Giriş

Gerekli tahmini süre: 20 dakika

Güvenli Geliştirme Ortamı uygulamalı laboratuvarına hoş geldiniz. Bu laboratuvar, kimlik bilgileri ve API anahtarları gibi sırları şifreleyerek geliştirme ortamınızı daha güvenli hale getirmenin yollarını öğreneceksiniz, böylece bu bilgiler açıkça saklanmaz.

Öğrenme Hedefleri

Bu laboratuvar sırasında şunları yapacaksınız:

- `pass` sır yöneticisi aracını kurun
- `pass`'i bir Gnu Privacy Guard (GPG) anahtarı ile başlatın
- `pass` CLI (komut satırı arayüzü) kullanarak sırları güvenli bir şekilde saklayın
- Saklanan sırları `pass` CLI (komut satırı arayüzü) kullanarak geri alın
- Bilgisayarınızı güvence altına almak için `pass`'i temizleyin

Güvenli Geliştirme Ortamı Nedir?

Bir geliştirici olarak, güvenliğin uygulamanızın geliştirme sürecinde bir düşünce sonrası olmamasını sağlamalısınız. Güvenlik, yazılım geliştirme yaşam döngüsü (SDLC) boyunca dahil edilmelidir, ancak bu yeterli değildir.

Eğer geliştirme ortamı güvenli değilse, orada geliştirilen kodun da güvenli olduğunu kabul etmek zordur. Geliştirme ortamınızı riske karşı güvence altına almak için kullanabileceğiniz birkaç basit adım vardır:

- Üretim uygulamanız için gereken gizli bilgileri güvenli bir şekilde saklayın.
- İnternet bağlantısını güvence altına alın. Gerekirse bir VPN kullanın.
- Güçlü giriş/çıkış politikalarına sahip bir güvenlik duvarı uygulayın.
- Açık portları düzenli olarak kontrol edin ve gerekli olmayan portları kapatın.
- Mümkünse geliştirme için Docker konteynerleri kullanın ve geliştirme görevleri ile iş görevleri için ayrı bilgisayarlar kullanın.
- Geliştirici ortamlarındaki davranışları kaydedin.
- Kimlik hırsızlığını önlemek için çok faktörlü kimlik doğrulama kullanın.
- Geliştirici makinelerinden üretim ortamına erişmesi gereken geliştiriciler için ek güvenlik önlemleri ekleyin.
- Geliştiriciler tarafından yapılan tüm taahhütleri ve değişiklikleri, gelecekte sorunlar ortaya çıkarsa referans almak için izleyin.

Ön Koşullar

Geliştiricilerin, günlük olarak kullandıkları bilgisayarlarla çalışmak için bulut API anahtarları ve diğer şifreler gibi gizli bilgilere ihtiyaçları vardır. Ancak, her geliştirici bu gizli bilgileri korumanın ve güvenli bir şekilde saklamanın yollarını anlamaz.

Diyelim ki, güvensiz kod uygulamalarını takip eden bir geliştiricisiniz. Önemli şifreleri düz metin olarak güvensiz bir şekilde saklayan `insecure.txt` adında bir dosyanız var.

Bu laboratuvar için gerekli olan güvensiz gizli bilgiler dosyasını indirelim.

Görev

Görev

1. Üst menü çubuğundan `Terminal > Yeni Terminal` seçeneği ile bir terminal açın ve `/home/project` klasöründe olduğunuzdan emin olun.

   ```
   cd /home/project
   ```

2. Laboratuvar için gerekli dosyayı edinmek üzere aşağıdaki `wget` komutunu çalıştırın:

   ```
   wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-CD0267EN-SkillsNetwork/labs/module4/data/insecure.txt -O ~/insecure.txt
   ```

3. Dosya içeriğini görüntülemek için aşağıdaki `cat` komutunu çalıştırın:

   ```
   cat ~/insecure.txt
   ```

Sonuçlar

```
$ cat ~/insecure.txt
IBM_CLOUD_API_KEY="OebUkIcSk9KbpeXnZ0z5bKiAj2G5uHeEgq49xd9vEXM"
PROD_ADMIN_PASS="UbJCN5dL46eNE6ecULp9DtNiQLWSxKxpZ6u3BzRsBKI"
```

Gördüğünüz gibi, dosya önemli sırları düz metin olarak saklıyor. Bu kesinlikle istenmeyen bir durum. Bir sonraki bölümde, bu sırları güvenli bir şekilde saklamayı `pass` kullanarak öğreneceksiniz.

Adım 1: Şifreleri pass ile güvence altına alma

Bu adımda, `pass` adlı Linux tabanlı bir şifre yöneticisini indirecek ve kuracağız. `pass` kullanarak gizli bir Gnu Privacy Guard (GPG) anahtarı oluşturacağız ve bunu bilgisayarınızda yerel bir kimlik bilgisi deposu oluşturmak için kullanacağız. `pass`, gizli bilgilerinizi güvenli bir şekilde saklamanıza olanak tanıyacaktır.

Göreviniz

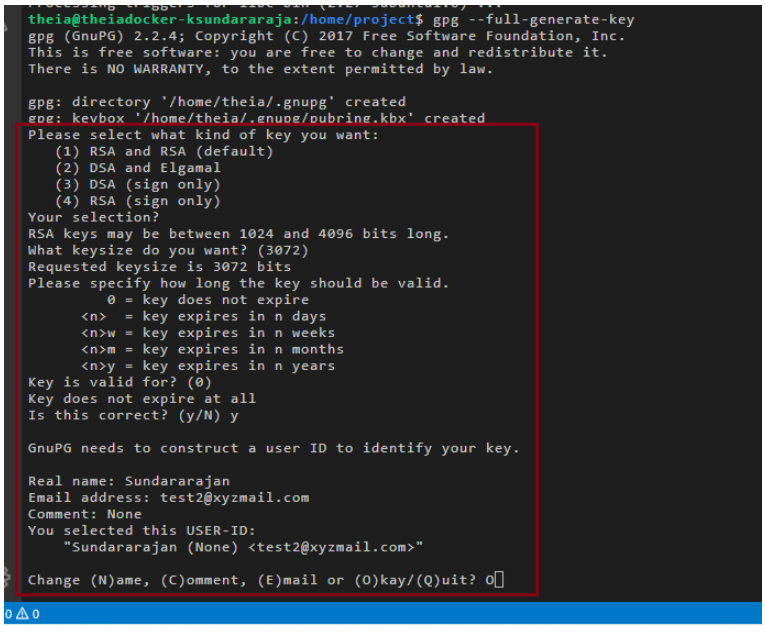
1. Öncelikle, aşağıdaki Linux komutlarını komut istemcisinde çalıştırarak ortamınıza `pass` yükleyin:

   ```
   sudo apt update
   sudo apt install -y pass
   ```

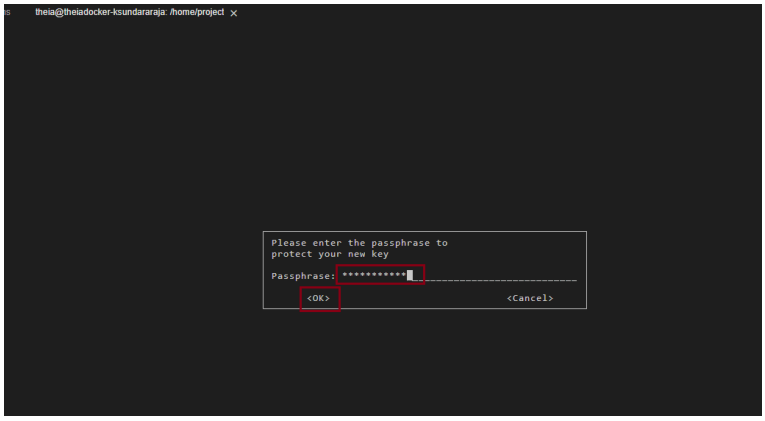
2. Ardından, yerel makinenizde gizli bir GPG anahtarı oluşturmak için `gpg` komutunu kullanın.

   ```
   gpg --full-generate-key
   ```

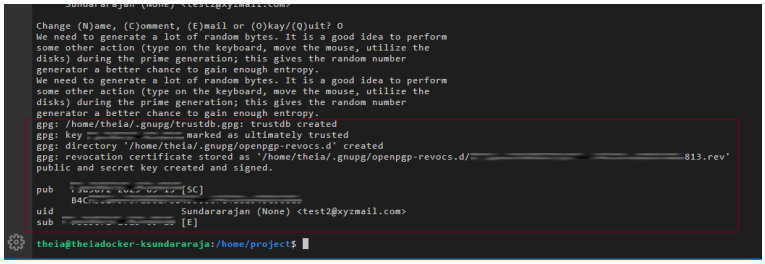
- Kurulum ve yapılandırma işlemi tamamlanana kadar tüm kurulum istemlerini takip edin. Anahtar türü ve diğer kalan ayarlar için varsayılan seçenekleri kabul etmek için `[enter]` tuşuna basın. Sonra onaylamak için `y` tuşuna basın.
- Daha sonra Gerçek adınızı, E-posta adresinizi ve herhangi bir Yorum girin. Ardından `O` (Tamam) tuşuna basın.



- Ardından, en az 8 karakterden oluşması ve en az 1 rakam veya özel karakter içermesi önerilen bir şifre girin.



3. Bu işlemle birlikte GPG anahtarınız oluşturulmuş olacak.



Not: Verdiğiniz şifreyi unutmayın, çünkü bu sizin ana şifreniz olacak.

Adım 2: Pass'i Başlatma

Artık bir GPG anahtarı oluşturduğumuza göre, `pass`'i GPG ID'si ile başlatabiliriz.

Bu adım yalnızca bir kez yapılması gereken bir işlemdir. Daha sonra ihtiyaç duyduğunuz kadar gizli bilgi kaydedebilirsiniz.

Göreviniz

1. Adım 1'de oluşturduğunuz GPG anahtarının ID'sini kullanın.

```
gpg --list-secret-keys --keyid-format LONG | grep sec
```

Elde edilen çıktıda, aşağıdaki gibi bir satır göreceksiniz:

```
sec   rsa2048/ABCDEFGH01234567 2019-07-31 [SC]
```

Bu, GPG anahtar ID'nizi içeren satırdır. Bu örnekte, GPG anahtar ID'si `ABCDEFGH01234567`'dir. Anahtarınızın ID'sini panoya kopyalayın.

2. `pass init` komutunu kullanarak `pass`'i GPG anahtar ID'niz ile başlatın.

```
pass init {gpg anahtarınızı buraya yapıştırın}
```

Not: GPG ID'nizi `{gpg anahtarınızı buraya yapıştırın}` ifadesinin olduğu yere yapıştırmalısınız, ancak GPG ID'nizin etrafındaki süslü parantezleri dahil etmenize gerek yoktur.

Sonuçlar

Sonuçlar

Aşağıda gösterilen çıktıya benzer bir çıktı görmelisiniz (ama kendi GPG anahtarınızla):

```
$ pass init ABCDEFGH01234567
Password store initialized for ABCDEFGH01234567
```

Hepsi bu kadar! `pass`'i başarıyla başlattınız.

Sonraki adımda, sırları nasıl saklayacağınızı ve alacağınızı öğreneceksiniz.

Adım 3: Gizli Anahtarlar Oluşturma

Artık gizli anahtarlar oluşturup bunları güvenli bir şekilde saklamaya hazırsınız. Bu adımda, gizli anahtarları gizli anahtar yöneticisi anahtar deposuna eklemek için `pass insert` komutunu kullanacaksınız.

Göreviniz

1. `~/insecure.txt` dosyasından `IBM_CLOUD_API_KEY` anahtarının değerini almak için aşağıdaki komutu çalıştırın:

   ```
   cat ~/insecure.txt| grep IBM_CLOUD_API_KEY | grep -o "\".*" | grep -o "[a-z,A-Z,0-9]*"
   ```

   Bu komutu çalıştırmak, Adım 2'de kullanacağınız değeri döndürecektir.

2. `pass` içinde `IBM_CLOUD_API_KEY` adında yeni bir gizli anahtar oluşturmak için aşağıdaki komutu çalıştırın:

   ```
   pass insert IBM_CLOUD_API_KEY
   ```

3. Adım 1'de elde ettiğiniz `IBM_CLOUD_API_KEY` değerini kopyalayıp yapıştırın.

   Not: Bu adımda, `pass` gizli anahtarı komut istemine yapıştırdığınızda veya yazdığınızda göstermeyecektir. Her iki değerin de aynı olduğunu onaylamak için değeri komut istemine iki kez yapıştırmanız istenecektir.

Adım 3: Gizli Anahtarlar Oluşturma (devam)

Göreviniz

Şifre deposu için başka bir gizli anahtar oluşturalım.

1. Anahtarın değeri için komutu çalıştırın. (Bu, önceki sayfadaki Adım 1'de izlediğiniz aynı prosedürdür). Bu sefer yeni gizli anahtar olarak `PROD_ADMIN_PASS` kullanacaksınız. Aşağıdaki komutu komut istemcisinde çalıştırın ve çıktıyı panoya kopyalayın:

   ```
   cat ~/insecure.txt| grep PROD_ADMIN_PASS | grep -o "\".*" | grep -o "[a-z,A-Z,0-9]*"
   ```

2. Yeni bir gizli anahtar `PROD_ADMIN_PASS` ekleyin:

   ```
   pass insert PROD_ADMIN_PASS
   ```

3. `PROD_ADMIN_PASS` için değeri komut istemcisine yapıştırın.

   Not: Bu adımda, `pass` gizli anahtarı yapıştırdığınızda veya komut istemcisine yazdığınızda görüntülemeyecektir. Her iki değerin de aynı olduğunu onaylamak için değeri komut istemcisine iki kez yapıştırmanız istenecektir.

Artık şifreleri ihtiyaç duyduğunuzda kullanabilirsiniz!

Laboratuvarın bir sonraki bölümünde, saklanan şifreleri nasıl alacağınızı öğreneceksiniz.

Adım 4: Gizli Anahtarları Alma

Artık birkaç gizli anahtar oluşturduğunuza göre, bunları gerektiğinde nasıl alacağınızı öğrenme zamanı.

Göreviniz

Göreviniz, gizli anahtarların `pass` içinde saklandığını doğrulamaktır.

1. Gizli anahtarın düzgün bir şekilde eklenip eklenmediğini doğrulamak için `pass` ile `IBM_CLOUD_API_KEY`'yi görüntülemek üzere `show` komutunu kullanalım:

   ```
   pass show IBM_CLOUD_API_KEY
   ```

   Büyük ihtimalle ana şifrenizi (GPG anahtarı) girmeniz istenecektir. Eğer öyleyse, girin ve devam edin. Elbette, `pass` ile kaydettiğiniz herhangi bir şifre için aynı işlemi gerçekleştirebilirsiniz.

2. Gizli anahtarın düzgün bir şekilde eklenip eklenmediğini doğrulamak için `pass` ile `IBM_CLOUD_API_KEY`'yi gösterelim:

   ```
   pass show PROD_ADMIN_PASS
   ```

`pass`, saklanan gizli anahtarları almayı kolaylaştırır. İstendiğinde GPG şifrenizi girmeniz, `pass`'in bunları sizin için görüntülemesine olanak tanır. Bir saldırgan veya başkaları bilgisayarınıza erişim sağlarsa, GPG şifrenizi bilmedikleri için şifrelerinize veya saklanan gizli anahtarlarınıza erişemezler.

Adım 5: Temizlik

Temizlik

Geliştirme ortamınızı güvence altına almak için bu laboratuvarın son bir adımı var.

Artık şifreleriniz `pass` ile güvenli bir şekilde saklandığına göre, son adım `~/insecure.txt` dosyasını silmektir.

1. Düz metin olarak kaydedilen şifreleri içeren dosyayı kaldırmak için aşağıdaki komutu çalıştırın:

   ```
   rm ~/insecure.txt
   ```

Artık şifreleriniz `pass` ile güvenli bir şekilde saklanıyor!

Sırlarınıza Erişim

Geliştirme için önemli bir sırınıza erişmeniz gerektiğinde, sadece aşağıdaki komutu çalıştırın:

```
pass show [SECRET_KEY_NAME]
```

Sonuç

Tebrikler! `pass`'in, gizli bilgileri güvenli bir şekilde saklayarak yalnızca sizin erişiminize açarak geliştirme ortamınızı nasıl güçlendirebileceğini öğrendiniz. `pass` uygulaması, şifre bilgilerinizi deneyimli bir kriptografi yazılımı olan GPG ile şifreler. Gizli bilgiler güvenlidir çünkü her şifre, yalnızca sizin bilmeniz gereken ana şifrenizin arkasında şifrelenmiştir.

Bu uygulamalı laboratuvar çalışmasında, `pass`'i nasıl kuracağınızı, yapılandıracağınızı ve başlatacağınızı öğrendiniz. Ayrıca, `pass` komut satırı arayüzünü (CLI) kullanarak gizli bilgileri güvenli bir şekilde nasıl saklayacağınızı öğrendiniz. `pass`'ten gizli bilgilerinizi nasıl alacağınızı ve bilgisayarınızı güvence altına almak için `pass`'i nasıl temizleyeceğinizi öğrendiniz.

Sonraki adımlar

Önerilen bazı sonraki adımlar, bilgisayarınıza `pass`'i kurarak bazı gizli bilgiler oluşturup almayı denemektir. Ayrıca, geliştirme ve dağıtım sırasında riskleri azaltmak için kullanabileceğiniz diğer yöntemler hakkında daha fazla bilgi edinmek için şu kaynağı ziyaret edebilirsiniz: Geliştirme Ortamınızı Güvence Altına Alın.

Yazar(lar)

Sam Prokopchuk

Diğer Katkıda Bulunan(lar)

Samaah Sarang
John J. Rofrano
Michelle R. Sanchez, Skill-up Technologies'de Eğitim Tasarımcısı, 25 yılı aşkın kurumsal düzeyde teknik destek ve teknik eğitim deneyimi ile.