlerdir,ona göre.. Bütün bu işlemlerden sonra ekranda şöyle bir uyarı belirebilir:

We need to generate a lot of random bytes.

It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 18 more bytes)

Bilgisayar çok sayıda ,farklı kombinasyonlarda random karakterler üretemiyordur.Bu durumda mause hareket ettirilip,ekrandan rastgele karakterler girebilirsiniz.Yada ,başka bir konsol açarak rastgele bir komut yazarak bilgisayar harddiskinin çalısması da işe yarayacaktır. Böylece bir anahtar çifti oluşturulmuş olur.

5.5.3.2 Anahtar Yönetimi - Revocatıon (İptal Etme) Belgesinin Oluşturulması :

Keypair oluşturulduktan sonra hemen bir revocation belgesi hazırlanmalıdır.Bunun için --gen-revoke parametresi yardımcı olacaktır.Şifrenizi(passphrase) unutur veya secret keyinizin herhangi birşekilde tehlikeye girmesi halinde,yapılacak en iyi ve tek Çözüm public keyi iptal edip,diğer iletişim kurduğunuz insanlara da bunu bildirmek olacaktır.Revocation belgesi de böyle bir durumda yardımcı olacaktır.Oluşturmanızda fayda var derim.

swordfish-root# gpg --output revoke.asc --gen-revoke mykey

"mykey" yerine secret keyinize özel herhangi birşey gelecektir.Bu UserID veya adresiniz olabilir.

Oluşturulan revocation belgesi,*revoke.asc* dosyasına bırakılacaktır.Revocation belgesi baskalarının ulaşamayacağı şekilde saklanmalıdır(şifrelenerek vs.).Çünkü revocation belgesini ele geçiren biri belgeyi public hale getirerek public keyinizi iptal edilebilir

5.5.3.3 Encryptıon Ve Decryptıon (Şifreleme ve Çözme):

Encryption Türkçe açıklamasıyla şifreleme,gizleme olarak ifade edilebilir.Decryption ise,encrypt edilmiş bir dökümanın açılması,okunabilir hale getirilmesi ,orjinalinin elde edilmesi olarak açıklanabilir.

secret key ve public key encryption/decryption olaylarında çok önemlidir.Bir döküman alıcının public keyi ile encrypt edilir ve alıcı da bu dökümanı kendi secret keyi ile açabilecektir.Bir örnekle açıklayalım :

swordfish-root# gpg --output doc.gpg --encrypt --recipient alici@gnuteam.org doc

Encryption için --encrypt parametresi kullanılır ve aynı zamanda alıcınınn public keyine

de ihtiyacınız vardır.Kişinin public keyi daha önceden import edilmiş olmalıdır

alici# gpg --output doc --decrypt doc.gpg

You need a passphrase to unlock the secret key for

user: "Alycy (Executioner) alici@gnuteam.org "

1024-bit ELG-E key, ID 5C8CBD41, created 1999-06-04 (main key ID 9E98BC16)

Enter passphrase:

Mesajın decrypt edilmesi için --decrypt seçeneği kullanılır.Kişi secretkeyi ile mesajı decrypt edebilecektir.

Ayrıca public key kullanmadanda mesajınızı,dosyanızı encrypt edip gönderebilirsniz.Bu yönteme "Symmetric Encryption(simetrik şifreleme)" deniyor.Dökğmanı bir algoritma ile şifreleyen Symmetric encryption çok güvenli değildir,ama kişinin public keyini elde edemediğiniz ya da public key zahmetine girmek istemediğiniz durumlarda --symmetric seçeneğiyle işinize yarayacaktır

swordfish-root # gpg --output doc.gpg --symmetric doc

Enter passphrase:

Bu şekilde encrypt edilmiş bir dosya oluşturdunuz,bu dosyayı kişiye gönderebilirsiniz.Ayrıca bu yöntem (Symmetric Encryption) kendinize özel,önemli dökümanları şifrelemek için de kullanılabilir.

5.5.3.4.Public Keyin Export Edilmesi :

Diğer insanlarla (gpg ile) iletişime geçebilmek için public key kullanılacaktır ve iletişime geçeceğiniz insanların public keylerini bilmelisiniz,tabii aynı durum karşı taraf için de geçerlidir.Public keyimizi başkalarına veya bir keyservera gönderebillmek için,export işlemi yapılmalı.

swordfish-root # gpg --output skeser.gpg --export skeser@tekfen.com.tr

Yukarıdaki komutla birlikte key,binary formatta export edilir.Bu da mail göndermede veya public keyi bir web üzerinde yayınlamada zahmetli olacaktır..Öyleyse asci formatta export edelmelidir Burada --armor parametresi kullanılacak.

swordfish-root# gpg --armor –export skeser@tekfen.com.tr

-----BEGIN PGP PUBLİC KEY BLOCK-----

Version: GnuPG v0.9.7 (GNU/Linux)

Comment: For info see http://www.gnupg.org

[...]

-----END PGP PUBLİC KEY BLOCK-----

5.5.3.5.Public Keyin Dağıtılması Ve Bulunması:

Public key email veya diger iletişim yollarıyla insanlara iletebilir.Fakat bu şekilde tek tek biraz zahmetli olacktır ,özellikle iletişimde bulunduğunuz insanlar çok fazla ise. Bu durumda keyin kopyasını web sayfanıza yerleştirebilir ya da en iyi çözüm olarak keyserverlara gönderilebilir Public key Server(sunucu)ları adından da anlasıldığı gibi,public keyleri toplar ve dağıtır. Keyservera basvurarak bir publickeyi arattırabilir veya kendi public keyinizi bu serverın databaseine eklettirirsiniz. Keylerin, bu keye güvendiğini göstermek isteyenler taafından sık sık imzalandığını(signing) düşünülürse keyserverlar , kişinin public keyinin en son halini elde etme bakımından da yararlıdır. Bir keyservera public keyinizi göndermek için --send-key parametresi kullanılır. public keyi ascii formatta gönderebilmek için de, *export* işleminin (--armor parametresiyle) yapıldığını daha önce söylemiştik

swordfish~root# gpg --keyserver www.keyserver.net --send-key ***skeser@tekfen.com.tr***
gpg: success sending to 'certserver.pgp.com' (status=200)

Bir keyserverdan public key almak için --recv-key parametresi aradığınız public keyin userID si ile kullanılacaktır.

swordfish~root# gpg --keyserver www.keyserver.net --recv-key 0xBB7576AC
gpg: requesting key BB7576AC from certserver.pgp.com ...
gpg: key BB7576AC: 1 new signaturegpg: Total number processed: 1 gpg: new signatures: 1

Herhangi bir keyserverdan alınan public key, keyringimize servera bağlı olarak ,import edilmiş olabilir yada serverdan aldığımız public key dosyasını kendimiz ,keyringimize import ederiz. Bu durumda "gpg --key-lists" komutu ile public keyin ,import edilip,edilmediği öğrenilebilir

5.5.3.6 Public Keyin İmport Edilmesi

Diğerleriyle güvenli iletişimde kullanılan encryption için,public keylerine ihtiyacınız olacağını artık biliyorsunuz. Public keyleri ,keyserverlardan,direk kişiden mail,posta vs..gibi çeşitli yollardan elde edilebilir.. Kişinin public keyini kendi keyringinizde geçerli hale getirmek için,import işlemi yapılamalı... Bir örnekle açıklayalım:

swordfish~root# gpg --import Someones_publickey

gpg: key 9E98BC16: public key imported

gpg: Total number processed: 1

gpg: imported: 1

--import seçeneği ile Someones_publickey import edildi.Ayrıca --list seçeneğiyle ,"Someones_publickey" in keyringimize eklenmiş olduğunu görürüz.

swordfish~root# gpg --list-keys

/users/ozlem/.gnupg/pubring.gpg

pub 1024D/BB7576AC 1999-06-04 Sevket Keser (skeser)

sub 1024g/78E9A8FA 1999-06-04

pub 1024D/9E98BC16 1999-06-04 Someone

sub 1024g/5C8CBD41 1999-06-04

5.5.3.7 Signature(İmzalama):

Bir imza public key/secret key anahtarçiflerinin ,encryptiondakinden farklı bir işleme tabii tutulmasıyla oluşturulur.Dökümanın sahibi ,kendi secret keyi ile imza olustururken,alıcı da ,gönderenin imzasını,kişinin public keyi ile kontrol edebilir. Signature (imza) oluşturmak için --sign parametresi kullanılır

swordfish~root# gpg --output doc.sig --sign doc

You need a passphrase to unlock the private key for

user: "Sevket Keser (skeser) "

1024-bit DSA key, ID BB7576AC, created 1999-06-04

Enter passphrase:

Döküman ,imzalanmadan önce sıkıştırılır ve çıktı da binary formatta olur.İmzalı bir dökümanda signature kontolu yapmak için --verify parametresi kullanılır. --decrypt seçeneği ile hem imza kontrolü yapılır hemde döküman açılır.

someone#gpg --output doc --decrypt doc.sig

gpg: Signature made Fri Jun 4 12:02:38 1999 CDT using DSA key ID BB7576AC

gpg: Good signature from "Sevket Keser (skeser) "

5.5.3.8 Public Keyin İmzalanması :

Mesajlarımızı secret key ile encrypt edip göndermek cesitli sebeblerden dolayi yeterince güvenli değilfir. Public keyi başkalarına dağıttığınızda birincil ve ikincil anahtarın genel özellikleri de dağıtmış olur. (örneğin userID).Bu özelliklerin dağıtılması da güvenlik açısından riskli olacaktır.Açık olarak dağıtılan public keyi herhangi biri üzerinde kurcalama yapılabilecek yada kendi public keyini sizinkine göre modife edebilecektir. Öyleyse key criptografyasından bahsetmek gerekir. Public key ,birincil secret key ile sign edilerek,özellikleri birincil key ile bağlanmış olur böylece public key dışarıdan kurcalamalara karşı korunur. Public key özelliklerinin bu şekilde birincil secret key ile sign edilmesine "Self Signing" denir. Keysigning ayrıca "web of trust" olarak adlandırılan bir modelde, key güvenirlğini artırmak için kullanılır. Bu modele göre bir public key ,keye güvenen kişiler tarafından imzalanarak ,keyin geçerliliği,güvenirliği artırılmış olur.

5.5.3.9 Web Of Trust Modeli:

İletişime geçtiğimiz insanların public keylerini email ile veya keyserverlardan kolayca elde edebileceğimizi belirtmiştik.

"gpg --keyserver www.keyserver.net --recv-key 0xBB7576AC"

komutu ile bir keyserverdan public key dosyasını indirebiliriz.Burada aradığınız public key in ID si veya diğer özelliklerden birini kullanılmalıdır.Dosya indirildikten sonra ,bu keyi kendi public key ringimizde geçerli hale getirmek için "import" işlemi yapılacaktır.

Bir public key, bu keye güvenen insanlar tarafından sign edilir,sorumluluk artık bu insanlarındır. Bu yüzden key imzalarken dikkatli ve dürüst olunmalıdır.Web of Trust modeli,public key ile keye ne kadar güvendiğinizi gösteren (sizin seçtiğiniz) belirteç atasındaki ilişkiye bakarak geçerliliği sağlayacaktır.Bu cümleyi biraz daha açarsak. Web of trust modeline göre,güvenirlik 4 düzeye ayrılır:

1.Unknown :Key sahibi hakkında hiçbirşey bilinmediğini gösterir

2.None:Key sahibine güvenilmediğini gösteriyor.

3.Marginal:marginally güveniliyor.Biraz tanıyorum gibi bişey...

4.Full:tam olarak güveniyorum.

GnuPG key editöründen bir publickey için trust(güvenirlik) ayarlarını yapılabilir

swordfish~root# gpg --edit-key murat

pub 1024D/8B927C8A created: 1999-07-02 expires: never trust: q/f

sub 1024g/C19EA233 created: 1999-07-02 expires: never

(1) Murat(Executioner)

Burada *murat* publickeyi bilgilerine bakacak olursak ; pub,birincil public key,sub da ikincil public key anlamına geliyor.expire zamanları gösteriliyor.Bir de trust seçeneğini görüyoruz (trust :q/f).q birinci seçeneği *skeser* tarafından seçilen güvenme düzeyidir(trust level) ve unknown anlamına gelir. ikinci seçenek (f) ise seçilen trust levela göre ayarlanan geçerlilik durumudur ve fully (tam) geçerli anlamına gelir. (q): unknown(bilinmiyor),(n): none(güven yok),(m): marginally,(f) : fully(tam güven)

Geçerlilik durumu ayarları ise şöyle yapılır ;bir key şu durumlarda keyringinizde geçerli hale getirilir:

1.Keyi kişisel olarak kendiniz imzalarsanız

2.Key bir tane fully trusted sign (tam güvenilir imza) veya 3 tane marginally trusted sign içerirse,web of trust modeline göre geçerlilik ayarı yapılır. command satırında trust komutuyla,bir keye karşı trust level ayarlarınızı yapabilirsiniz.

Command> trust

pub 1024D/8B927C8A created: 1999-07-02 expires: never trust: q/f

sub 1024g/C19EA233 created: 1999-07-02 expires: never

(1) Murat(Executioner)

Please decide how far you trust this user to correctly

verify other users' keys (by looking at passports,

checking fingerprints from different sources...)?

1 = Don't know

2 = I do NOT trust

3 = I trust marginally

4 = I trust fully

s = please show me more information

m = back to the main menu

Your decision? 3

pub 1024D/8B927C8A created: 1999-07-02 expires: never trust: m/f

sub 1024g/C19EA233 created: 1999-07-02 expires: never

(1) Murat(Executioner)

Command>
quit [...]

skeser kullanıcısı murat publickeyinin trust levelini m(marginally) olarak değiştiriyor.

5.5.4 GNUPG'nin Diğer Araçlarla Kullanılması

Bu bölümde gpg'nin grafik arabirim yardımıyla ve mail istemcilerinde kullanılması üzerinde durulmuştur.

5.5.4.1 GPA - GNU Privacy Assistant

GPA, UNIX sistemlerde GnuPG'yi kullanabileceğiniz bir grafik arayüzdür. Bu program ile yukarıda komut satırında yaptıklarınızı fare yardımı ile birkaç tıklama ile yapabilirsiniz. Ama daha geliştirme aşamasında olduğundan kullanırken gene de dikkat etmek gerekli olduğu http://www.gnupg.org/gpa.html sitesinde açıklanmaktadır. Bu programın nasıl kullanıldığını açıklamaya pek gerek yok çünkü yukarıda komut satırında yaptıklarımızın aynısını sadece güzel bir arayüzde yapmamızı sağlayan bir programdır.[9]

5.5.4.2 E-Posta İstemcisi Programlarında Gnupg Kullanımı

Anahtarlarınızı ürettikten sonra gelişmiş e-posta programlarında bu sistemi kullanmak epey kolaydır. Ancak şifrelenmiş bir e-postayı uzaktan erişerek bir başka sunucuda açmak güvenliğiniz için tehlikelidir. Bu işlemlerin hepsini masaüstü bilgisayarınızda yapmanız önerilir. Bir çok popüler e-posta istemcisi GnuPG'yi desteklemektedir. Hangileri olduğunu kullandığınız e-posta istemcisinin web sayfasından veya birçok e-posta istemcilerinin hangi plug-in'lerle çalıştığını gösteren http://www.geocities.com/openpgp/courrier_en.html adresinden ulaşabilirsiniz. Sylpheed gibi başarılı e-posta istemcilerinde GnuPG desteği hala deneysel olduğundan dolayı bu gibi e-posta istemcilerini GnuPG ile kullanırken dikkat etmek gerekiyor.

GnuPG veya PGP'yi destekleyen e-posta istemcilerinin hepsi aynı mantıkta çalışmaktadır. İlk önce yukarıda anlatıldığı gibi anahtar çiftinizi yaratmalısınız ardından e-posta istemcinize kullandığınız kişisel pgp/gnupg kimliğinizi girmelisiniz. Şifreli

yazışmak istediğiniz kişilerin açık anahtarları sizin açık anahtar veritabanınızda bulunması gerekiyor. E-posta istemcinizin öntanımlı ayarlarında bir değişiklik yapmadıysanız e-postanızı göndermeden önce şifreli ve/veya imzalı olup olmayacağını işaretleyiniz, eğer gönderdiğiniz adresin açık şifresi sizde bulunuyorsa bu isteğiniz yerine gelecektir.

Dikkat edilmesi gereken bir başka konu, genelde e-posta istemcileri e-posta ile beraber yollanılan (attach) dosyaları şifrelememektedir. Yollamadan önce sözkonusu olan dosyayı yukarıda anlatıldığı gibi alıcının açık anahtarı ile şifrelemeniz gerekmektedir.

Aşağıdaki listede Gnupg'yi destekleyen e-mail istemcilerinin bir listesi bulunmaktadır.

- KMail 2.2.x
- 2.2 Evolution 1.x
- 2.3 Mozilla/Netscape
- 2.4 Outlook
- 2.5 Outlook Express 5.x/6.0

Sonuç olarak diyebiliriz ki ; GnuPG güvenli iletişim için önemli bir araçtır. GnuPG programı hem Windows'da, hem de Linux'da çok rahatlıkla kullanılabilmektedir. E-posta istemcilerinizde kullanmak için de KMail ve Evolution'da gerekli araçlar gömülü olduğundan dolayı diğer plugin gerektiren e-posta istemcilerine göre GNU/Linux gibi, kurulumu ve kullanımı daha kararlı ve kolaydır.

Projede ise Gnupg serverda dataların şifrelenmesi sırasında kullanılmıştır.

5.6 MySQL Veritabani

MySQL, T.c.X DataKonsultAB firmasi tarafindan üretilmis, performansi cok yüksek bir SQL veritabanidir. MySQL son zamanlarda hazirlanan veritabani destekli

websitelerde üstün performansini ortaya koyarak, kücük ve büyük boyuttaki projeler icin yeterli oldugu ispatlamistir.[10]

MySQL yüksek performansını multithreaded calismasiyla sağlamaktadır. Multithreaded nedir? Multithreaded özelligi olan bir program icinde, programsin cesitli bölümleri ayni anda paralel olarak calisabilme özelliğidir. Multithreading programin hizini ve performansini arttirir. Multithreaded özelligi, MySQL veritabanina ayni anda birden fazla kullanicinin baglanip, sorgulama (query) yapmasi imkani verir.

SQL (Strutured Query Language), SQL bazinda calisan veritabanlari (örnegin Oracle, MSSql) icin baglayici bir standarttir. Bir veritabani sisteminin "ben SQL standartina uygunum" diyebilmesi icin SQL Entry Level olarak bilinen SQL komut/direktif/fonksiyonlari kümesini tanimasi gerekmektedir. MySQL cok zengin bir SQL komut/direktif/fonksiyon kümesine sahiptir ve SQL standardina uyumludur. MySQL bir düzineye yakin veritipi ve cesitli SQL fonksiyonlari icermektedir. MySQL yaptigi eklemelerde SQL standartini genisletistir. ENCRYPT, WEEKDAY, IF, AUTO_INCREMENT, LAST_INSERTE_ID bunlardan bazilaridir.

MySQL icinde kasitli olarak, performans düsüsünü engellemk icin, SQL standartinin öngördügü bazi komut/direktif/fonksiyonlar kullanilmamistir.

MySQL veritabani sisteminin bazi özelliklerini söyle siralayabiliriz:

- Multithreaded. Ayni anda birden fazla kullanici baglanti yapabilir.
- Her MySQL bilgibankasi 50.000.000 kayit (record = satir) icerebilir.
- Cok hitli SQL komut sürümü.

MySQL'i hizi ve performansi yaninda projede tercih edilmesini sağlayan en önemli özelligi bedava olmasidir. T.c.X firmasi herhangi bir ücret almadan MySQL veritabanini General Public Lisansi (Açik Kod)altinda kullanima sunmaktadir. Özel sahislar ve firmalar ücret ödemeden MySQL'i kullanabilirler.

Öncelikle kullandiginiz Linux sürümüne bagli olarak gerekli MySQL sürümünü http://www.mysql.com adresinden tedarik edilebilir.Projede Gelecek 2.0 ile beraber gelen MySQL sürümü kullanılmıştır.

5.7.. Expect Scriptleri

Expect scriplreri Linux altında kullanılan ve komut satırında onceden hazırlanmış girdilere göre işler yapılmasını sağlayan program parçacıklarıdır.Projede üç adet expect scripti kullanılmış ve bunlarla ilgili ayrıntılı bilgi 6. bölümde verilmiştir.

Expect Scripti oluşturmak için

[swordfish@root]#autoexpect –f script_name.exp

komutu kullanılır.

Daha sonra bu scripti çalıştırmak içinse

[swordfish@root]#expect script_name.exp

komutu kulalnılmaktadır.Expect scriptleri yardimiyla komut satırında daha bir çok uygulama gerçekleştirilebilmektedir.

6. Internet Ortamında Linux Işletim Sistemi Tabanlı Güvenli Data Transferi (Open Source Security Data Transfer – OSSDT)

Bu bölümde proje; teknik altyapısı ve çalışma sisitemiyle birlikte ayrıntılı olarak anlatılmıştır.

6.1 OSSDT - Proje Mimarisi ve Teknik Özellikler

Projede her adımda güvenlik ön plandadır.Sistem taşınacak data bakımından esnek bir yapıya (.doc.txt e-mail) sahiptir.Desteklenen dosya sistemlerinin listesi aşağıda görülmektedir.

- Doc => Word dosyası
- `txt =>` text
- `php =>` php dosyası`.gif',`
- `inc =>` php header dosyası
- `zip =>` sıkıştırılmış formatta dosya
- `bz2 =>` sıkıştırılmış formatta dosya
- `gz =>` sıkıştırılmış formatta dosya
- `ps =>` post script dosya
- `pdf =>` pdf dodyası
- `png =>` resim dosya formatı
- `bmp =>` resim dosya formatı
- gif `=>` resim dosya formatı
- `jpg` => resim dosya formatı

Kullanılacak dosya koruma yontemiyle, her türlü veri, güvenli bir şekilde şifrelenebilir ve yine guvenli şekilde istenen kullanıcıya erişimi sağlanabilir. Proje sırasında sourceforge.net [11] ve freshmeat.net [12]sitelerinden yarlanılmıştır.

Projenin üzerine kurulduğu kavramlar aşağıda maddelenmiştir.

6.1.1 Platform Bağımsızlığı ;

Kullanılacak yöntem sayesinde hemen hemen her dosya sistemindeki data kullanılabilir.Sistem server – client tabanlı olması nedeniyle, hizmet alacak kullanıcılara hiçbir teknik yük getirmez (istemcilerde program kurma vs..) Sıradan PC sahipleri rahatlıkla OSSDT'i kullanabilirler.

6.1.2 Güvenlik ;

Sistemde güvenlik hayati önem taşımaktadır.Aşağıda sistem güvenliği hakkında ayrıntılı şekil ve açıklamalar bulunmakatadır.

Sistem güvenliği aşağıdaki yontemlere oluşturulmaktadır.

6.1.2.1 Taşınacak veri güvenliği

Taşınacak verinin korunması uluslar arası bir standart ve algoritma kullanan PGP (OpenGPG) ile sağlanır.GPG ile ilgili ayrıntılı bilgi proje bileşenleri bölümünde ayrıntılı olarak anlatılmıştır.GPG programının getirdiği şifreleme yontemleri sayesinde taşınacak veride platform bağımsızlığı sağlanır,hemen hemen her türlü dosya yapısı, e-mail iletişimi etkin olarak korunur ve şifrelenir.

6.1.2.2 İstemci güvenliği

İstemci tarafında güvenlik; firewall kullanarak sistem yoneticisi tarafından sağlanır.Eğer söz konusu istemci bir kişisel bilgisayarsa güvenlik kullanılan işletim sitemi tarafından veya entegre kişisel firewall tarafından sağlanır.

6.1.2.3 İnternet ortamında güvenlik

İnternet ortamında güvenlik ise, uluslar arası sıtandart olan 128 bit (OpenSSL) şifrelemeye dayanır.SSL ile ilgili bilgi proje bileşenleri bölümünde ayrıntılı olarak verilmiştir.

6.1.2.4 Sunucu güvenliği

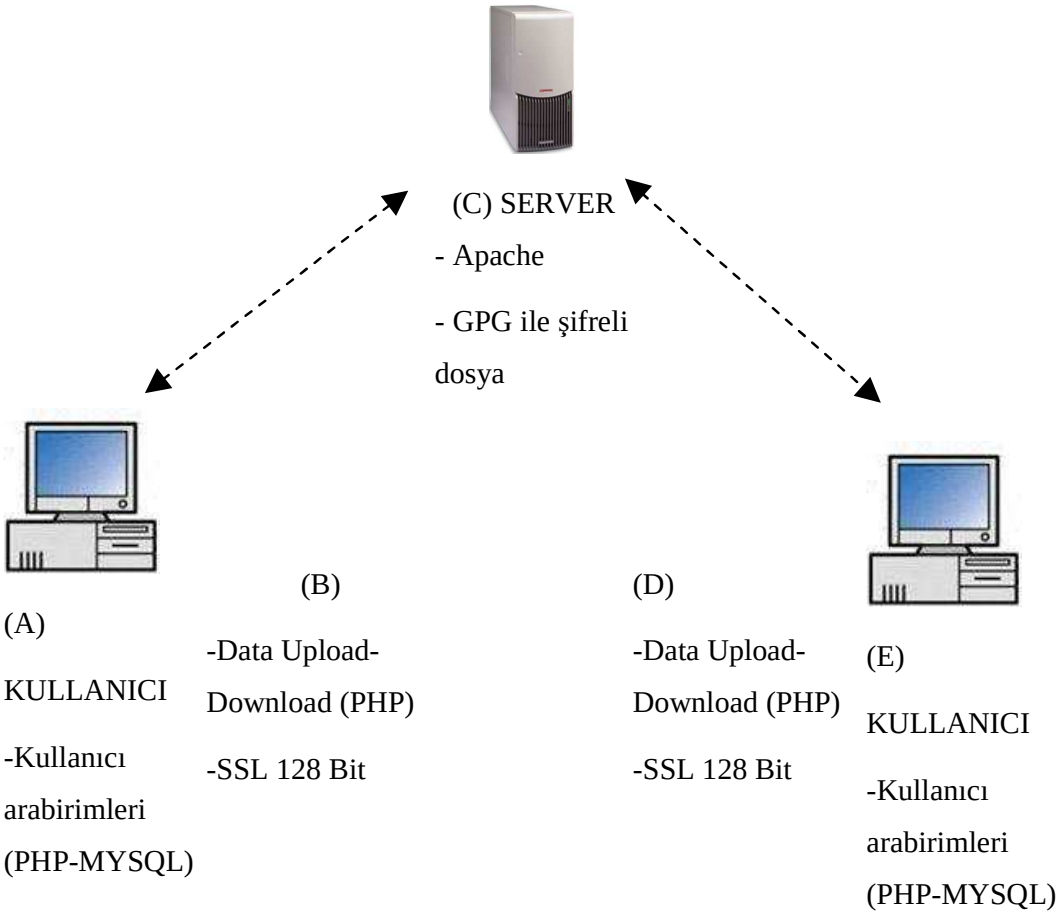
Sunucu güvenligi ise, sunucu üzerinde çalşan güvenlik duvarları (ipchains ve iptables) ve paket filtreleyicilerle sağlanır.Bunun dışında kullanılan upload sırasında kullanılan dosya oluşturma yöntemi sayesinde güvenlik daha da arttırılmıştır.

6.1.2.5 Programlama Tekniği ve Sağlamlığı

Sistemin işleyişinde ve kullanıcı arabirimleri tasarlamada kullanılacak programlama dili (PHP) ile veri tabanı dili (MYSQL) uluslar arası kabul görmüşlükleriyle sisteme ayrıca bir sağlamlık katmaktadırlar.

6.2 OSSDT – Genel Sistem İşleyişi

Sitemin çalışma mantığı aşağıdaki şekilde anlatılmıştır.



Şekil IV Proje Genel Şeması

6.2.1 (A) .. Kullanıcı Arabirimleri (Gonderen)

Kullanıcı kendisine sunulan özel web tabanlı arabirim sayesinde, göndermek istediği dosyayı server'a B arabirimi sayesinde SSL desteğinde gönderir.

6.2.2 (B) .. İnternet Uplaod

Aşağıda programda kullanılan upload fonksiyonun kodları verilmiştir.

client_display.inc dosyasından…

```
/* Upload bolumu !!! */
function phpftp_display_upload_bar ()
{
?>
<!-- Upload a file -->
<h2>Dosya Gonderme - File Upload </h2>
<hr>
<center>
<form enctype="multipart/form-data" name="ftp_upload" method="post" action=""> <!--action neden bos cunku kendi sayfasini cagirmakta -->
<input type="hidden" name="phpftp_cmd" value="put"/>
<input type="hidden" name="MAX_FILE_SIZE" value="10000000"/>
<!-- 10 MB Boyutunda ... formda kisitlama .. -->
<table style="width:560px">
<?php
/* display hidden login variables bu degiskenlere dikkat.. burdan datalar gitmekte.. */
phpftp_display_hidden_vars();
?>
<tr class="even">
<td style="width:150px">Gonderilecek Kisi</td>
<td>
```

```
        <input type="text" name="phpftp_cozdosya" size="30" value="" />
      </td>
    </tr>
   <tr class="odd">
    <td style="width:150px">Dosya Goderme</td>
    <td style="width:335px">
      <input type="file" name="phpftp_file" size="35"/>
    </td>
    <td style="width:75px">
      <input type="submit" value="Gonder"/>
    </td>
   </tr>
  </table>
 </form>
 </center>
<?php
 }
/**************Dosya gosterme fonksiyonu  bitti **********/
```

Yukarıdaki kod tarafından, upload için üretilen parametlreler client_command.inc dosyasındaki dizi değişkeni tarafından alınır ve yine o sayada verilen fonksiyonlar tarafından işlenir.

```
'file'      => $phpftp_file,
'from_file'  => $phpftp_from,
..
/* put a file */
   } else if ($args[0] == "put") {
```

```
    $command['cmd']  = "unsupported";
    $command['name'] = "put";
..
case "put":
      return $phpftp_conn->put($command['file'],
                        $command['put_name']);
```

Client_command.inc den gelen parametre ve secime gore client_connection_class.inc dosyasıdanki asagidaki kodlar çalışır.

```
  function put ($path, $name=false, $force_upload=false,
            $mode=false, $log_level=PHPFTP_LOG_ALL)
  {
    /*Daha Once Login OLunmus ise  */
    if (!$this->autologin())
      return false;
    /*isim kontrolu  */
        $uzunpath=$path;
    if (!$name) {
      $path_parts = pathinfo($path);
      $name = $path_parts['basename'];
    }
    /*Upload modu kontrol ediliyor.. */
    if ($mode != FTP_BINARY and $mode != FTP_ASCII) {
      $mode        = FTP_BINARY;
      $this->myMode = $mode;
    }
```

```
    if (!$force_upload) {
      if (!is_uploaded_file($path)) {
        $this->addErrorMessage(
          "<p class='error'>
           Hata  -Upload dosya path'in de problem var .. yeniden deneyim
           </p>", $log_level);
        return false;
      }
    }
/*************** Dosya Gonderme Blogu !!!!!! Scriptler Burda !!! **********/
           /* Upload the file
            * $name dosya adi*/
    if (!ftp_put($this->myConn, $name, $path, FTP_BINARY)) {
      exit;
      $this->addErrorMessage(
        "<p class='error'>
         Hata -- Dosya transferi gerceklestirilemedi... Error - Could not upload the file ($name), check the the
         specified path was correct. Path - $path
         </p>", $log_level);
      return false;
    }
/* sifreleme scriptleri is here !! */
          $source="/home/proje";
          $slash="/";
```

```
$bos=" ";
$exp="/usr/bin/expect";
$expect_location="/usr/local/bin/sifrele_php.exp";
$user="proje";
$upass=123456;
/*$sifrelenecek="/home/proje/skeser/komut.txt"; */
$homepath="/home/proje/.gnupg/";
$sahip="proje";
$sifrelenecek=("$source"."$slash"."$name");
$gpg=".gpg";
$sifreli_isim=("$source"."$slash"."$name"."$gpg");
$silinecek=$sifrelenecek;
$command=("$exp"."$bos"."$expect_location"."$bos"."$user"."$bos"."$upass"."$bos"."$sifrelenecek"."$bos"."$homepath"."$bos"."$sifreli_isim"."$bos"."$sahip"."$bos"."$silinecek");
$result=exec($command);
/*  ********    sifreleme kodları biti  -- end of cryted code !!  **********/
/*Temp file siliniyor  ediliyor ..*/
if (!$force_upload)
  unlink($path);
$this->addSuccessMessage(
  "<p class='success'>
  name : $name basariyla gonderildi... | <BR>
        temppathi: $path <BR>
  bide uzunpath : $uzunpath <BR>
        sifrelenecek :: $sifrelenecek <BR>
```

*sifreli isim :: $sifreli_isim
*

*sil :: $sil
*

sifreleme komutu :: $command

</p>", $log_level);

return true;

}

Kullanınıcı tarafından upload edilen data SSL desteğinde güvenli bir şekilde server'a ulaştırılır.

6.2.3 (C)..Server

Kullanıcıdan gelen veri burada PGP prgramı sayesinde şifrelenir ve burada bekletilir.Bu aşamada da güvenlik I. Öncelikte tutulmaktadır.Sistem Apache web sunucusu üzerinde çalışmaktadır.

Sifreleme Ve cozme kodlari asagida verilmistir..

Aşağıdaki kod, /usr/local/bin/ac_php.exp pathindedir ve php tarafından calistitirlip kendisine gelen parametreye gore sifreleme islemini gerceklestitir.

set user [lindex $argv 0]

set userpass [lindex $argv 1]

set acilacak [lindex $argv 2]

set acildi [lindex $argv 3]

set homepath [lindex $argv 4]

set sahip [lindex $argv 5]

set keypass [lindex $argv 6]

set silinecek [lindex $argv 7]

set timeout -1

spawn /bin/sh

match_max 100000

expect -exact "$ "

send -- "su - $user\r"

expect -exact "Password: "

sleep .2s

send -- "$userpass\r"

expect -exact "$ "

send -- "gpg --homedir=$homepath --no-secmem-warning --passphrase-fd 0 -o $acildi -r $sahip -d $acilacak \r"

expect -exact "..."

send -- "$keypass\r"

expect -exact "$ "

send -- "rm -f $silinecek\r"

expect -exact "$ "

send -- "exit\r"

expect -exact "$ "

send -- "exit\r"

expect eof

Aşağıdaki kod, /usr/local/bin/ sifrele_php.exp pathindedir ve php tarafından calistitirlip kendisine gelen parametreye göre desifreleme islemini gerceklestitir.

set user *[lindex $argv 0]*

set upass *[lindex $argv 1]*

set sifrelenecek *[lindex $argv 2]*

set homepath *[lindex $argv 3]*

```
set sifreli_isim        [lindex $argv 4]
set sahip               [lindex $argv 5]
set silinecek           [lindex $argv 6]
set timeout -1
spawn /bin/sh
match_max 100000
expect -exact "$ "
send -- "su - $user\r"
expect -exact "Password: "
sleep .01s
send -- "$upass\r"
expect -exact "$ "
send -- "gpg --homedir=$homepath --no-secmem-warning -o $sifreli_isim -e -r $sahip $sifrelenecek\r"
expect -exact "$ "
send -- "rm -f $silinecek\r"
expect -exact "$ "
send -- "exit\r"
expect -exact "$ "
send -- "exit\r"
expect eof
```

6.2.4 (D) .. Internet Download (Upload)

Serverda kullanıcı kimliğine göre ozel olarak şifrelenmiş dosyaya alıcı ulaşmak istediğinde,datanın şifresi severda çözülerek SSL desteğinde güvenli olarak alıcıya ulaştırılır.

Asagida Download yapan kodlar bulunmaktadır.

```
/*****     FILE DOWNLOAD *******/
/*        Burdan client_command.in dosoyasina 3 degiskengider ..
          1- phpftp_cozdosya    : cozulecek dosyanin adi..
          2- phpftp_cozsahip    : cozulecek dosyanin sahibi..
          3- phpftp_cozkeypass : cozen anahtarin paorolasi .. */
function phpftp_display_coz () {
?>
  <!-- Download a file -->
  <h2>Dosya Indirme - File Download</h2>
  <hr>
  <h4>Asagidaki alanlara indirmek istediginiz dosyanin bilgilerini girin..</h4>
   <form name="phpftp_display_coz" method="post" action="">
          <input type="hidden" name="phpftp_cmd" value="coz"/>
     <?php
      /* gizli giris degiskenlerini goster */
      phpftp_display_hidden_vars();
     ?>
<center>
     <table>
       <tr class="even">
```

```
        <td style="width:150px">Dosya Adi</td>
        <td>
          <input type="text" name="phpftp_cozdosya" size="30" value="" />
        </td>
      </tr>
<tr class="odd">
        <td style="width:150px">Yeni Ad</td>
        <td>
          <input type="text" name="phpftp_coz_newfilename" size="30" value="" />
        </td>
      </tr>
      <tr class="even">
        <td style="width:150px">Sahip</td>
        <td>
          <input type="text" name="phpftp_cozsahip" size="30" value="" />
        </td>
      </tr>
<tr class="odd">
        <td style="width:150px">Anahtar Parolasi</td>
        <td>
          <input type="password" name="phpftp_cozkeypass" size="30"/>
        </td>
      </tr>
    </table>
        <table width="400">
```

```
      <tr>
        <td style="width:200px" align="center">
          <input name="clear" type="reset" value="Temizle"/>
        </td>
        <td style="width:200px" align="center">
          <input type="submit" value="Decrypted File"/>
        </td>
      </tr>
    </table>
  </center>
 </form>
<?php
}
client_command.inc icerisinde.. dizi degiskenleri
 'coz_dosya'     => $phpftp_cozdosya,         /* download fonksiyonu degiskenleri */
‘coz_sahip'  => $phpftp_cozsahip,   /* download fonksiyonu degiskenleri */
'coz_keypass'=> $phpftp_cozkeypass, /* download fonksiyonu degiskenleri */
'coz_newfilename' => $phpftp_coz_newfilename,/* download fonksiyonu degiskenleri*/
/* dosya coz */
        } else if ($args[0] == "coz") {
          $command['cmd']  = "coz";
          $command['coz_dosya'] = $args[1];
          $command['coz_sahip'] = $args[2];
          $command['coz_keypass'] = $args[3];
          $command['coz_newfilename']=$args[4];
```

```
....
case "coz":
        return $phpftp_conn->coz($command['coz_dosya'],
                                 $command['coz_sahip'],
                                $command['coz_keypass'],
                                $command['coz_newfilename']);
client_connection_class.inc dosyasında ..
function coz ($coz_dosya, $coz_sahip, $coz_keypass, $coz_newfilename,
$log_level=PHPFTP_LOG_ALL)
        {
        /* Daha once login olunup olunmadigi session kontrolu */
    if (!$this->autologin())
     return false;
        $cwd = $this->myCwd;
        $home="/home/proje";
        $slash="/";
        $bos=" ";
        $exp="/usr/bin/expect";
        $expect_location="/usr/local/bin/ac_php.exp";
        $user="proje";
        $userpass=123456;
        $acilacak=("$home"."$slash"."$coz_dosya");
        $acildi=("$home"."$slash"."$coz_newfilename");
        $homepath="/home/proje/.gnupg";
        /* formdan gelen degiskenler */
```

```
        $sahip=$coz_sahip;
        $keypass=$coz_keypass;
        $silinecek=$acilacak;
        $command=("$exp"."$bos"."$expect_location"."$bos"."$user"."$bos"."$userpa
        ss"."$bos"."$acilacak"."$bos"."$acildi"."$bos"."$homepath"."$bos"."$sahip"."$
        bos"."$keypass"."$bos"."$silinecek");
        $result=exec($command);
        if(!$result){
        $this->addSuccessMessage(
      "<p class='success'>
       Basariyla cozuldu..$coz_dosya, $coz_sahip, $coz_keypass.
       </p>", $log_level);
   return true; }
else {
         $this->addErrorMessage(
       "<p class='error'>
        Error - $coz_dosya, $coz_sahip, $coc_keypass
                  <BR>
                  $command
        </p>", $log_level);
      return false;
exit;
    }
}
```

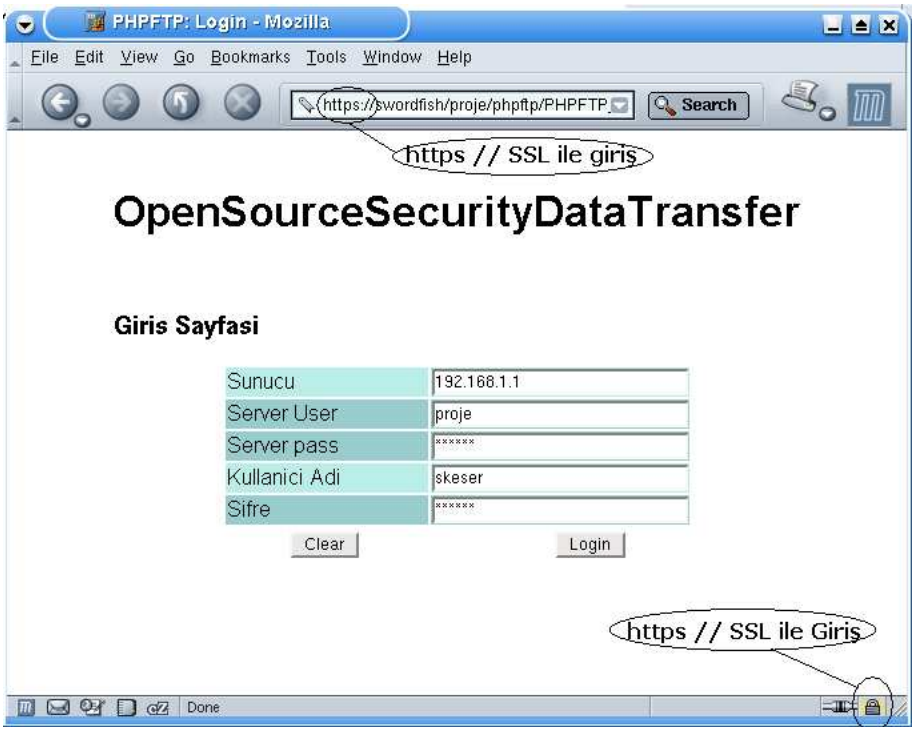
6.2.5 .. (E)İstemci Arabirilmi..(Alıcı)

Kullanıcı kendisine sunulan özel web tabanlı arabirim sayesinde,sisteme giriş yapar ve dosyalarına erişir.Eriştiği dosyaları serverdan güvenli olarak indirir.Diğer kullanıcılara dosya gonderebilir.

6.3 .. OSSDT – Ekran görüntüleri ile Örnek Senaryo

Giriş sayfası şekil I ' de görüldüğü gibisir.Kullanıcı gerekli kimlik bilgilerini sisteme girerek şekil Iide görülen home dizinine erişir.

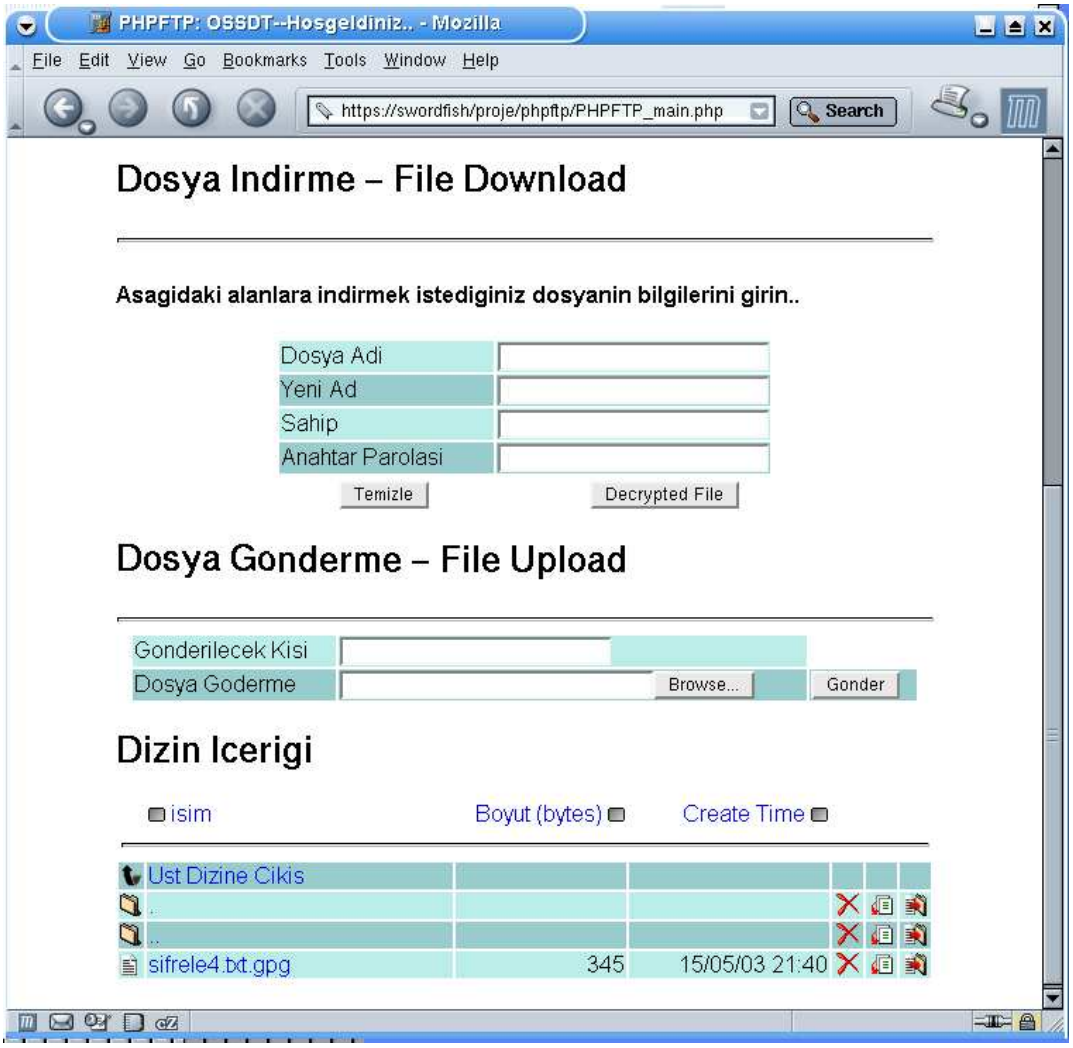
Diyelimki biz *skeser* kullanıcısıyla sisteme login olmak isteyelim aşağıdaki login ekranından girilecektir.Şekil üzerinde de gösterildiği gibi giriş SSL Desteğinde (128 bit şifreleme) yapılmaktadır. Bu konuyla ilgili ayrıntılı bilgi proje bileşenkleri bölümünde ayrıntılı olarak verilmiştir.



Şeki IV OSSDT Giriş sayfası

Aşağıda login olunduktan sonraki safha olan upload ve download alanlarıyla en altta ise login olunan kullanıcının dizin alanı görülmektedir.Dosya gönderme işlemi için

Browse butonuna tıklanıp diskten gönderilmek istenen dosyanın pathi şekilde görüldüğü gibi text box üzerine düşürülür.Gönderilecek kişinin kullanıcı adi da yazıldıktan sonra *Gönder* butonuna tıklanır.



Şekil V Kullanıcı Dizini

Upload isleminden sonra ekrandaki gibi sifrele.txt.gpg dosyası sifreli olarak görülmektedir.

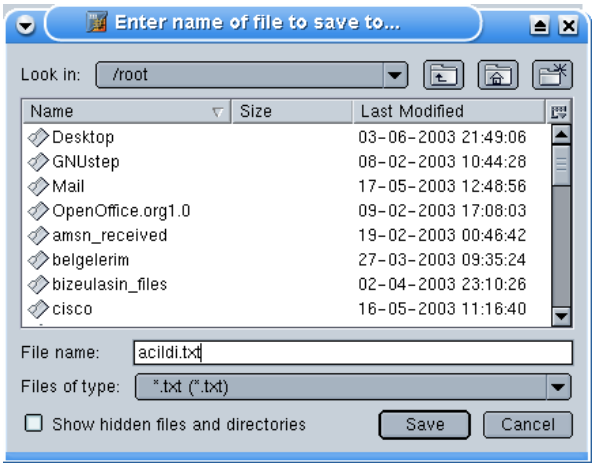
Bir sonraki sayfada ise dosyanın acilmasi anlatilmistir.

Açmak istediğimiz dosyanın ismi ve gerekli kimlik bilgileri şekilde görüldüğü gibi *Dosya İndirme* bölümüdeki gerekli yerlere yazılır ve *Decrypted file* butonuna tıklanır.Dosya program tarafından çözülür ve istenilen isimdeyine ekranda görüldüğü gibi yine diske yazılır.

Şekil VI :Decrypted file

Bu asamadan sonra acidi.txt dosyasının uzerine tıklanarak dosya diskte istenilen yere indirilebilir.

Aşağıda, indirilecek dosya için istemci bilgisayarında yer seçimi yapılır.



Şekil VII Dosya için yer seçimi

Şekil VIII : Diğer ozellikler

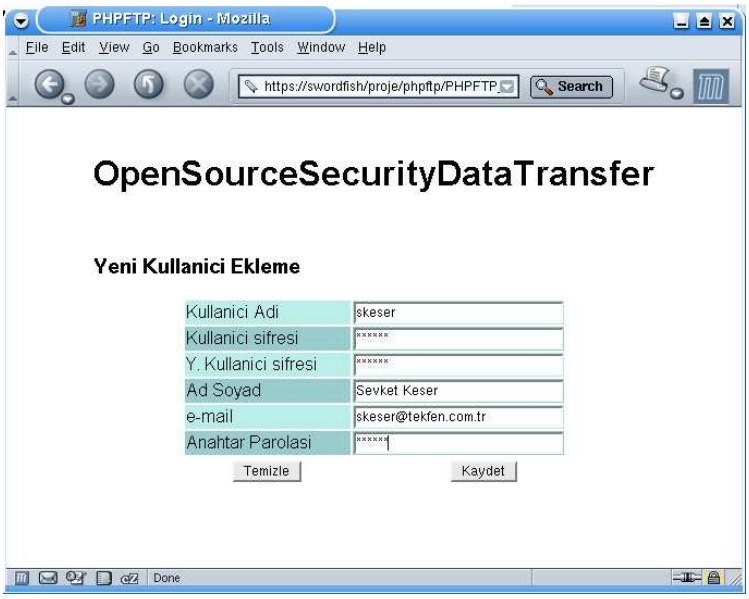
Yukarıdaki şekilde ise bilgilendirme mesajları , dizin oluşturma ve dizin değiştirme ekranları görülmektedir.Bilgilendirme mesajları bölümünden programın işleyişi sırasında alınabilecek olası hatalar vs.. ekrana bastırılır.Dizin oluşturma bölümündeyse,

kullancı yeni dizinler oluşturabilri.Dizin değiştirme bölümüyle ise, oluşturulan bu dizinler arasında dolaşım sağlanır.

Şekil IX Çıkış Ekranı

6.4 Yeni Kullanıcı Ekleme ve Key Oluşturulması

Aşağdaki resimde ise yeni kullanıcı ekleme bölümü görülmektedir.Ekranda görüldüğü gibi yeni kullanıcı bilgileri girilerek bilgiler mysql databaseine eklenir.



Şekil X Yeni Kullanıcı Ekleme

Aşağda ise yeni kayıt üretmek için genkey scriptlerini görmektesiniz.Bu scriptler yardımıyla yeni kullanıcılar için gpg keyleri oluştırulur.GPG keyleri oluşturmak ile ilgilia ayrıntılı bilgi proje bileşenleri kısmında verilmiştir.

/usr/local/bin/generate_key.exp

```
set adsoyad         [lindex $argv 0]
set email           [lindex $argv 1]
set username        [lindex $argv 2]
set ukeypass        [lindex $argv 3]
set timeout -1
spawn $env(SHELL)
match_max 100000
expect -exact "$ "
send -- "su - proje\r"
expect -exact "Password: "
sleep 0.1s
send -- "123456\r"
expect -exact "$ "
send -- "gpg --gen-key\r"
expect -exact "Seçiminiz? "
send -- "1\r"
expect -exact "(1024) "
send -- "1024\r"
expect -exact "(0) "
send -- "0\r"
expect -exact "(e/h)? "
send -- "e\r"
```

```
expect -exact "Soyadýnýz: "
send -- "$adsoyad\r"
expect -exact "E-posta adresiniz: "
send -- "$email\r"
expect -exact "Önbilgi: "
send -- "$username\r"
expect -exact "(T)amam/Ç?(k)? "
sleep 0.1s
send -- "T\r"
expect -exact "girin: "
send -- "$userkeypass\r"
expect -exact "Tekrar: "
send -- "$userkeypass\r"
expect -exact "$ "
send -- "exit\r"
expect eof
```

7. SONUÇ VE ÖNERİLER

7.1 Sonuçlar

İnternet ortamında, linux işletim sistemi tabanlı, cok kullanıcılı, güvenli data transferi uygulaması gerçekleştrilmiş, internet üzerinde güvenli bir ortam sağlanarak, bir yerden başka bir yere gonderilmek istenen verinin,(.txt,.doc ..vs) taransferi gerçekleştirilmiştir.Transfer edilmek istenen veri, ozel yontemlerle korunmuş ve şifrelenmiştir.Şifreli-korunan verinin internet ortamında güvenliği sağlanmış ve t asarlanan kullanıcı arabirimleri sayesinde, sistem kolay kullanılabilir hale getirilmiştir.Proje da amaçlanan tüm bu hedeflere ulaşılmıştır.

7.2 Öneriler

Projeye mevcut özelliklerinin yanında yeni özellikler de ekelenebilir.Bunlardan bir kaçı aşağıda verilmiştir.

- Daha fazla güvenlik, özellikle temp dizini altındaki mekanizma snifferlardan daha fazla korunmak için iyileştirilebilir.Bunun için FIFIO dosya yapısı kullanılarak testler yapılmalıdır.
- Yeni kullanıcı eklerken IP tabanlı bir güvenlik mekanizmasi apache'nin conf dosyaları aracılığıyla sağlanabilir.
- Her kullanıcı için istatistiksel veriler tutularak, projenin kullanım aşaması konusnda daha fazla veri edinilip, bu yönde yeni kararlar alınabilir.
- Sifreleme ve deşifreleme işlemleri scriptler yardimiyla değilde php tarafından yaptırılabilir.Bu da projenin sağlamlığını artıracaktır.
- Hata Kontrol mekanizmaları daha fazla kullanıcıyı bilgilendiren tarzda geliştirilebilir.

7.3. OSSDT Kulanım Alanları

"İnternet Ortamında Linux İşletim Sistemi Tabanlı Güvenli Data Transferi" , aşağıda belirtilen yerlerde kullanım alanına sahiptir.

- Büyük şirketlerin ve taşra şubeleriyle aralarında gerçekleşen her türlü veri transferinde (VPN maliyetlerinden kurtulmak için),
- Şirletkerin bürolari ve kendi iç birimleri arasındaki evrak transferi işelemlerinde,
- Devlet kurumları arasıdanki evrak transferinde,
- Kişisel kullanımlarda
- Üniversite ve Eğitim kurumları arasıdanki evrak transferi gibi alanlarda geniş bir kullanılabilirlik niteliğine sahiptir.

KAYNAKLAR

[1] “Serbest Yazılım Nedir?” http://cisn.odtu.edu.tr/2002-6/free.php

[2] Fatih Özavcı , “Neden, Ne Kadar Güvenlik ? ve Güvenlik Politikaları”, , www.siyahsapka.com

[3] Şevket Keser “Eğitimde Linux İşletim Sisteminin Kullanılması” Isparta 2003

[4] “PHP'ye Genel bakış”

http://hadi.ulakbim.gov.tr/myworks/internethaftasi2001/classic/php_2.myhtml

[5] Türkiye Linux Kullanıcıları Web Sayfaları www.linux.org.tr

[6] “FTP Nedir ?” http://bbankasi.nyg.ege.edu.tr/index.php

[7] Yasin KAPLAN “SSL (Secure Sockets Layer)”, www.yasinkaplan.net/

[8] Özlem ATAŞ “PGP(Pretty Good Privacy) NEDİR?” http://www.debian-tr.org/print.php?sid=4#PGP

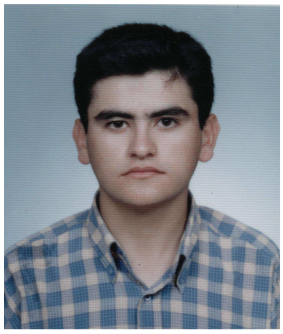
[9] Altay Ş. Özaygen “GPG - GNU Privacy Guard ile Güvenli İletişim” http://www.metu.edu.tr/~ozaygen/gpg.html

[10] MySQL Veri Tabanı , http://www.turkcgi.com/mssql7

[11] www.sourceforge.net

[12] www.freshmeat.net

[13] Luke Welling , Laura Thomson , “Php Ve Mysql”, Alfa Yayınları İstanbul 2002



Şevket Keser

KISA ÖZGEÇMİŞ

1981 Yılında Manisa ili Salihli ilçesi Yukarı Poyraz köyünde dünyaya geldi.

Sırasıyla; 1987 – 1992 Yukarı Poyraz Köyü İlk Okulunu, 1992 – 1995 Salihli 50. Yıl Orta Okulunu , 1995 – 1998 Salihli Türk Birliği Lisesini bitirdi.

Yüksek öğrenimine ise 1999 yılında başladığı S. Demirel Üniversitesi Teknik Eğitim Fakültesi Elektronik- Bilgisayar Eğitimi Bölümü Bilgisayar Sistemleri Öğretmenliği programında devam etmektedir.

YAZ STAJLARI

- 2001 – 2002 TEKFEN İnşat ve Tesisat A.Ş Bilgi Teknolojileri Bölümü
- 2002 – 2003 TEKFEN İnşat ve Tesisat A.Ş Bilgi Teknolojileri Bölümü

LİSANS TEZİ

- İnternet Ortamında Linux İşletim Sistemi Tabanlı Güvenli Data Transaferi (Open Source Security Data Transfer – OSSDT)

YAYINLAR

- VERİ TABANI SUNUCU KÜMELERİNDE YÜK DENGELEME SİSTEMLERİN BULANIK MANTIKLA MODELLENMESİ, Şevket Keser , Tuncay AYDOĞAN , Mehmet Albayrak , **XII International Twelfth Turkish Symposium on Artificial Intelligence and Neural Networks ,** Çanakkale Turkey 2003