linux kernel hooking , veri yaniltmalari root exploitlerin mantigi

linux kernel hooking , data manipulations and making root exploits

ONUR TURKESHAN—CAGRI TEPEBASILI

dusunenturk secure company

bu pdf 2012 yılının 2. günü ONUR TÜRKEŞHAN ve Çağrı TEPEBAŞILI tarafından linux2011 işletim sistemlerindeki veri döngülerinden meydana gelenveri arttirma , veri yanıltmalari ve root exploit kodlamanın mantığıyla beraber 2011 root exploitlerin yaygın olmadığı içinyararlı olabilmek adına yazılmıştır.

Katkısı Geçenler : Cyber-Warrior TiM , DusunenTURK Secure Company, ZEMANA , Unique-Sec , MythSec , Bildirgec , kodaman

Arch Linux Nedir ?

Arch Linux, i686 ve x86-64 mimarileri üzerinde çalışmak üzere uyarlanmış hafif bir Linux dağıtımıdır. Arch BSD tarzı girdi sistemi kullanır ve kendine özgü "pacman" paket yöneticisine sahiptir, güncellemelerinde düzeltilmiş versiyon yayınlamaktan ziyade rolling release (her zaman güncel) gelişim sistemini kullanır.[Rolling release güncelleme sistemi, bir kere kurulum yaptıktan sonra, sistem güncelleme komutu (pacman -Syu) ile sürekli güncel kalma mantığına dayanmaktadır. Dağıtımın yeni bir sürümü çıktığında onun ISO'sunu indirip kurmak yerine, güncelleme yolu ile üst sürüme terfi edilmektedir.Arch ne kadar kendini "sade" olarak tanımlasa da, Arch Linux; Linux dağıtımları ile ilk kez tanışacaklar ya da deneyimsiz kullanıcılar için uygun bir sistem olmadı.Arch Linux'un sadelik tanımı; zerafeti ve sistem konfigürasyonunda "olabildiğince az" teknik karmaşayı bir arada sunmaktır. "Olabildiğince az" teknik karmaşa ile kastedilen, kolay kurulum ya da kolay kullanım değildir, gerçi kolay kullanım Arch Linux kurulumunda yardımcı olabilirdi.

Ne Değildir ?

Arch'ın babası ve baş geliştiricisi Aaron Griffin'in Arch için söylediği birkaç cümle şöyle; "Arch, "kullanıcı dostu" olmak için üretilmedi. Arch, bir platform olması için üretildi - kullanıcının dilediğini gerçekleştirebileceği bir platform. Bu şu demek, **biz kullanıcıyı bizim yolumuzdan gitmesi, bizim konfigürasyon araçlarımızı kullanması, bizim düşüncelerimizi taşıması için zorlamıyoruz. Bu (Arch), onların düşüncesi (rüyası) olmalı.**"Kullanıcıların ne istiyorsa onu yapmasın fırsat tanıdığından olsa gerek, zaman içinde Arch tabanlı birçok varyasyon oluştu. En bilineni ArchBang, Arch temelinin üstünde Openbox pencere yöneticisi kullanarak fark yarattı. Yine Arch'ın, Hurd'a uyarlanması sonucu ArchHurd oluştu.Bundan başka denemeye değer Arch varyasyonları arasında Chakra (**Arch Linux + KDE4 + Shaman** ve birkaç artistik dokunuş) ve Parabola (tamamen özgür yazılım) var.

Linuxun geliştiricisi acaba bu lafları söylerken ,hiç bir hackerin arc linux ile uğraşmayacağını düşünmedi? Şimdi burdaya biz ve süperzeki beyinlerimiz devreye giriyor.

ROOT EXPLOİT NEDİR?

Root ismi , linux sistemlerde en üst düzey yetkiye sahip kullanıcılara verilen genel addır.Root exploitlerde ki genel mantık : yapmaya hakkın olmayan bir işlemi exploit ederek en üst düzey kullanıcı OLMAKTIR.

Yukarıdaki kelimelere bakarsak Chakra ve KDE4 ü görüyoruz. 2011 Exploit Kodlamamaızdaki Anahtarlar bunlar olacak.

KDE4 Grafik arayüzü kullanıcılara hata mesajlarını düzenleme imkanı sunmaktadır.Aslında bu hatalar kernelde çakılı olarak DEĞİŞTİRİLEMEYECEK şekilde kodlanmıştır.KDE4 görüntü arayüzünde KERNELDE bulunmayan hata mesajını kullanıcının belirttiği şekilde düzenler . bu işlemin yapıldığı döngü pam_setcred() üzerinden onaylama yapar.Linuxun güvenlik yada onaylama sistemleri bu işlemde herhangi bir yanlışlık görmez hata vermez yada kullanıcı kimliği gerektirmez.Zaten PAM kullanmayan kullanıcılar etkilenmez.

/proc/kdu/envsys 'da tanımlanmış PROT_READ PROT_WRITE, MAP_SHARED değişkenlerine kendi verileri
mizi gönderelim ve ve onaydan gecirmeye çalışalım.

```
chmod ("/proc/kdu/envsys", 04755);
c = mmap (0, 4096, PROT_READ , PROT_WRITE, MAP_SHARED , 0, 0);
memset ((void *) c, 0, 4099);
```

Verilerimizin aktarım aralığı 4096 ve 4099 olsun.biz zaten bu artan 3 alanı dolduracağız.

```
char thoughts[] = { onur turkeshan root exploitl ,onur turkeshan root exploitl ,onur turkeshan root
onur turkeshan root exploitl ,onur turkeshan root exploitl ,onur turkeshan root exploitl ,
onur turkeshan root exploitl ,onur turkeshan root exploitl ,onur turkeshan root exploitl ,
onur turkeshan root exploitl ,onur turkeshan root exploitl ,
};
void RANDOM_THOUGHT(void){  int i;
  char thought;
char p, p2;
 char c;
 int size_of_thought;
 srand(time(NULL));
 thought = strdup(thoughts[rand() % (sizeof(thoughts)sizeof(thoughts[0]))]);
```

Onur turkeshan root exploit yazisini 4096 ve 4099 arasındaki 3 kısımda hata aldırmadan yada linuxun onayından geçip geçmeyeceğini anlamak için veriyi gönderdik ve çektik. Onaylamadan bir sorun çıkmadı ; şimdi ne yapacağız ?

PROT_READ , PROT_WRITE VE MAP_SHARED 'a rastgele veri gönderip ondan her cevap geldiğinde aynı veriyi tekrar göndereceğiz. onaylamayı manipükle etmesi için. Manipulasyon bittiğinde işlemler KDE4'üzerinden değil KERNEL üzerinden vurgulanır ve başlatılır !

```
if ((unsigned long)current =
(orig_current + 0x1000 - 17 )) {      onurturkeshanrootexploit_onur turkeshan__club(orig_current);
      cred_support = 1;       return;     }
```

Üstteki Betiktende anlaşıldığı üzere verimiz aslında kernelin kabul etmeyeceği bir haldeyken KDE4 exploitimiz sayesinde işaretli yani güvenilir veri olarak aktarıma devam ediliyor.
Bundan sonra yapabilecekleriniz tamamen sizin üretgenliğinize bağlı. 2011 kernelin direkt kendisinde açık aramak yerine içindeki updatelerdeki modüller bakmayı deneyin , en fazla rağbet görenler mesela:)

bir sonraki yazımız :
2.6.32 2011 EXPLOİTLERE BAKIŞ

Onur TÜRKEŞHAN
DÜŞÜNENTÜRK SECURE COMPANY

KERNEL HOOKİNG NEDİR?

Hooking tamamen bir karmaşa üstüne kurulmuştur . Bu yazımızda "Hooking" terimini inceleyeceğiz ve Linux Sistemlerde hooking nasıl yapılır ona göz atacağız.Hooking nedir ? Hooking , Sistemde bulunan modülleri isteklerimiz doğrultusunda değiştirmektir f
Genellikle kernel ve ve uygulama tabanlı olarak uygulanır. Nasıl Yani ? Yani , Örneğin Kernelde Bulunan bir uygulama var , biz bu uygulamanın içine kendi çıkarlarımız için dinamik kodlar ekleyeceğiz.
Windows Sistemlerini çekirdeği kapalı oldugu için Windows Sistemlerde Hooking yapmak gerçekten yetenek ister :) .Fakat Linux Sistemlerini Kerneli açık oldugu için elleşmek serbesttir.Örneğin "www.kernel.org" adresinden istediğini sürüm kerneli indirerek incelemeler yapabilirsiniz.Genel olarak Hooking terimi kafanızda oturmuştur sanırım. Şimdi biz Linux Kernellerde Hooking işlemini nasıl yaparız ne yapabiliriz o konulara biraz bakacağız. Hooking terimini anladınız fakat şu noktaya değinelim . Hooklama işlemini kafanıza göre yapamassınız çunkü , Kernellerinde bir güvenlik modülü bulunmaktadır . Linux kernellerin neredeyse hepsinde Security adlı klasör içinde bulunan yazılımlar güvenliği sağlamaktadır, fakat biraz beyin fırtınası yaparak bunuda aşabileceğimizi göreceksiniz.
Sizlere direk hooklama işlevini göstermeyecem fazla kolaya alışmak iyi değildir fakat bu makaleyi okuduktan sonra ne yapacağınızı biliyor olacaksınız.
Aklınızda birşeyler oluştu fakat halen tam olarak ne yapacağınızı bilmediğinizden eminim . O yüzden kernele giriş yapmadan normal bir uygulamada hook işlevselliği nasıldır ona bakacaz oradan kernele bir bağlantı yapacağız.

Kısa bir C uygulaması ,

```
#include<stdio.h> // stdio.h kütüphanesini çektik
#include<string.h> // string.h kütüphanesini çektik
#include<stdlib.h> // stdlib.h kütüphanesini çektik
security(int,int); // fonksiyonumuzu belirttik iki adet integer değeri oldugunu tanımladık
main(){ // Gövde kısmı
int a,b; // a ve b adında iki adet integer değişken tanımladık.
     printf("Numaranızı Giriniz : \n"); // Numara girmesini istedik
     scanf("%d",&a); // Onayladık
     printf("Sifrenizi giriniz : \n"); // Sifre girmesini istedik
     scanf("%d",&b); // Onayladık
return 0; // geri-dönüş olmassa olmaz
}security(int x,int y){ // belirlediğimiz fonksiyonun içeriğini belirtmek için açıyoruz
if(x == 1 && y == 123456){ // x >> numarayı y >> sifreyi belirtiyor
          //mesela giris sağlansın DİKKAT}
     else{exit(1); // Şart sağlanmassa programı sonlandır.
}}
```

Şimdi bu kod bir yerde dursun. "www.kernel.org" girin ve istediğiniz bir kernel indirin ve inceleyin.Göreceksiniz herşey açık mesela.c gibi yazılımlarla dolu bu ne demek ;) Şimdi tekrar koda dönüp böyle bir uygulama oldugunu varsayalım . "Hooking" işlevini devreye sokalım ve programı bozmadan istediğimiz gibi değiştirelim. Ne yapabiliriz mesela ? >>> "//mesela giris sağlansın DİKKAT" bu kısmı istediğiniz bir yere yollamasını sağlayabilirsiniz mesela = ;)
Genelde bu kadar kolay değildir hooking işlemi nedeni uygulama içeriği değil güvenllik modülüdür . Bu durumlarda Güvenlik Modülü'de açıkta oldugu için iyi bir "C" bilgisi ile isteğinize göre donatabilirsiniz.FAKAT bu bize bir getiri sağlamayacaktır . Çünkü kullanılan bir Kernele hooking işlemi yapacağız Açıktada olsa kalsör içinden yazılımı seçim editlemeyeceğiz ;)Ne Yapacağız ? Bu sorunun Cevabı çok fazla metodda geçerlidir . Buffer Overflow(Stack Based), Binary Planting gibi . Anlatmadı demeyin bu metodlardan nasıl yapacaz demeyin bunuda anlatalım :)
Kısaca Buffer Overflow dan bahsedip uygulamayı anlatacagım.
Buffer Overflow zaafiyetini bildiğinizi varsayarak bir beyin fırtınası yapacağız.
Buffer Overflow genellikle 2 alanda incelenir .

1.Stack Based
2.Heap Based
Bizim Burda Kullanacağımız İnceleyeceğimiz , Stack Based.

>>> Şimdi char gir[4]; tanımlanmış biz buraya AAAAAAAAAA girersek boflamış olacağız . Sonra Metasploit ile reverse tcp bind vs ile uzaktan bağlantı isteriz bu devran böyle döner fakat biz AAAAAAAAAAcat /etc/passwd dersek etc passwd okuruz !?

>>> Dank etmiş olması gerekiyor durumun :)

>>> Girdi-Çıktı sorunları dolayısıyla kod çalıştırabiliyoruz .

>>> Bu Olayı Programa dönük yaparsak program kontrolümüz altına girmiş olacaktır.

>>> Yok bu yöntem uygun gelmedi çakamadım durumu diyorsanız .

>>> Metasploiti biraz karıştırmanız gerecektir.Payloadlarda "Hook" ile Anti-Virüs Bypasslamaya yarayan exploitler bulunmaktadır.(bkz:rootkit)

Muhammet Çağrı TEPEBAŞILI | ÇağrıTEPEBAŞILI
ÇağrıTEPEBAŞILI | Vulnerability Researcher | LOJISTIK