Gbde ile FreeBSD sunucularının güvenliğinin arttırılması

Yazılarımda dikkat ederseniz devamlı şekilde FreeBSD`nin ne kadar güvenli bir işletim sistemi olduğundan bahsedip duruyorum.Hepimiz o veya bu şekilde güvenlik duvarları, router`lar, switchler ile ağımızı gelecek saldırılara karşı sanal olarak koruyoruz.Peki ya fiziksel güvenlik? Dolaştığım çoğu sunucu odalarında bu güvenlik çeşidinin hiç mi hiç dikkate alınmadığını gördüm.Dışarı ülkelerde "Disaster recovery" sözcüğü ile birlikte anılan hırsızlık gibi olaylar ülkemizden daha ciddiye alınmakta.Internet üzerinden ağınıza girmeyi başaramayan ama sunucunuzdaki bilgileri almayı kafasına koymuş olan bir kişi, istediği bilgilerin bulunduğu sunucularınızdan birisini açıp sabit diski çıkartıyor ve kayıplara karışıyor."Yok daha neler", veya "Adamın işi yok gelip benim sunucumdan sabit diskimi çalacak" şeklinde düşünenleriniz olabilir.Belki ufak ölçekli firmalarda kimsenin sizin diskinizle uğraşmak istememesi doğru olabilir fakat içinde yüzlerde kişisel bilgi ve kredi kartı numarası gibi değerli bilgilerin bulunduğu bir banka sunucusu bu kişiler için "Ağız sulandırıcı" şeklinde tanımlandırılabilinir.

Peki fiziksel olarak sunucu kapılarının kapatılması, anahtarın güvenli yerlerde tutulması gibi bütün önemler dışında yapılabilecek bir şey var mi?

Biz yine o paranoyak sistem yöneticisi tutumumuzu devam ettirip, oldu da birisi bütün aldığımız önlemleri geçip diskimizi çaldı diyelim.İşte o zaman bu makalede size bahsedeceğim yöntem bizi koruyacak.

Güvenlik konusunda paranoyak denebilecek seviyede birisi olarak disklerin şifrelenmesi konusunda araştırmalarım sonunda FreeBSD el kitabında da bahsedilmiş olan gbde programını bayağı başarılı buldum.

Uygulama seviyesinde şifreleme

Bu tür şifreleme uzun yıllardan beridir PGP ile uygulanılıp kullanılmakta.Tüm şifreleme ve şifrelerin açılmasından sorumlu olan kullanıcı her işlem başında şifresini girerek kimliğini doğrulamak zorundadır.

Dosya sistemi seviyesinde şifreleme

Çok fazla uygulanılmaya çalışılan bir yöntem olmakla beraber maliyeti ve "Sistem yöneticilerinin problemlerini arttıran işlemler" kategorisindeki yerini almıştır.

Her işletim sisteminin kendi dosya sistemini kullanması ve genel geçer dosya sistemlerinin çok az olması yüzünden şirket bazında şifrelemeye giriş, sistem yöneticisi olarak size baş ağrı kat sayınızın yükselmesi ve daha fazla mesai şeklinde geri dönecektir.

Diğer bir sorun ise veritabanı gibi önemli bilgilerin bulunduğu disklerin şifrelenmesindedir.Bu sistemlere test edilmeden yapılan şifreleme sonucunda yanlış giden bir şey bütün diske ulaşımınızın engellemesi anlamına gelebilir.

Sabit disk seviyesinde şifreleme

Disk seviyesindeki şifreleme işletim sistemi, dosya sistemine ve kullanılan programlara bakılmaksızın bütün verilerinizi koruyabilir.

Kullanıcı bakımından disk seviyesindeki bir şifreleme mekanizması kullanıcının sadece bir kere kimliğini belirtmesini gerektirir.Bundan sonra yapılacak tüm işlemler şifrelenmiş olarak diske kaydedilir.

Diğer şifreleme yöntemlerinin aksine bu tür şifreleme hangi sektörlerin kullanılıp hangi sektörlerin kullanılmadığını bilmek gibi bir özelliği yoktur.Bu yüzden uygulama ve dosya sistemi şifreleme yöntemleri her dosyayı teker teker anahtarlarken disk seviyesindeki şifreleme tüm sektörleri şifrelemektedir.

Disk seviyesinde şifreleme çalışmalarıyla kişiden alınan şifrenin bütün diski şifrelemede kullanılan anahtar olarak atanması yöntemi uygulanmaya başlandı.Bunun da kötü yönü sadece tek bir şifrenin kullanılabiliniyor olmasıydı.Bunu büyük organizasyon bazında düşünürsek; her disk için ayrı bir şifrenin oluşturulması ya elinizde yüzlerce şifrenin olması ve kullanılan şifrenin kullanıcıya özel olması sonucunda o iş istasyonunu şifreyi bilen kullanıcı dışında kimsenin kullanamaması gibi bir sorunla karşılaşabiliriz..

Disk şifreleme konusunda beklediklerimizi toparlayalım:

- ✓ Sabit disk seviyesinde şifrelemeyi desteklemeli.
- ✓ İkinci olarak güvenlik için aylık veya haftalık olarak değiştirilen şifreleri desteklemesi yanında bu değişiklik tüm diskin yeni şifre ile tekrar şifrelenmesi anlamına gelmemeli.
- ✓ Üçüncü olarak diskin şifrelenmesi için kullanılan şifrenin unutulması sonucunda diskteki bilgileri tamamen kaybetmek yerine bu disk üzerinde şifreyi açacak birkaç yöntemin daha bulunması.
- ✓ Son olarak kullanıcı için en iyi korumayı sağlamalı.

Gbde işte tam burada işe girip bu tür şifrelemenin hem kişisel hem de organizasyon bazında kullanabilir hale gelmesini sağlıyor.

Şifreleme kullanımında en zayıf halkayı kullanıcılar oluşturmaktadır.En zayıf halka olarak adlandırılmaları para veya kişisel fayda için şifrelerini satmaları değil, fiziksel şiddet veya şantaj yoluyla bilgilerinin kötü niyetli kişiler tarafından elde edilebilir halde olmasından kaynaklanmaktadır.İste bu yüzdendir ki bankalar açılması için iki kişiye ihtiyaç duyan sistemler kullanmaktadırlar.

Her şeyi bir yana bırakıp çok uzak bir tahmin gibi görünen bir senaryo düşünelim.Kullanıcılardan biri bilgileriyle yakalanıyor.Yapılaması gereken kullanıcıyı fiziksel şiddet vs... ile bilgiyi vermeye iterek elindeki bilgiyi öğrenmek.Burada kullanılan şifreleme sistemini de Steganographic dosya sistemi (STEGFS) olarak farz edelim.Stegfs kullanıcılara ayrı seviyelerde şifreleme yapma şansı vermektedir.Genel olarak mantık o kadar önemli olmayan dosyalarına alt, önemli olanlara ise üst seviyede koruma sağlayabilmektir.

Bunu öğrenen saldırgan elinde Stegfs ile şifrelenmiş bilgi ile yakaladığı kişinin birden fazla bilgiyi değişik seviyelerde saklayabileceği sansını da düşünüp, bütün şifreleri kullanıcıdan öğrendiğinden emin olana dek fiziksel şiddet vs. gibi islemelere devam edecektir.Tabiki

bilgiyi öğrenmeye çalışan kısım polis ise kullanıcıyı hiç kimseye hesap vermeden uzunca bir sure hapiste tutup ondan bu bilgiyi eninde sonunda alacaktır.

Daha kötüsü kullanıcı bütün şifreleri verse bile bildiği tüm şifreleri verdiğini gösterebilecek bir mekanizma Stegfs'de bulunmamaktadır.

Bu kulağa çok uzak gelse de çok önemli kategorisine, örneğin devlet sırları, giren bilgiler hesaba katıldığında bu senaryonun hiç te o kadar uzak olmadığı görülebilir.

Hatta buna gerçek hayattan bir örnek olarak 1980`de Iran başkenti Tehran konsolosluğunda rehine olarak tutulmuş CIA ajanlarını verebiliriz.(1)

Bu yüzden kullanıcının tüm bilgileri yok ettiğini kanıtlayabileceği ve bu şekilde kullanıcıyı da koruyabilen bir şifreleme sisteminin seçilmesi gerekmektedir.Buda tabiki gbde`nin içerdiği özelliklerden biridir.Bunun yanında böyle durumlarda gbde tarafından yok edilen anahtar şifrelerin en pahalı brute-force yöntemleri de dahil olmak üzere hiçbir şekilde geri kazanılabilme konusu söz konusu değildir.

GBDE kısıtlamaları

Tabi ki her şifreleme yönteminde olduğu gibi gbde`nin de kısıtlamaları bulunmakta.

1. Gbde sadece "Soğuk disk" olarak adlandırdığımız ve diske giriş yapılmasını sağlamaya yarayan tüm anahtarlardan arınmış diskler üzerindeki bilgileri korumak amacıyla dizayn edilmiştir.örnek olarak sistemden çıkarılmış diskler verilebilir.

2. Eğer gbde tarafından şifrelenmiş bir aygıt çalışan bir sisteme bağlanılırsa, kendi üzerinde şifreleme için kullanılmış olan anahtarları taşıyor olacaktır.Bu da tabiki diske ancak işletim sisteminin sunabildiği güvenliği sağlayacaktır.Buna da örnek olarak sisteme interet'te iken giren saldırganın kullanıcı sistemi aktif ram`inde tutulan bilgilere ulaşarak bu anahtarları elde etmeye çalışması verilebiliyor.

GBDE nasıl çalışıyor?

Gbde'nin çalışma mantığını kısaca kullanıcı tarafından girilen şifrenin kullanıcıya ana şifreye giriş izni vermesi ve ana şifreye ulaşma izni olan kullanıcının tüm şifrelenmiş sektör bilgilerine ulaşabilirliğin sağlanması olarak açıklanabilinir.

FreeBSD altında GBDE

Bu şifreleme yöntemini kullanabilmek için tek şart FreeBSD 5.0 veya daha yüksek bir versiyonunu kullanıyor olmanız.Daha fazla detaya girmeden FreeBSD üzerinde nasıl kullanabileceğimize bakalım:

1 Tehran'daki olayın detaylarına ulaşmak için:
http://news.bbc.co.uk/onthisday/hi/dates/stories/april/25/newsid_2503000/2503899.stm

Her zamanki gibi süper kullanıcı oluyoruz.Ardından kernel’a ekleme yapmak için `/usr/src/sys/i386/conf` dosyası içinde bulunan kernel dosyamıza girip:

```
options GEOM_BDE
```

Parametresini kernel'a ekliyoruz.

FreeBSD 5.0 versiyonu veya sonrası bir versiyon kullanıyor olacağımız için “Yeni” şekilde kernelimizi derleyebiliriz:

```
quasar$: su

passwd:

quasar$:make buıldkernel KERNCONF=quasar && make installkernel
KERNCONF=quasar && reboot
```

Kernel derlenmesi bittikten ve sistem tekrar başladıktan sonra artık yeni şifreleme yöntemimizi uygulamaya sokabiliriz.

Gbde’nin bütün şifrelenmiş disk sektörlerine girebilmesi için bir kilit (lock) dosyasına ihtiyacı vardır.Her şifrelenmiş bölüm (partition) için ayrı bir kilit dosyası hazırlanır.Bunu yaratmak için :

```
quasar$:mkdir /etc/gbde
```

Komutu ile bu klasörü oluşturuyoruz.

Kullanılmadan önce her bölüm gbde 'ye gösterilmek ve gbde`nin bu dosya bölümünü şifrelemesinin istendiğinin belirtilmesi gerekir.Sadece bir defaya mahsus yapılacak bu işlemi ise şu komutla verebiliriz:

```
quasar$:gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c
```

Burada dikkat edeceğiniz bölüm disk isimleridir.Ben kullandığım ikinci diskin şifrelenmesini istediğim için kendi diskime uygun gelen ismi seçtim.Sisteminizdeki disk isimlerini `mount` komutuyla görebilirsiniz.

Bu İşlem sonrasında gbde size üzerinde değişiklik yapabilmeniz için bir menu sunacaktır.burada yapılacak tek değişiklik sektör büyüklüğü (sector_size) parametresinde olabilir.512 olarak gelen sektör büyüklüğünü, çok ufak kalıp performansı etkilemesini engellemek için 2048 olarak değiştiriyorum.

```
sector_size     =       2048
```

Unutmayın bu dosya değerini çok düşük olarak ayarlamak performans ve boş alanınızın azalmasına yol acarken çok büyük kullanmak dosya sisteminizin çalışmamasına yol açabilir.

Daha sonra gbde size şifreleme için kullanılacak şifrenizi iki defa soracaktır.Tabi buradaki şifrenin 8 karakter veya ustu büyük,küçük,rakam ve işaretlerden oluşturulması her zaman diskinizin güvenliğini arttırıcı etkenlerin başında gelmektedir.

Burada da dikkat etmeniz gereken şey gbde kilit dosyanız hazırladıktan sonra hemen bir yedeğinin alınıp güvenli bir yerde saklanmasıdır.Unutmayın bu dosyanın silinmesi diski alan kısının sisteme girmesi yanında sizin de erişiminizin engellenmesi anlamına gelmektedir.

Şimdi de şifrelenmiş bölümü kernel'a bağlayalım :

```
quasar$:gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

Bölüm şifrelenmesi sırasında sizin belirlediğiniz şifre size tekrar sorulacak ve bunun doğrulanması sonunda şifrelenmiş bölümünüz `/dev` içinde .bde şeklinde bir dosya halinde gözükecektir.

```
quasar$: ls /dev/ad*

/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1

/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c

/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

Şifrelenmiş diski kernel'a bağladıktan sonra artık üzerinde yeni bir dosya sistemi oluşturabiliriz.Disk üzerinde yeni bir dosya sistemi oluşturmak için `newfs` komutunu kullanacağız:

```
quasar$: newfs -U /dev/ad4s1c.bde
```

Üzerine bilgi yazıp okuyabilmek için yeni diskimizi artık mount edebiliriz:

```
quasar$:mkdir /mnt/sifreli

quasar$:mount /dev/ad4s1c.bde /mnt/sifreli/
```

Her şeyi ayarladığımıza göre artık son bir test yapabiliriz.Bakalım `df` ile diskimiz görüntüleniyor mu:

```
quasar$: df -H

Filesystem        Size   Used  Avail Capacity  Mounted on

/dev/ad0s1a      1037M    72M   883M     8%    /

/devfs            1.0K   1.0K     0B   100%    /dev

/dev/ad0s1f       8.1G    55K   7.5G     0%    /home

/dev/ad0s1e      1037M   1.1M   953M     0%    /tmp

/dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr

/dev/ad4s1c.bde   150G   4.1K   138G     0%    /mnt/sifreli
```

Gbde'nin en kötü olarak adlandırılabilecek noktalarından biri ise her boot sonrasında şifrelenmiş bölümün tekrar kernel'a bağlanması ve mount edilmesi gerekmekte.Maalesef `/etc/fstab` dosyasına da eklenmesi mümkün olmadığından tek çözüm bir script hazırlayıp bunların otomatiğe bağlanmasının sağlanması olacaktır.

Bunu yazarken sizin belki de bana soracağınız soruyu kendi kendime sordum.Yazdığımız scriptte bu dosya şifresinin yazılması script üzerinde şifreyi okuyan saldırganın hiçbir zorluğa gerek kalmadan şifrelenmiş bölümüne ulaşmasına olanak vermeyecek midir?Evet bu doğru o yüzden script yazmamız da pek işe yaramayacak.

Benim kullanma şeklim sizin kullanım amaç ve alanlarınıza uygun olur mu bilmiyorum fakat sistem kapanışına yakın elimdeki onamlı bilgileri buraya aktaracağım veya bunları okuyacağım zaman bu diski kernel'a bağlayıp mount etmek açıkçası en mantıklı olanı gibi geliyor.

Uyumluluk problemi açısından gbde hiç sorunsuz çalışabilmektedir.Tek uyumsuzluk problemi sysinstall kullanımı sırasında ortaya çıkmaktadır. sysinstall kullanımı sırasında “Probing for devices” bölümünde eğer şifrelenmiş bölüm kernel'dan ayrılmamış durumda ise sistemin hata vermesine ve hatta yeniden başlamasına sebep olabiliyor..Bunu önlemek için kernel'a bağladığımız bölümünün `gbde detach /dev/ad4s1c` komutu ile kernel'dan ayrılması gerekmektedir.

Kolay gelsin.

Özgür Özdemircili
http://www.enderunix.org
http://news.enderunix.org
http://haber.enderunix.org

Sorularınız için : dionypheles@gmx.net

Kaynaklar

GBDE – GEOM based disk encryption http://phk.freebsd.dk/pubs/bsdcon-03.slides.geom-tutorial.pdf

FreeBSD handbook http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/disks-encrypting.html