27 Aralık 2024

Take-Home

Teslim tarihi: 30 Aralık 2024, 23:59 (geç teslimler kabul edilmeyecektir).

Problem 1: Jump Oriented Programming (JOP) (20p)

İçinde hataların (bug) olduğu bilinen bir programda karşılaşılan zafiyete bağlı olarak `ecx`, `edx` ve `eip` registerlerinin değerleri ile `0x9000` ve `0x9014` adreslerindeki bellek içeriğinin kontrol edilebildiği ortaya çıkıyor. Bu zafiyetten return-oriented-programming (ROP) ile yararlanılmak isteniyor ancak programın ret komutu kullanılmadan derlendiği anlaşılıyor. Buna rağmen, program içerisinde aşağıda verilmiş kod parçalarının yer aldığı da görülüyor.

```
0x3000: add edx, 4       ; edx = edx + 4
        jmp [edx]        ; jump to *edx

0x4000: add edx, 4       ; edx = edx + 4
        mov eax, [edx]   ; eax = *edx
        jmp ecx          ; jump to ecx

0x5000: mov ebx, eax     ; ebx = eax
        jmp ecx          ; jump to ecx

0x6000: mov [eax], ebx   ; *eax = ebx
        ...              ; don't worry about what happens
after this
```

Bu şartlar altında yukarıda belirtilen registerlerin ve bellek adreslerinin değerlerinin uygun şekilde değiştirilmesi ile `0x8888` adresine `0x2222` değerinin nasıl yazılabileceğini gösteriniz.

ecx	
edx	
eip	0x4000

0x9000	
0x9004	
0x9008	
0x900c	
0x9010	
0x9014	

`eip` instruction pointer, yani bir sonra çalıştırılacak komutun adresini gösteriyor. `ecx` ve `edx` de genel amaçlı registerlar.

Problem 2: Stack canaries (20p)

a. Derste konuştuğumuz gibi GCC ile bir C programı derlendiğinde eğer `-fstack-protector` bayrağı kullanılırsa her stack frame içerisine koruma amaçlı bir kanarya (canary) değeri yerleştirilmektedir. Stack smashing (stackde yer alan geri dönüş adresinin ezilmesi) saldırısına açık olan ve bunu göstermek için komut satırından giriş kabul eden kısa bir C programı yazın ve `-fstack-protector` bayrağı ile derlenmiş olsa bile ilgili saldırının yapılabildiğini gösterin. (İpucu olarak yazacağınız programda stack bölgesinde içinde bir pointer ve bir de string olan bir struct yapısının olduğundan yola çıkabilirsiniz. Bu yapı içerisindeki alanların birbirlerinin ardı sıra geldiğini ve ilk alanın bellekte ikinci alandan daha düşük adreste saklandığını kabul edebilirsiniz. Bu durumda string değişkeninin taşırılmasını sağlayabilirseniz pointer alanını da ezebilirsiniz ve kanarya ile korumaya rağmen saldırının gerçekleşebilmesi saldırganın stackteki bu alanları ezebilmesi üzerine kurulu olmalı.)

b. İşletim sisteminin bütün stack alanlarını non-executable olarak işaretlediği bir durumda stack smashing saldırısı yapılabilir mi? Cevabınızı kısaca açıklayınız.

Problem 3: Integer underflow vulnerability (15p)

Aşağıda basitleştirilmiş olarak verilen kod parçası oldukça yaygın olarak kullanılan bir routera aittir.

```
uint32_t nlen, vlen;    /*  values in 0 to 2^32-1  */
char buf[8264];

nlen = 8192;
if ( hdr->nlen <= 8192 )
    nlen = hdr->nlen;

memcpy(buf, hdr->ndata, nlen);
buf[nlen] = ':';

vlen = hdr->vlen;
if (8192 - (nlen+1) <= vlen )    /*  DANGER  */
    vlen = 8192 - (nlen+1);

memcpy(&buf[nlen+1], hdr->vdata, vlen);
buf[nlen + vlen + 1] = 0;
```

Eğer `hdr->ndata = "ab"` ve `hdr->vdata = "cd"` ise yukarıdaki kod parçası buf değişkenine `"ab:cd"` değerini yazacaktır. Saldırganın `hdr` değişkenini istediği gibi kontrol edebildiğini varsayarsak, yukarıdaki kod ile `buf` değişkeni üzerinden bir taşmanın (overflow) nasıl gerçekleştirilebileceğini gösteriniz.

Problem 4: Privilege Escalation (10p)

Unix tabanlı bir sistemde root yetkileri olmayan kendinize ait kullanıcı ile file sistemi incelerken `/sbin` içerisinde aşağıdaki içerik ile karşılaştığınızı varsayın.

```
-rwsrwxr-x   1 root laura   234K Apr 01 21:32 ping
```

Buradaki potansiyel güvenlik zafiyeti nedir? Bu zafiyetten yararlanılarak root yetkilerine nasıl sahip olunabilir. (ping komutunda herhangi bir zafiyet veya güvenlik açığı olmadığını varsayabilirsiniz.)

Problem 5: Android Isolation (10p)

Android'de her uygulama kendi prosesi içerisinde farklı bir UID ile çalışmaktadır. Güvenlik açısından her uygulamaya farklı bir UID atamanın faydası nedir? Kısaca açıklayınız.

Problem 6: Race conditions (15p)

Aşağıdaki kod parçası verilmiştir:

```
if (!stat("file.dat", buf)) return;   // abort if file exists
sleep(10);                            // sleep for 10 seconds
fp = fopen("file.dat", "w" );         // open file for write
fprintf(fp, "Hello world" );
close(fp);
```

a. Bu kod parçasının setuid root olarak çalıştığını varsayarsak güvenlik açığı oluşturabilecek beklenmedik bir çalışma nasıl oluşabilir ? Bir örnek ile açıklayınız. (İpucu olarak derste benzer bir kod parçasını incelememizden yola çıkabilirsiniz.)

b. sleep(10) kısmı uzaklaştırıldığında da (a) şıkkında tartışılan güvenlik açığı ortaya çıkar mı? Kısaca açıklayınız.

c. (a) şıkkındaki olası güvenlik probleminin oluşmaması nasıl sağlanabilir? (İpucu olarak open Unix sistem çağırısına aktarılan *O_CREAT* and *O_EXCL* bayraklarının görevlerinden yola çıkabilirsiniz.)

Problem 7: Setuid (10p)

Devreye alınacak yeni bir web sunucusunda aşağıdaki kod parçası bulunmaktadır. `serve` fonksiyonundaki bir hata saldırganlar tarafından nasıl kullanılabilir?

Sistemde `www-data` isimli UID değeri 100 olan bir servis kullanıcısı mevcuttur ve web sunucusu çalışmaya root kullanıcısı olarak başlamaktadır.

```
if (fork() == 0) {
    int socket = socket(":80");
    if (socket == -1) {
       perror("unable to open socket: ");
       exit(-1);
    }
    seteuid(100);
    serve(socket);
}
```

Dikkat edilmesi gerekenler:

1. Teslim PDF formatında yapılacaktır.
2. Tüm soruların cevapları tek bir PDF dosyasında verilecektir. Her sorunun cevabının yeni bir sayfada başlamasına dikkat ediniz.
3. Göndereceğiniz dosyanın isminin <öğrenci numaranız>-take-home şeklinde olması gerekmektedir. Öğrenci numarası etrafındaki <> işaretleri aradaki alana öğrenci numaranızın geleceğini belirtmektedir. <> işaretleri kullanılmayacaktır.
4. Bu take-home kişisel olarak cevaplanacaktır, grup çalışması yapılmayacaktır.
5. Kullanılan tüm kaynakların referans ile belirtilmesi gerekmektedir. Referansı belirtilmeyen ancak doğrudan Internet'ten alıntı olarak verilen cevaplar geçersiz sayılacaktır.
6. Internet'ten copy-paste şeklinde yapılan alıntılar, referans verilmiş olsa bile kabul edilmeyecektir. Copy-paste işlemi sırasından değişiklik olsun diye değişken adlarının değiştirilmesi veya satırların yerlerinin değiştirilmesi de kabul edilmeyecektir.
7. Yapay zeka destekli (ChatGPT vb.) olarak üretilen cevaplar kabul edilmeyecektir.