

## Nmap İleri Seviye Part 3

Nmap, ağ tarama ve zafiyet tespiti için kullanılan açık kaynaklı bir araçtır. Bu araç birçok sisteme yönelik taramaları gerçekleştirerek esnek, hızlı ve anlamlı bir şekilde sonuç üretmektedir. Sistemlerin açık olup olmadığını, açık olan sistemlerin portlarını durumları, hangi servislerin çalıştığı ve kullanılan işletim sistemi gibi birçok bilgiyi verebilmektedir.

Nmap ile tespit edilen servislerin güvenlik açığı barındırıp barındırmadığı ve kullanılan servisler hakkında bilgi elde edilebilir. Ayrıca içerisinde barındırmış olduğu scriptler ile hedef sisteme yönelik tarama gerçekleştirildiğinde hedef sistem hakkında detaylı bilgi ve güvenlik açığı olup olmamasına yönelik sonuç üretmektedir. Nmap aracı, alanının en iyi araçları araçları arasında yer almaktadır.

### NMAP SCRIPTING ENGINE

Nmap Scripting Engine (NSE), Nmap'in en güçlü ve esnek özelliklerinden biridir. Kullanıcıların ağ üzerinde yapmak istediği işlemleri otomatize eder. İçerisindeki komut dosyaları, Nmap ile paralel çalışmakta olup taramaya hız ve verimlilik katar. Nmap Scripting Engine, ağ keşiflerinde hedef hakkında bilgiler toplamak, açık olan portlara yönelik gelişmiş sürüm tespitini yapmak, hedef sistemde bulunan güvenlik açıklıklarının tespitini yapmak, sistem üzerinde çalışan backdoorların tespitini yapmak ve tespit edilen güvenlik açıklıklarını exploit etmek için Lua dilinde yazılmış olan bazı scriptleri kullanmak gibi işlemleri gerçekleştiren modülleri içerir.

NSE, -sC parametresi ile kullanılır. Ayrıca özel script belirtilmek istendiğinde “-script” veya “-sC” parametresinden sonra kullanılacak script adının yazılması gereklidir. Elde edilen sonuçları Nmap normal ve XML çıktısına eklenir.

Şekil 7.a – NSE ( -sC ) parametresinin Kullanılması

```

root@priviasecurity:~# nmap -sC -p22,111,139 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-05 09:05 EST
Nmap scan report for 192.168.16.128
Host is up (0.00033s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
111/tcp   open  rpcbind
| rpcinfo:
|   program  version  port/proto  service
|   100000    2          111/tcp    rpcbind
|   100000    2          111/udp    rpcbind
|   100003    2,3,4      2049/tcp   nfs
|   100003    2,3,4      2049/udp   nfs
|   100005    1,2,3      38733/udp mountd
|   100005    1,2,3      57134/tcp  mountd
|   100021    1,3,4      46047/udp nlockmgr
|   100021    1,3,4      57723/tcp  nlockmgr
|   100024    1          45015/udp status
|   100024    1          56701/tcp  status
139/tcp   open  netbios-ssn
MAC Address: 00:0C:29:9C:96:DD (VMware)

Host script results:
|_clock-skew: mean: 4h59m54s, deviation: 0s, median: 4h59m54s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:

```

Şekil 7.a'da çıktı üreten servis komut dosyaları, sistemin RSA ve DSA SSH anahtarlarını sağlayan ssh-hostkey ve portmapper'ı mevcut hizmetleri numaralandırmayı sorgulayan rpcinfo'dur. Bu örnekte, çıktı üreten tek host komut dosyası, SMB sunucularından çeşitli bilgiler toplayan smb-os-discovery'dir.

## Kullanım Şekilleri

**-script <filename>|<category>|<directory>|<expression>[,...]**: Virgülle ayrılmış dosya adı, script kategorileri ve dizin listesini kullanarak script taraması yapılır. Her öğe önce bir ifade, sonra bir kategori ve en sonunda bir dosya ya da dizin adı olarak yorumlanır. Script ifadesi listesindeki her öğeye, verilen script/kodları, script veya hostrule işlevlerindeki koşullardan bağımsız olarak çalışması için bir + karakteri eklenebilir. Nmap'ın mssql servisini tanıyabilmesi için kapsamlı sürüm tespiti (-sV –version-all) çalıştırarak Nmap taramasını yavaşlatmak yerine, **ms-sql-config** scriptini tüm hedeflenen hostlara ve portlara karşı çalıştırabilir. **-script + ms-sql-config** olarak yapılabilir.

**-script-args <args>, -script-args-file<filename>**: Scriptlere argümanlar sağlanır. “-script-args-file” parametresi, argümanların bir dosyada belirtilmesinde kullanılır.

**-script-help <filename>|<category>|<directory>|<expression>|all[,...]**: Scriptler hakkında bilgi elde etmek için –script-help kullanılır. Belirtilen öğe ile eşleşen her bir script için Nmap, script adını, kategorilerini ve açıklamasını yazdırır. Özellikler **-script** tarafından kabul edilenlerle aynıdır; Örneğin, **nmap –script-help ssl-enum-ciphers** komutunu çalıştırılabilir.

**-script-trace**: Bu parametre, **-packet-trace** ögesine benzemektedir. Ancak paket yerine uygulama düzeyinde çalışır. Bu parametre belirtilirse, scriptler tarafından gerçekleştirilen, tüm gelen ve giden iletişim yazdırılır. Görüntülenen bilgiler iletişim protokolünü, kaynak ve hedef adreslerini ve iletilen verileri içerir. İletilen verilerin %5'inden fazlası yazdırılamazsa, bunun yerine hex dökümleri yapılır. –packet-trace parametresinin belirlenmesi, script izlemeyi de sağlar. Kullanımı, Şekil 7.b'de gösterilmiştir.

Şekil 7.b – (–script-trace) parametresinin kullanımı

```

root@priviasecurity:~# nmap --script smb-os-discovery --script-trace 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-05 11:38 EST
NSE: UDP 192.168.16.129:48515 > 192.168.16.128:137 | CONNECT
NSE: UDP 192.168.16.129:48515 > 192.168.16.128:137 | 00000000: 13 37 00 00 00 00 01 00 00 00 00 00 00 00 00 00
00 20 43 4b 41 7 CKA
00000010: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA
00000020: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21 AAAAAAAA !
00000030: 00 01

NSE: UDP 192.168.16.129:48515 > 192.168.16.128:137 | SEND
NSE: UDP 192.168.16.129:48515 > 192.168.16.128:137 | 00000000: 13 37 84 00 00 00 00 00 01 00 00 00
00 20 43 4b 41 7 CKA
00000010: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAA
00000020: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21 AAAAAAAA !
00000030: 00 01 00 00 00 00 00 ad 07 4d 45 54 41 53 50 4c METASPL
00000040: 4f 49 54 41 42 4c 45 20 00 04 00 4d 45 54 41 53 OITABLE METAS
00000050: 50 4c 4f 49 54 41 42 4c 45 20 03 04 00 4d 45 54 PLOITABLE MET
00000060: 41 53 50 4c 4f 49 54 41 42 4c 45 20 20 04 00 01 ASPLOITABLE
00000070: 02 5f 5f 4d 53 42 52 4f 57 53 45 5f 5f 02 01 84 MSBROWSE
00000080: 00 57 4f 52 4b 47 52 4f 55 50 20 20 20 20 20 20 WORKGROUP
00000090: 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 WORKGROUP

```

**-script-updatedb:** Bu parametre, mevcut varsayılan scriptleri ve kategorileri belirlemek için Nmap tarafından kullanılan, script/script.db içinde bulunan script veritabanını günceller. Kullanımı Şekil 7.c'de gösterilmiştir.

Şekil 7.c – (–script-updatedb) parametresinin kullanımı

```

root@priviasecurity: ~
File Edit View Terminal Help
root@priviasecurity:~# nmap --script-updatedb
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-05 11:42 EST
NSE: Updating rule database.
NSE: Script Database updated successfully.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.95 seconds
root@priviasecurity:~#

```

## Script Kategorileri

NSE scriptleri, ait oldukları kategorilerin bir listesini tanımlar. Şu anda tanımlanmış kategoriler; auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version ve vuln. Kategori adlarının büyük/küçük harf duyarlılığı yoktur.

- ✓ **Auth:** Hedef sisteme yönelik kimlik doğrulama işlemini yapan scriptlerin kategorisidir.
- ✓ **Broadcast:** Genellikle listelenmeyen hostları yerel ağıda yayinallyarak keşfeden scriptlerdir.
- ✓ **Brute:** Uzak bir sunucunun kimlik doğrulama bilgilerini tahmin etmek için bruteforce saldıruları kullanan script kategorisidir.
- ✓ **Default:** Nmap'in –A parametresi ile kullanılan varsayılan scriptlerin kategorisidir. Bu kategori, –script=default kullanılarak belirtilir.

**Speed:** Bruteforce, kimlik doğrulama, crackleme, web dizin tespiti ve tek bir hizmeti taramak için hızlı sonuç veren default scriptleri kullanılır.

**Usefulness:** Default kategorisindeki değerli ve işlem yapılabilir bilgiler üreten scriptleri belirtir.

**Verbosity:** Nmap çıktısı çok çeşitli amaçlar için kullanılıp okunaklı ve özlü olması gereklidir. Sık sık çıktılarla dolu sayfalar üreten bir scripti, default kategorisinden çıkartır.

**Reliability:** Hedef host veya hizmetle ilgili sonuçlara ulaşmak için sevgisel ve bulanık imza eşleştirme için kullanılan scriptleri belirtir. Örneğin, sniffer tespiti ve sql injection scriptlerini içerir.

**Intrusiveness:** Sistemi veya hizmeti çektirebilen veya uzak yöneticilerin saldırısı olarak algılanan scriptleridir.

**Privacy:** Üçüncü şahıslara bilgi veren scriptlerdir. Örneğin, whois betiği hedef IP adresini bölgesel whois kayıtlarına göstermelidir.

✓ **Discovery:** Ağ ve ağa bağlı bütün cihazlar hakkında bilgi elde etmek için kullanılan scriptlerin kategorisidir.

✓ **Dos:** Denial Of Service scriptlerinden oluşur. Servislerin çökeltilmesine yönelik bir zafiyet olup olmadığını test etmek için kullanılan scriptlerdir.

✓ **Exploit:** Hedef sisteme bulunan bir güvenlik açığını sömürmek için kullanılan scriptlerdir. Örnekler arasında **jdwp-exec** ve **http-shellshock** bulunur.

✓ **External:** Bir üçüncü taraf veritabanına veya başka bir ağ kaynağına veri gönderen scriptlerdir. Örneğin, whois sunucularına hedefin adresi hakkında bilgi edinmek için bir bağlantı kuran whois-ip'tir.

✓ **Fuzzer:** Hedefin cevap vermemesine yönelik rastgele hazırlanmış paketlerle istek yapılmasını sağlayan scriptlerdir. Örneğin, bir DNS sunucusunu, sunucu çökene veya kullanıcı tarafından belirlenen bir zaman sınırı doluncaya kadar yavaşça hatalı DNS istekleriyle bombalayan **dns-fuzz**'dır.

✓ **Intrusive:** Hedef sistemi çökertmeyecek veya hedef sistemin zararlı olarak algılayacağı scriptlerdir. Örnekler **http-open-proxy** (hedef sunucusunu bir HTTP proxy'si olarak kullanmaya çalışır) ve **snmp-brute** (genel, özel ve cisco gibi ortak değerler göndererek bir cihazın SNMP topluluk dizesini tahmin etmeye çalışır).

✓ **Malware:** Hedef platformun zararlı yazılımlara veya backdoorlara bulaşıp bulaşmadığını tespit eden scriptler malware kategorisinde yer almır. Örneğin, 25 numaralı port dışındaki farklı portlarda çalışan SMTP sunucularını izleyen **smtp-strangeport** ve herhangi bir istek almadan, istek almış gibi davranışını cevap veren kimlik doğrulama scripti **auth-spoof**'tur.

✓ **Safe:** Servisleri çökertmek, büyük miktarda ağ bant genişliği kullanmak veya güvenlik açıklarından yararlanmak için tasarlanmamış scriptlerdir. Örneğin, **ssh-hostkey** (bir SSH host anahtarı alır) ve **html-title** (başlığı bir web sayfasından alır). Sürüm kategorisindeki scriptler güvenlik açısından sınıflandırılmaz.

✓ **Version:** Sürüm tespit etme özelliğinin bir uzantısıdır. Açıkça seçilemez. Yalnızca sürüm tespiti (-sV) istendiğinde çalışacak şekilde seçilirler. Çıktıları sürüm tespit çıktısından ayırt edilemez ve hizmet ya da host script sonuçları üretmezler. Örneğin, **skypev2 sürümü**, **pptp sürümü** ve **iax2 sürümü**.

✓ **Vuln:** Bilinen bazı güvenlik açıklarının hedef platformda olup olmadığını tespit etmek için kullanılan scriptlerdir. Örneğin, **realvnc-auth-bypass** ve **afp-path-vuln**, **smb-vuln-ms17-010** scriptleri bulunur.

## Script Seçimi

**-script** parametresi, virgülle ayrılmış kategorilerin, dosya adlarının ve dizin adlarının gösterimiyle kullanılır.

**nmap -script default, safe:** Default ve Safe kategorilerindeki scriptler kullanılır. Örnek olarak Şekil 7.3'te gösterilmiştir.

Şekil 7.3 – Varsayılan script kategorisinin belirlenmesi

```

root@priviasecurity:~# nmap --script default,safe 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-05 11:51 EST
Pre-scan script results:
| broadcast-dhcp-discover:
  Response 1 of 1:
    IP Offered: 192.168.16.130
    Server Identifier: 192.168.16.254
    Subnet Mask: 255.255.255.0
    Router: 192.168.16.2
    Domain Name Server: 192.168.16.2
    Domain Name: localdomain
    Broadcast Address: 192.168.16.255
    NetBIOS Name Server: 192.168.16.2
| broadcast-igmp-discovery:
  192.168.16.1
    Interface: eth0
    Version: 2
    Group: 224.0.0.252
    Description: Link-local Multicast Name Resolution (rfc4795)
    Use the newtargets script-arg to add the results as targets
| broadcast-listener:
  ether
    ARP Request
      sender ip      sender mac      target ip
      192.168.16.2  00:50:56:FD:9E:49  192.168.16.131
  udp
    DHCP
      srv ip      cli ip      mask      gw      dns      vendor
      192.168.16.254  192.168.16.131  255.255.255.0  192.168.16.2  192.168.16.2  -
      192.168.16.254  192.168.16.130  255.255.255.0  192.168.16.2  192.168.16.2  -
| broadcast-netbios-master-browser:

```

**nmap –script smb-os-discovery:** Yalnızca smb-os-discovery scripti kullanılır. .nse doya uzantısını eklemek zorunlu değildir.

**nmap –script default, banner, /home/user/customscripts:** Default kategorisindeki scriptler, banner scripti ve /home/user/customscripts dizininde bulunan scriptler kullanılır.

**nmap –script “http-\*”:** Http-auth ve http-open-proxy gibi, adı http- ile başlayan tüm scriptler kullanılır. –script parametresi, wildcard karakterleri kabuktan korumak için tırnak içinde olmalıdır. Boolean ifadeleri oluşturmak için ve/veya operatörleri kullanılarak daha karmaşık komut dosyası seçimi yapılabilir.

**nmap –script “not intrusive”:** intrusive kategorisi dışındaki scriptlerin kullanılmasını sağlar.

**nmap –script “default or safe”:** İşlevsel olarak nmap –script “default, safe” komutu ile eşdeğerdir. Default ya da safe kategorisindeki scriptler veya her ikisindeki scriptler kullanılır.

**nmap –script “default and safe”:** Hem default hem de safe kategorisindeki scriptler kullanılır.

**nmap –script “(default or safe or intrusive) and not http-\*”:** http- ile başlayan scriptler hariç, default, safe ya da intrusive kategorisindeki scriptler kullanılır.

## Script Tipleri ve Aşamaları

NSE, alındıkları hedeflerin türü ve çalıştırıldığı tarama aşaması ile ayrı edilen dört tür scripti destekler. Bireysel scriptler, çoklu işlem türlerini destekleyebilir.

**Prerule Scriptleri:** Herhangi bir Nmap'ın tarama aşamasından önce yayınlanır. Bu nedenle Nmap henüz hedefleri hakkında herhangi bir bilgi toplamayabilir. DHCP ve DNS SD sunucularını sorgulamak için ağ yayın istekleri yapmak gibi belirli tarama hedeflerine bağlı olmayan görevler için faydalı olabilirler. Bu scriptlerden bazıları, Nmap'ın taraması için yeni hedefler oluşturabilir. Örneğin, **dns-zone-transfer**, zone transfer isteğini kullanarak bir alandaki IP'lerin listesini alabilir ve bunları otomatik olarak Nmap'ın tarama hedefi listesine eklemektedir.

**Host Scriptleri:** Bu aşamadaki scriptler, Nmap host bulma, port tarama, sürüm tespiti ve hedef hostlara karşı işletim sistemi tespiti gerçekleştirdikten sonra Nmap'ın normal tarama işlemi sırasında çalışır. Bu tür scriptler, **hostrule** işleviyle eşleşen her hedef hosta karşı bir kez çağrırlar. Örneğin, bir hedef IP için ownership bilgilerini arayan whois-ip ve parçalanma gerektirmeden hedefe ulaşabilecek maksimum IP paket boyutunu belirlemeye çalışan yol **mtu**'dur.

**Servis Scriptleri:** Bu scriptler, bir hedef hostta dinleyen belirli servislere karşı çalıştırılır. Örneğin, Nmap, web sunucularına karşı çalıştmak için 15'ten fazla http hizmeti scripti içerir. Bir hostta birden fazla portta çalışan web sunucusu varsa, bu komut dosyaları birden çok kez çalışabilir. Bunlar en çok kullanılan Nmap komut dosyası türüdür ve bir komut dosyasının hangi algılayıcılarla karşı çalıştırılacağına karar vermek için bir **portrule** işlevi içерerek ayrı edilirler.

**Postrule Scriptleri:** Nmap tüm hedeflerini taradıktan sonra yayılanır. Nmap çıktısını biçimlendirmek ve sunmak için faydalı olabilirler. Örneğin, **ssh-hostkey**, SSH sunucularına bağlanan, genel anahtarlarını keşfeden ve bunları basan hizmet (**portrule**) scriptleriyle en iyi bilinir. Ancak, taranan tüm hostlar arasında yinelenen anahtarları denetleyen, ardından bulunanları yazdırın bir posta kodu da içerir. Bir **postrule** scripti için bir başka kullanım, Nmap çıktısının ters bir indeksini basmaktadır. Hangi hostların yalnızca her bir hosttaki hizmetleri listelemek yerine belirli bir servisi çalıştığını gösterir. Postrule scriptleri, postrule işlevi içерerek tanımlanır. Birçok script, bir ön hazırlık ya da önyükleme dosyası olarak çalıştırılabilir. Bu durumlarda, tutarlılık için bir primer kullanmanız faydalı olacaktır.

## Scriptlerden Bağımsız Değişkenler

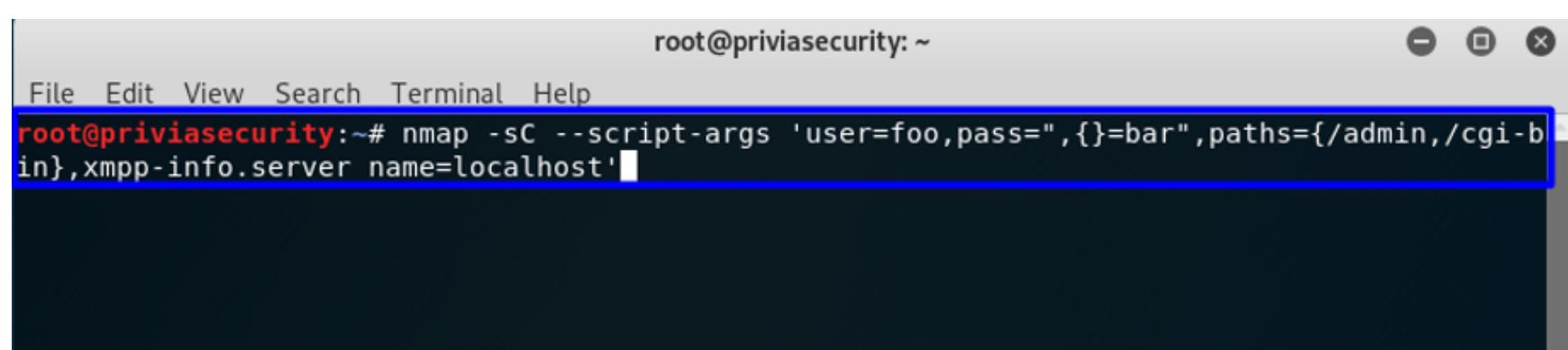
Argümanlar **-script-args** parametresi kullanılarak NSE scriptlerine iletilebilir. Argümanlar bir **key-value** çiftleri tablosunu ve muhtemelen dizi değerlerini açıklar. Argümanlar, scriptlere **nmap.registry.args** adlı kayıt defterinde bir tablo olarak tutulur.

Ancak normalde **stdnse.get\_script\_args** işleviyle erişilebilirler. Komut satırı bağımsız değişkenlerinin sözdizimi, Lua'nın tablo yapıcısının sözdizimine benzemektedir.

Bağımsız değişkenler, virgülle ayrılmış bir ad listesidir: Değer çiftleridir. Adlar ve değerler, boşluk içermeyen dizeler veya '{', '}', '=' veya ',' karakterleri olabilir. Alıntı yapılan bir dizgede '\' bir alıntıdan kaçar.

Bir ters eğik çizgi yalnızca bu özel durumda tırnak işaretlerinden kaçmak için kullanılır; Diğer tüm durumlarda, ters eğik çizgi tam anlamıyla yorumlanır. Komut satırındaki **-script-args** komutundaki argümanları iletmek yerine, bunları bir dosyada (virgül veya yeni satırlarla ayrılmış) saklayabilir ve **-script-args-file** ile sadece dosya adını belirleyebilirsiniz. Komut satırında **-script-args** ile belirtilen parametreler, dosyada verilenlerden önceliklidir. Dosya adı mutlak bir yol olarak veya Nmap'ın normal arama yoluna (NMAPDIR, vs.) göre görülebilir. Argüman kullanımına örnek olarak Şekil 7.5 verilmiştir.

Şekil 7.5- Nmap komut satırında argüman kullanımı



```
root@priviasecurity: ~
File Edit View Search Terminal Help
root@priviasecurity:~# nmap -sC --script-args 'user=foo,pass=",{}=bar",paths={/admin,/cgi-bin},xmpp-info.server name=localhost'
```

## Script Dili

Nmap Scripting Engine çekirdeği gömülebilir bir Lua yorumlayıcısıdır. Lua, genişletilebilirlik için tasarlanmış hafif bir dildir. Nmap gibi diğer yazılımlarla arayüz oluşturmak için güçlü ve iyi belgelenmiş bir API sunmaktadır. Nmap Scripting Enginenin ikinci kısmı, Lua ile Nmap'ı birbirine bağlayan NSE kütüphanesidir. Bu katman, Lua yorumlayıcısının başlatılması, paralel komut dosyası çalışma zamanlaması, komut dosyası alımı ve daha fazlası gibi sorunları ele alır. Aynı zamanda NSE ağ I/O framework ve yönetim mekanizmasının kalbidir. Ayrıca, komut dosyalarını daha güçlü ve kullanışlı hale getirmek için yardımcı program kitaplıkları da içerir.

Nmap script dili, Nmap ile arabirim oluşturmak için kütüphanelerle genişletilen gömülü bir Lua yorumlayıcısıdır. Nmap API, Lua namespace'inde nmap adıyla bulunur. Bu, Nmap tarafından sağlanan tüm kaynaklara yapılan çağrıların bir nmap önekinde sahip olduğu anlamına gelmektedir. **nmap.new socket()**, örneğin, yeni bir soket sarmalayıcı nesnesi döndürür. Nmap kütüphane katmanı aynı zamanda Lua içeriğini başlatmayı, paralel kodları planlamayı ve tamamlanmış kodlar tarafından üretilen çıktıyı toplamayı da önemsemektedir. Planlama aşamalarında, Nmap script için temel olarak birkaç programlama dili düşünüldü. Diğer bir seçenek ise tamamen yeni bir programlama dili uygulamaktır. NSE'nin kullanımı kolay, küçük boyutlu, Nmap lisansı ile uyumlu, ölçeklenebilir, hızlı ve paralelleştirilebilir olması gerekiyordu. Perl, Python ve Ruby dillerde daha kapsamlı interpreterler vardır. Ancak, Nmap içerisinde verimli bir şekilde gömülmesi zordur. Ama, Lua Nmap içerisinde gömülü olduğu gibi, Wireshark sniffer ve Snort IDS gibi diğer popüler açık kaynaklı güvenlik araçlarında bile gömülü geliyor.

Lua'nın önemli yerleşik özelliklerine ek olarak, senaryo yazmayı daha güçlü ve kullanışlı hale getiren birçok uzantı kütüphanesi yazılmış ve entegre edilmiştir. Gerekirse bu kütüphaneler (bazen modüller olarak da adlandırılır) derlenir ve Nmap ile birlikte kurulur. Yapılandırılmış Nmap veri dizinine yüklenen kendi dizinleri neslib vardır. NSE Kütüphaneleri Şekil 7.6'da gösterilmiştir.

Şekil 7.6 – NSE Kütüphaneleri

1. afp	28. giop	55. ndmp	82. snmp
2. ajp	29. gps	56. netbios	83. socks
3. amqp	30. http	57. nmap	84. srvloc
4. asn1	31. httpspider	58. nrpc	85. ssh1
5. base32	32. iax2	59. nsedebug	86. ssh2
6. base64	33. imap	60. omp2	87. sslicert
7. bin	34. informix	61. openssl	88. stdnse
8. bit	35. ipOps	62. ospf	89. strbuf
9. bitcoin	36. ipp	63. packet	90. strict
10. bittorrent	37. iscsi	64. pcre	91. stun
11. bjnp	38. isns	65. pgsql	92. tab
12. brute	39. jdwp	66. pop3	93. target
13. cassandra	40. json	67. pppoe	94. tftp
14. citrixxml	41. ldap	68. proxy	95. tns
15. comm	42. lfs	69. rdp	96. unpwdb
16. creds	43. listop	70. redis	97. upnp
17. cvs	44. match	71. rmi	98. url
18. datafiles	45. membase	72. rpc	99. versant
19. dhcp	46. mobileme	73. rpcap	100. vnc
20. dhcp6	47. mongodb	74. rsync	101. vulns
21. dns	48. msrpc	75. rtsp	102. vuzedht
22. dnsbl	49. msrpcperformance	76. sasl	103. wsdd
23. dnssd	50. msrpctypes	77. shortport	104. xdmcp
24. drda	51. mssql	78. sip	105. xmpp
25. eap	52. mysql	79. smb	
26. eigrp	53. natpmp	80. smbauth	
27. ftp	54. ncp	81. smtp	

## GÜVENLİK DUVARLARI VE IDS SİSTEMLERİNİN TESPİTİ

Yapılan taramalarda güvenlik duvarları ağın haritalanmasını zorlaştırmaktadır. Sistemler hakkında keşif taramaları gerçekleştirildiğinde güvenlik duvarları ve IDS sistemleri bu tarama işlemlerini engelleyebilmektedir. Nmap, bu karmaşık ağları anlamaya yardımcı olmak ve filtrelerin amaçlandığı gibi çalıştığını doğrulamak için birçok özellik sunmaktadır. Tüm büyük IDS'ler Nmap taramalarını tespit etmek için tasarlanmış kurallarla birlikte gelmektedir. Çünkü taramalar bazen saldırırlara öncülük etmektedir.

## Güvenlik Duvarı Kurallarını Belirlemek

Güvenlik duvarı kurallarını atlamak için nasıl çalışıklarını bilmek gerekmektedir. Nmap ulaşılabilir, ancak kapalı olan ve aktif olarak filtrelenen portları birbirinden ayırmaktadır. Etkili bir teknik, normal bir SYN port taraması ile başlamak, daha sonra ağın daha iyi anlamak için ACK taraması ve IP ID sıralaması gibi teknikler kullanmaktadır.

## TCP SYN Scan

TCP SYN(Stealth) Scan, TCP portlarını taramanın en hızlı yolu olduğu için en popüler tarama türüdür. Connect taramasından daha gizlidir ve tüm işlevsel TCP yığınlarına karşı çalışmaktadır. SYN veya Stealth taraması, bir SYN paketi gönderip gelen cevabı göz önüne alarak bu prosedürü kullanmaktadır. SYN/ACK geri gönderilirse, TCP connection bağlantısı ile açık porta uzaktan bağlanmaya çalışır. Tarayıcı tamamen kurulmadan önce bağlantıyı down etmek için bir RST gönderir; genellikle uygulama logları bağlantı girişiminin görünmesini önlemektedir. Port kapalıysa, bir RST gönderilir. Eğer filtre edilirse, SYN paketi düşürülmüş olacak ve herhangi bir cevap gönderilmeyecektir. Bu şekilde, Nmap portun açık, kapalı ya da filtreli olup olmadığını tespit etmektedir. Şekil 8.1.1.a'da 956 tane portun kapalı, 40 tane portun filtreli olduğu belirtilmektedir. Varsayılan tarama yaptığımız için 1000 portu taradığına göre 4 portu da açık olarak belirledi.

Şekil 8.1.1.a – Varsayılan SYN Scan Örneği

```
root@priviasecurity: ~
File Edit View Search Terminal Help
root@priviasecurity:~# nmap -sS 192.168.16.160

Starting Nmap 7.50 ( https://nmap.org ) at 2019-02-01 12:38 EST
Nmap scan report for 192.168.16.160
Host is up (0.0014s latency).
Not shown: 956 closed ports, 40 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
MAC Address: 00:0C:29:A1:79:1C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.04 seconds
root@priviasecurity:~#
```

**RST döndüren gizli güvenlik duvarları:** Kapalı TCP portları (bir RST paketi döndüren) ve filtrelenmiş portlar (hiçbir şey veya bir ICMP hatası döndürmeyen) arasındaki Nmap ayrimı genellikle doğru olsa da, birçok güvenlik duvarı cihazı şimdi RST paketlerini hedef hosttan geliyormuş gibi yapabilir ve portun kapalı olduğunu iddia edebilir. Bu özelliğin bir örneği, istenmeyen paketleri reddetmek için birçok yöntem sunan Linux iptables sistemidir. Iptables kılavuzu özelliği, **-reject-with** tipi olarak belirtmektedir. Verilen tip uygun ICMP hata mesajını döndüren icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited veya icmp-host-prohibited olabilir, unreachable varsayılandır). **Tcp-reset** parametresi yalnızca TCP protokolüyle eşleşen kurallarda kullanılabilir: bu bir TCP RST paketinin geri gönderilmesine neden olur. Bu, çoğunlukla bozuk posta hostlarına, posta gönderirken sıkça oluşan ident (113/tcp) problemini engellemek için kullanışlıdır (aksi halde postanızı kabul etmeyecektir). RST paketlerini güvenlik duvarları ve IDS/IPS ile oluşturmak, ağ operatörlerinin kafasını karıştırabildiğinden ve tarayıcıların, düşmüş paketlerin neden olduğu zaman aşımını beklemeden hemen bir sonraki porta geçmesine izin verdiginden özellikle 113 numaralı port dışında yaygın değildir. Böyle bir sahtecilik, RST paketinin, makine tarafından gönderilen diğer paketlerle karşılaşıldığında dikkatli bir şekilde analiz edilmesiyle tespit edilebilir.

## TCP ACK Scan

TCP ACK Scan, güvenlik duvarı kurallarının durumlu olup olmadığını ve hangi portların filtrelendiğini belirlemek için kullanılır. Dezavantajı ise açık portları kapalı portlardan ayırt edememesidir. **-scanflags** parametresini kullanmıyorsanız, varsayılan olarak bu tarama türü problkardaki ACK bayrağını ayarlamaktadır. Filtrelenmemiş sistemleri tararken, açık ve kapalı portların her ikisi de bir RST paketi döndürmektedir. Nmap aracı da bunları unfiltered olarak etiketlemektedir. Bu durum, ACK paketleri tarafından erişilebilecekleri anlamına gelmektedir. Ancak Açık veya kapalı olup olmamaları belirsizdir. Yanıt vermeyen veya belirli ICMP hata mesajlarını geri gönderen portlar (type 3, codes 0, 1, 2, 3, 9, 10 veya 13),filtrelenmiş olarak etiketlenir. Ayrıca bu taramayı gerçekleştirmek için **-sA** parametresini kullanmanız yeterli olacaktır. Şekil 8.1.2 'de gösterilmiştir.

Şekil 8.1.2 – TCP ACK Scan Gösterimi

```
root@priviasecurity:~# nmap -sA -n -v 192.168.16.160
Starting Nmap 7.50 ( https://nmap.org ) at 2019-02-09 11:06 EST
Initiating ARP Ping Scan at 11:06
Scanning 192.168.16.160 [1 port]
Completed ARP Ping Scan at 11:06, 0.22s elapsed (1 total hosts)
Initiating ACK Scan at 11:06
Scanning 192.168.16.160 [1000 ports]
Increasing send delay for 192.168.16.160 from 0 to 5 due to 248 out of 825 dropped
t increase.
Increasing send delay for 192.168.16.160 from 5 to 10 due to 11 out of 23 dropped p
increase.
Increasing send delay for 192.168.16.160 from 10 to 20 due to 11 out of 33 dropped
t increase.
Increasing send delay for 192.168.16.160 from 20 to 40 due to 11 out of 33 dropped
t increase.
Increasing send delay for 192.168.16.160 from 40 to 80 due to 11 out of 32 dropped
t increase.
Completed ACK Scan at 11:06, 25.14s elapsed (1000 total ports)
Nmap scan report for 192.168.16.160
Host is up (0.00060s latency).
All 1000 scanned ports on 192.168.16.160 are unfiltered
MAC Address: 00:0C:29:A1:79:1C (VMware)
```

## IP ID Püf Noktaları

IP başlıklarındaki ID alanında şartsızı miktarda bilgi açığa çıkabilir. Idle Scan tekniğide kullanıldığı gibi güvenlik duvarlarını kandırmak için RST paketlerinin saldırıcı bir sistemden gelmediğini göstermeye çalışılır. Bir başka yol da, güvenilir kaynak adreslerinin güvenlik duvarının kontrol mekanizmalarına takılmadan çalışmasıdır. Örneğin, bir şirketin üretim ağındaki makineler, şirket ağındaki IP adreslerine güvenebilir veya bir sistem yöneticisinin kişisel makinesine güvenebilir. Güvenilir kaynak adres sorununun somut bir örneği, bir şirketin özel UDP hizmetinin, bir yapılandırma dosyasına girilen özel bloklardan geliyorsa, kullanıcıların kimlik doğrulaması mekanizması atlatılabilir. Bu ağ blokları farklı kurumsal konumlara karşılık gelip bu özellik yönetimi ve hata ayıklamayı kolaylaştırmaktadır. Bu güvenlik duvarı kurallarını belirleme tekniği olarak Nmap kullanılmamalıdır. Ama Nmap taramalarının çıktıları önemlidir. Örneğin, bu test bazı kod çözüçüler kullanılıp kullanılmayacağını gösterebilir (-D).

## UDP Sürüm Taraması

UDP ile çalışmak genellikle daha zordur. Çünkü protokol, TCP gibi açık portların onaylanması yapmamaktadır. Birçok UDP uygulaması beklenmedik paketleri görmezden gelmektedir. Nmap, portların açık veyafiltrelenmiş olup olmadığını öğrenmek amacıyla açık portlardan yanıt alabilmek için birbirinden farklı UDP hizmetine birçok UDP probu gönderir. Şekil 8.1.4'te 53 numaralı portun açık ve üzerinde bir servisin çalışıyor olduğu görülmektedir. Diğer portlar hala open|filtered,'dır. Çünkü probların hiçbirine cevap dönmemektedir.

Şekil 8.1.4 – UDP sürüm taraması örneği

```
root@priviasecurity:~# nmap -sV -sU -p 50-59 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 12:06 EST
Nmap scan report for 192.168.16.128
Host is up (0.00032s latency).

PORT      STATE SERVICE      VERSION
50/udp    closed  re-mail-ck
51/udp    closed  la-maint
52/udp    closed  xns-time
53/udp    open   domain      ISC BIND 9.4.2
54/udp    closed  xns-ch
55/udp    closed  isi-gl
56/udp    closed  xns-auth
57/udp    closed  priv-term
58/udp    closed  xns-mail
59/udp    closed  priv-file
MAC Address: 00:0C:29:9C:96:DD (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.70 seconds
```

## Güvenlik Duvarı Kurallarını Atlatma Teknikleri

Güvenlik duvarı kurallarını belirlemek değerli olsa da kuralları atlama çoğu zaman öncelikli hedeftir. Nmap bunu yapmak için birçok teknik kullanarak kötü yapılandırılmış güvenlik duvarlarını atlayabilmektedir. Saldırganın başarılı olabilmesi için oluşturulan yanlış yapılandırmaların birini bulması gereklidir, ağ yöneticilerinin yanlış yapılandırmaların hepsini tespit edip düzeltmeleri gerekmektedir.

## Egzotik Tarama Bayrakları

Ağ portlarından hangisininfiltrelendiğini belirlemek için bir ACK taraması kullanılabilir. Ancak, erişilebilir portlardan hangisinin açık veya kapalı olduğu bilinmemektedir. Nmap, istenen port durumu bilgilerini verirken güvenlik duvarlarını gizlice atlatmak için iyi olan birkaç tarama yöntemi sunmaktadır. FIN taraması böyle bir tekniktir. `-sF` parametresi kullanılarak FIN taraması gerçekleştirilir. Şekil 8.2.1'de, bu kez bir FIN taraması kullanarak FIN taraması yapılmıştır. Bir FIN paketi ayarlandığında, SYN paketlerini engelleyen kuralların ötesine geçmektedir. Bir SYN taraması 100'ün altında yalnızca bir açık port bulurken, FIN taraması ikisini de bulmaktadır.

Şekil 8.2.1 – Egzotik tarama bayraklarının kullanımı

```
root@priviasecurity:~# nmap -sF -p1-100 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 12:44 EST
Nmap scan report for 192.168.16.128
Host is up (0.00024s latency).

Not shown: 94 closed ports
PORT      STATE      SERVICE
21/tcp    open|filtered  ftp
22/tcp    open|filtered  ssh
23/tcp    open|filtered  telnet
25/tcp    open|filtered  smtp
53/tcp    open|filtered  domain
80/tcp    open|filtered  http
MAC Address: 00:0C:29:9C:96:DD (VMware)
```

Diğer birçok tarama türü denenmeye değer tarama türleridir. Çünkü hedef güvenlik duvarı kuralları ve hedef host türü hangi tekniklerin çalışacağını belirlemektedir. Bazı değerli tarama türleri FIN, Maimon, Window, SYN/FIN ve NULL taramalıdır.

## Kaynak Port Manipülasyonu

Yaygın olarak yapılan yanlış yapılandırmaların biri, yalnızca kaynak port numarasına dayanan trafiğe güvenmektir. Bir yönetici, uygulamaları durdurulan kullanıcıların şikayetleri ile ilgilenecek şekilde yeni bir güvenlik duvarı kurarken, UDP DNS harici sunuculardan gelen yanıtlar artık ağa giremediğinden DNS çökebilir. Ayrıca FTP protokolü üzerinden dosya paylaşımında proxyler, sunucular görevlendirilir. Bazı sistem ve ağ yöneticilerinin FTP protokolü üzerinden saldırı yapılmayacağını varsayılmaktadır. Aynı durum 53 numaralı DNS portu içinde geçerlidir. Nmap, bu zayıflıklardan

yararlanmak için **-g** ve **-source-port** seçeneklerini sunmaktadır. Bir port numarası verip bu porta paketler gönderilmektedir. Nmap, belirli işletim sistemi tespit testlerinin düzgün çalışması için farklı port numaraları kullanmalıdır. SYN taraması dâhil olmak üzere çoğu TCP taraması, UDP taraması gibi seçeneği tamamen desteklemektedir. Şekil 8.2.2'de gösterildiği gibi kaynak port manipülasyon işlemini gerçekleştirmiştir. Bunun için **-g** parametresi kullanılmıştır. **-sS** parametresi ile SYN taraması, **-v** parametresi ile detaylı sonuç getirmesini, ikinci **-v** parametresi ile çıktıları detaylı bir şekilde getirmesi sağlanmıştır. Ayrıca **-n** parametresi ile dns çözümleme yapmaması için kullanılmıştır. **-Pn** parametresini kullanarak pingsiz tarama yapmasını ve **-p1-100** parametresini belirterek ilk 100 portu taraması istenilmiştir.

Şekil 8.2.2 – Kaynak port manipülasyonu gösterimi

```
Raw packets: 0 sent, 0 received, 0% loss, 0Kb/s
root@priviasecurity:~# nmap -sS -v -n -Pn -g 25 -p1-100 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 13:01 EST
Initiating ARP Ping Scan at 13:01
Scanning 192.168.16.128 [1 port]
Completed ARP Ping Scan at 13:01, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 13:01
Scanning 192.168.16.128 [100 ports]
Discovered open port 80/tcp on 192.168.16.128
Discovered open port 23/tcp on 192.168.16.128
Discovered open port 21/tcp on 192.168.16.128
Discovered open port 25/tcp on 192.168.16.128
Discovered open port 53/tcp on 192.168.16.128
Discovered open port 22/tcp on 192.168.16.128
Completed SYN Stealth Scan at 13:01, 0.05s elapsed (100 total ports)
Nmap scan report for 192.168.16.128
Host is up, received arp-response (0.00062s latency).
Scanned at 2019-03-06 13:01:22 EST for 0s
Not shown: 94 closed ports
Reason: 94 resets
PORT      STATE SERVICE REASON
21/tcp    open  ftp      syn-ack ttl 64
22/tcp    open  ssh      syn-ack ttl 64
23/tcp    open  telnet   syn-ack ttl 64
25/tcp    open  smtp     syn-ack ttl 64
53/tcp    open  domain   syn-ack ttl 64
80/tcp    open  http     syn-ack ttl 64
MAC Address: 00:0C:29:9C:96:DD (VMware)
```

## IPv6 Saldırıları

IPv6 tam olarak dünyanın her yerinde olmasa bile, Japonya ve diğer bazı bölgelerde oldukça popülerdir. IPv6'yi filtrelemek bazen IPv4'ten daha kritik olabilir. Çünkü genişletilmiş adres alanı, genel olarak adreslenebilir IPv6 adreslerinin genellikle RFC 1918 tarafından belirtilen özel IPv4 adreslerini kullanmak zorunda kalacak makinelere tahsis edilmesini sağlar. IPv4 varsayılan yerine IPv6 taraması gerçekleştirmek çoğu zaman komut satırına **-6** eklemek kadar kolaydır. İşletim sistemi tespiti ve UDP tarama gibi bazı özellikler bu protokol için henüz desteklememektedir. Ancak en popüler özellikler çalışmaktadır.

## IP ID Idle Scan

IP ID Idle scan, hedefinize hiçbir adres göndermez. Çünkü en gizli tarama türlerinden biridir. Açık portlar, seçilen bir zombi makinesinin IP ID sequencelerinden çıkarılır. Idle scan'ın daha az bilinen bir özelliği, elde edilen sonuçların, eğer zombinin doğrudan hedef host taraması durumunda elde edeceğiniz sonuçlardır. **-g** parametresinin güvenilir kaynak portlarından yararlanmasına izin verdiği gibi, Idle scan bazen güvenilir kaynak IP adreslerinden de yararlanabilir.

## Çoklu Ping Problemleri

Güvenlik duvarı ağlarını taramaya çalışırken, atılan ping problemleri hostlardan cevap alamayabilir. Bu sorunu azaltmak için Nmap, çok çeşitli problemlerin paralel olarak gönderilmesine izin verir.

## Fragmentation (Parçalama)

Parçalama işlemi güvenlik duvarlarını atlatmak için kullanılan tekniklerden biridir. Bazı güvenlik duvarları paketleri boyutlarını göz önüne alarak paketin incelenmesini yapmaktadır. Paket boyutlarının parçalanması ile güvenlik duvarı paketleri tek bir paket halinde inceleyemediği için kolaylıkla güvenlik duvarı atlatılır. Paketlerin parçalanması için **-f** parametresi kullanılır. Nmap, paketlerin parçalanması işleminde her parçalanan paketin boyutunu 8 bayt olarak belirler. Bu nedenle tipik bir 20 veya 24 bayt TCP paketi üç küçük parça halinde gönderilir. **-f** parametresinin kullanımı her parçanın 8 bayt olduğunu belirtir. Yani **-f -f**, her bir parça içerisinde 16'ya kadar veri baytı sağlar. Alternatif olarak, **-mtu** parametresini belirtebilir. **-mtu** argümanı sekizin bir katı olmalı ve **-f** parametresiyle birleştirilmemelidir.

Şekil 8.2.6 – Varsayılan bir taramada fragmentation kullanımı

```
root@priviasecurity:~# nmap -f -f -p1-100 192.168.16.128
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 13:34 EST
Nmap scan report for 192.168.16.128
Host is up (0.00080s latency).

Not shown: 94 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
MAC Address: 00:0C:29:9C:96:DD (VMware)
```

IP katmanını atlamak ve raw ethernet frameleri göndermek için **-send-eth** parametresi kullanılabilir. Fragmentation, Nmap'ın yalnızca TCP ve UDP port taramaları (connect scan ve FTP bounce scan hariç) ve işletim sistemi tespiti içeren ham paket özellikleri için destekler. Sürüm tespiti ve Nmap Scripting Engine gibi özellikler genellikle parçalanmayı desteklemez. Çünkü hedef hizmetlerle iletişim kurmak için makinedeki TCP stack'e güvenir. Nmap, herhangi bir çakışma olmadan sırayla parçaları gönderir. Parçalanan bir port taraması geçerse, ana makineye saldırmak için kullanılan diğer araçları ve istismarları parçalamak için fragroute gibi bir araç kullanılabilir.

## Proxyler

Web için uygulama düzeyinde proxy'ler, algılanan güvenlik ve ağ verimliliği (önbelleklemeye yoluyla) yararları nedeniyle popüler hale gelir. Güvenlik duvarları ve IDS gibi, yanlış yapılandırılmış proxy'ler çözümlerinden çok daha fazla güvenlik sorununa neden olabilir. En sık karşılaşılan sorun uygun erişim kontrollerinin ayarlanamamasıdır. Internette yüzbinlerce geniş açık proxy sunucusu bulunur. Bu sayede herhangi birinin başka internet sitelerine isimsiz atlamalı noktalar olarak kullanılmasına izin verir. Birçok kuruluş açık proxy'leri bulmak ve IP adreslerini dağıtmak için otomatik tarayıcılar kullanır. Açık proxy'ler daha çok sitelere sızmak, kredi kartı sahtekârlığı yapmak ya da interneti spam ile doldurmak isteyen daha kötü insanlar tarafından kötüye kullanılır. Internet kaynaklarında açık bir proxy barındırmak çok sayıda soruna neden olabilir. Ancak açık proxy'lerin korumalı ağa yeniden bağlantı kurmasına izin verilmesi daha ciddi bir durumdur. Dâhili ana makinelere Internet kaynaklarına erişmek için bir proxy kullanması gerektiğine karar veren yöneticiler, istemeden de olsa trafiğe ters yönde de izin verir. **Hacker Adrian Lamo, genellikle bu reverse-proxy tekniğinden yararlanarak Microsoft, Excite, Yahoo, WorldCom, New York Times ve diğer büyük ağlara girdiği için ünlüdür.**

## MAC Address Spoofing Tekniği

Ethernet cihazları, benzersiz bir altı bayt Media Access Control (MAC) adresiyle tanımlanır. İlk üç bayt organizasyonel olarak benzersiz bir tanımlayıcı (OUI) oluşturur. Bu ön ek, bir satıcıya IEEE tarafından atanır. Satıcı daha sonra kalan üç byte'ı sattığı adaptörlere ve cihazlara benzersiz bir şekilde atamaktan sorumludur. Nmap, OUI'leri atandıkları satıcı adlarıyla eşleştirilen bir veritabanı içerir. Bu, bir ağı tararken aygıtları tanımlamaya yardımcı olur. OUI veritabanı dosyası, nmap-mac-prefixes bulunur. MAC adresleri ethernet cihazlarına önceden atanmış olsalar da, mevcut donanımların çoğu sürücü ile değiştirilebilir. Örneğin, çoğu Wireless Access Point, belirli bir MAC adresi grubuna erişimi sınırlamak için bir yapılandırma seçeneği sunar. Benzer şekilde, bazı ücretli veya özel ağlar, sizi bir web formu kullanarak bağlandıktan sonra doğrulamaya veya ödeme yapmaya zorlar. Ardından, MAC adresinize dayanarak ağın geri kalanına erişmenizi sağlar. MAC adreslerini spooflamadan ve MAC'in ağa yetkisiz erişim sağlama için spoofing yapmasının kolay olduğu göz önüne alındığında, bu erişim kontrolü şekli oldukça zayıftır. Bir yönlendiriciyi geçerken son sunucunun MAC adresi değiştirilir. Erişim kontrolüne ek olarak, MAC adresleri bazen hesap verebilirlik için kullanılır. Ağ yöneticileri, DHCP lease aldıklarında veya yeni bir makine ağda iletişim kurduğunda MAC adreslerini kaydedeler. Ağ

kötüye kullanılması veya korsanlık şikayetleri alınırsa, IP adresini ve olay saatini temel alarak MAC adresini bulur. Sonra sorumlu makineyi ve sahibini bulmak için MAC kullanılır. Nmap, **-spoof-mac** parametresiyle MAC Address Spoofing işlemi yapar. Verilen argüman birkaç şekilde olabilir. Eğer sadece 0 ise, Nmap oturum için tamamen rasgele bir MAC adresi seçilir. Verilen dize çift sayıda onaltılık bir rakam ise, Nmap bunları MAC olarak kullanır. 12 hex basamaktan daha az rakam sağlanmışsa, Nmap altı baytin kalanını rasgele değerlerle doldurur. Argüman sıfır veya hexadecimal dize değilse, Nmap verilen dizeyi içeren bir satıcı adı bulmak için nmap-mac-öneklerine bakılır(büyük/küçük harf duyarsızdır). Bir eşleşme bulunursa, Nmap satıcının OUI'sini kullanır ve kalan üç baytı rastgele doldurur. Geçerli **-spoof-mac** argüman örnekleri Apple, 0, 01:02:03:04:05:06, deadbeefcafe, 0020F2 ve Cisco'dur. Bu parametre, Nmap'ın aslında ethernet düzeyinde paketler göndermesini sağlamak için **-send-eth** anlamına gelir. Bu parametre, sürüm tespiti veya Nmap Scripting Engine gibi bağlantı yönelik özellikleri değil, yalnızca SYN taraması veya işletim sistemi algılama gibi ham paket taramalarını etkiler. MAC Address Spoofing ağ erişimi için gerekli olmasa bile, aldatma için kullanılabilir.

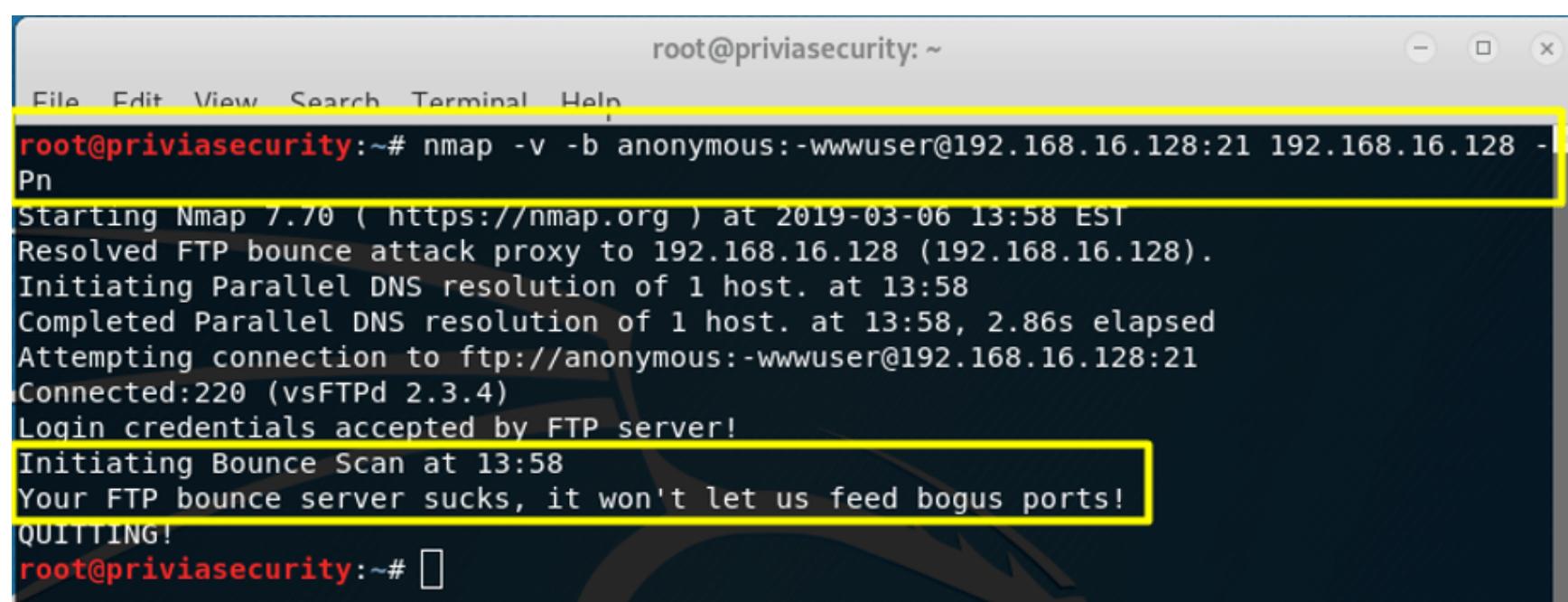
## Source Routing (Kaynak Yönlendirme)

Hedef ile aranızdaki bir router sorun yaşamana neden oluyorsa, çevresinde bir rota bulmaya çalışılır. Bu tekniğin etkinliği sınırlıdır. Çünkü paket filtreleme sorunları genellikle hedef ağ üzerinde veya yakınında meydana gelir. Bu makinelerin tüm kaynak yönlendirmeli paketlerini düşürmesi veya ağa girmesinin tek yoludur. Nmap, **-ip-options** parametresini kullanarak hem gevşek hem de katı kaynak yönlendirmesini desteklemektedir. Örneğin, **-ip-options** parametresi "L 192.168.0.7 192.168.30.9" olarak belirtilmesi, paketin, verilen iki IP yol noktasından bu serbest kaynağın yönlendirilmesini talep eder. Kesin kaynak yönlendirmesi için L yerine S belirtilir. Sıkı kaynak yönlendirmeyi seçerseniz, yol boyunca her bir sekmeyi belirtmek zorunda kalacaktır. IPv4 kaynak yönlendirmesi çok sık engellenirken, kaynak yönlendirmenin IPv6 biçimi çok daha yaygındır. Nmap ile bir hedef makineye yönlendirilmiş bir kaynak yolu keşfedilirse, exploitability port taramasıyla sınırlı değildir. Ncat, kaynak yönlendirilmiş yollar üzerinden TCP ve UDP iletişimini etkinleştirebilir (-g parametresi kullanılabilir).

## FTP Bounce Scan

FTP protokolünün bir özelliği (RFC 959) proxy FTP bağlantıları için destek sağlar. Bu, kullanıcının bir FTP sunucusuna bağlanması ve ardından dosyaların üçüncü taraf bir sunucuya gönderilmesini istemesini sağlar. Böyle bir özellik birçok düzeyde saldırganlar tarafından kullanılmıştır. Bu nedenle çoğu sunucu bu özelliği destekler. Özelliğin diğer bir dezavantajı, FTP sunucusunun diğer hostları taramasına izin vermesidir. FTP sunucusundan, sırayla bir hostun portuna bir dosya göndermesini istemektedir. Hata mesajı portun açık olup olmadığını açıklamaktadır. Bu, güvenlik duvarlarını atlamanın iyi bir yoldur. Çünkü kurumsal FTP sunucuları, genellikle diğer tüm ana bilgisayarlarla, eski herhangi bir Internet ana bilgisayardan daha fazla erişebilecekleri bir yere yerleştirilir. Nmap, **-b** seçeneğiyle birlikte FTP bounce taramasını deskeklere. "**kullanıcıadı:parola@ftpserver:port**" biçiminin bir argümanını alır. Server, güvenlik açığı bulunan bir FTP sunucusunun adı veya IP adresidir. Normal bir URL'de olduğu gibi, "**kullanıcıadı:parola**" ögesi atlanılabilir; bu durumda adsız oturum açma kimlik bilgileri (**kullanıcı: anonymous parola: -wwwuser@**) kullanılır. Port numarası ihmal edilebilir, bu durumda **<server>** üzerindeki varsayılan FTP port (21) kullanılır. Bir güvenlik duvarını atlamanızı, hedef ağ bağlantı noktası 21 için tarayın (veya sürüm algılamasıyla tüm bağlantı noktalarını tararsanız herhangi bir FTP hizmeti için bile) ve **ftp-bounce** NSE komut dosyası kullanılabilir. Nmap hedefin savunmasız olup olmadığını söyler. İzlerinizi sadece kapatmaya çalışıyorsanız, kendinizi hedef ağdaki ana bilgisayarlarla sınırlandırmanız gerekmekz. Güvenlik açığı bulunan FTP sunucuları için rasgele internet adresleri taramaya gitmeden önce, sysadmins'in sunucularını bu şekilde kötüye kullanmanızı yardımcı olmaz.

Şekil 8.2.10 – TCP FTP Bounce Scan Gösterimi

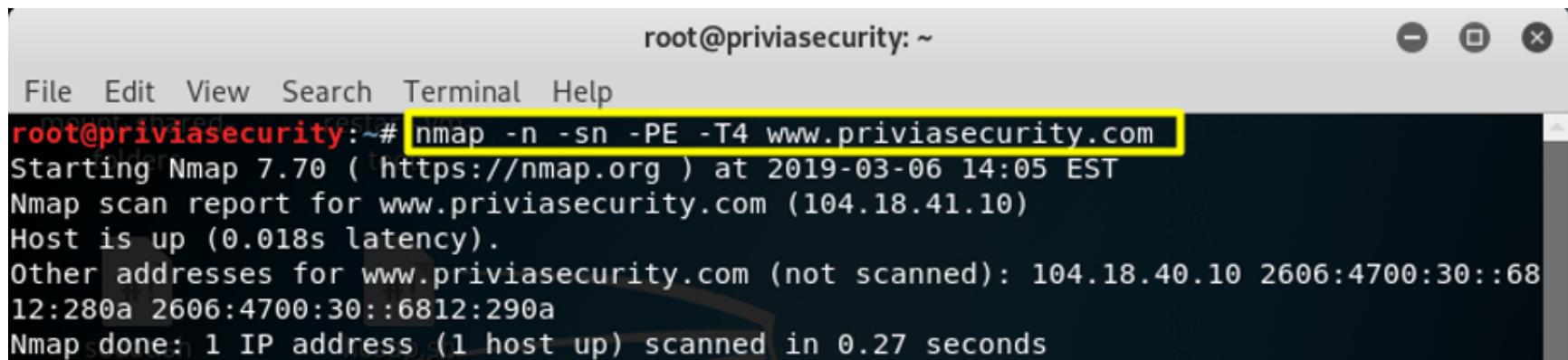


```
root@priviasecurity:~# nmap -v -b anonymous:-wwwuser@192.168.16.128:21 192.168.16.128 -Pn
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 13:58 EST
Resolved FTP bounce attack proxy to 192.168.16.128 (192.168.16.128).
Initiating Parallel DNS resolution of 1 host. at 13:58
Completed Parallel DNS resolution of 1 host. at 13:58, 2.86s elapsed
Attempting connection to ftp://anonymous:-wwwuser@192.168.16.128:21
Connected:220 (vsFTPd 2.3.4)
Login credentials accepted by FTP server!
Initiating Bounce Scan at 13:58
Your FTP bounce server sucks, it won't let us feed bogus ports!
QUITTING!
root@priviasecurity:~#
```

Şekil 8.2.10 da gösterildiği gibi tarama işlemi gerçekleştirildi. Birazda FTP sunucusuna yönelik bir saldırı olarak da gösterilmektedir. Böyle bir girişimde yukarıdaki gibi kullanıcı adı parola FTP sunucusu adresi sonra hedefin adresi ve pingsiz tarama için -Pn parametresini kullanmıştır.

## Güvenlik Duvarını Atlama Örneği

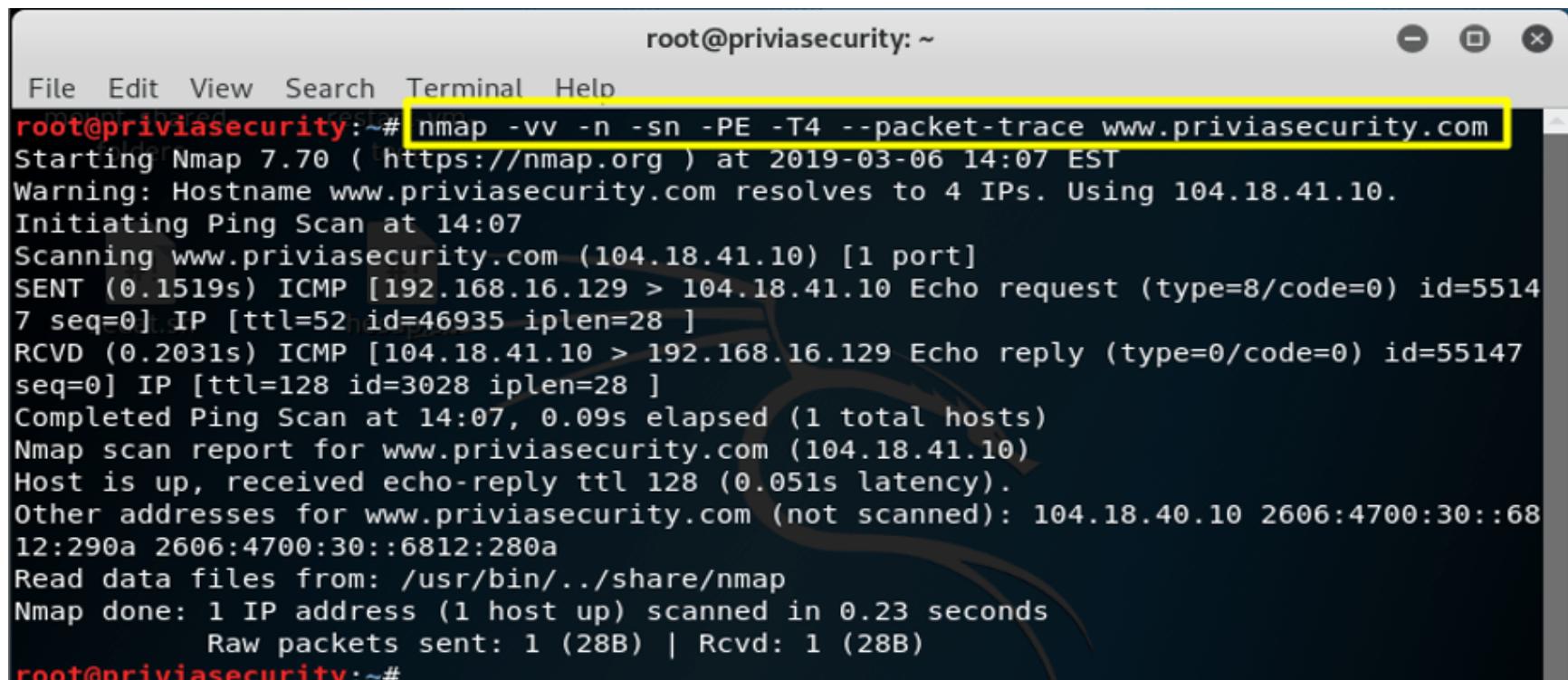
Şekil 8.2.11.a – Genel bir ağ taraması



```
root@priviasecurity:~# nmap -n -sn -PE -T4 www.priviasecurity.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 14:05 EST
Nmap scan report for www.priviasecurity.com (104.18.41.10)
Host is up (0.018s latency).
Other addresses for www.priviasecurity.com (not scanned): 104.18.40.10 2606:4700:30::68
12:280a 2606:4700:30::6812:290a
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

Şekil 8.2.11.a'da bir ağ taraması gerçekleştiriliyor. Bu taramada –n parametresi ile dns çözümlemesini yapmamasını sağlanır. –sn parametresini kullanarak port tarama işlemini devre dışı bırakılır. –PE parametresini kullanarak bir ICMP echo taraması gerçekleştirilmesi ve –T4 parametresini kullanarak zamanlamasını ve performansını ayarlama işlemi gerçekleştiriliyor. Çünkü varsayılan olarak buradaki amaç ağdaki sistemlerin varlığını keşfetmektedir. Tabi, örnek olması amacıyla www.priviasecurity.com domain veriliyor. Ayrıca 192.168.16.0/24 veya 104.18.41.10 makinasının subnetinin hesaplandıktan sonra /24, /32 vb. opsiyonları kullanarak veriliip ağdaki sistemlerin keşfi sağlanabilir.

Şekil 8.2.11.b – Hedefin ek olarak –packet-trace parametresi ile taranması



```
root@priviasecurity:~# nmap -vv -n -sn -PE -T4 --packet-trace www.priviasecurity.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 14:07 EST
Warning: Hostname www.priviasecurity.com resolves to 4 IPs. Using 104.18.41.10.
Initiating Ping Scan at 14:07
Scanning www.priviasecurity.com (104.18.41.10) [1 port]
SENT (0.1519s) ICMP [192.168.16.129 > 104.18.41.10 Echo request (type=8/code=0) id=55147 seq=0] IP [ttl=52 id=46935 iplen=28 ]
RCVD (0.2031s) ICMP [104.18.41.10 > 192.168.16.129 Echo reply (type=0/code=0) id=55147 seq=0] IP [ttl=128 id=3028 iplen=28 ]
Completed Ping Scan at 14:07, 0.09s elapsed (1 total hosts)
Nmap scan report for www.priviasecurity.com (104.18.41.10)
Host is up, received echo-reply ttl 128 (0.051s latency).
Other addresses for www.priviasecurity.com (not scanned): 104.18.40.10 2606:4700:30::68
12:290a 2606:4700:30::6812:280a
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
      Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
root@priviasecurity:~#
```

Şekil 8.2.11.b'de ise gösterildiği –packet-trace parametresinin sağladığı avantajları kullanarak genel bir tarama gerçekleştiriliyor.

Şekil 8.2.11.c – Idle Scan İşleminin Kullanılması

```
root@priviasecurity:~  
File Edit View Search Terminal Help  
root@priviasecurity:~# nmap -vv -n -Pn -sI google.com:80 -p 80 www.priviasecurity.com  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 14:08 EST  
Warning: Hostname www.priviasecurity.com resolves to 4 IPs. Using 104.18.40.10.  
Initiating idle scan against www.priviasecurity.com (104.18.40.10) at 14:08  
Idle scan using zombie google.com (172.217.22.14:80); Class: Incremental  
WARNING: idle scan has erroneously detected phantom ports -- is the proxy google.com (172.217.22.14) really idle?  
Completed idle scan against www.priviasecurity.com (104.18.40.10) at 14:09, 6.90s elapsed (1 ports)  
Nmap scan report for www.priviasecurity.com (104.18.40.10)  
Host is up, received user-set (0.19s latency).  
Other addresses for www.priviasecurity.com (not scanned): 104.18.41.10 2606:4700:30::68  
12:290a 2606:4700:30::6812:280a  
Scanned at 2019-03-06 14:08:58 EST for 7s  
  
PORT      STATE      SERVICE      REASON  
80/tcp    closed|filtered  http      no-ipid-change  
  
Read data files from: /usr/bin/../share/nmap  
Nmap done: 1 IP address (1 host up) scanned in 7.16 seconds  
    Raw packets sent: 23 (1.012KB) | Rcvd: 17 (680B)  
root@priviasecurity:~#
```

Şekil 8.2.11.c'de ise idle scan işlemini kullanarak hedefin 80. portuna yönelik tarama gerçekleştiriliyor. Burada google.com adresinin bulunduğu makineyi zombie makina olarak belirleyip google.com makinesi üzerinden iletişime geçip tarama gerçekleştiriliyor.

Şekil 8.2.11.d – (ip-options) parametresinin kullanılması

```
root@priviasecurity:~  
File Edit View Search Terminal Help  
root@priviasecurity:~# nmap -n -sn -PE --ip-options=L www.priviasecurity.com --reason g  
oogle.com  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 14:11 EST  
Nmap scan report for www.priviasecurity.com (104.18.41.10)  
Host is up, received echo-reply ttl 128 (0.047s latency).  
Other addresses for www.priviasecurity.com (not scanned): 104.18.40.10 2606:4700:30::68  
12:280a 2606:4700:30::6812:290a  
Nmap scan report for google.com (172.217.21.238)  
Host is up, received echo-reply ttl 128 (0.069s latency).  
Other addresses for google.com (not scanned): 2a00:1450:4001:817::200e  
Nmap done: 2 IP addresses (2 hosts up) scanned in 0.91 seconds  
root@priviasecurity:~#
```

Şekil 8.2.11.d'de gösterildiği gibi –ip-options parametresini avantajlarını kullanarak hedef makineye yönelik tarama işlemleri gerçekleştiriliyor.

Şekil 8.2.11.e – (-Pn) parametresinin kullanılması ile tarama

```
root@priviasecurity:~  
File Edit View Search Terminal Help  
root@priviasecurity:~# nmap -n -sn -PE --ip-options=L www.priviasecurity.com --reason g  
oogle.com  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-06 14:11 EST  
Nmap scan report for www.priviasecurity.com (104.18.41.10)  
Host is up, received echo-reply ttl 128 (0.047s latency).  
Other addresses for www.priviasecurity.com (not scanned): 104.18.40.10 2606:4700:30::68  
12:280a 2606:4700:30::6812:290a  
Nmap scan report for google.com (172.217.21.238)  
Host is up, received echo-reply ttl 128 (0.069s latency).  
Other addresses for google.com (not scanned): 2a00:1450:4001:817::200e  
Nmap done: 2 IP addresses (2 hosts up) scanned in 0.91 seconds  
root@priviasecurity:~#
```