

NMAP BÖLÜM 11-15

Chapter 11. Defenses Against Nmap (Bölüm 11. Nmap'e Karşı Savunmalar)

- Introduction (Giriş)
- Scan Proactively, Then Close or Block Ports and Fix Vulnerabilities (Proaktif Olarak Tarayın, Ardından Bağlantı Noktalarını Kapatın veya Engelleyin ve Güvenlik Açıklarını Düzeltin)
- Block and Slow Nmap with Firewalls (Güvenlik Duvarları ile Nmap'i Engellemeye ve Yavaşlatma)
- Detect Nmap Scans (Nmap Taramalarını Algılama)
- Clever Trickery (Zeki Hileler)
 - Hiding Services on Obscure Ports (Belirsiz Bağlantı Noktalarındaki Hizmetleri Gizleme)
 - Port Knocking (Liman Vuruntusu)
 - Honeypots and Honeynets (Honeypotlar ve Honeynetler)
 - OS Spoofing (İşletim Sistemi Sahtekarlığı)
 - Tar Pits (katran çukurları)
 - Reactive Port Scan Detection (Reaktif Port Tarama Algılama)
 - Escalating Arms Race (Tırmanan Silahlanma Yarışı)

Introduction (Giriş)

Bölüm 10, Güvenlik Duvarlarını ve Saldırı Tespit Sistemlerini Tespit Etmek ve Yıkmak Nmap'in (diğer birkaç açık kaynak güvenlik aracıyla birlikte) güvenlik duvarlarını aşmak ve saldırı tespit sistemlerini alt etmek için kullanabileceğimiz sayısız yolu tartışırdı. Şimdi duruma čitin diğer tarafından bakıyoruz: Güvenlik duvarları ve IDS'ler gibi teknolojiler Nmap'e karşı nasıl savunma yapabilir. Olası savunmalar arasında problemlerin engellenmesi, döndürülen bilgilerin kısıtlanması, Nmap taramasının yavaşlatılması ve yaniltıcı bilgilerin döndürülmesi yer almaktadır. Bazı savunmaların tehlikeleri de ele alınmıştır. Ağınızı saldırganların ne olup bittiğini anlayamayacağı ölçüde gizlemek, yöneticileriniz de artık anlamıyorsa net bir kazanç değildir. Benzer şekilde, port tarayıcılarının kafasını karıştırmak ya da onları engellemek için kullanılan savunma yazılımları da daha ciddi güvenlik açıklarına yol açıyorsa faydalı değildir. Burada açıklanan tekniklerin çoğu, yalnızca Nmap ile üretilenlere değil, genel olarak aktif problere karşı koruma sağlar.

Scan Proactively, Then Close or Block Ports and Fix Vulnerabilities (Proaktif Olarak Tarayın, Ardından Bağlantı Noktalarını Kapatın veya Engelleyin ve Güvenlik Açıklarını Düzeltin)

Sıklıkla en iyi savunmanın iyi bir hücum olduğu söylenir. Saldırganlara karşı savunma yapmanın mükemmel bir yolu onlar gibi düşünmektir. Ağlarınızı düzenli olarak tarayın ve güvenlik açıkları için çıktıları dikkatle analiz edin. Unix'te crontab'ı veya Windows'ta Görev Zamanlayıcısı'nı, Ndiff veya nmap-report gibi bir sistemle birlikte kullanın ("MadHat Harikalar Diyarında" adlı bölümde bakın) ve değişiklikleri size bildirin.

Proaktif tarama, güvenlik açıklarını saldırganlardan önce bulma ve düzeltme fırsatı sağlar. Henüz bilmemiş olduğunuz güvenlik açıklarından yararlanılmasını önlemek için gereksiz yere kullanılan bağlantı noktalarını kapatmak ve engellemek de aynı derecede önemlidir. Proaktif tarama ayrıca saldırganların hangi bilgileri elde edebileceği konusunda daha bilinçli olmanızı sağlar. Sonuçları zayıflıklar açısından kendiniz incelediğinizde ve güvenlik duruşunuzdan emin olduğunuzda, port tarayıcılar çok daha az tehdit edici hale gelir. Port tarayıcılar konusunda en paranoid olan ve en çok savunma ve tespit yazılımı kullanan kişiler genellikle ağ

güvenliklerine en az güvenen kişilerdir. Kimseyi bu bölüm boyunca anlatılan teknikleri kullanmaktan caydırırmak istemiyorum, sadece öncelikle mevcut ağ risklerini ve güvenlik açılarını araştırmalarını ve düzeltmelerini öneriyorum. Bir açığı düzeltmek, onu gizlemeye çalışmaktan çok daha etkilidir. Bu yaklaşım aynı zamanda saldırganların güvenlik açılarını bulabileceğinden sürekli endişe etmekten daha az streslidir.

Proaktif tarama devreye girdiğinde, ilk adım bilinen tüm güvenlik açılarını düzeltmektir. Daha sonra, güvenlik duvari aracılığıyla dışarıdan veya dahili ağıda bulunan her açık bağlantı noktasını denetlemek gelir. Halkın erişmesi gerekmeyen hizmetler güvenlik duvarında engellenmelidir. Çalışanların bunlara ulaşması gerekiyorsa, belki de bunun yerine VPN kullanabilirler. Dahili hizmetler genellikle kullanılmadıkları zaman bile dinlenirler. Varsayılan olarak yüklenmiş veya etkinleştirilmiş olabilirler ya da geçmiş kullanım nedeniyle etkinleştirilmiş ve hiç devre dışı bırakılmamış olabilirler. Bu tür gereksiz hizmetler devre dışı bırakılmalıdır. Hizmette bir güvenlik açığı olduğunu bilmeseniz bile saldırganlar bunu fark edebilir. Gelecekte de hizmet için güvenlik hataları bulunabilir. Kapalı bir port açık bir porttan çok daha küçük bir risktir. Bilinen açıklar giderildikten, özel hizmetler güvenlik duvarı tarafından engellendikten ve gereksiz hizmetler devre dışı bırakıldıktan sonra, sıfırıncı gün istismarlarına, iç tehditlere ve güvenlik açığı analiz sisteminizin gözden kaçıldığı açıklara karşı koruma sağlamak için saldırı önleme sistemleri gibi daha fazla savunma teknolojisi gerekebilir.

Proaktif ağ taraması ve denetimi tek seferlik bir denetimden ziyade bir rutin haline gelmelidir. Herhangi bir karmaşık ağıda, ana bilgisayarlar ve hizmetler düzenli olarak eklenir ve değiştirilir. Ağın güvenli kalması için bunları takip etmeniz gereklidir.

Bazı kötü uygulanmış ve test edilmiş sistemlerin bağlantı noktasını taramalarına, işletim sistemi algılamasına veya sürüm algılamasına olumsuz tepki verebileceğini unutmayın. İnternet üzerinden tarama yaparken bu nadiren bir sorundur, çünkü tarandığında çöken makineler böyle düşmanca bir ortamda uzun süre dayanmazlar. Dahili makineler genellikle daha kırılgandır. Proaktif bir tarama programına başlarken, bunun onaylandığından ve etkilenen taraflara önceden iletildiğinden emin olun. Ağın nispeten küçük bir bölümüyle başlayın ve herhangi bir sorun olmadığından emin olun, ardından aşamalı olarak ilerleyin. Basit port taramasıyla başlayıp daha sonra istediğiniz şekilde işletim sistemi tespiti veya sürüm tespitine geçmek isteyebilirsiniz.

Block and Slow Nmap with Firewalls (Güvenlik Duvarları ile Nmap'i Engelleme ve Yavaşlatma)

Taramaya karşı en iyi savunma önlemlerinden biri iyi yapılandırılmış bir güvenlik duvarıdır. Daha sonra açıklanan bazı tekniklerin yaptığı gibi ağ yapılandırmasını basitçe gizlemek yerine, iyi yapılandırılmış güvenlik duvarları birçok saldırı yolunu etkili bir şekilde engelleyebilir.

İyi bir güvenlik duvari kitabı şu temel kuralı vurgular: varsayılan olarak reddet. Kötü niyetli olduğundan şüphelenilen trafiği engellemeye çalışmak yerine, önce her şeyi engelleyin, ardından gerekli trafiğe izin vermek için bunu özellikle geçersiz kılmak. Kötü niyetli bir şeyi engellemeyi gözden kaçırmak, aynı seye yanlışlıkla açıkça izin vermekten çok daha kolaydır. Ayrıca, kötü trafiği engellememek bir saldırının tarafından istismar edilene kadar fark edilmeyebilirken, meşru trafiğe izin vermemek genellikle etkilenen kullanıcılar tarafından hızlı bir şekilde keşfedilir. Ve düzeltilene kadar size hatırlatmaya devam edeceklerdir.

Yukarıdaki iki neden herkesi varsayılan olarak reddetmeye ikna etmek için yeterli olmalıdır, ancak başka faydaları da vardır. Bunlardan biri Nmap gibi araçların büyük ölçekli keşiflerini yavaşlatmaktadır. Bir Nmap TCP SYN taraması kapalı bir portla karşılaşlığında, hedef makine bir RST paketi geri gönderir ve bu portun durumu yalnızca bir gidiş-dönüş süresi içinde belirlenir. Bu, Kaliforniya'daki web sunucumdan Moskova'daki bir ISP'ye dünyanın öbür ucundan bile saniyenin çeyreğinden daha az bir süredir. Öte yandan, bir güvenlik duvari probu düşürerek portu filtrelerse, Nmap pes etmeden önce en kötü durum zaman aşımını beklemek zorundadır. Nmap daha sonra paketin bir güvenlik duvari kuralı yerine aşırı kapasite nedeniyle bir yönlendirici tarafından düşürülmesi ihtimaline karşı birkaç yeniden iletişim yapar. Büyük ölçekli taramalarda fark oldukça önemli olabilir. Örneğin, kablosuz ağındaki bir makineye karşı 1.000 portlu TCP SYN taraması (nmap -sS -T4 para) tüm portlar açık veya kapalı olduğunda yalnızca beş saniye sürüyor. Yaygın olarak kullanılan bir düzine kadar bağlantı noktasını filtrelemek tarama süresini 12 saniyeye çıkarıyor. Default-deny (beş açık bağlantı noktası hariç tüm bağlantı noktalarını filtreleme) seçeneğine geçmek tarama süresini neredeyse

üçe katlayarak 33 saniyeye çıkarır. 28 saniyelik bir fark kulağa anlamlı gelmeyebilir, ancak büyük ölçekli taramalar için fazladan günler ekleyebilir.

UDP protokolü kullanıldığında filtrelenmiş portlar saldırıcılar için daha da sinir bozucudur. Güvenlik duvarı söz konusu olmadığından, Nmap kapalı bir bağlantı noktasını araştırdığında neredeyse tüm sistemler ICMP bağlantı noktasına ulaşılamıyor yanıtını verir. Açık portlar genellikle hiç yanıt vermez. Bu nedenle, bir varsayılan güvenlik duvarı bir prob paketini düşürürse, Nmap portun açık mı yoksa filtrelenmiş mi olduğunu söyleyemez. Port asla yanıt vermeyeceğinden, yeniden iletişim burada yardımcı olmaz. Saldırganlar daha sonra UDP portlarını anlamlandırmak için Nmap sürüm tespiti ve SNMP topluluk dizesi kaba zorlama gibi daha yavaş ve çok daha dikkat çekici tekniklere başvurmalıdır.

Nmap'i gerçekten yavaşlatmak için, güvenlik duvarının bir ICMP hatası veya TCP RST ile yanıt vermek yerine paketleri düşürdüğünden emin olun. Aksi takdirde Nmap, portlar kapalımiş gibi hızlı ve doğru bir şekilde çalışacaktır, ancak yine de problemleri engellemenin avantajını elde edersiniz. Bu ayrıca bir örnek olarak, Linux iptables güvenlik duvarı DROP ve REJECT hedef eylemlerini sunar. Adlarından da anlaşılacağı gibi, DROP paketi engellemenin ötesinde bir şey yapmazken, REJECT bir hata mesajı gönderir. Birincisi keşifleri yavaşlatmak için daha iyidir ve genellikle tavsiye edilir, ancak REJECT, güvenlik duvarının belirli trafiği engellediğini açıkça ortaya koymak ağ sorunlarını teşhis etmeyi kolaylaştırabilir.

Güvenlik duvarlarının bir başka ilkesi de derinlemesine savunmadır. Bağlantı noktaları güvenlik duvarı tarafından engellenmiş olsa bile, yine de kapalı olduklarıdan (hiçbir uygulamanın dinlemediğinden) emin olun. Kararlı bir saldırının eninde sonunda güvenlik duvarını aşacağını varsayıñ. Bölüm 10, Güvenlik Duvarlarını ve Saldırı Tespit Sistemlerini Tespit Etme ve Yıkma'daki bir tekniği kullanarak geçseler bile, güçlü bir savunma sunmak için tek tek makineler kilitlenmelidir. Bu, herkesin zaman zaman yaptığı hataların kapsamını ve zararını azaltır. Saldırganların hem güvenlik duvarında hem de bireysel makinelerde zayıflıklar bulması gerekecektir. Bir port tarayıcısı hem kapalı hem de filtrelenmiş portlara karşı oldukça etkisizdir. Özel adres alanı kullanmak (ağ adresi çevirisi gibi) ve ek güvenlik duvarları daha da fazla koruma sağlar.

Detect Nmap Scans (Nmap Taramalarını Algılama)

Bazı insanlar port taramalarını tespit etmenin zaman kaybı olduğuna inanır. Bunlar o kadar yaygındır ki internete bağlı herhangi bir kuruluş düzenli olarak taranacaktır. Bunların çok azı hedefli saldıruları temsil eder. Birçoğu Windows'un bir açığını bulmak için durmaksızın çalışan internet solucanlarıdır. Bazı taramalar internet araştırma projelerinden, bazıları ise interneti keşfeden meraklı ya da sıkılmış kişilerden geliyor. Bu kitap için iyi örnekler ve deneyimsel veriler bulmak amacıyla on binlerce IP'yi taradım. Diğer taramalar aslında kötü niyetlidir. Script kiddie'ler düzenli olarak, günün istismarına açık sistemler için geniş bir alanı tararlar. Bu kişilerin kötü niyetleri olsa da, ağınızda savunmasız bir hizmet bulamadıklarında kendi başlarına hareket etmeleri muhtemeldir. En büyük tehdit, özellikle kuruluşunuzu hedef alan saldırganlardır, ancak bunlar tespit edilen taramaların o kadar küçük bir yüzdesini temsil eder ki ayırt edilmeleri son derece zordur. Bu yüzden pek çok yönetici port taramalarını kaydetme zahmetine bile girmez.

Düzenli yöneticiler farklı bir görüşe sahiptir. Port taramalarının genellikle saldıruların habercisi olduğunu ve yanıt verilmese bile en azından kaydedilmesi gerektiğini iddia ederler. İnternet bağlantı noktası tarama faaliyeti selini azaltmak için genellikle dahili ağlara algılama sistemleri yerleştirirler. Günlükler bazen trendler için analiz edilir ya da dünya çapında korelasyon ve analiz için Dshield gibi 3. taraflara gönderilir. Bazen yeterli bütçeleri haklı çıkarmak için saldıruları ölçen kapsamlı günlükler ve korkutucu grafikler yönetime sunulur.

Sistem günlükleri tek başına port taramalarını tespit etmek için nadiren yeterlidir. Genellikle yalnızca tam TCP bağlantıları kuran tarama türleri günlüğe kaydedilirken, varsayılan Nmap SYN taraması gizlice geçer. Tam TCP bağlantıları bile yalnızca belirli bir uygulama bunu açıkça yaparsa günlüğe kaydedilir. Bu tür hata mesajları, mevcut olduğunda, genellikle şifrelidir. Ancak, bir grup farklı hizmetin aynı anda hata mesajları vermesi, tarama faaliyetinin yaygın bir göstergesidir. Özellikle Nmap sürüm tespiti kullanan müdahaleci taramalar genellikle bu şekilde tespit edilebilir. Ancak sadece yöneticiler sistem günlüklerini düzenli olarak okurlarsa. Günlük mesajlarının büyük çoğunluğu sonsuza kadar okunmadan kalır. Logwatch ve Swatch gibi günlük izleme araçları kesinlikle yardımcı olabilir, ancak gerçek şu ki, sistem günlükleri Nmap etkinliğini tespit etmede yalnızca marjinal olarak etkilidir. Özel amaçlı port tarama dedektörleri Nmap aktivitesini tespit etmek için daha etkili bir yaklaşımındır. İki yaygın örnek PortSentry ve Scanlogd'dur. Scanlogd 1998'den

beri kullanılmaktadır ve güvenlik için dikkatlice tasarlanmıştır. Kullanım ömrü boyunca hiçbir güvenlik açığı rapor edilmemiştir. PortSentry benzer özelliklerin yanı sıra şüpheli tarayıcıların kaynak IP'sini engelleyen reaktif bir yetenek sunar. "Reaktif Port Tarama Tespiti" adlı bölümde gösterildiği gibi, bu reaktif tekniğin tehlikeli olabileceğini unutmayın.

"Saldırı Tespit Sistemlerinden Kaçınma" başlıklı bölümde tartışılan eşik tabanlı saldırılara maruz kalmalarına rağmen, bu port tarama tespit araçları oldukça iyi çalışır. Yine de port taramalarını takip edecek kadar önemseyen yönetici tipi, istismar girişimleri ve yüklü arka kapılar gibi daha ciddi saldırılar hakkında da bilgi sahibi olmak isteyecektir. Bu nedenle, çok çeşitli şüpheli davranışlar konusunda uyarı veren saldırı tespit sistemleri bu özel amaçlı araçlardan daha popülerdir.

Birçok satıcı artık saldırı tespit sistemleri satmaktadır, ancak Nmap kullanıcıları Snort adlı açık kaynaklı hafif bir IDS'ye yönelmektedir. Snort, 3.243 Nmap kullanıcısından oluşan bir anket grubu arasında en popüler üçüncü güvenlik aracı olarak yer almıştır (<https://sectools.org>). Nmap gibi Snort da küresel bir geliştirici topluluğu tarafından geliştirilmektedir. Port taramaları da dahil olmak üzere her türlü şüpheli etkinliği tespit etmek için iki binden fazla kuralı destekler.

Düzgün bir şekilde kurulmuş ve izlenen bir IDS muazzam bir güvenlik varlığı olabilir, ancak "Saldırı Tespit Sistemlerini Yıkmak" adlı bölümde tartışılan riskleri unutmayın. Snort'un ve ticari rakiplerinin birçoğunun uzaktan istismar edilebilen birçok güvenlik açığı vardır. Ayrıca, yetenekli bir saldırgan çoğu IDS kuralını yenebilir, bu nedenle gardınızı düşürmeyin. IDS'ler çoğu zaman yanlış bir güvenlik hissine yol açar.

Clever Trickery (Zeki Hileler)

Nmap, diğer aktif problama araçları gibi, bilgilerini hedef sistemlere paketler göndererek ve ardından yanıtları yorumlamaya ve yararlı raporlar halinde düzenlemeye çalışarak elde eder. Nmap, düpedüz düşmanca ortamlar olabilecek sistem ve ağlardan gelen bilgilere güvenmek zorundadır. Bazı yöneticiler taranmaktan rahatsız olur ve küçük bir kısmı da daha önce tartışılan güvenlik duvarı ve IDS tekniklerinin ötesinde aktif önlemlerle Nmap'i karıştırmaya veya yavaşlatmaya çalışır.

Bu aktif müdahale yöntemlerinin birçoğu oldukça zekice. Ben birçoğunun fazla akıllı olduğunu ve çözümlerinden daha fazla soruna neden olduklarını iddia ediyorum. Bu sorunlardan biri de istismar edilebilirliktir. Bu özel aktif yanıt yazılımlarının çoğu, dikkatli bir güvenlik değerlendirmesi yapılmadan yazılmış hızlı hack'lerden ibarettir. Örneğin, Paul adında bir yönetici arkadaşım FakeBO'yu makinesine yüklemekten oldukça gurur duyuyordu. Paul aslında sadece girişimlerini günlüğe kaydederken, komut dosyası çocukların Back Orifice virüsü bir makine bulduklarını düşünerek onları kandırma ihtimaline gülüyordu. Bir FakeBO arabellek taşıması keşfedildiğinde ve bir saldırgan bunu kutusunu tehlikeye atmak ve gerçek bir arka kapı kurmak için kullandığında Paul'e şaka yapıldı.

Bu teknolojilerde ortak olan diğer büyük risk ise başka bir yerde harcanması daha iyi olan zamanın yer değiştirmesidir. Saldırganların kafasını karıştırmak eğlenceli ve tatmin edici olabilir ve hatta bazı durumlarda saldırıcıları engeller. Ancak sonuçta bu teknikler çoğunlukla belirsizliğe dayalı güvenliktir. Yine de faydalı olsalar da, güvenlik duvarları ve güvenlik açığı yamaları gibi daha dayanıklı teknolojiler kadar önemli değildirler. Gelişmiş saldırganlar her halükarda şaşırtmacanın arkasını göreceklere ve script kiddie'ler ve solucanlar nadiren keşif yapmakla uğraşırlar. Apache web sunucuma karşı her gün yapılan IIS istismar girişimleri bunun kanıdır. Bu teknikler yalnızca güvenlik duruşunuzdan son derece emin olduğunuzda dikkate alınmalıdır. Çok fazla insan bunları ağlarını gerçekten güvence altına almanın yerine kullanmaktadır.

Hiding Services on Obscure Ports (Belirsiz Bağlantı Noktalarındaki Hizmetleri Gizleme)

Zaman zaman yöneticiler, saldırganların kendilerini bulmasını zorlaştırmak için hizmetleri alıgilmadık bağlantı noktalarında çalıştırmayı savunurlar. Özellikle, bazı yazılımların savunmasız bir sürümünü arayan saldırganların adres alanlarındaki tek portlu taramaların sıklığına dikkat çekiyorlar. Otonom solucanlar da sıklıkla aynı şeyi yapıyor.

Bu tür bir şaşırtmacanın bazı solucanların ve script kiddie'lerin hizmet bulmasını engelleyebileceği doğrudur, ancak bunlar güvenlik açıklarını hızla yamayan şirketler için nadiren marginal bir tehditten daha fazlasıdır. Ve hızlı bir şekilde yama yapmayan şirketler bu basit port şaşırtmacası tarafından kurtarılmayacaktır. Savunucular genellikle daha becerikli saldırganların bile buna kanacağını iddia ediyor. Hatta bazları güvenlik listelerine 65.536 TCP portunun tamamının

taranmasının mümkün olmadığını yazmıştır. Yanlılıyorlar. Saldırganlar tüm TCP portlarını tarayabilir ve taramaktadır. Buna ek olarak, Nmap sürüm tespiti gibi teknikler, olağanüstü bir bağlantı noktasında hangi hizmetin dinlendiğini belirlemeyi kolaylaştırır. Örnek 11.1 böyle bir taramayı göstermektedir. Sadece sekiz dakika sürmesi ve bunun başka bir eyaletteki yavaş bir konut aDSL hattından yapılması dikkat çekicidir. Daha hızlı bir makineden aynı tarama sadece üç dakika sürüyor. Varsayılan durum filtrelenmiş olsaydı, tarama daha yavaş olurdu, ancak mantıksız bir şekilde değil. Bir tarama 10 veya 20 dakika sürse bile, bir saldırının oturup izlemesi gerekmez. Bir şirkete yönelik hedefli bir saldırısı kolayca bir gecede bırakılabilir ve kitlesel saldırular en son veri dosyalarını periyodik olarak indirerek bir tarayıcıyı haftalarca çalışır durumda bırakabilir.

Örnek 11.1. Bir tüm-TCP bağlantı noktası sürüm taraması

```
# nmap -sSV -T4 -O -p0-65535 apollo.sco.com

Starting Nmap ( https://nmap.org )
Nmap scan report for apollo.sco.com (216.250.128.35)
Not shown: 65524 closed ports
PORT      STATE    SERVICE VERSION
0/tcp      filtered unknown
21/tcp     open     ftp      WU-FTPD 2.1WU(1)+SCO-2.6.1+-sec
22/tcp     open     ssh      SSH 1.2.22 (protocol 1.5)
199/tcp    open     smux?
457/tcp    open     http    NCSA httpd 1.3
615/tcp    open     http    NCSA httpd 1.5
1035/tcp   filtered unknown
1521/tcp   open     oracle   Oracle DB Listener 2.3.4.0.0 (for SCO System V/386)
13722/tcp  open     inetd   inetd exec err /usr/openv/netbackup/bin/bpjava-msvc
13782/tcp  open     inetd   inetd exec err /usr/openv/netbackup/bin/bpcd
13783/tcp  open     inetd   inetd exec err /usr/openv/bin/vopied
64206/tcp  open     unknown
Device type: general purpose
Running: SCO UnixWare
OS details: SCO UnixWare 7.0.0 or OpenServer 5.0.4-5.0.6

Nmap done: 1 IP address (1 host up) scanned in 501.90 seconds
```

Bu yaklaşımın en büyük dezavantajı, meşru kullanıcılar için büyük bir rahatsızlıktır. SMTP ve DNS gibi bazı hizmetler, pratik nedenlerden dolayı neredeyse her zaman iyi bilinen portlarında çalışmak zorundadır. HTTP ve SSH gibi daha kolay değiştirilebilen hizmetler için bile bunu yapmak, tüm kullanıcıların hizmete her bağlandıklarında 52,147 gibi alışılmadık bir bağlantı noktası numarasını hatırlamaları gereği anlamına gelir. Birkaç "gizli" hizmet olduğunda, hangisinin hangisi

olduğunu hatırlamak özellikle zordur. Her makinede farklı bağlantı noktaları kullanmak daha da kafa karıştırıcı hale gelir, ancak kuruluş genelinde alışılmadık bağlantı noktası eşlemelerini standartlaştırmak bu planın sözde faydasını azaltır. Saldırganlar SSH'nin her zaman 52.147'de olduğunu fark edebilirler. Sonuç olarak, hayal kırıklığına uğramış meşru kullanıcılar temel hizmetlerin nerede gizlendiğini bulmaya çalışıkça sunucularınıza karşı tüm port Nmap taramaları artabilir. Daha az bilgili kullanıcılar ise sizi telefon yağmuruna tutabilir.

Port Knocking (Liman Vuruntusu)

Port knocking adı verilen bir teknik, hizmetleri potansiyel saldırılardan gizlemenin bir yolu olarak son zamanlarda popüler hale gelmiştir. Yöntem, <http://www.portknocking.org/> adresinin ön sayfasında iyi bir şekilde açıklanmıştır:

- Port knocking, açık portu olmayan ağa bağlı bir bilgisayarla bağlantı kurma yöntemidir. Bir bağlantı kurulmadan önce, kapalı bağlantı noktalarına bir dizi bağlantı denemesi olan bir bağlantı noktası vuruş dizisi kullanılarak bağlantı noktaları açılır. Uzak bir ana bilgisayar, sunucunun güvenlik duvarı kurallarını bir veya daha fazla belirli portu açacak şekilde manipüle etmek için özgün bir knock dizisi oluşturur ve gönderir. Bu manipülasyonlara sunucuda çalışan ve otantik knock dizilerine dönüştürülebilecek bağlantı girişimleri için güvenlik duvarı günlük dosyasını izleyen bir port knock daemon aracılık eder. İstenen portlar açıldıktan sonra, uzak ana bilgisayar bir bağlantı kurabilir ve bir oturum başlatabilir. Portun kapanmasını tetiklemek için başka bir vuruş dizisi kullanılabilir.

Bu yöntem yepyeni değildir, ancak 2003 yılında Martin Krzywinski'nin port knocking ifadesini bulması, bir uygulama yazması, kapsamlı bir web sitesi oluşturulması ve Sys Admin ve Linux Journal dergileri için bu konuda makaleler yazmasıyla popülerlik kazanmıştır. Port knocking hizmetlere ikinci bir koruma katmanı ekler, ancak kimlik doğrulama genellikle SSH gibi birincil hizmetler tarafından sağlanandan daha zayıftır. Uygulamalar genellikle koklama ve tekrarlama saldırılara maruz kalır ve genellikle kaba kuvvet ve hizmet redi tehditlerinden de muzdariptir.

Bunun avantajı, daha önce açıklanan basit ve etkisiz port gizleme tekniğinden çok daha güçlü bir hizmet gizlemesidir. Port çalma yoluyla yetkin bir şekilde gizlenen bir portun Nmap tarafından gönderilenler gibi aktif probalar kullanılarak keşfedilmesi neredeyse imkansızdır. Öte yandan, izinsiz giriş tespit sistemleri ve pasif ağ

eşleyiciler gibi koklayıcı tabanlı sistemler bu şemayı önemsiz bir şekilde tespit eder.

Bağlantı noktası gizlemenin uygulanıp uygulanmayacağına karar vermek, önerilen uygulama için geçerli olan fayda ve maliyetlerin analizini gerektirir. Hizmet gizleme yalnızca küçük bir uygulama grubu için faydalıdır. Buradaki motivasyon, dünyanın her yerinden yetkili kullanıcıların bağlantılarına izin verirken saldırganların savunmasız hizmetlere bağlanması (ve bunlardan yararlanması) önlemektir. Yalnızca belirli IP adreslerinin bağlanması gerekiyorsa, bağlantıları bu belirli IP'lerle sınırlayan güvenlik duvarı kısıtlamaları genellikle daha iyi bir yaklaşımdır. İdeal bir dünyada, uygulamalar kimlik doğrulama işlemini güvenli bir şekilde kendileri gerçekleştirir ve istismarı önlemek için bunları gizlemeye gerek kalmazdı. Ne yazık ki, SSH gibi güvenlik bilincine sahip programlar bile uzaktan istismar edilebilen çok sayıda kimlik doğrulama öncesi hataya maruz kalmıştır. Bu hataların her halükarda mümkün olan en kısa sürede düzeltilmesi gereklidir, port çalma yeni bir hata ortaya çıkmadan önce ekstra bir zaman aralığı sağlayabilir. Sonuçta, bazı SSH açıkları resmi yamalar mevcut olmadan çok önce yeraltına yayılmıştır. Bir hata duyurulduğunda, en bilinçli yöneticinin bile hatayı öğrenmesi, düzeltmeyi test etmesi ve tüm savunmasız örnekleri bulup yamalaması için birkaç saat veya gün gerekebilir. Bir ev bilgisayarı sahibinin yanıt süresi daha da uzun olabilir. Sonuçta, bilgisayar kullanıcının büyük çoğunluğu Bugtraq'a abone değildir.

Hizmet gizlemeden yararlananlar sadece iyi adamlar değildir. Gri şapka ve düpedüz suç teşkil eden kullanıcılar için de en az o kadar popülerdir (daha fazla değilse bile). Birçok İSS, kullanıcıların web veya SSH hizmetleri gibi herhangi bir sunucu daemon'u çalıştırmasını kısıtlar. Müşteriler port knocking teknolojisini kullanarak kişisel SSH daemon'larını ya da web sunucularını gizleyebiliyorlardı (sadece çok sınırlı kullanım için, çünkü halk kolayca bağlanamıyordu). Benzer şekilde, arkadaşım Tom'un işvereni de sadece Windows VPN istemcisi kullanarak evden bağlantıya izin veriyordu. Tom buna, uygun problemleri aldıktan sonra iş sunucusundan evindeki Linux kutusuna ters bir SSH tüneli kurup bir port vurma sistemi (henüz böyle adlandırılmadan önce) kurarak yanıt verdi. Bu sayede evden iş ağına tam erişimle ve Windows kullanmanın zahmetine katlanmak zorunda kalmadan çalışabiliyordu. Hem İSS hem de işveren örneklerinde servis sağlayıcının bir sniffer ya da netflow kullanarak bu hileyi tespit edebileceğini tekrarlamakta fayda var. Daha da karanlık kullanımlara geçecek olursak, bilgisayar suçluları ele geçirdikleri sistemlerdeki arka kapıları gizlemek için bu gibi teknikleri sıkılıkla

kullanırlar. Script kiddies, bir sonraki Nmap taraması tarafından tespit edilmeye açık, yüksek bir portta dinleyen bariz bir SSH daemon veya hatta ham kök kabuğu bırakabilir. Daha temkinli saldırganlar, arka kapılarında ve rootkit'lerinde port çalma gibi gizleme teknikleri kullanırlar.

Bu sistem tarafından sağlanan hizmet gizleme değerli olsa da, birçok sınırlamayla birlikte gelir. Hiç kimse sadece web sitenizi ziyaret etmek için özel bir knock istemci kurmayacağından, genel kullanıma yönelik hizmetler uygun değildir. Ayrıca, erişim talimatlarının kamuya açıklanması sistemin birincil amacını ortadan kaldıracaktır. Herkese açık olmayan hizmetler genellikle port kapama ile korunmak yerine bir güvenlik duvarı tarafından engellenmelidir. Bir grup insanın erişime ihtiyacı olduğunda, VPN'ler şifreleme ve kullanıcı düzeyinde erişim kontrolü sundukları için genellikle daha iyi bir çözümüdür. VPN'ler ayrıca paketlerin düşürülebildiği, çoğaltılabildiği ve yeniden sıralanabildiği gerçek dünya ağlarını idare etmek üzere tasarlanmıştır. [Portknocking.Org](#) uygulamasını kullanan nispeten basit bir prob, hepsi hedefe sırayla ulaşması gereken 30'dan fazla port probu gerektirebilir. Bu kadar çok prob için özel bir istemciye ihtiyacınız olacaktır. Telnet veya web tarayıcısı kullanmak çok sıkıcıdır. Ayrıca, yoldaki tüm güvenlik duvarları bu olağanüstü bağlantı noktalarına bağlanmanızı izin vermelidir. Bu kısıtlamalar ve zorluklar göz önüne alındığında, bir VPN kullanmak da aynı derecede uygun olabilir.

Ek bir risk de port knocking uygulamalarının hala olgunlaşmamış olmasıdır. Martin Krzywinski tarafından yazılan en iyi bilinen uygulama, indirme sayfasında "bu bir prototiptir ve başlamak için en azını içerir. Bunu üretim ortamları için kullanmayın." Ayrıca, kendi ağınıza envanterini çıkarmak için proaktif tarama yapmanın bu gibi programlar yüklenliğinde daha zor olacağını unutmayın.

Bu uzun sınırlamalar listesinin sizi port çalmayı düşünmekten bile caydırmasına izin vermeyin. Özellikle gizli arka kapılar veya kişisel bir makinenin uzaktan yönetimi ile ilgili olanlar gibi belirli durumlar için uygun olabilir.

Honeypots and Honeynets (Honeypotlar ve HoneyNetler)

Saldırganların kafasını karıştırmak için giderek daha popüler hale gelen bir yöntem, bir ağa yem sistemleri yerleştirmek ve bunları saldırılara karşı izlemektir. Bunlar bal küpü olarak bilinir. Ben, araştırma amacıyla bunlardan oluşan ağlar kuran HoneyNet Projesi'nin bir üyesiyim. Birçok şirket bu sistemleri kurumsal güvenlik amacıyla kurmuştur, ancak bunu yapmak risklidir. Gereken kapsamlı izleme, bu sistemlerin

yüksek bakım gerektirmesine neden olur ve saldırganların makinelere girip ciddi suçlar işlemek için kullanma riski her zaman vardır. Bir sonraki bölümde açıklanan Honeyd veya hatta bir IDS gibi daha az bakım gerektiren çözümler daha uygun olabilir. Her durumda, bal küpleri basit Nmap taramalarından daha invaziv saldıruları yakalamak için tasarlanmıştır, bu nedenle daha fazla tartışılmamaktadır.

OS Spoofing (İşletim Sistemi Sahtekarlığı)

Nmap OS algılamasını kandırmak için özel olarak çeşitli programlar geliştirilmiştir. Nmap problemlerine özel yanıtları desteklemek için ana işletim sistemini manipüle ederler. Bu şekilde, bir Linux PC bir Apple LaserWriter yazıcıya ya da hatta bir web kamerasına benzetilebilir. 2000'de piyasaya sürülen IP Personality en popüler sistemlerden biridir. Bu saçmalıkları desteklemek için Linux Netfilter çerçevesini genişletir. Ne yazık ki, Nisan 2002'den beri güncellenmemiştir ve 2.4.18'den sonraki çekirdek sürümlerinde çalışmamaktadır.

Araç kullanılabilirliği tek başına işletim sistemi sahteciliğini iyi bir fikir haline getirmez. Çabayı bir şekilde gerekçelendirmek gereklidir. IP Personality SSS, "Buna neden ihtiyacınız olsun ki?" sorusunu "Bunu soruyorsanız, yok demektir" şeklinde yanıtlayarak geçiştirmektedir. Yine de, bazı insanlar bu araçları yazmayı ve kullanmayı yeterince değerli buluyor. Bunun bir nedeni, belirli işletim sistemi bilgilerinin saldırganların ağınızdaki güvenlik açıklarını çıkarmasını kolaylaştırması ve ayrıca ne tür bir istismarın çalıştırılacağına karar vermeye yardımcı olmasıdır. Elbette buradaki asıl sorun güvenlik açığının kendisidir ve düzeltilmesi gereklidir. Diğer insanlar bu tür bir aracı çalıştırırlar çünkü kullandıkları işletim sisteminden utanırlar veya gizlilik konusunda son derece bilinçlidirler. Eğer işletim sisteminiz bir şirketin IP ihlali iddiası ve kullanıcılarına karşı dava açması nedeniyle yasal gri bir alandaysa, işletim sistemi sahteciliği böyle bir davaya karşı koruma sağlayabilir.

Bir ana bilgisayar işletim sistemini bu şekilde maskelemenin ciddi bir sorunu, güvenlik ve işlevsellik sorunlarına neden olabilmesidir. Nmap, TCP başlangıç sıra numarası ve IP kimlik numarası tahmin edilebilirliği gibi birkaç önemli güvenlik özelliğini test eder. Yazıcı gibi farklı bir sistemi taklit etmek, bu sayı dizilerini zayıflatmayı gerektirebilir, böylece öngörülebilir ve tüm saldırılara karşı savunmasız hale gelirler. İşletim sisteminizin parmak izini taklit ederek elde edilen belirsizlik, değerli güvenlik mekanizmalarını feda etmeye değmez. Bu tür bir sahtekarlık işlevselligi de bozabilir. Birçok Nmap OS tespit testi, sisteme hangi TCP seçeneklerinin desteklendiğini sormayı içerir. Zaman damgaları ve pencere

ölçeklendirme gibi belirli seçenekleri desteklemiyormuş gibi davranışmak, bu seçeneklerin verimlilik avantajlarını ortadan kaldıracaktır. Kullanılamayan seçenekleri destekliyormuş gibi yapmak felaket olabilir.

Örnek 11.2'de Nmap, IP Personality tarafından bir Linux kutusunun gerçekten bir Sega Dreamcast oyun konsolu olduğuna inandırılmaktadır. David Barroso Berrueta tarafından yazılan Nmap OS-Fingerprinting'i yenmek için pratik bir yaklaşım başlıklı makaleden alınmıştır. Bu mükemmel makale çok daha fazla örneğin yanı sıra ayrıntılı yapılandırma talimatları da içermektedir. Ayrıca, "kod çok kararlı değil" gibi kullanışlı uyarılarla birlikte birçok benzer sistemi açıklamaktadır. Modülü yükledim ve birkaç dakika içinde Linux kutum dondu."

Örnek 11.2. IP Kişişi ile Nmap'i Aldatmak

```
# nmap -sS -O -oN nmap2.log 192.168.0.19
Nmap scan report for 192.168.0.19
(The 1597 ports scanned but not shown below are in state: closed)
Port      State    Service
22/tcp    open     ssh
25/tcp    open     smtp
80/tcp    open     http
143/tcp   open     imap
Remote operating system guess: Sega Dreamcast
Nmap finished: 1 IP address (1 host up) scanned in 5.886 seconds
```

İşletim sistemi sahtekarlığı için (diğer özelliklerin yanı sıra) daha yeni ve daha popüler bir program Honeyd'dir. Yazar Niels Provos tarafından aktif olarak sürdürülmektedir ve IP Personality'e göre birkaç önemli avantaj sunmaktadır. Bunlardan biri yapılandırmanın çok daha kolay olmasıdır. Yukarıda IP Personality kullanarak Dreamcast sahteciliği için neredeyse 100 yapılandırma satırı gerekliydi. Öte yandan Honeyd, Nmap işletim sistemi algılama veritabanını okur ve kullanıcının seçtiği herhangi bir işletim sistemini taklit eder. (Honeyd'in Nmap'in 2007'de kullanıldından kaldırılan 1. nesil işletim sistemi algılama veritabanını kullandığını unutmayın). Honeyd ayrıca emülsyon için sentetik ana bilgisayarlar oluşturarak işletim sistemi sahteciliğinin güvenlik ve işlevsellik sorunlarını da çözer. Honeyd'den bir kuruluştaki yüzlerce kullanılmayan IP adresini devralmasını isteyebilirsiniz. Bu IP'lere gönderilen problara yapılandırmasına göre yanıt verir. Bu, bir ana bilgisayarın kendi TCP yiğinini maskelemeye çalışmanın güvenlik ve

işlevsellik risklerini ortadan kaldırır. Bunun yerine bir grup sentetik ana bilgisayar oluşturuyorsunuz, bu nedenle bu, mevcut ana bilgisayarların işletim sistemini gizlemeye yardımcı olmuyor. Sentetik ana bilgisayarlar temel olarak saldırılardan izlenebilecek az bakım gerektiren bir bal ağını oluşturur. Çoğunlukla araştırma amaçlıdır, örneğin yeni solucanları tanımlamak ve spam gönderenlerin faaliyetlerini izlemek için dünya çapındaki Honeyd kurulum ağını kullanmak gibi.

Bu bölümdeki diğer tekniklerde olduğu gibi, işletim sistemi sahteciliğini yalnızca güvenlik durumunuzdan tamamen memnun olduğunuzda denemenizi öneririm. Tek bir işletim sistemini yanılmak, hatta yüzlerce sahte Honeyd örneği eklemek, savunmasız sistemleri yamalamanın yerini tutmaz. Birçok saldırıcı (ve özellikle solucanlar) istismar kodunu göndermeden önce işletim sistemi tespitiyle bile uğraşmaz.

Bu sistemlerin yetenekli saldırıcılar tarafından tespit edilmesinin kolay olduğunu da belirtmek gereklidir. İşletim sistemleri arasındaki tüm uygulama ve TCP yiğini farklılıklar göz önüne alındığında, ikna edici bir görünüm sunmak olağanüstü zordur. IMAP, SMTP ve SSH sunan Örnek 11.2, "IP Kişiliği ile Nmap'i Aldatmak" taki sistemin gerçekten kendi işletim sistemini çalıştırın bir Dreamcast olduğuna kimse inanmayacaktır. Buna ek olarak, 0.8'e kadar olan tüm sürümlerdeki bir hata, tek bir sonda paketiyle basit Honeyd tanımlamasına izin veriyordu. Honeyd'in henüz işleyemediği birçok TCP özelliği de vardır. Bunlar Honeyd'i tespit etmek için kullanılabilir, ancak Nmap bu işi otomatikleştirmez. Honeyd yaygınlaşırsa, algılama işlevi muhtemelen Nmap'e eklenecektir.

Honeyd gibi aldatma programları, Nmap kullanıcılarının Nmap sonuçlarını dikkatli bir şekilde yorumlamaları ve özellikle kontrol etmediğiniz ağları tararken tutarsızlıklara dikkat etmeleri için sadece bir nedendir.

Tar Pits (katran çukurları)

Saldırıcıları kandırmak yerine, bazı insanlar sadece onları yavaşlatmayı hedefler. Katran çukurları uzun zamandır Internet solucanlarını ve spam gönderenleri yavaşlatmak için kullanılan popüler yöntemlerdir. Bazı yöneticiler sıfır boyutlu alma pencereleri ya da veriyi yavaşça bayt bayt geri gönderme gibi TCP teknikleri kullanmaktadır. LaBrea bunun popüler bir uygulamasıdır. Diğerleri SMTP komutlarına yanıt vermeden önce uzun gecikmeler gibi uygulama düzeyinde teknikler kullanır. Bunlar çoğunlukla anti-spamcılar tarafından kullanılsa da, benzer teknikler Nmap taramalarını yavaşlatmak için de kullanılabilir. Örneğin, kapalı

portlar tarafından gönderilen RST paketlerinin oranını sınırlamak tarayıcıları önemli ölçüde yavaşlatabilir.

Reactive Port Scan Detection (Reaktif Port Tarama Algılama)

Daha önce Scanlogd gibi araçları kullanarak tarama tespitinden bahsetmiştik. Diğer araçlar bundan çok daha ileri gider ve taramalara gerçekten yanıt verir. Bazı kişiler tarama kaynağına karşı istismar veya hizmet reddi saldıruları başlatarak karşılık vermeyi önermektedir. Bu, birçok nedenden dolayı korkunç bir fikirdir. Birincisi, taramalar genellikle sahtedir. Kaynak adres doğruysa, saldırganın günah keçisi olarak kullandığı önceki bir kurban olabilir. Ya da tarama bir İnternet araştırma anketinin parçası olabilir veya meşru bir çalışan veya müşteriden gelebilir. Kaynak adres gerçek bir saldırgana ait bir bilgisayar olsa bile, karşılık vermek yol boyunca masum sistemleri ve yönlendiricileri bozabilir. Ayrıca yasa dışı da olabilir.

Geri saldırı fikri güvenlik camiasında yaygın olarak reddedilse de, güvenlik duvarı kurallarını saldırgan IP adresini engelleyecek şekilde ayarlayarak tespit edilen saldırılara yanıt verme konusunda çok daha fazla ilgi var. Buradaki fikir, taramayı gerçek bir saldırıyla takip etmelerini önlemektir. Bu yaklaşımda birkaç risk vardır. Birincisi elinizi göstermiş olursunuz. Saldırganlar için engellendikleri açık olacaktır ve çoğunun problamaya devam etmek için kullanabilecekleri çok sayıda başka IP adresi vardır. Böylece reaktif sisteminiz hakkında bilgi sahibi olacaklar ve kendi saldırılарını artırabileceklerdir. Daha önemli bir sorun ise taramaların çok kolay bir şekilde taklit edilebilmesidir. "Saldırı Tespit Sistemlerini Yanıltmak" adlı bölümde bunu yapmak için çeşitli yöntemler açıklanmaktadır. Bir saldırgan engelleme fark ettiğinde, büyük web siteleri ve DNS sunucuları gibi önemli sistemlerin taramalarını taklit edebilir. Daha sonra bu IP'leri engelleyen bir hedef ağ, kendisine yönelik bir hizmet reddi saldırısı gerçekleştirmiş olacaktır. Güvenlik duvarı bloklarını tam TCP bağlantısı başlatan taramalarla sınırlamak sahtecilik sorununu azaltır, ancak bu varsayılan Nmap SYN taramasını bile durduramaz.

Escalating Arms Race (Tırmanan Silahlanma Yarışı)

Bu kitabın birincil odak noktası açık kaynak araçları olsa da, bir dizi ticari satıcı Nmap'i aldatmaya çalışan ürünler sunmuştur. Cisco Security Agent buna bir örnektir. Değerlendirme kılavuzu Nmap'e karşı aşağıdaki korumaları talep etmektedir.

- Network Mapper (Nmap), bir dizi ağ probu göndererek bir ağda hangi cihazların bulunduğu ve hangi işletim sistemi ve hizmetleri çalıştırdıklarını tanımlar. Bir cihazın ağdaki varlığı ve çalıştığı portlar, Nmap probleme verdiği yanıtla duyurulur. Döndürülen hata mesajlarının modeli işletim sistemini tanımlar. Nmap şaşırtıcı derecede doğrudur. Bir saldırının veya soruşturmanın ilk aşamasında, hangi sistemlerin bir saldırganın istismarlarına yanıt verebileceğini belirlemek için sıkılıkla kullanılır.
- Cisco Security Agent korumalı sistemlere karşı Nmap taramasının beklenen sonucu: Nmap, varsayılan sunucu veya varsayılan masaüstü ilkelerini çalıştırın sistemlerin hedef işletim sistemini tanımlayamıyor. Nmap taramaları, güvenlik testleri zaman aşımına uğrarken askıda kalıyor gibi görünüyor. Cisco Security Agent tarafından korunmayan sistemlere karşı yapılan Nmap taramaları sonuçları çok hızlı bir şekilde bildiriyor

CSA'nın nasıl çalıştığını ve Nmap'in bunu otomatik olarak algılayıp ayarlayıp ayarlayamayacağını araştırıyorum. Tarama teknolojisi bir silahlanma yarışıdır. Açık kaynak ve ticari şirketler Nmap ve diğer araçları yavaşlatmak, engellemek veya aldatmak için tasarlanmış ürünler yaratmaya devam edecktir. Bu arada, Nmap sürekli olarak gelişmekte ve bu zorluklar karşısında esneklik geliştirmektedir.

Chapter 12. Zenmap GUI Users' Guide (Bölüm 12. Zenmap GUI Kullanıcı Kılavuzu)

İçindekiler

- Introduction (Giriş)
 - The Purpose of a Graphical Frontend for Nmap (Nmap için Grafiksel Önyüzün Amacı)
- Scanning (Tarama)
 - Profiles (Profiller)
 - Scan Aggregation (Tarama Birleştirme)
- Interpreting Scan Results (Tarama Sonuçlarını Yorumlama)

- Scan Results Tabs (Tarama Sonuçları Sekmeleri)
 - The "Nmap Output" tab ("Nmap Çıktısı" sekmesi)
 - The "Ports / Hosts" tab ("Bağlantı Noktaları / Ana Bilgisayarlar" sekmesi)
 - The "Topology" tab ("Topoloji" sekmesi)
 - The "Host Details" tab ("Ana Bilgisayar Ayrıntıları" sekmesi)
 - The "Scans" tab ("Taramalar" sekmesi)
- Sorting by Host (Ana Bilgisayara Göre Sıralama)
- Sorting by Service (Hizmete Göre Sıralama)
- Saving and Loading Scan Results (Tarama Sonuçlarını Kaydetme ve Yükleme)
 - The Recent Scans Database (Son Taramalar Veritabanı)
- Surfing the Network Topology (Ağ Topolojisinde Gezinme)
 - An Overview of the "Topology" Tab ("Topoloji" Sekmesine Genel Bir Bakış)
 - Legend (Gösterge)
 - Controls (Kontroller)
 - Action controls (Eylem kontrolleri)
 - Interpolation controls (Enterpolasyon kontrolleri)
 - Layout controls (Düzen kontrolleri)
 - View controls (Görünüm kontrolleri)
 - Fisheye controls (Balık gözü kontrolleri)
 - Keyboard Shortcuts (Klavye Kısayolları)
 - The Hosts Viewer (Ana Bilgisayarlar Görüntüleyici)
- The Profile Editor (Profil Düzenleyici)
 - Editing a Command (Komut düzenleme)
 - Script selection (Senaryo Seçimi)
 - Creating a New Profile (Yeni Profil Oluşturma)

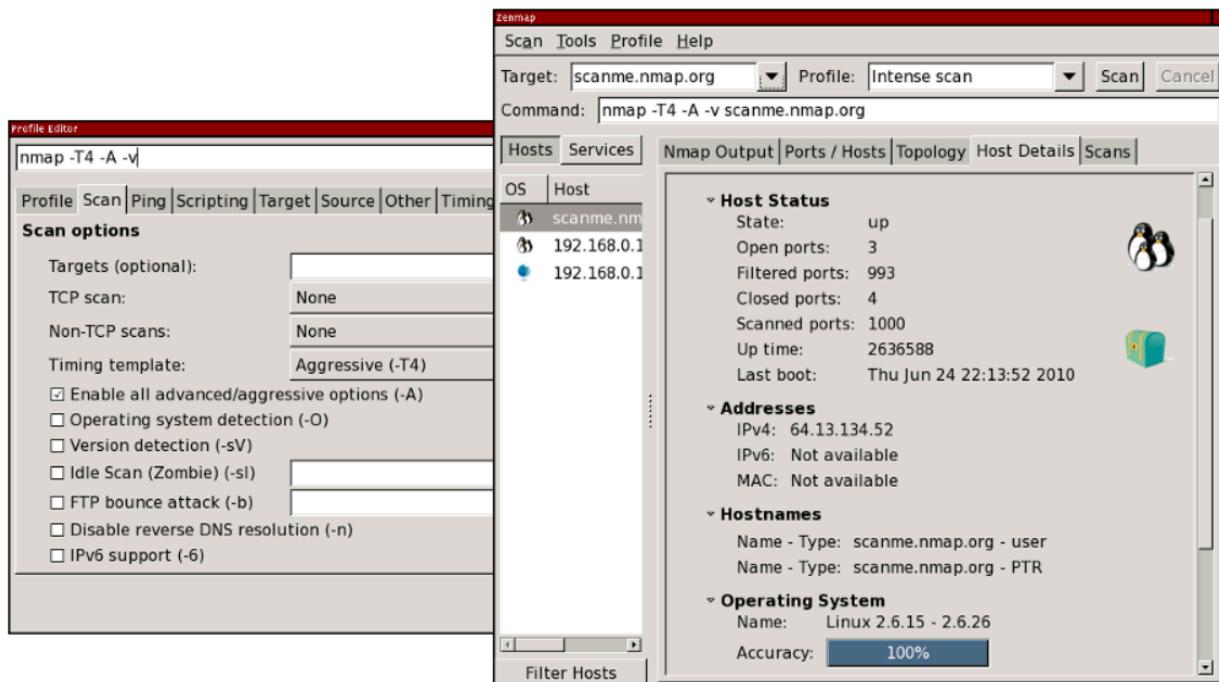
- Editing or Deleting a Profile (Profil Düzenleme veya Silme)
- Host Filtering (Ana Bilgisayar Filtreleme)
- Searching Saved Results (Kaydedilen Sonuçları Arama)
- Comparing Results (Sonuçları Karşılaştırma)
- Zenmap in Your Language (Kendi Dilinizde Zenmap)
 - Creating a new translation (Yeni bir çeviri oluşturma)
- Files Used by Zenmap (Zenmap Tarafından Kullanılan Dosyalar)
 - The `nmap` Executable (nmap Çalıştırılabilir)
 - System Configuration (Files Sistem Yapılandırma Dosyaları)
 - Per-user Configuration Files (Per-kullanıcı Yapılandırma Dosyaları)
 - Output Files (Çıktı Dosyaları)
- Description of `zenmap.conf` (zenmap'in açıklaması.conf)
 - Sections of `zenmap.conf` (zenmap.conf'un Bölümleri)
- Command-line Options (Komut Satırı Seçenekleri)
 - Synopsis (Özet)
 - Options Summary (Seçenekler Özeti)
 - Error Output (Hata Çıktı)
- History (Geçmiş)

Introduction (Giriş)

Zenmap, Nmap Güvenlik Tarayıcısı için resmi grafik kullanıcı arayüzüdür (GUI). Deneyimli Nmap kullanıcıları için gelişmiş özellikler sunarken yeni başlayanlar için Nmap'i kolaylaştırmak üzere tasarlanmış çok platformlu, ücretsiz ve açık kaynaklı bir uygulamadır. Sık kullanılan taramalar, tekrar tekrar çalıştırılmalarını kolaylaştırmak için profil olarak kaydedilebilir. Bir komut oluşturucu, Nmap komut satırlarının etkileşimli olarak oluşturulmasını sağlar. Tarama sonuçları kaydedilebilir

ve daha sonra görüntülenebilir. Kaydedilen taramalar birbirleriyle karşılaştırılarak ne kadar farklı oldukları görülebilir. Son taramaların sonuçları aranabilir bir veritabanında saklanır. Tipik bir Zenmap ekran görüntüsü Şekil 12.1'de gösterilmektedir. Daha fazla ekran görüntüsü için resmi Zenmap web sayfasına bakın.

Şekil 12.1. Tipik Zenmap ekran görüntüsü



Bu kılavuz, daha önce ikisini de kullanmamış olsanız bile, Nmap ve Zenmap'i birlikte kullanmayı kolaylaştırmak için hazırlanmıştır. Bu kılavuzun özellikle Nmap ile ilgili kısımları için (komut satırı seçenekleri vb.), Bölüm 15, Nmap Başvuru Kılavuzu'na bakın.

The Purpose of a Graphical Frontend for Nmap (Nmap için Grafiksel Önyüzün Amacı)

Hiçbir önyüz eski komut satırı Nmap'in yerini alamaz. Bir önyüzün doğası, işini yapmak için başka bir araca bağlı olmasıdır. Bu nedenle Zenmap'in amacı Nmap'in yerini almak değil, Nmap'i daha kullanışlı hale getirmektir. Zenmap'in düz Nmap'e göre sunduğu avantajlardan bazıları şunlardır.

İnteraktif ve grafiksel sonuç görüntüleme ⇒ Nmap'in normal çıktısını göstermenin yanı sıra, Zenmap ekranını bir ana bilgisayardaki tüm bağlantı noktalarını veya belirli bir hizmeti çalıştırın tüm ana bilgisayarları gösterecek şekilde düzenleyebilir. Tek bir ana bilgisayar veya tam bir tarama hakkındaki ayrıntıları uygun bir ekranda özetler. Zenmap keşfedilen ağların bir topoloji haritasını bile çizebilir. Birkaç taramanın sonuçları bir araya getirilebilir ve aynı anda görüntülenebilir.

Comparison ⇒ Zenmap iki tarama arasındaki farkları gösterme özelliğine sahiptir. Farklı günlerde çalıştırılan aynı tarama arasında, iki farklı ana bilgisayar taraması arasında, aynı ana bilgisayarların farklı seçeneklerle taraması arasında veya başka herhangi bir kombinasyonda nelerin değiştiğini görebilirsiniz. Bu, yöneticilerin ağlarında ortaya çıkan yeni ana bilgisayarları veya hizmetleri ya da mevcut olanların çöküşünü kolayca izlemelerine olanak tanır.

Kolaylık ⇒ Zenmap, siz onları atmayı seçene kadar tarama sonuçlarınızı takip eder. Bu, bir tarama çalıştırabileceğiniz, sonuçları görebileceğiniz ve ardından bunları bir dosyaya kaydedip kaydetmeyeceğinize karar verebileceğiniz anlamına gelir. Önceden bir dosya adı düşünmenize gerek yoktur.

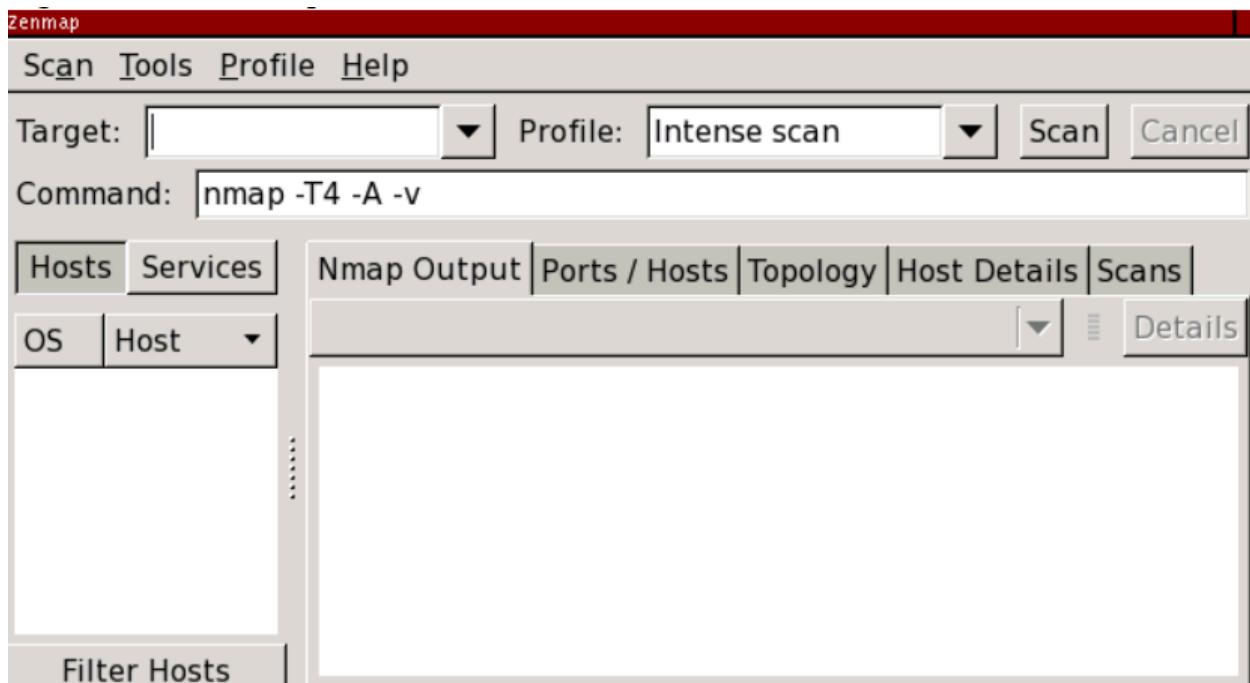
Tekrarlanabilirlik ⇒ Zenmap'in komut profilleri, aynı taramayı birden fazla kez çalıştırmayı kolaylaştırır. Ortak bir tarama yapmak için bir kabuk komut dosyası oluşturmaya gerek yoktur.

Keşfedilebilirlik ⇒ Nmap kelimenin tam anlamıyla yüzlerce seçeneğe sahiptir ve bu da yeni başlayanlar için göz korkutucu olabilir. Zenmap'in arayüzü, ister bir profilden gelsin ister bir menüden seçenekler seçilerek oluşturulmuş olsun, her zaman çalıştırılacak komutu gösterecek şekilde tasarlanmıştır. Bu, yeni başlayanların ne yaptıklarını öğrenmelerine ve anlamalarına yardımcı olur. Ayrıca uzmanların "Tara" düğmesine basmadan önce tam olarak neyi çalıştırılacağını iki kez kontrol etmelerine yardımcı olur.

Scanning (Tarama)

Zenmap'i bir terminalde zenmap yazarak veya masaüstü ortamında Zenmap simgesine tıklayarak başlatın. Şekil 12.2'de gösterildiği gibi ana pencere görüntülenir.

Şekil 12.2. Zenmap'in ana penceresi



Zenmap'in hedeflerinden biri de güvenlik taramasını yeni başlayanlar ve uzmanlar için kolay hale getirmektedir. Bir taramayı çalıştırırmak "Hedef" alanına hedefi yazmak, "Yoğun tarama" profilini seçmek ve "Tara" düğmesine tıklamak kadar basittir. Bu Şekil 12.3'te gösterilmektedir.

Şekil 12.3. Hedef ve profil seçimi



Bir tarama çalışırken (ve tamamlandıktan sonra), Nmap komutunun çıktısı ekranda gösterilir.

Boşluklarla ayrılmış herhangi bir sayıda hedef, hedef alanına girilebilir. Nmap tarafından desteklenen tüm hedef özellikleri Zenmap tarafından da desteklenir, bu nedenle 192.168.0.0/24 ve 10.0.0-5.* gibi hedefler çalışır. Zenmap en son taranan hedefleri hatırlar. Bir ana bilgisayarı yeniden taramak için, "Hedef" metin alanına bağlı birleşik giriş kutusundan ana bilgisayarı seçin.

Profiles (Profiller)

"Yoğun tarama" Zenmap ile birlikte gelen çeşitli tarama profillerinden sadece biridir. "Profil" açılan kutusundan seçerek bir profil seçin. Profiller birkaç yaygın tarama için mevcuttur. Bir profil seçildikten sonra onunla ilişkili Nmap komut satırı ekranda görüntülenir. Elbette bu profilleri düzenlemek ya da yenilerini oluşturmak mümkündür. Bu konu "Profil Düzenleyici" adlı bölümde ele alınmaktadır.

Bir Nmap komutu yazmak ve bir profil kullanmadan çalıştırılmasını sağlamak da mümkündür. Sadece komutu yazın ve return tuşuna basın ya da "Tara" ya tıklayın. Bunu yaptığınızda, taramanın herhangi bir profil kullanmadığını belirtmek için "Profil" girişi boş olur - doğrudan komut alanından gelir.

Scan Aggregation (Tarama Birleştirme)

Zenmap, tarama birleştirme olarak bilinen bir özellik olan birçok Nmap taramasının sonuçlarını tek bir görünümde birleştirme yeteneğine sahiptir. Bir tarama bittiğinde, aynı pencerede başka bir tarama başlatabilirsiniz. İkinci tarama tamamlandığında, sonuçları ilk taramanın sonuçları ile birleştirilir. Toplu bir görünüm oluşturan taramalar koleksiyonuna ağ envanteri denir.

Bir toplama örneği kavramı daha açık hale getirecektir. Scanme.nmap.org'a karşı hızlı bir tarama yapalım.

The screenshot shows the Zenmap application window. At the top, there's a menu bar with 'Scan', 'Tools', 'Profile', and 'Help'. Below the menu is a toolbar with 'Target' (set to 'scanme.nmap.org'), 'Profile' (set to 'Quick scan'), 'Scan' button, and 'Cancel' button. A command line field shows the command: 'nmap -T4 -F scanme.nmap.org'. The main area has several tabs: 'Hosts' (selected), 'Services', 'Nmap Output' (active), 'Ports / Hosts', 'Topology', 'Host Details', and 'Scans'. On the left, there's a sidebar with 'OS' and 'Host' sections, showing 'scanme.nmap.org' under the host section. At the bottom left is a 'Filter Hosts' button. The central part displays a table of network services:

	Port	Protocol	State	Service	Version
●	22	tcp	open	ssh	
●	25	tcp	closed	smtp	
●	53	tcp	open	domain	
●	80	tcp	open	http	

Şimdi aynı işlemi localhost'a karşı yapın:

Zenmap

Scan Tools Profile Help

Target: localhost Profile: Quick scan Scan Cancel

Command: nmap -T4 -F localhost

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans

OS	Host	Port	Protocol	State	Service	Version
	scanme.r	22	tcp	open	ssh	
	localhost	25	tcp	closed	smtp	
		53	tcp	open	domain	
		80	tcp	open	http	

Filter Hosts

The screenshot shows the Zenmap interface with a quick scan of the localhost target. The Nmap output table displays the following results:

Port	Protocol	State	Service	Version
22	tcp	open	ssh	
25	tcp	closed	smtp	
53	tcp	open	domain	
80	tcp	open	http	

Şimdi hem scanme hem de localhost için sonuçlar gösterilmektedir. Bu, tüm hedefleri önceden düşünmek zorunda kalmamak için kullanışlı olsa da, her iki hedefi de vererek tek bir Nmap taramasıyla yapabileceğiniz bir şeydir. Şimdi scanme hakkında daha fazla bilgi istediğimizi varsayıyalım, bu yüzden üzerinde yoğun bir tarama başlatalım.

Zenmap

Scan Tools Profile Help

Target: scanme.nmap.org Profile: Intense scan Scan Cancel

Command: nmap -T4 -A -v scanme.nmap.org

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans

OS	Host	Port	Protocol	State	Service	Version
	scanme.r	22	tcp	open	ssh	OpenSSH 4.3 (prc)
	localhost	25	tcp	closed	smtp	
		53	tcp	open	domain	

Filter Hosts

The screenshot shows the Zenmap interface with an intense scan of the scanme.nmap.org target. The Nmap output table displays the following results:

Port	Protocol	State	Service	Version
22	tcp	open	ssh	OpenSSH 4.3 (prc)
25	tcp	closed	smtp	
53	tcp	open	domain	

Şimdi scanme'de işletim sisteminin Linux olarak algılandığını gösteren küçük bir penguen simgesi var. Ayrıca hizmetlerinden biri de tanımlanmıştır. Şimdi tek bir Nmap taramasıyla yapamayacağınız bir şey yapıyoruz, çünkü bizim yaptığımız gibi daha yoğun tarama için bir ana bilgisayarı seçemezsiniz. Localhost için sonuçlar

hala mevcut, ancak daha derinlemesine bir tarama yapmaya karar vermediğimiz sürece daha önce bildiğimizden daha fazlasını bilmeyeceğiz.

Diğerine başlamadan önce bir taramanın bitmesini beklemek gereklidir. Birkaç tarama aynı anda çalışabilir. Her biri tamamlandığında sonuçları envanter eklenir. Herhangi bir sayıda tarama bir envanter oluşturabilir; taramaların toplanması "Taramalar" sekmesi bölümünde tam olarak açıklandığı gibi "Taramalar" tarama sonuçları sekmesinde yönetilir.

Aynı anda birden fazla envanterin açık olması mümkündür. Zenmap, bir pencerenin bir ağ envanterini temsil ettiği kuralını kullanır. Yeni bir envanter başlatmak için "Tara" menüsünden "Yeni Pencere"yi seçin veya **ctrl+N** klavye kısayolunu kullanın. "Tara" düğmesiyle bir tarama başlatmak, taramayı geçerli penceredeki envanter ekleyecektir. Farklı bir envanter koymak için ayrı bir pencere açın ve taramayı oradan çalıştırın. Tarama sonuçlarını bir dosya veya dizinden yüklemek, "Taramayı Bu Pencerede Aç" menü öğesini kullanmadığınız sürece yeni bir envanter başlatacaktır. Ağ envanterlerini ve tek tek taramaları kaydetme ve yükleme hakkında daha fazla bilgi için "Tarama Sonuçlarını Kaydetme ve Yükleme" başlıklı bölüme bakın.

Bir pencereyi kapatmak için "Tara" menüsünden "Pencereyi Kapat"ı seçin veya **ctrl+W** tuşlarına basın. Tüm açık pencereler kapatıldığında uygulama sonlanacaktır. Tüm açık pencereleri kapatmak için "Quit" seçeneğini seçin veya **ctrl+Q** tuşlarına basın.

Interpreting Scan Results (Tarama Sonuçlarını Yorumlama)

Nmap'in çıktısı bir tarama sırasında ve sonrasında görüntülenir. Bu çıktı Nmap kullanıcılarına tanıdık gelecektir. Zenmap'in renk vurgulaması dışında, bu, Nmap'i bir terminalde çalıştırıma göre herhangi bir görselleştirme avantajı sunmaz. Bununla birlikte, Zenmap'in arayüzünün diğer bölümleri terminal çıktısını, tarama sonuçlarının anlaşılması ve kullanılmasını kolaylaşacak şekilde yorumlar ve toplar.

Scan Results Tabs (Tarama Sonuçları Sekmeleri)

Her tarama penceresi, her biri tarama sonuçlarının farklı yönlerini gösteren beş sekme içerir. Bunlar "Nmap Çıktısı", "Bağlantı Noktaları / Ana Bilgisayarlar", "Topoloji", "Ana Bilgisayar Ayrıntıları" ve "Taramalar". Bunların her biri bu bölümde ele alınmaktadır.

The "Nmap Output" tab (The "Nmap Output" tab)

The screenshot shows the 'Nmap Output' tab selected in a software interface. The command entered is 'nmap -T4 -A -v scanme.nmap.org'. The output details the scan process, including OS detection, traceroute, DNS resolution, and NSE script execution. It identifies the host as up with 0.074s latency. A table lists open ports: 22/tcp (ssh, OpenSSH 4.3 protocol 2.0) and 25/tcp (closed, smtp). The interface includes tabs for Ports / Hosts, Topology, Host Details, Scans, and a 'Details' panel on the right.

```
nmap -T4 -A -v scanme.nmap.org
host)
Initiating OS detection (try #1) against scanme.nmap.org
(64.13.134.52)
Initiating Traceroute at 12:05
Completed Traceroute at 12:05, 0.29s elapsed
Initiating Parallel DNS resolution of 12 hosts. at 12:05
Completed Parallel DNS resolution of 12 hosts. at 12:05, 6.64s
elapsed
NSE: Script scanning 64.13.134.52.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 12:05
Completed NSE at 12:05, 4.17s elapsed
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.074s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE VERSION
22/tcp      open  ssh      OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey: 1024
|   60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)
|_ 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)
25/tcp      closed  smtp
```

"Nmap Çıktısı" sekmesi, bir tarama çalıştırıldığında varsayılan olarak görüntülenir. Bilinen Nmap terminal çıktısını gösterir. Ekran, çıktının bazı kısımlarını anlamlarına göre vurgular; örneğin, açık ve kapalı bağlantı noktaları farklı renklerde görüntülenir. Özel vurgular zenmap.conf dosyasında yapılandırılabilir ("zenmap.conf'un açıklaması" başlıklı bölüme bakın).

Birden fazla taramanın sonuçlarının bir pencerede gösterilebileceğini hatırlayın ("Tarama Birleştirme" adlı bölüme bakın). Sekmenin üst kısmındaki açılır kutu, görüntülenecek taramayı seçmenize olanak tanır. "Ayrıntılar" düğmesi, tarama

hakkında zaman damgaları, komut satırı seçenekleri ve kullanılan Nmap sürüm numarası gibi çeşitli bilgileri gösteren bir pencere açar.

The "Ports / Hosts" tab ("Bağlantı Noktaları / Ana Bilgisayarlar" sekmesi)

	Port	Protocol	State	Service	Version
●	22	tcp	open	ssh	OpenSSH 4.3 (protocol 2.0)
●	25	tcp	closed	smtp	
●	53	tcp	open	domain	
●	70	tcp	closed	gopher	
●	80	tcp	open	http	Apache httpd 2.2.3 ((CentOS))
●	113	tcp	closed	auth	

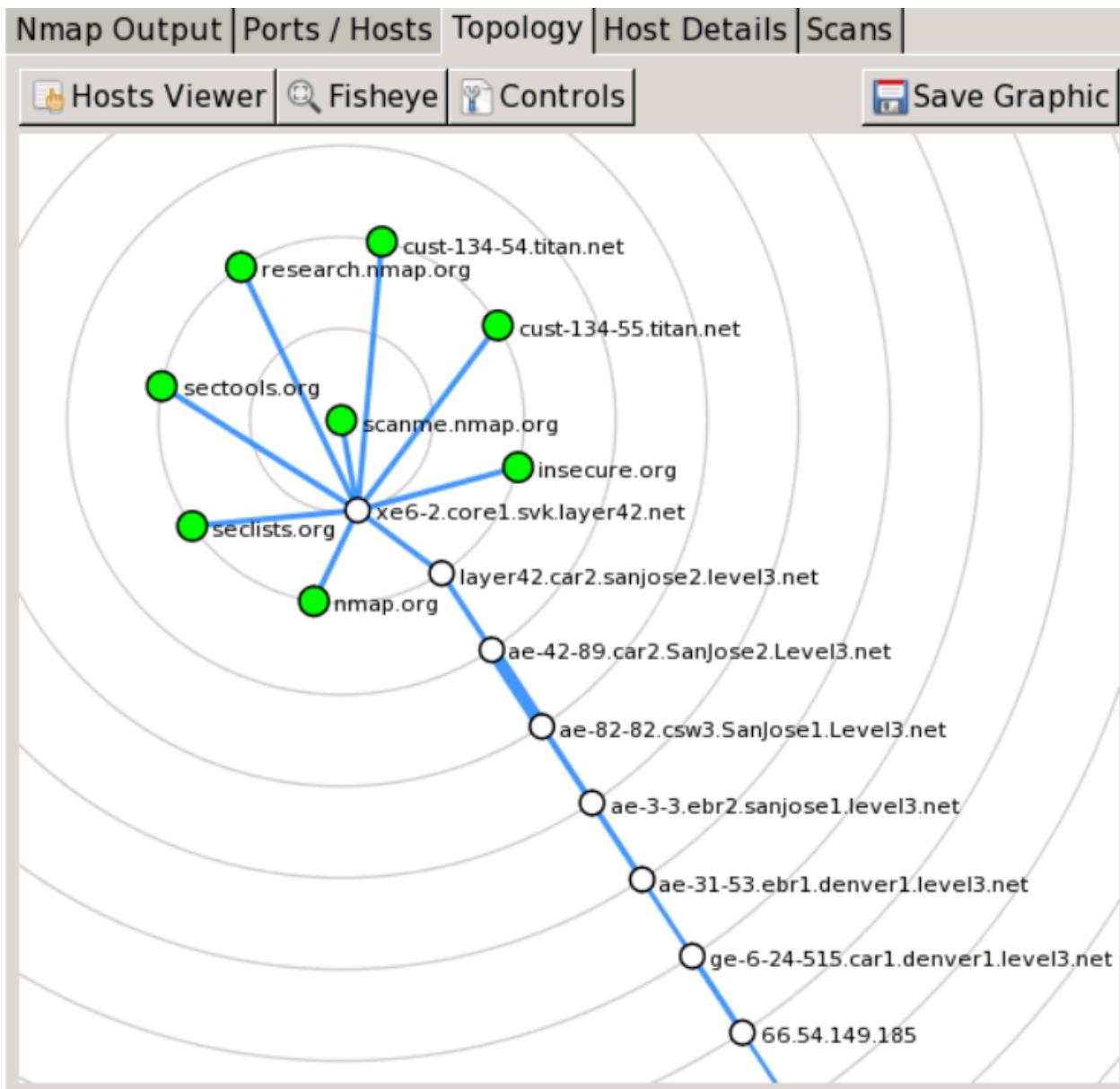
"Bağlantı Noktaları / Ana Bilgisayarlar" sekmesinin görüntüsü, bir ana bilgisayarın mı yoksa bir hizmetin mi seçili olduğuna bağlı olarak farklılık gösterir. Bir ana bilgisayar seçildiğinde, bu ana bilgisayardaki tüm ilginç bağlantı noktalarını, mevcut olduğunda sürüm bilgileriyle birlikte gösterir. Ana bilgisayar seçimi "Ana Bilgisayara Göre Sıralama" adlı bölümde daha ayrıntılı olarak açıklanmaktadır.

	Hostname	Port	Protocol	State	Version
●	scanme.nmap.org (64.13.134.52)	80	tcp	open	Apache http
●	192.168.0.1	443	tcp	open	ActionTec DS
●	192.168.0.1	80	tcp	open	ActionTec DS

Bir hizmet seçildiğinde, "Bağlantı Noktaları / Ana Bilgisayarlar" sekmesi o bağlantı noktasının açık veya filtrelenmiş olduğu tüm ana bilgisayarıları gösterir. Bu, "Hangi bilgisayarlar HTTP çalıştırıyor?" sorusuna hızlı bir şekilde cevap vermenin iyi bir

yoludur. Hizmet seçimi "Hizmete Göre Sıralama" adlı bölümde daha ayrıntılı olarak açıklanmaktadır.

The "Topology" tab ("Topoloji" sekmesi)



"Topoloji" sekmesi, bir ağdaki ana bilgisayarlar arasındaki bağlantıların etkileşimli bir görünümüdür. Ana bilgisayarlar eşmerkezli halkalar halinde düzenlenmiştir. Her halka, merkez düğümden ek bir ağ atlamasını temsil eder. Bir düğüme tıklamak onu merkeze getirir. Ana bilgisayarlar arasındaki ağ yollarının bir temsilini gösterdiğinden, "Topoloji" sekmesi --traceroute seçeneğinin kullanımından

yararlanır. Topoloji görünümü "Ağ Topolojisinde Gezinme" başlıklı bölümde daha ayrıntılı olarak ele alınmaktadır.

The "Host Details" tab ("Ana Bilgisayar Ayrıntıları" sekmesi)

The screenshot shows the Nmap interface with the 'Host Details' tab selected. The main content area displays details for the host 'scanme.nmap.org (64.13.134.52)'. The 'Host Status' section shows the following information:

State:	up
Open ports:	3
Filtered ports:	993
Closed ports:	4
Scanned ports:	1000
Up time:	1961342
Last boot:	Thu Jun 24 19:16:43 2010

Two small icons are displayed next to the status information: a penguin icon and a briefcase icon.

The 'Addresses' section lists:

- IPv4: 64.13.134.52
- IPv6: Not available
- MAC: Not available

The 'Hostnames' section shows:

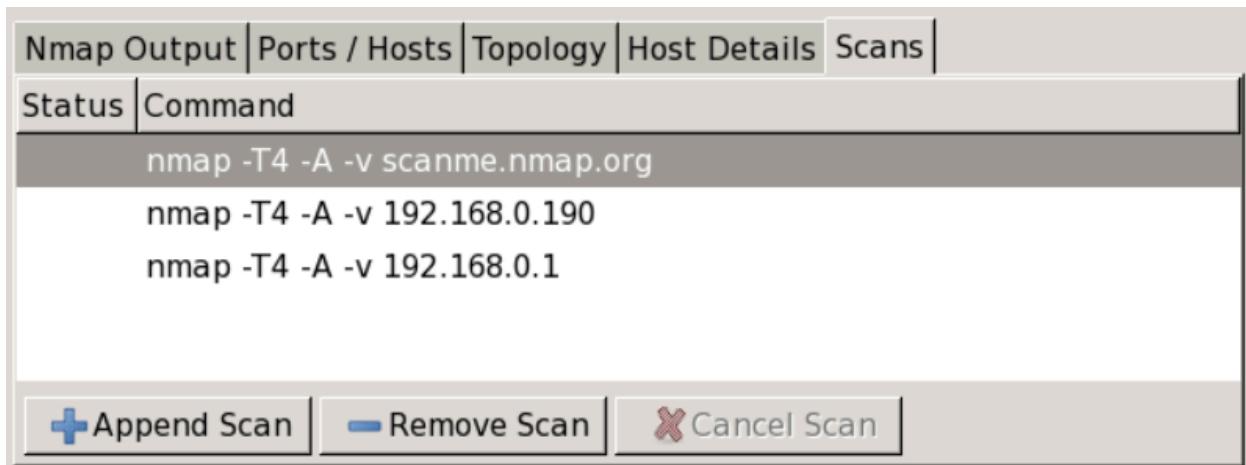
Name - Type: scanme.nmap.org - user

"Ana Bilgisayar Ayrıntıları" sekmesi, tek bir ana bilgisayar hakkında tüm bilgileri hiyerarşik bir ekrana ayırır. Ana bilgisayarın adları ve adresleri, durumu (yukarı veya aşağı) ve taranan bağlantı noktalarının sayısı ve durumu gösterilir. Ana bilgisayarın çalışma süresi, işletim sistemi, işletim sistemi simgesi (bkz. Şekil 12.5, "İşletim sistemi simgeleri") ve diğer ilişkili ayrıntılar mevcut olduğunda gösterilir. Tam işletim sistemi eşleşmesi bulunamadığında, en yakın eşleşmeler görüntülenir. Ayrıca, tarama bir dosyaya kaydedildiğinde kaydedilecek olan ana bilgisayarla ilgili bir yorumu saklamak için katlanabilir bir metin alanı da vardır ("Tarama Sonuçlarını Kaydetme ve Yükleme" başlıklı bölüme bakın).

Her ana bilgisayar, yalnızca açık bağlantı noktalarının sayısına dayanan çok kaba bir "güvenlik açığı" tahmini sağlayan bir simgeye sahiptir. Simgeler ve karşılık geldikleri açık port sayıları şunlardır

-  0-2 open ports,
-  3-4 open ports,
-  5-6 open ports,
-  7-8 open ports, and
-  9 or more open ports.

The "Scans" tab ("Taramalar" sekmesi)



The screenshot shows the Nmap interface with the "Scans" tab selected. The top navigation bar includes tabs for "Nmap Output", "Ports / Hosts", "Topology", "Host Details", and "Scans". Below the tabs, there are two sub-tabs: "Status" and "Command". Under the "Command" sub-tab, three scan commands are listed:

- nmap -T4 -A -v scanme.nmap.org
- nmap -T4 -A -v 192.168.0.190
- nmap -T4 -A -v 192.168.0.1

At the bottom of the window, there are three buttons: "Append Scan" (with a plus sign icon), "Remove Scan" (with a minus sign icon), and "Cancel Scan" (with a red X icon).

"Taramalar" sekmesi, ağ envanterini oluşturmak için bir araya getirilen tüm taramaları gösterir. Bu sekmeden taramalar ekleyebilir (bir dosya veya dizinden) ve taramaları kaldırabilirsiniz.

Bir tarama yürütülürken ve henüz tamamlanmamışken, durumu "Çalışıyor" dur. "Taramayı İptal Et" düğmesine tıklayarak çalışan bir taramayı iptal edebilirsiniz.

Sorting by Host (Ana Bilgisayara Göre Sıralama)

Şekil 12.4. Ana bilgisayar seçimi

Hosts		Services
OS	Host	
Windows	scanme.nmap.org	
Windows	192.168.0.1	
Ubuntu	192.168.0.190	

Zenmap'in ana penceresinin sol tarafında "Ana Bilgisayarlar" ve "Hizmetler" etiketli iki düğmenin bulunduğu bir sütun vardır. "Ana Bilgisayarlar" düğmesine tıklandığında, Şekil 12.4'te olduğu gibi taranan tüm ana bilgisayarların bir listesi görüntülenir. Genellikle bu liste sadece tek bir ana bilgisayar içerir, ancak büyük bir taramada binlerce ana bilgisayar içerebilir. Ana bilgisayar listesi, listenin üst kısmındaki başlıklara tıklanarak işletim sistemi veya ana bilgisayar adı/IP adresine göre sıralanabilir. Bir ana bilgisayar seçildiğinde "Portlar / Ana Bilgisayarlar" sekmesi o ana bilgisayardaki ilginç portları gösterecektir.

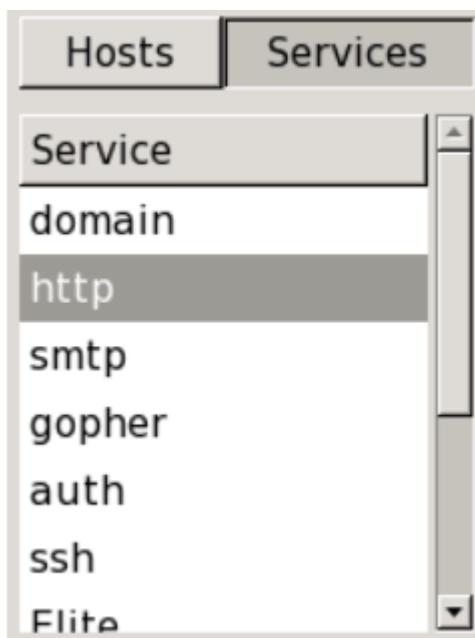
Her ana bilgisayar, ana bilgisayar adı veya IP adresi ile etiketlenir ve o ana bilgisayar için algılanan işletim sistemini gösteren bir simgeye sahiptir. Simge yalnızca işletim sistemi tespiti (-O) yapıldıysa anlamlıdır. Aksi takdirde, simge işletim sisteminin bilinmediğini gösteren varsayılan bir simge olacaktır. Şekil 12.5 olası tüm simgeleri göstermektedir. Nmap'in işletim sistemi algılamasının her zaman simgelerin imaj ettiği özgüllük düzeyini sağlayamayacağını unutmayın; örneğin bir Red Hat Linux ana bilgisayarı genellikle genel Linux simgesiyle görüntülenecektir.

Şekil 12.5. İşletim sistemi simgeleri



Sorting by Service (Hizmete Göre Sıralama)

Şekil 12.6. Hizmet seçimi



Taranan tüm ana bilgisayarları içeren aynı listenin üzerinde "Hizmetler" etiketli bir düğme bulunmaktadır. Buna tıklandığında, Şekil 12.6'da gösterildiği gibi, liste hedeflerden herhangi birinde açık, filtrelenmiş veya açık/filtrelenmiş tüm bağlantı noktalarının bir listesine dönüşecektir. (Nmap çıktısında açıkça listelenmeyen bağlantı noktaları dahil edilmemiştir.) Bağlantı noktaları hizmet adına göre tanımlanır (http, ftp, vb.). Liste, listenin başlığına tıklanarak sıralanabilir.

Bir ana bilgisayar seçildiğinde "Bağlantı Noktaları / Ana Bilgisayarlar" sekmesi söz konusu hizmetin açık veya filtrelenmiş olduğu tüm ana bilgisayarları görüntüleyecektir.

Saving and Loading Scan Results (Tarama Sonuçlarını Kaydetme ve Yükleme)

Tek bir taramayı bir dosyaya kaydetmek için "Tara" menüsünden "Taramayı Kaydet"'i seçin (veya **ctrl+S** klavye kısayolunu kullanın). Eğer envanterde birden fazla tarama varsa hangisini kaydetmek istediğiniz sorulacaktır. Kaydet iletişim kutusunda, "Nmap XML formatında" (.xml uzantılı) veya "Nmap metin formatında" (.nmap uzantılı) kaydetme seçeneğiniz vardır. XML biçimi Zenmap tarafından tekrar açılabilen tek biçimdir; metin biçiminde kaydederseniz dosyayı tekrar açamazsınız. Nmap çıktı formatları "XML Çıktısı (-oX)" adlı bölümde ele alınmıştır.

Bir envanterdeki her taramayı "Tara" menüsü altındaki "Tüm Taramaları Dizine Kaydet" (**ctrl+alt+S**) ile kaydedebilirsınız. Bir envanteri ilk kez kaydederken, kaydetme iletişim kutusundaki "Klasör Oluştur" düğmesini kullanarak genellikle yeni bir dizin oluşturacaksınız. Sonraki kaydetmelerde aynı dizine kaydetmeye devam edebilirsınız. İlgisiz tarama dosyalarının üzerine yazma olasılığını azaltmak için, seçilen dizin envantere ait olmayan bir dosya içeriyorsa, dizine kaydetme işlevi devam etmemeyi reddedecektir. Bu dizine kaydetmek istediğinizden eminseniz, rahatsız edici dosyaları silin ve ardından tekrar kaydedin.

Kaydedilen sonuçlar "Tara" menüsünden "Taramayı Aç" seçilerek veya **ctrl+O** klavye kısayolu yazılarak yüklenir. Dosya seçicide, "Aç" düğmesi tek bir taramayı açarken, "Dizini Aç" düğmesi seçilen dizindeki her dosyayı açar (belki de "Tüm Taramaları Dizine Kaydet" kullanılarak oluşturulmuştur).

"Taramayı Aç" yüklenmiş taramaları yeni bir pencerede açar ve böylece yeni bir envanter oluşturur. Bunun yerine yüklenen taramaları mevcut envanterle birleştirmek için "Taramayı Bu Pencerede Aç" seçeneğini kullanın.

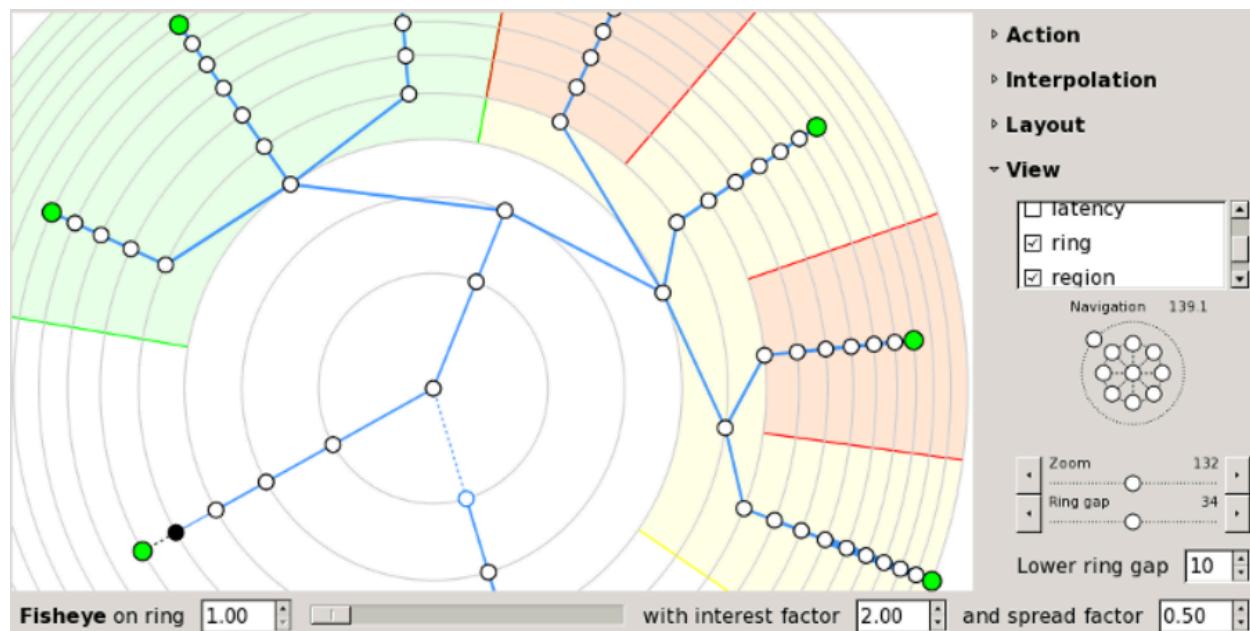
The Recent Scans Database (Son Taramalar Veritabanı)

Bir dosyaya kaydedilmeyen tarama sonuçları otomatik olarak bir veritabanında saklanır. Bir dosyadan yüklenen ve daha sonra değiştirilen (örneğin bir ana bilgisayar yorumu eklenerek) ancak yeniden kaydedilmeyen tarama sonuçları da veritabanında saklanır. Veritabanı zenmap.db adlı bir dosyada saklanır ve konumu platforma bağlıdır ("Zenmap Tarafından Kullanılan Dosyalar" adlı bölüme bakın). Varsayılan olarak, taramalar 60 gün boyunca veritabanında tutulur ve daha sonra kaldırılır. Bu zaman aralığı zenmap.conf dosyasının [search] bölümündeki

`save_time` değişkeninin değeri değiştirilerek değiştirilebilir ("zenmap.conf dosyasının açıklaması" başlıklı bölüme bakın).

Zenmap'in arama arayüzü, varsayılan olarak son taramalar veritabanının içeriğini aradığı için, bir veritabanı görüntüleyicisi olarak iki katına çıkar. Arama penceresi açıldığında veritabanındaki her tarama gösterilir. Tarama listesi daha sonra bir arama dizesi ilefiltrelenebilir. "Kaydedilen Sonuçların Aranması" başlıklı bölüme bakın.

Surfing the Network Topology (Ağ Topolojisinde Gezinme)



An Overview of the "Topology" Tab ("Topoloji" Sekmesine Genel Bir Bakış)

Zenmap'in "Topoloji" sekmesi, bir ağdaki ana bilgisayarlar arasındaki bağlantıların etkileşimli, animasyonlu bir görselleştirmesini sağlar. Ana bilgisayarlar, merkezden radyal olarak uzanan bir grafik üzerinde düğümler olarak gösterilir. Ekranı kaydirmak için tıklayın ve sürükleyn ve yakınlaşımak ve uzaklaşımak için sağlanan kontrolleri kullanın. Bir ana bilgisayara tıkladığınızda yeni merkez haline gelir. Grafik, ağır yeni görünümünü yansıtmak için yumuşak bir animasyonla

kendini yeniden düzenler. Yeni bir tarama çalıştırın ve her yeni ana bilgisayar ve ağ yolu topolojiye otomatik olarak eklenecektir.

Topoloji görünümü en çok Nmap'in --traceroute seçeneği ile birlikte kullanıldığından kullanışlıdır, çünkü bu seçenek bir ana bilgisayara giden ağ yolunu keşfeder. Topolojide traceroute bilgisi olmayan bir ağ envanterini görüntüleyebilirsiniz, ancak ağ yolları görünmeyecektir. Yine de, Zenmap'in tarama toplama özelliği sayesinde başka bir tarama çalıştırarak bir ağ envanterine traceroute bilgisi ekleyebileceğinizi unutmayın.

Başlangıçta topoloji, siz merkezde olacak şekilde localhost'un bakış açısından gösterilir. Bir ana bilgisayarı merkeze taşımak için üzerine tıklayın ve ağın onun bakış açısından nasıl göründüğünü görün.

Topoloji görünümü João Paulo S. Medeiros'un RadialNet programının bir uyarlamasıdır.

Legend (Gösterge)

Topoloji görünümünde birçok simbol ve renk kuralı kullanılır. Bu bölümde bunların ne anlama geldiği açıklanmaktadır.

- Aşağıda her normal ana bilgisayar küçük bir daire ile temsil edilir. Dairenin rengi ve boyutu ana bilgisayardaki açık port sayısına göre belirlenir. Ne kadar çok açık bağlantı noktası varsa, daire o kadar büyük olur. Beyaz daire, ağ yolunda port taraması yapılmamış bir ara ana bilgisayarı temsil eder. Bir ana bilgisayarın üçten az açık bağlantı noktası varsa yeşil; üç ila altı arasında açık bağlantı noktası varsa sarı; altıdan fazla açık bağlantı noktası varsa kırmızı olacaktır.
- Bir ana bilgisayar bir yönlendirici, anahtar veya kablosuz erişim noktasıysa, daire yerine kare ile çizilir.

Ağ mesafesi eşmerkezli gri halkalar olarak gösterilir. Her ek halka, merkez ana bilgisayardan bir ağ atlaması daha anlamına gelir.

Ana bilgisayarlar arasındaki bağlantılar renkli çizgilerle gösterilir. Birincil traceroute bağlantıları mavi çizgilerle gösterilir. Alternatif yollar (farklı bir yolun zaten mevcut olduğu iki ana bilgisayar arasındaki yollar) turuncu renkle çizilir. Hangi yolun birincil ve hangi yolların alternatif olduğu keyfidir ve yolların kaydedildiği sıraya göre kontrol edilir. Bir çizginin kalınlığı gidiş-dönüş süresiyle orantılıdır; daha yüksek RTT'ye sahip ana bilgisayarlar daha kalın bir çizgiye

sahiptir. Traceroute bilgisi olmayan ana bilgisayarlar localhost etrafında kümelenmiştir ve kesikli siyah bir çizgi ile bağlanmıştır.

Bir atlama için RTT yoksa (eksik bir traceroute girişi), bağlantı mavi kesikli bir çizgi ile gösterilir ve bağlantıyı yapan bilinmeyen ana bilgisayar mavi bir taslak ile gösterilir.

Bazı özel amaçlı ana bilgisayarlar, ne tür bir ana bilgisayar olduğunu açıklayan bir veya daha fazla simge taşıyabilir:

- A router.
- A switch.
- A wireless access point.
- A firewall.
- A host with some ports filtered.

Controls (Kontroller)

"Kontroller" düğmesine tıklandığında kontroller bir sütunda görünür. Kontroller bölmelere ayrılmıştır.

Action controls (Eylem kontrolleri)

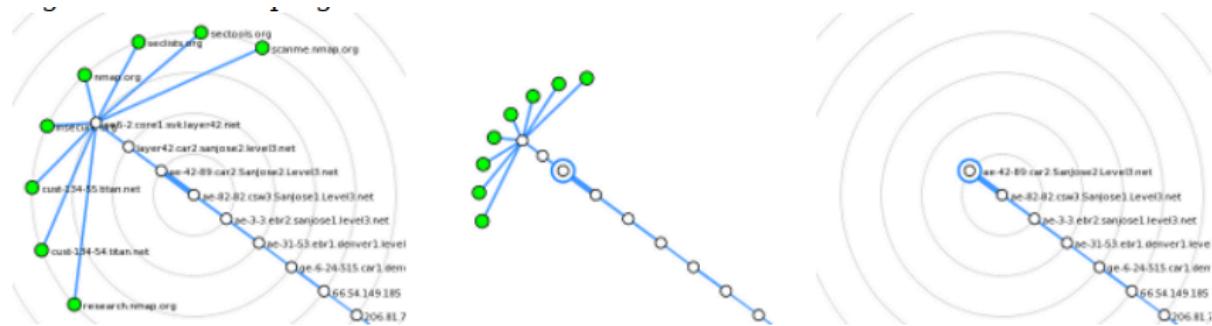


"Eylem" bölümündeki kontroller, bir ana bilgisayara tıkladığınızda ne olacağını kontrol eder. Bu bölümdeki düğmeler soldan sağa "Odağı değiştir", "Bilgileri göster", "Çocukları grupta" ve "Bölgeyi doldur" şeklindedir. Mod "Odağı değiştir" olduğunda, bir ana bilgisayara tıklandığında, seçili ana bilgisayarı merkeze koymak için ekran yeniden düzenlenir. Mod "Bilgi göster" olduğunda, bir ana bilgisayara tıklandığında, ana bilgisayar hakkında bilgi içeren bir pencere açılır.

Mod "Çocukları grupta" olduğunda, bir ana bilgisayara tıklandığında, merkezden daha uzakta olan düğümler tüm çocukların içine çöker. Bir ana bilgisayar

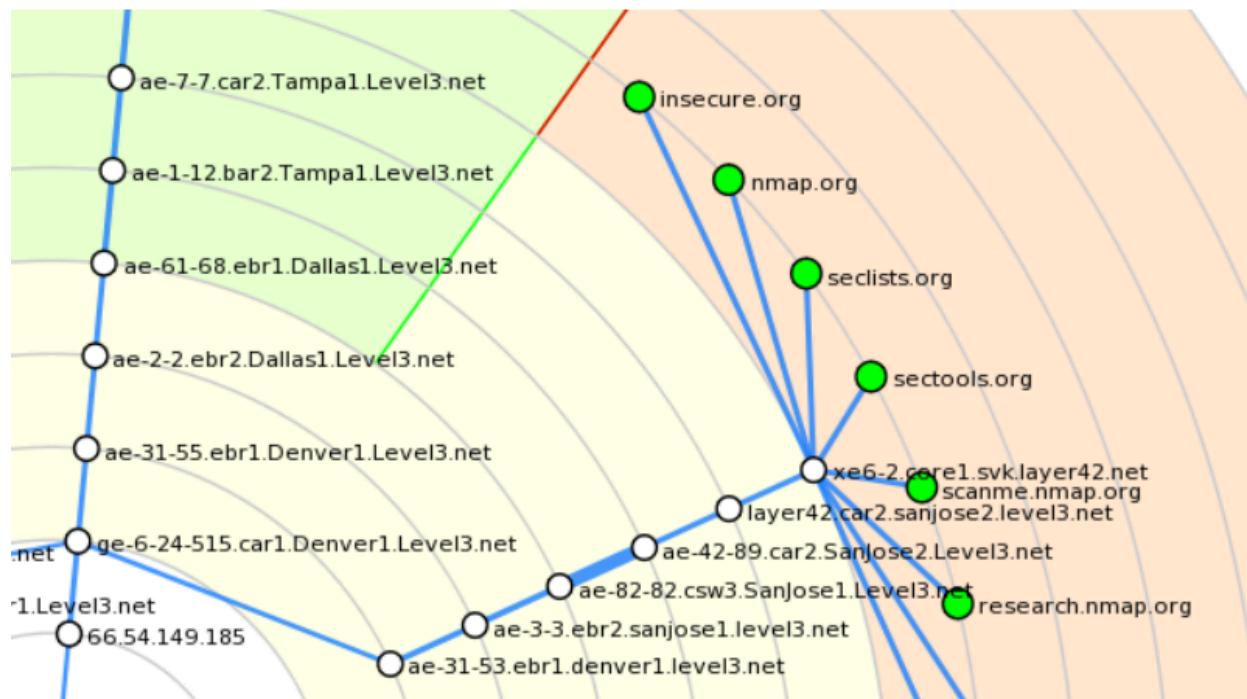
gruplandığında şu şekilde görünür: . Gruplanmış bir düğüme tıklandığında düşümün grubu tekrar açılır. Bu diyagram gruplama işlemini göstermektedir.

Şekil 12.7. Bir ana bilgisayarın çocuklarınını gruplama

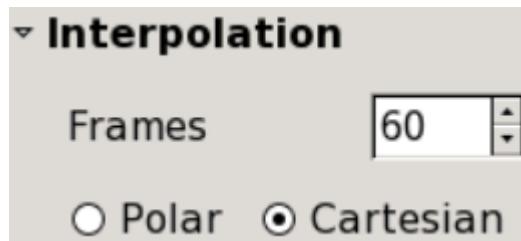


Mod "Bölgeyi doldur" olduğunda, bir ana bilgisayara tıklandığında ekranın ana bilgisayar ve alt bilgisayarları tarafından işgal edilen bölgesi vurgulanır. Vurgulanan ana bilgisayarlar, "Çocukları gruplandır" modunda gruplandırılacak olanlarla tamamen aynıdır. Farklı bölgeleri vurgulamak için farklı renkler seçebilirsiniz. Bu diyagram, farklı renklerle vurgulanmış birkaç bölgenin bir örneğini göstermektedir.

Şekil 12.8. Topolojinin bölgelerini vurgulama

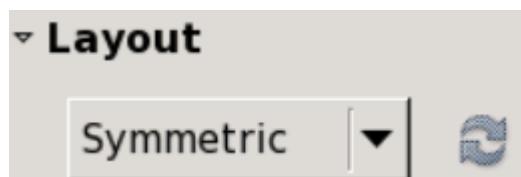


Interpolation controls (Enterpolasyon kontrolleri)



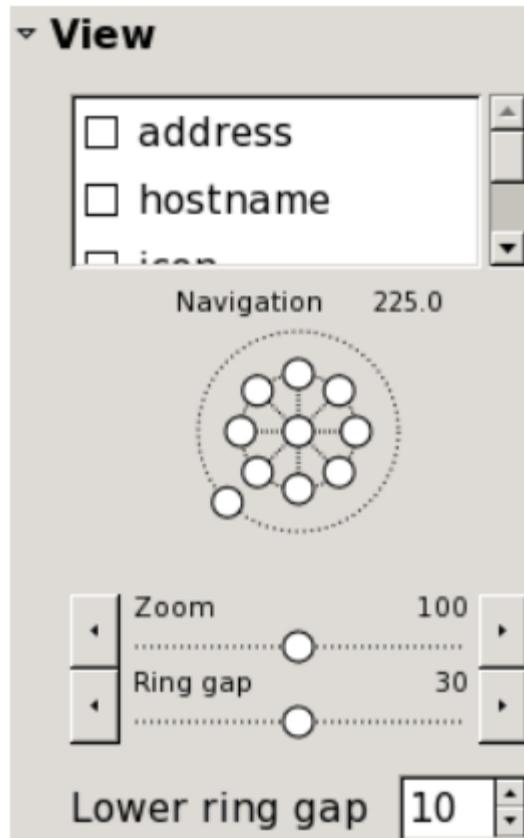
"Enterpolasyon" bölümündeki kontroller, grafiğin bir kısmı değiştiğinde animasyonun ne kadar hızlı ilerleyeceğini kontrol eder.

Layout controls (Düzen kontrolleri)



Düğümlerin otomatik yerlesimi için iki seçenek vardır. Simetrik mod, bir ana bilgisayarın her alt ağacına grafiğin eşit büyüklükte bir dilimini verir. Ağ hiyerarşisini iyi gösterir ancak merkezden uzaktaki ana bilgisayarlar birbirine yakın sıkıştırılabilir. Ağırlıklı mod, daha fazla çocuğu olan ana bilgisayarlara grafikte daha büyük bir parça verir.

View controls (Kontrolleri Görüntüle)



"Görünüm" bölümündeki onay kutuları ekranın bazı kısımlarını etkinleştirir ve devre dışı bırakır. Örneğin, her ana bilgisayar için yalnızca bir IP adresi göstermek için "ana bilgisayar adı" seçeneğini devre dışı bırakın veya hiç etiket kullanmamak için "adres" seçeneğini devre dışı bırakın. "Latency" seçeneği, Nmap'in --traceroute seçeneği tarafından belirlendiği gibi, her bir ana bilgisayara gidiş-dönüş sürelerinin görüntülenmesini etkinleştirir veya devre dışı bırakır. "Yavaş giriş/çıkış" seçeneği işaretlenirse animasyon doğrusal olmaz, animasyonun ortasında daha hızlı, başında ve sonunda ise daha yavaş ilerler.

Pergel benzeri widget ekranı sekiz yönde kaydırır. Merkez ana bilgisayara dönmek için merkeze tıklayın. Dış tarafın etrafındaki halka tüm grafiğin dönüşünü kontrol eder.

"Yakınlaştır" ve "Halka boşluğu" her ikisi de grafiğin genel boyutunu kontrol eder. "Yakınlaştırma" her şeyin boyutunu değiştirir - ana bilgisayarlar, etiketler, bağlantı çizgileri. "Halka boşluğu" sadece eşmerkezli halkalar arasındaki boşluğu artırır, diğer her şeyi aynı boyutta tutar. "Düşük halka boşluğu" halkalar için minimum bir aralık verir, özellikle balık gözü etkinleştirildiğinde kullanılmalıdır.

Fisheye controls (Balık gözü kontrolleri)

Fisheye on ring with interest factor and spread factor

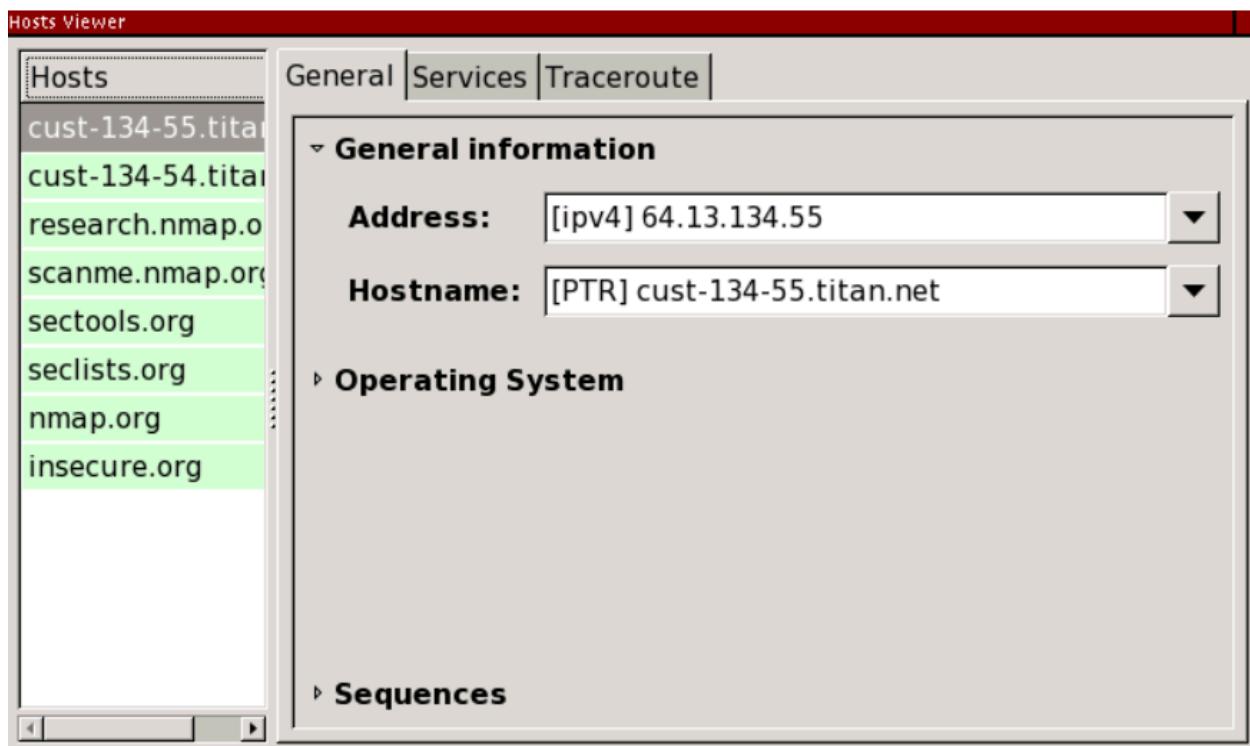
Balık gözü kontrolleri, seçilen bir halkaya daha fazla yer vererek diğerlerini sıkıştırır. Kaydırıcı hangi halkanın daha çok ilgi çekeceğini kontrol eder. "İlgı faktörü", seçilen halka için halka aralığının balık gözü olmadığından olacağından kaç kat daha fazla olduğunu belirtir. "Yayılma faktörü" -1 ile 1 arasında değişir. Seçilen halkanın etrafında kaç komşu halkanın genişletileceğini kontrol eder, daha yüksek sayılar daha fazla yayılma anlamına gelir.

Keyboard Shortcuts (Klavye Kısıyolları)

Topoloji ekranı bu klavye kısayollarını tanır:

Tuş	İşlev
c	Ekranı orta ana bilgisayara döndürür.
a	Ana bilgisayar adreslerini gösterir veya gizler.
h	Ana bilgisayar adlarını gösterir veya gizler.
i	Ana bilgisayar simgelerini gösterir veya gizler.
l	Gecikme süresini gösterir veya gizler.
r	Halkaları gösterir veya gizler.

The Hosts Viewer (Ev Sahipleri Görüntüleyicisi)



Ana bilgisayar görüntüleyici, ana bilgisayarlar hakkında ayrıntılı bilgi almak için alternatif bir yoldur. "Hosts Viewer" düğmesine tıklayarak görüntüleyiciyi etkinleştirin. Envanterdeki tüm ana bilgisayarlar bir liste halinde sunulur. Hakkında ayrıntılı bilgi almak için herhangi bir ana bilgisayarı seçin.

The Profile Editor (Profil Editörü)

Nmap ile aynı taramayı tekrar tekrar çalıştırın istemek yaygındır. Örneğin, bir sistem yöneticisi işleri takip etmek için ayda bir kez tüm ağları tarayabilir. Zenmap'in bunu kolaylaştıran mekanizmasına profiller denir.

Şekil 12.9. Bir profil seçme

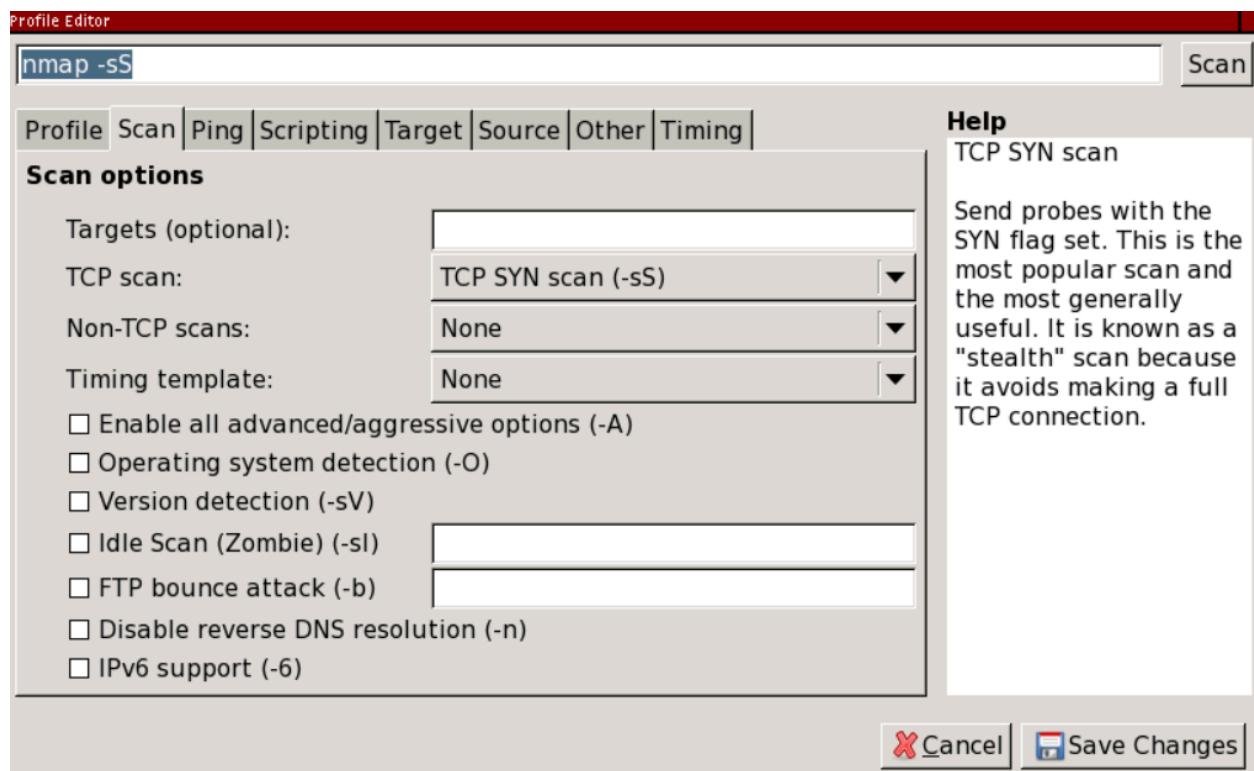


Her pencere "Profil" etiketli bir birleşik giriş kutusu içerir. Açıldığında hangi profillerin mevcut olduğu gösterilir. Bir profilin seçilmesi "Komut" alanında çalıştırılacak komut satırının görüntülenmesine neden olacaktır. Zenmap ile birlikte gelen profiller birçok tarama amaci için yeterlidir, ancak er ya da geç kendi profilinizi oluşturmak isteyeceksiniz.

Editing a Command (Komut Düzenleme)

Profil editörü kullanışlı bir interaktif Nmap komut editörü olarak kullanılabilir. "Profil" menüsü altından "Yeni Profil veya Komut" u seçin veya ctrl+P klavye kısayolunu kullanın. Profil düzenleyici, ana pencerede hangi komut gösteriliyorsa onu görüntüleyerek görünecektir.

Şekil 12.10. Profil düzenleyici



Üstteki metin girişи düzenlenmekte olan komutu gösterir. Kullanmak istediğiniz seçenekleri biliyorsanız doğrudan bu alana yazabilirsiniz. Ortadaki kontroller, kutuları işaretleyerek veya menülerden seçim yaparak seçenekleri belirlemenizi sağlar. Komut dizesi ile kontroller arasında iki yönlü bir ilişki vardır: kontrollerden birini değiştirdiğinizde komut dizesinde anında bir değişikliğe neden olur ve komut dizesini düzenlediğinizde kontroller kendilerini eşleşecek şekilde günceller. Ne

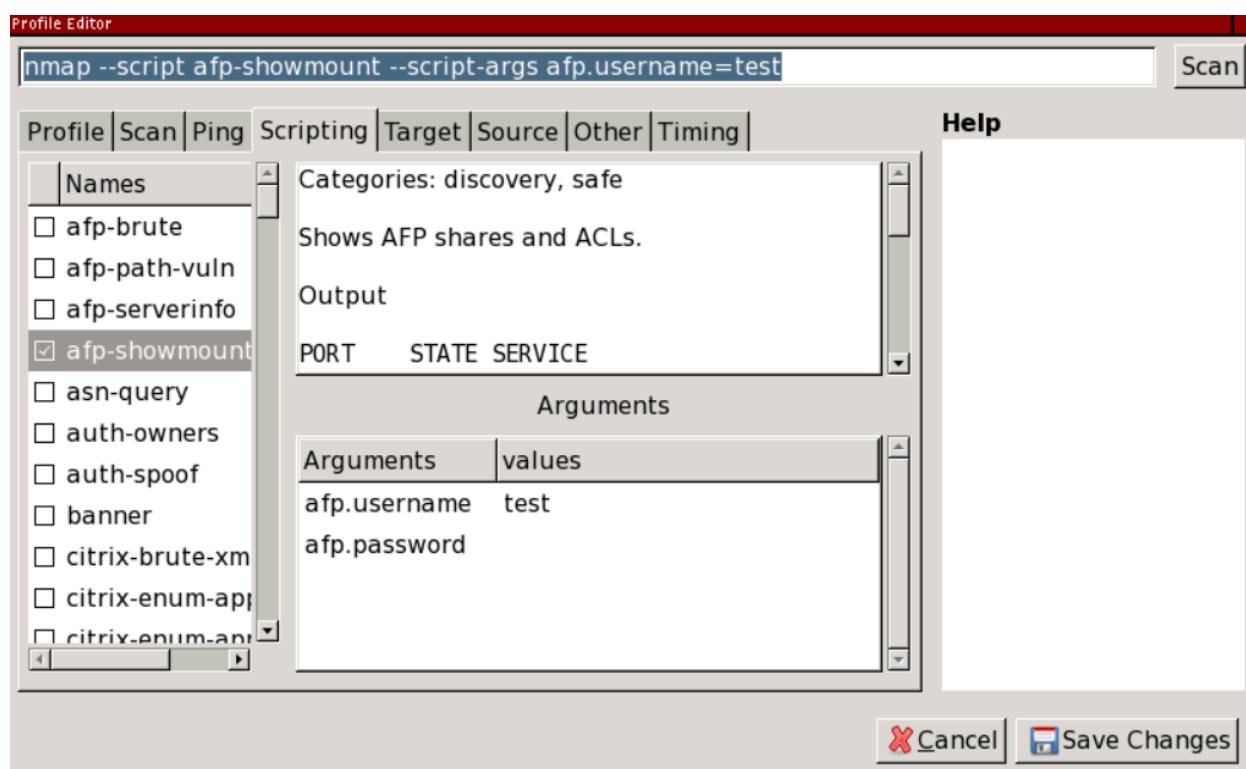
yaptığının ve ne tür bir girdi beklediğinin açıklamasını görmek için fare imlecini bir seçenekin üzerine getirin.

Yeni komut satırını çalıştırmak için "Tara" düğmesine tıklayın. Bu, komutu ana pencereye kopyalayacak, profil düzenleyiciyi kapatacak ve taramayı çalışmaya başlayacaktır. Komutta daha fazla değişiklik yapmak için, ekranda hangi komut gösteriliyorsa onu kullanacağını hatırlayarak tekrar "Yeni Profil veya Komut "u seçin.

Script selection (Senaryo seçimi)

"Komut Dosyası" sekmesi, birçok seçenekin nedeniyle özel olarak bahsedilmeyi hak ediyor. Soldaki kayan liste script.db'de yüklü olan tüm komut dosyalarını gösterir. Komut dosyaları, komut dosyası adının yanındaki onay kutusuna tıklanarak tek tek seçilebilir veya seçimleri kaldırılabilir. Bir komut dosyası vurgulandığında, açıklaması ve bağımsız değişkenleri görüntülenir. Bağımsız değişkenler düzenlenenebilir. Yardımı görmek için fare imlecini bir bağımsız değişkenin üzerine getirin. Şekil 12.11, "Komut Dosyası" profil düzenleyici sekmesi" örnek bir komut dosyası seçim oturumunu göstermektedir.

Şekil 12.11. "Scripting" profil düzenleyici sekmesi



"Komut Dosyası Seçimi" başlıklı bölümde açıklandığı gibi komut dosyalarını kategorilere veya Boole operatörlerine göre seçmek için, üstteki komut girişinde --script seçeneğinin argümanını düzenleyin. Seçilen komut dosyalarının kayan listesi kısa bir gecikmeden sonra kendini güncelleyecektir.

Creating a New Profile (Yeni Profil Oluşturma)

Yeni bir profil oluşturma prosedürü, bir komutu düzenleme ile hemen hemen aynıdır. "Profil" menüsünden "Yeni Profil" veya Komut "u seçin ve komutu istediğiniz gibi düzenleyin. Ardından, "Tara" ya tıklamak yerine "Profil" sekmesine gidin ve profile bir isim verin. Ardından yeni profili kaydetmek için "Değişiklikleri Kaydet" e tıklayın.

Bir profil tarama hedeflerini içerebilir veya içermeyebilir. Aynı taramayı sıkılıkla aynı hedef kümesine karşı çalıştırıyorsanız, hedefleri profil içinde listelemeyi uygun bulacaksınız. Aynı taramayı farklı hedeflere karşı çalıştmayı planlıyorsanız, "Hedefler" alanını boş bırakın ve hedefleri daha sonra taramayı çalıştırığınızda doldurun.

Editing or Deleting a Profile (Profili Düzenleme veya Silme)

Bir profili düzenlemek için, düzenlemek istediğiniz profili seçin, ardından "Profil" menüsünden "Seçili Profili Düzenle" yi seçin veya **ctrl+E** klavye kısayolunu kullanın. Profil düzenleyici, bu kez seçilen profilden doldurulan ad ve açıklama ile açılacaktır. Değişiklikleri kaydetmek için "Değişiklikleri Kaydet" e veya kaydetmeden çıkmak için "İptal" e tıklayın.

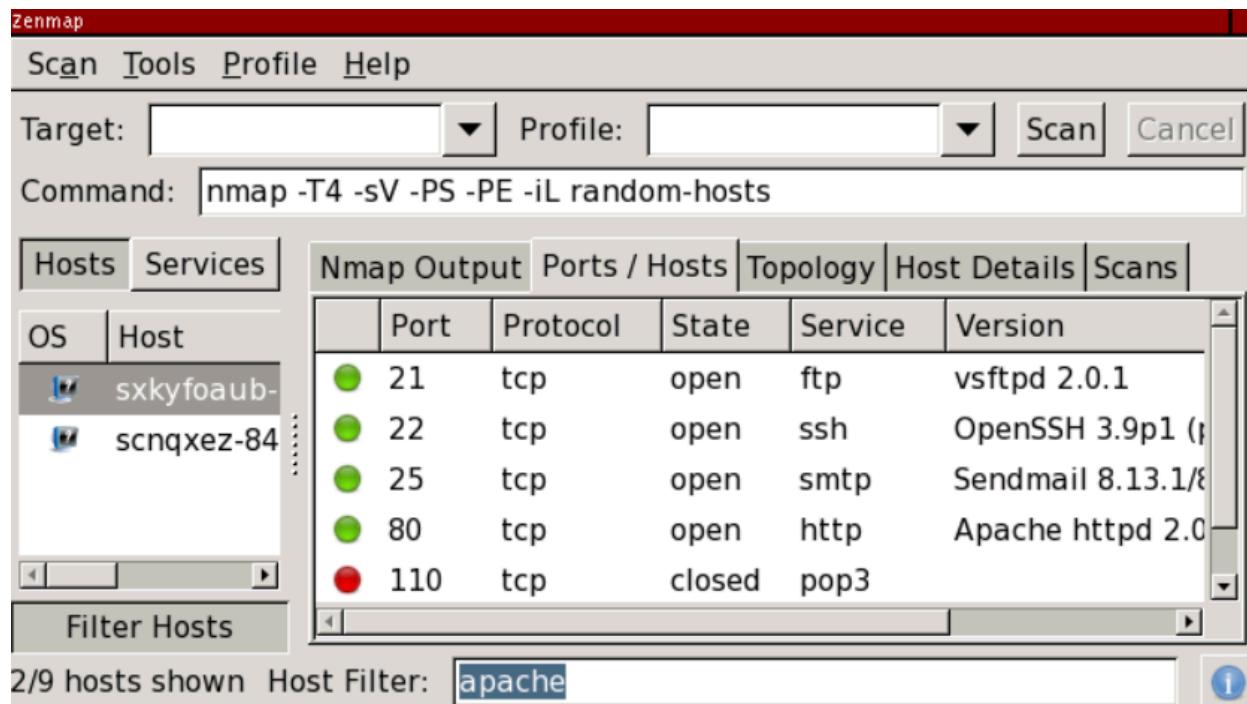
"Seçili Profili Düzenle" seçeneğini kullanarak profil düzenleyiciyi açtığınızda, alt kısmda ek bir "Sil" düğmesi bulunacaktır. Zenmap profili silmeden önce bir uyarı sunacaktır.

Host Filtering (Ana Bilgisayar Filtreleme)

Ana bilgisayar listesi bir filtre dizesi ile filtrelenebilir. Filtre etkin olduğunda, yalnızca filtre dizesiyle eşleşen ana bilgisayarlar gösterilir. Diğer ana bilgisayarlar envanterde hala mevcuttur, ancak gizlidir. Bir filtre dizesi kullanarak, geniş bir taramayı yalnızca ilgilendiğiniz ana bilgisayarlara hızlı bir şekilde daraltabilirsiniz.

Şekil 12.12, apache ile eşleşen dokuz ana bilgisayardan ikisini bulan ana bilgisayar filtresini göstermektedir.

Şekil 12.12. Ana bilgisayar滤resi



Ana bilgisayar滤resi "Ana Bilgisayarları Filtrele" düğmesine tıklanarak, "Araçlar" menüsünden "Ana Bilgisayarları Filtrele" seçilerek veya **ctrl+L** tuşlarına basılarak etkinleştirilir. Bu, ekranın altındaki滤re çubuğuunu yükseltecektir. Ana bilgisayarlar siz yazdıktan sonra canlı olarak filtrelenir. Filtreyi devre dışı bırakmak için "Ana Bilgisayarları Filtrele" seçeneğine tıklayın veya滤re çubuğuunu gizlemek için **ctrl+L** tuşlarına tekrar basın.

Ana bilgisayar filterlemenin bu en basit şekli temel anahtar kelime eşleştirmidir. Apache, linux veya telnet gibi bir veya daha fazla dize girin ve bilgilerinin herhangi bir yerinde (ana bilgisayar adı veya IP adresi, bağlantı noktası adı veya numarası, işletim sistemi eşleşmesi veya hizmet sürümü dizesi olarak) bu dizeye sahip ana bilgisayarlarla eşleşeceklerdir. Filter dizesinde birden fazla sözcük varsa, bir ana bilgisayarın gösterilmesi için tüm sözcüklerin eşleşmesi gereklidir.

Daha spesifik sonuçlar için, ana bilgisayar filteri her bir ana bilgisayarda yalnızca belirli veri alanlarının eşleştirilmesini destekler. Desteklenen sözdizimi,

"Kaydedilen Sonuçların Aranması" bölümünde ayrıntılı olarak ele alınan arama kriterlerinin bir alt kümesidir. Ana bilgisayar filtresinde izin verilen kriterler şunlardır anahtar kelime eşleştirme;

port states(liman eyaletleri): `open`, `closed`, `filtered`, `open|filtered`, `closed |filtered`, and `unfiltered`;

`os:` (operating system)(İşletim Sistemi);

`service:` (service version); and ((hizmet sürümü); ve)

`inroute:` (host in route). ((rotadaki ana bilgisayar).)

İşte bazı ana bilgisayar filtre dizesi örnekleri.

apache ⇒ Bu anahtar sözcük araması, Apache web sunucusunu çalıştırıldığı tespit edilen ana bilgisayarlarla eşleşir, ancak içinde "apache" geçen herhangi bir ana bilgisayarla da eşleşir, örneğin apache.example.com adlı bir ana bilgisayar.

service:apache ⇒ Bu, önceki dize gibidir, ancak yalnızca bir hizmet sürümündeki "apache" ile eşleşecektir.

os:linux ssh ⇒ SSH çalıştırılan Linux ana bilgisayarlarını gösterir.

open:445 ⇒ Port 445 açık olan tüm ana bilgisayarıları gösterir.

Searching Saved Results (Kaydedilen Sonuçları Arama)

Zenmap kayıtlı tarama sonuçları dosyalarında ve son taramaların veritabanında arama yapmanızı sağlar. Aramaya başlamak için "Araçlar" menüsünden "Tarama Sonuçlarını Ara" yi seçin veya ctrl+F klavye kısayolunu kullanın. Arama iletişim kutusu Şekil 12.13'te gösterildiği gibi görünür.

Şekil 12.13. Arama iletişim kutusu

Search Scans	
Search: <input type="text"/>	Expressions
Scan	Date
Intense Scan on scanme.nmap.org	2008-07-01 11:26
Quick Scan on localhost	2008-07-01 11:26
nmap -T Aggressive -v localhost	2008-07-01 16:10
Regular Scan on scanme.nmap.org	2008-07-01 16:10

Matched **4** out of **4** scans.

Close Append Open

Arama arayüzü başlangıçta son taramalar veritabanındaki tüm taramaları gösterir (bunun için "Son Taramalar Veritabanı" adlı bölüme bakın). Tüm taramaların gösterilmesinin nedeni basittir - aramaya henüz herhangi bir kısıtlama getirilmemiştir, bu nedenle olası her sonuç döndürülür.

Aramalar çeşitli arama kriterleri açısından verilebilir, ancak en basit arama sadece bir anahtar kelime aramasıdır. Ana bilgisayar adı, işletim sistemi adı, profil veya başka bir şey olarak çıktılarının bir parçası olarak bu kelimeyi içeren tüm taramaları bulmak için "Ara" alanına scanme gibi bir kelime yazmanız yeterlidir. Bunun bir örneği Şekil 12.14'te gösterilmektedir.

Şekil 12.14. Anahtar kelime arama

Search Scans	
Search: <input type="text" value="scanme"/>	Expressions
Scan	Date
Intense Scan on scanme.nmap.org	2008-07-01 11:26
Regular Scan on scanme.nmap.org	2008-07-01 16:10

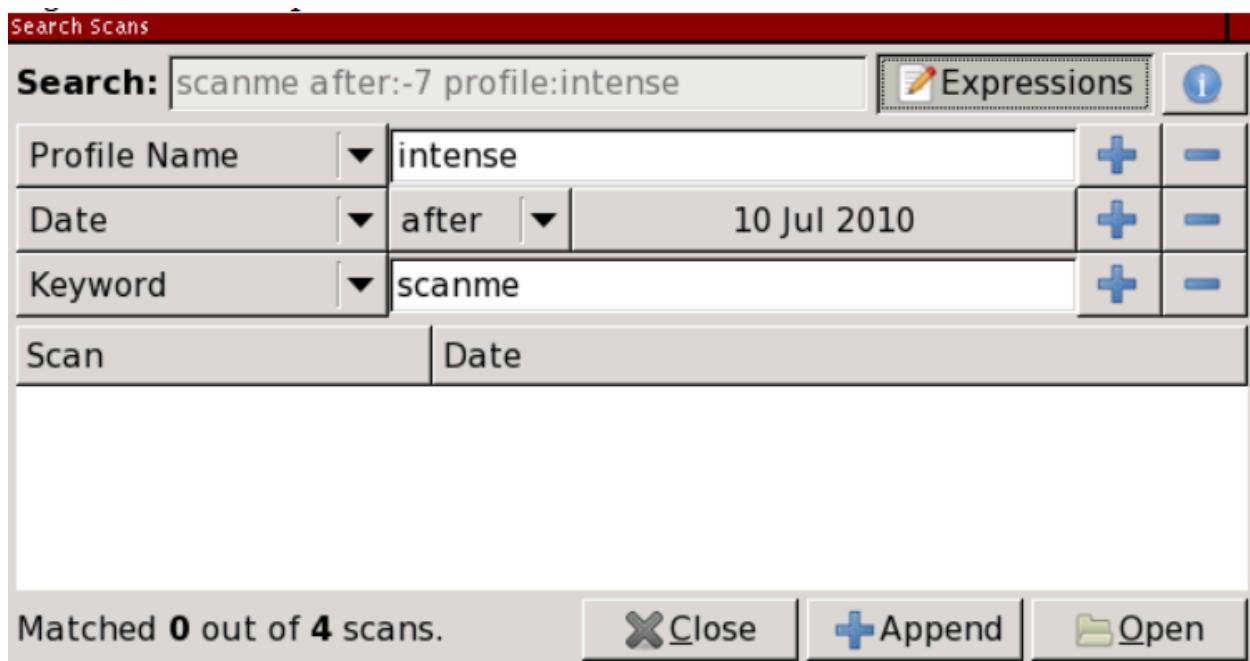
Matched **2** out of **4** scans.

Close Append Open

Aramalar siz yazarken canlı olarak gerçekleşir. İstediğiniz taramayı bulduğunuzda "Aç" düğmesine tıklayın veya tarama adına çift tıklayın.

"İfadeler" arayüzü kullanılarak daha karmaşık aramalar oluşturulabilir. "İfadeler" düğmesine tıklayın ve mevcut aramanın grafiksel gösterimi görünecektir. Görüntülenen birleşik kutulardan seçim yaparak aramayı değiştirebilir. Kriter eklemek için "+", kaldırma için "-" düğmesine tıklayın. Kriterleri gizlemek için "İfadeler" düğmesine tekrar tıklayın (arama dizesinde hala mevcutturlar). İfadeler gösterilirken arama metninin düzenlenmesi devre dışı bırakılır. Daha karmaşık bir arama örneği Şekil 12.15'te gösterilmektedir.

Şekil 12.15. İfade arama



Aramalar ve tabanlıdır, yani bir taramanın eşleşmesi ve sonuç listesinde görünmesi için tüm kriterlerin doğru olması gereklidir. Çoğu arama büyük/küçük harfe duyarsızdır. (Büyük/küçük harfe duyarlı tek kriter option:'dır.) Varsayılan olarak yalnızca son taramalar veritabanındaki taramalar aranır. Bir dizindeki dosyaları özyinelemeli olarak aramak için "Dizini Dahil Et" ifadesini kullanın.

Bir arama ifadesi seçtiğinizde, arama girişinde bunun bir metin temsilinin göründüğünü fark etmişsinizdir. "Arama" alanındaki dize aramayı gerçekten kontrol eden şeydir; "İfadeler" arayüzü sadece bunu ayarlamak için uygun bir yoldur. Hangi arama dizelerinin hangi ifadelere karşılık geldiğini öğrendiğinizde, ifadeler

arayüzüünü atlayabilir ve doğrudan bir arama dizesi yazabilirsiniz. Aşağıda, arama arayüzü tarafından tanıyan tüm metinsel arama kriterlerinin bir listesi bulunmaktadır. Çoğu kriterin kısa bir formu vardır: d:-5, date:-5 ile aynıdır ve op:80, open:80 ile aynıdır. Her bir kriterin kısa formu aşağıdaki listede verilmiştir.

`<keyword>` ⇒ Süslenmemiş bir kelime taramadaki her şeyle eşleşir. Örneğin, apache tüm Apache sunucularıyla ve linux tüm Linux ana bilgisayarlarıyla eşleşir. Anahtar kelime aramasını kullanırken, bir ana bilgisayarın apache veya linux olarak adlandırılması gibi yanlış pozitif olma ihtimali vardır.

Port states (Liman devletleri) ⇒ Her olası liman durumu aynı zamanda bir arama kriteridir. Bunlar

```
open:<ports>  (op: for short)
closed:<ports>  (cp: for short)
filtered:<ports>  (fp: for short)
unfiltered:<ports>  (ufp: for short)
open|filtered:<ports>  (ofp: for short)
closed|filtered:<ports>  (cfp: for short)
```

80 numaralı bağlantı noktası açık olan bir ana bilgisayara sahip taramalarla eşleşmek için open:80 kullanın. `<ports>` bağımsız değişkeni virgülle ayrılmış bir liste de olabilir.

Ayrıca scanned:<ports> (kısaca sp:) kriteri, son durumları ne olursa olsun, verilen portların tarandığı taramalarla eşleşir.

`date: <YYYY-MM-DD>` or `date:- <n>` (`d:` for short) ⇒ `<YYYY-MM-DD>` biçiminde verilen tarihte gerçekleşen taramaları eşleştirir. Veya `<n>` gün önce herhangi bir günde gerçekleşen taramaları eşitmek için `date:-<n>` kullanın. Dün gerçekleştirilen taramaları bulmak için `date:-1` kullanın.

- `<YYYY-MM-DD>` biçimini kullanıldığında, tarihin ardından bir veya daha fazla ~ işaretü gelebilir ve bunların her biri eşleşen tarih aralığını her iki tarafta da bir gün genişletir. `date:2007-12-23`, 23 Aralık 2007 tarihinde 00:00 ile 24:00 arasında gerçekleşen taramalarla eşleşir. `date:2007-12-23~`, 22 Aralık 00:00 ile 24 Aralık 24:00 arasında gerçekleşen taramalarla eşleşir. Bu "bulanık" tarih

eşleştirme, bir taramayı tam olarak ne zaman çalıştırıldığınızı hatırlayamadığınızda kullanışlıdır.

`after: <YYYY-MM-DD>` or `after:- <n>` (`a:` for short) ⇒ `<YYY-MM-DD>` biçiminde verilen tarihte veya sonrasında gerçekleşen taramalarla eşleşir. Veya son `<n>` gün içinde gerçekleşen taramaları eşlestirmek için `after:-<n>` kullanın. Örneğin, `after:-7` son bir hafta içinde gerçekleşen taramalarla eşleşir.

`before: <YYYY-MM-DD>` or `before:- <n>` (`b:` for short) ⇒ `<YYY-MM-DD>` biçiminde verilen tarihte veya öncesinde gerçekleşen taramaları eşleştirir. Veya `<n>` gün öncesinden önce herhangi bir zamanda gerçekleşen taramaları eşlestirmek için `before:-<n>` kullanın.

`target: <name>` (`t:` for short) ⇒ Verilen ada sahip tüm ana bilgisayarların taramalarını eşleştirir. Ad, taramada belirtilen ad veya herhangi bir ana bilgisayarın ters-DNS adı olabilir.

`option: <option>` (`o:` for short) ⇒ Verilen komut satırı seçeneğini kullanan taramalarla eşleşir. Baştaki - veya -- işaretini atlayın: `option:A`, `-A` seçeneğini kullanan taramalarla eşleşir.

- Bu kriter yalnızca tam anlamıyla eşleşir. `option:O`, `-A -O` anlamına gelse bile `-A` kullanılan taramalarla eşleşmeyecektir. Benzer şekilde `option:sU`, `-sSU` kullanılan taramalarla eşleşmeyecektir. Seçenek eşleştirme büyük/küçük harfe duyarlıdır.

`os: <string>` ⇒ İşletim sistemi açıklamalarının herhangi bir bölümünde verilen dizeye sahip ana bilgisayarların taramalarını eşleştirir. `os:windows`, Microsoft Windows ana bilgisayarlarının taramalarını genel olarak döndürür.

`service: <string>` (`s:` for short) ⇒ Portlarından herhangi birinin hizmet açıklamasının herhangi bir bölümünde verilen dizeye sahip ana bilgisayarların taramalarını eşleştirir. `service:ssh`, herhangi bir SSH türünü çalıştırın ana bilgisayarların taramalarını döndürür.

`profile: <name>` (`pr:` for short) ⇒ Adı verilen profili kullanan taramalarla eşleşir, örneğin profil: "yoğun tarama".

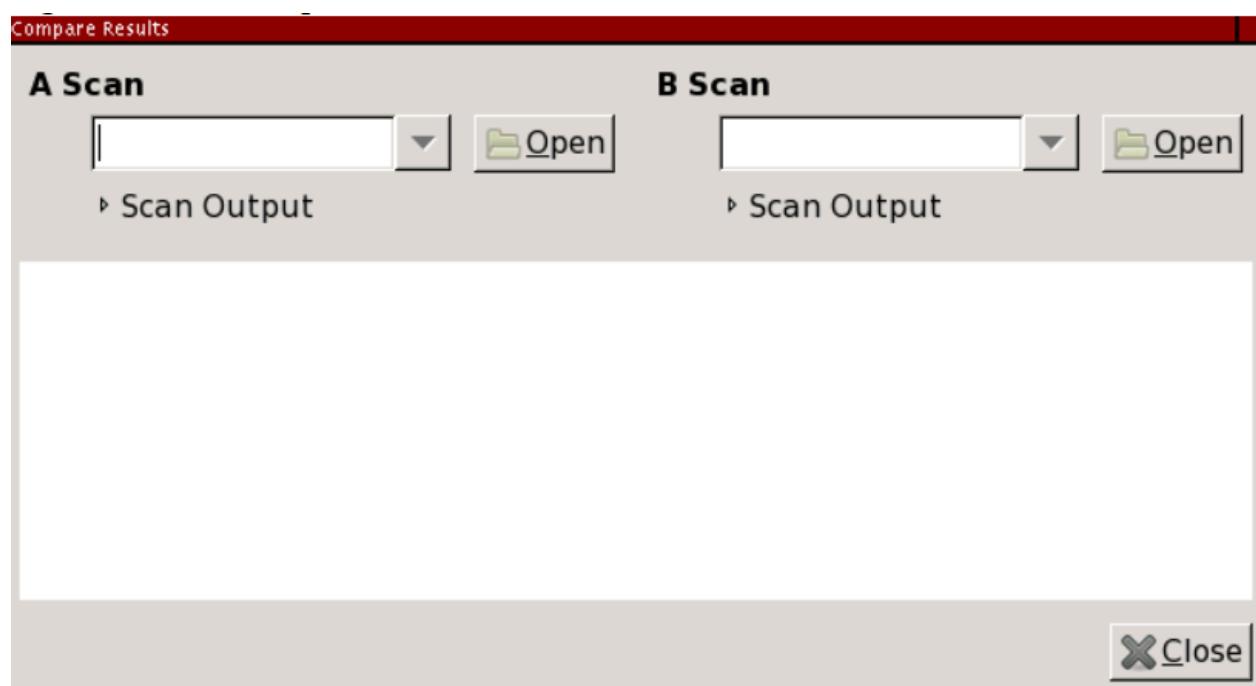
`inroute: <host>` (`ir:` for short) ⇒ Verilen ana bilgisayarın --traceroute çıktısında bir ara yönlendirici olarak göründüğü taramalarla eşleşir.

`dir: <directory>` ⇒ dir: gerçekten bir arama kriteri değildir. Daha ziyade, son taramalar veritabanındaki lere ek olarak dosya sistemindeki bir dizini aramanın yoludur. Dizinler, varsayılan olarak yalnızca xml olmak üzere belirli uzantılarla biten dosyalar için özyinelemeli olarak aranır. Daha fazla dosya adıyla eşleşmek için zenmap.conf dosyasının [search] bölümündeki file_extension değişkenini "zenmap.conf dosyasının bölümleri" başlıklı bölümdeki talimatlara göre değiştirin.

Comparing Results (Sonuçların Karşılaştırılması)

Aynı taramayı farklı zamanlarda iki kez çalıştmak veya biraz farklı iki taramayı aynı anda çalıştmak ve ne kadar farklı olduklarını görmek yaygın bir istektir. Zenmap tarama sonuçlarını karşılaştırmak için Şekil 12.16'da gösterilen bir arayüz sağlar. "Araçlar" menüsünden "Sonuçları Karşılaştır" seçeneğini seçerek veya **ctrl+D** ("diff" olarak düşünün) klavye kısayolunu kullanarak karşılaştırma aracını açın. Zenmap bir seferde iki tarama sonucunun karşılaştırılmasını destekler.

Şekil 12.16. Karşılaştırma aracı



Karşılaştırma yapmanın ilk adımı, karşılaştırılacak iki tarama seçmektir; bunlar "A taraması" ve "B taraması" olarak adlandırılır. Birleşik giriş kutuları açık taramalar arasından seçim yapmanızı sağlar. Ya da bir dosyadan tarama sonuçlarını almak için "Aç" düğmelerine tıklayın. Son taramalar veritabanındaki sonuçları karşılaştırmak için, önce arama arayüzüne kullanarak bu taramaları açmalısınız ("Kaydedilen Sonuçların Aranması" bölümüne bakın).

İki taramanın sırası önemlidir. Karşılaştırma, dosyalarda kaydedilen sürelerle bakılmaksızın her zaman A taramasından B taramasına "yapılır". İki sonuç seçildikten sonra karşılaştırma hemen başlar. Şekil 12.17'de birkaç gün arayla birkaç Internet ana bilgisayarının iki taraması arasında yapılan bir karşılaştırma gösterilmektedir.

Şekil 12.17. Karşılaştırma çıktısı

The screenshot shows the 'Compare Results' window in Zenmap. It has two main sections: 'A Scan' and 'B Scan'. Each section contains a command line input field ('nmap -T4 -sV -PS -F'), a 'Scan Output' button, and a scrollable terminal window. The 'A Scan' terminal shows:

```
-Nmap 4.76 at 2008-09-16 13:59
+Nmap 4.75 at 2008-09-11 11:39
```

The 'B Scan' terminal shows:

```
- 10.189.71.117:
+scnqxez-842.example.com (10.189.71.117):
Host is up.
Not shown: 995 filtered ports
PORT      STATE      SERVICE      VERSION
```

Below the terminals is a horizontal scrollbar. In the bottom right corner, there is a 'Close' button with a red X icon.

Farkı çıktısı Nmap'in çıktısına benzer. Her satırda önce ' ', '-' veya '+' gelir, bu da sırasıyla bazı bilgilerin değişmediğini, kaldırıldığını veya eklediğini gösterir. Renk kodlaması da farklılıklarını gösterir; silme için kırmızı ve ekleme için yeşil.

Zenmap'in karşılaştırma işlevinin altında yatan motor, Nmap ile birlikte dağıtılan grafiksel olmayan bir araç olan Ndifftir. Ndifft, Zenmap'in üzerinde çalıştığı tüm platformlarda çalışır. Eğer ndifft çalıştırılabilir dosyasını varsayılan konumundan

başka bir yere yüklediyseniz, zenmap.conf dosyasının [paths] bölümündeki nmap_command_path değişkenini değiştirmeniz gerekebilir.

Zenmap in Your Language (Kendi Dilinizde Zenmap)

Zenmap İngilizce dışında birkaç dile daha çevrilmiştir. Şekil 12.18 Zenmap'in Almanca'da nasıl göründüğünü göstermektedir. Bu bölümde Zenmap'in çevirilerinin nasıl kullanılacağı gösterilmektedir.

Şekil 12.18. Almanca Zenmap



Unix benzeri sistemlerde, tercih ettiğiniz dili LANG ortam değişkenini ayarlayarak seçersiniz. Farklı dil seçim olanaklarına sahip diğer işletim sistemlerinde bile LANG ayarı, diğer yöntemler işe yaramadığında çevirileri almanın en kusursuz yoludur. Unix benzeri işletim sisteminiz LANG'ı dil yapılandırmasının bir yan etkisi olarak ayarlayabilir. Eğer ayarlamıyororsa, .login veya .profile dosyanızda aşağıdaki gibi bir satır ekleyin ve de yerine yerel ayar adınızı yazın:

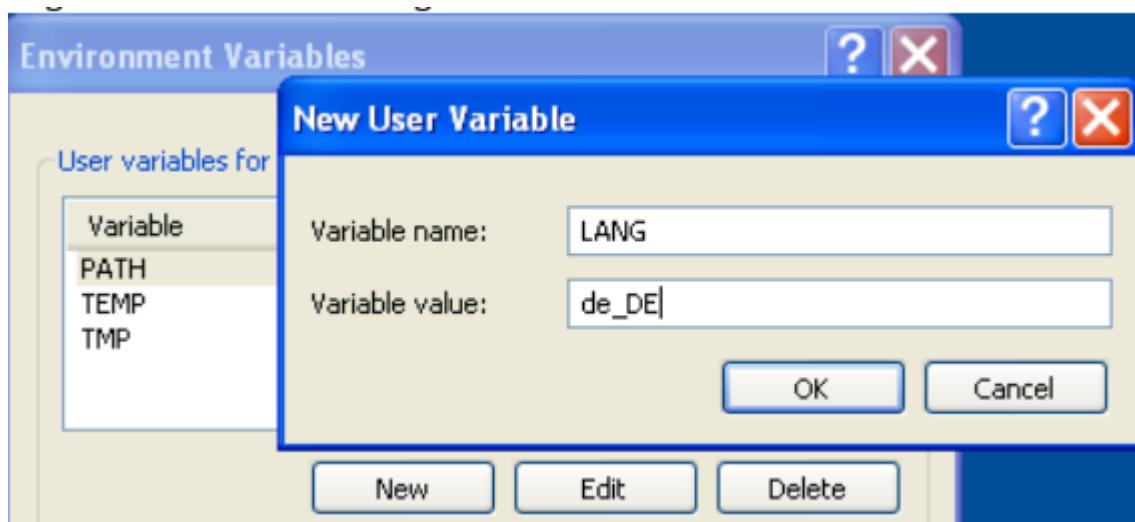
```
export LANG=de
```

Bir yerel ayar adı, isteğe bağlı olarak bir ülke kodu ve bazen başka bilgiler tarafından takip edilen bir dil kodudur. Dil kodları ISO 639'dan, ülke kodları ise ISO

3166'dan alınmıştır. Burada "de" Almanca anlamına gelir; başka bir örnek Brezilya Portekizcesi için "pt_BR" dir. Yerelleştirmeyi tamamen devre dışı bırakmak ve varsayılan İngilizce metni kullanmak için LANG=C olarak ayarlayın.

Windows XP'de LANG'ı ayarlamak için aşağıdaki adımları izleyin. Denetim Masası'nı açın ve "Sistem" öğesini seçin. "Gelişmiş" sekmesine ve ardından "Ortam Değişkenleri" düğmesine tıklayın. Yeni bir ekran açılacaktır; "Kullanıcı değişkenleri" altında "Yeni" ye tıklayın. Görünen formda değişken adı için "LANG" ve değer için yerel ayar adınızı girin. İşlem Şekil 12.19'da gösterilmektedir.

Şekil 12.19. Windows XP'de LANG ortam değişkeninin ayarlanması



Mac OS X'te, yukarıda açıklandığı gibi bir kabuk başlangıç dosyasında LANG ayarının yalnızca Zenmap bir terminalden başlatıldığından etkisi vardır. Grafik Finder arayüzü ortam değişkenlerini ayrı bir dosyada tutar, .MacOSX/environment.plist. Bunu oluşturmak içinTextEdit uygulamasını açın ve de yerel ayar adınızı yazarak aşağıdakileri girin:

```
{ LANG=de; }
```

Ardından "Biçim" menüsünden "Düz Metin Yap"ı seçin. "Kaydet" iletişim kutusunu açın, ev dizinizi seçin ve "Yeni Klasör" e tıklayın. .MacOSX adında bir klasör oluşturun ve beliren uyarıyı geçin. Dosyayı environment.plist adıyla kaydedin ve görünen bir sonraki uyarıda .plist uzantısında ısrar edin. Son olarak, değişikliğin etkili olması için oturumu kapatıp tekrar açın. Bu işlemin bir kısmı Şekil 12.20'de gösterilmektedir.

Şekil 12.20. Mac OS X üzerinde LANG ortam değişkeninin ayarlanması



Creating a new translation (Yeni bir çeviri oluşturma)

Zenmap için yeni bir çeviri oluşturmak veya var olanı güncellemek teknik olarak zor değildir, ancak elbette biraz bilgi birikimi ve İngilizce dışında en az bir dil bilgisi gerektirir. Zenmap'in çevirileri, <http://www.gnu.org/software/gettext/manual/> adresinde tam olarak belgelenmiş olan GNU gettext sistemi tarafından gerçekleştirilir. Bu bölüm manuel çeviri sürecinin bir özetidir. Bu adımları otomatik olarak gerçekleştiren özel çeviri düzenleme programları da mevcuttur.

Diyelim ki "es" dil koduna sahip İspanyolcaya bir çeviri yapacaksınız. Zenmap kaynak ağacında, uygulamadaki tüm çevrilebilir dizeleri içeren bir düz metin dosyası share/zenmap/locale/zenmap.pot vardır. msginit -I es.po -i zenmap.pot komutunu çalıştırarak yeni bir taşınabilir nesne (.po) dosyası oluşturursunuz. Yeni es.po dosyası, uygulamanın tüm İngilizce dizelerini ve ardından uygun çevirileri dolduracağınız boş dizeleri içerir.

Taşınabilir nesne dosyasını test etmek için onu bir makine nesnesi (.mo) dosyasına dönüştürmelisiniz. es/LC_MESSAGES dizinini oluşturun ve ardından msgfmt es.po -o es/LC_MESSAGES/zenmap.mo komutunu çalıştırın. Bu dosya yerindeyken Zenmap, LANG doğru ayarlandığında çevirinizi kullanacaktır. Zenmap.pot değiştiğinde taşınabilir nesne dosyasını güncellemek için msgmerge -U es.po zenmap.pot komutunu çalıştırın. Bu, yeni dizeler ekleyecek ve kaldırılabilmeleri için eski dizeleri işaretleyecektir.

Yeni bir çeviriye başladığınızda, niyetinizi nmap-dev posta listesine duyurun. Posta listesi çeviri tavsiyesi almak için de iyi bir yerdır. İşiniz bittiğinde .po dosyasını gönderin.

Files Used by Zenmap (Zenmap Tarafından Kullanılan Dosyalar)

Zenmap bir dizi yapılandırma ve kontrol dosyası kullanır ve elbette Nmap'in kurulu olmasını gerektirir. Dosyaların nerede saklandığı platforma ve Zenmap'in nasıl yapılandırıldığına bağlıdır. Yapılandırma dosyaları iki kategoriye ayrılır: sistem dosyaları ve kullanıcı başına dosyalar.

The `nmap` Executable (nmap Çalıştırılabilir)

Zenmap, nmap komut satırı çalıştırılabilir dosyasının yüklü olmasına bağlıdır. Program ilk olarak PATH ortam değişkeninde belirtilen tüm dizinlerde aranır.

Bazı platformlarda nmap komutu genellikle YOL'daki dizinlerin hiçbirinde yüklü değildir. Bu platformlar için bir kolaylık olarak, komut PATH'de bulunmazsa aşağıdaki ek dizinler aranacaktır:

- Mac OS X'te /usr/local/bin dizini aranır.
- Windows'ta, Zenmap yürütülebilir dosyasını içeren dizin aranır.

Yürüttürebilir dosyaya mutlak bir yol kullanmak için veya yürütülebilir dosya nmap dışında bir adla yüklenmişse, zenmap.conf dosyasının [paths] bölümündeki nmap_command_path değişkenini değiştirin. Örneğin, nmap'i /opt/bin içine kurduysanız, şunu kullanın

```
[paths]
nmap_command_path = /opt/bin/nmap
```

Veya Nmap'in nmap-custom adlı özel olarak derlenmiş bir sürümüne sahipseniz

```
[paths]
nmap_command_path = nmap-custom
```

"zenmap.conf'un açıklaması" adlı bölüme bakın.

System Configuration Files (Sistem Yapılandırma Dosyaları)

Bu dosyalar Zenmap'in tüm kurulum boyunca yüklenmesini etkiler. Unix ve Mac OS X'te, <prefix>/share/zenmap içindedirler, burada <prefix> Zenmap'in derlendiği dosya sistemi önekidir. Önek muhtemelen /usr veya /usr/local'dır, bu nedenle Zenmap'in dosyası muhtemelen /usr/share/zenmap veya

/usr/local/share/zenmap içindedir. Windows'ta, konum Zenmap'in nereye kurulduğuna da bağlıdır. Muhtemelen C:\Program Files\Nmap\share\zenmap içindedirler. Zenmap sistem yapılandırma dizini aşağıdakileri içerir:

`config/` ⇒ config altındaki dosyalar kullanıcı başına yapılandırma dizinlerine kopyalanır. "Kullanıcı Başına Yapılandırma Dosyaları" adlı bölüme bakın.

`docs/` ⇒ docs alt dizinindeki dosyalar Zenmap'in dokümantasyon dosyalarıdır.

`locale/` ⇒ locale/ alt dizinindeki dosyalar Zenmap tarafından kullanılan metnin diğer dillere çevirilerini içerir.

`misc/profile_editor.xml` ⇒ Bu dosya, profil düzenleyicisi tarafından hangi seçeneklerin sunulacağını tanımlar ("Profil Düzenleyicisi" adlı bölüme bakın). Profil düzenleyiciyi sistem genelinde değiştirmek için dikkatle yenilenebilir.

Per-user Configuration Files (Kullanıcı Başına Yapılandırma Dosyaları)

Bu dosyalar Zenmap'in yalnızca bir kullanıcısını etkiler. Bazıları Zenmap ilk kez çalıştırıldığında sistem dosyalarının config alt dizininden kopyalanır. Varsayılan olarak, kullanıcı başına dosyalar Unix ve Mac OS X'te <HOME>/.zenmap içindedir, burada <HOME> geçerli kullanıcının ev dizini anlamına gelir. Windows Vista ve Windows 7'de C:\Users\<USER>\.zenmap içindedirler. Windows'un önceki sürümlerinde C:\Documents and Settings\<USER>\.zenmap içindedirler. Burada <USER> geçerli kullanıcının adıdır. Farklı bir dizin kullanmak için --confdir seçeneğini kullanın.

`recent_scans.txt` ⇒ Bu, son kaydedilen taramaların dosya adlarının bir listesini içerir. Bu taramalar "Tara" menüsü altında gösterilir. Taramaların burada görünmesi için bir dosyaya kaydedilmiş olması gereklidir. "Tarama Sonuçlarını Kaydetme ve Yükleme" başlıklı bölüme bakın. Bu dosya mevcut değilse Zenmap çalıştırıldığında oluşturulur.

`scan_profile.usp` ⇒ Bu dosya, varsayılanlar ve kullanıcı tarafından oluşturulan profiller de dahil olmak üzere tarama profillerinin açıklamalarını içerir. Bu dosyada değişiklik yapmak için profil düzenleyiciyi ("Profil Düzenleyici" adlı bölüme bakın) kullanmanızı öneririm. Bu dosya Zenmap ilk kez çalıştırıldığında sistem yapılandırma dizininden kopyalanır.

`target_list.txt` ⇒ Bu dosya son taranan hedeflerin bir listesini içerir. Eğer mevcut değilse Zenmap çalıştırıldığında oluşturulur.

`zenmap.conf` ⇒ Bu Zenmap'in ana yapılandırma dosyasıdır. Belirli bir kullanıcının Zenmap kopyası için ayarları tutar ve "zenmap.conf'un açıklaması" adlı bölümde daha ayrıntılı olarak ele alınmıştır.

`zenmap.db` ⇒ Bu, "Son Taramalar Veritabanı" adlı bölümde açıklanacağı gibi son taramaların veritabanıdır. Halihazırda mevcut değilse oluşturulur.

`zenmap_version` ⇒ Bu dosya, bu kullanıcı başına yapılandırma dizinini oluşturmak için kullanılan Zenmap sürümünü içerir. Bir sürüm çıkışmasından şüpheleniyorsanız, bu dosyadaki sürüm numarasını sistem yapılandırma dizinindeki aynı adlı dosyayla karşılaştırmak yararlı olabilir. Zenmap ilk kez çalıştırıldığında sistem yapılandırmasından kopyalanır.

Output Files (Çıktı Dosyaları)

Bir tarama çalıştırıldığında, Zenmap Nmap'e XML çıktısını geçici bir dosyaya koymasını söyler, böylece Zenmap bunu ayırtılabilir. Normalde XML çıktı dosyası tarama bittiğinde silinir. Ancak, Zenmap komut satırında -oX veya -oA seçeneği varsa, XML çıktısı bunun yerine adlandırılmış dosyaya yazılır ve tarama tamamlandığında bu dosya silinmez. Başka bir deyişle, -oX ve -oA beklediğiniz şekilde çalışır. Zenmap bu seçenekler tarafından üretilen çıktı dosyalarını kullanmasa da -oG, -oN ve -oS de çalışır.

Zenmap'in bu dosya adlarını işlemesinde dikkat edilmesi gereken önemli bir nokta vardır. Yüzde karakterleri (%), strftime benzeri biçim belirticileri olarak yorumlanmalarını önlemek için öncelenir ("Çıktı Türünü Kontrol Etme" adlı bölüme bakın). Bunun nedeni Zenmap'in Nmap'in çıktı dosyası için tam olarak hangi ismi kullanacağını bilmesi gerektidir. Zenmap'te -oX scan-%T-%D.xml yazarsanız, çıktı dosyası scan-144840-121307.xml dosyasına değil scan-%T-%D.xml dosyasına kaydedilir veya Nmap'i doğrudan çalıştırıyor olsaydınız geçerli saat ve tarihe göre ne olacaksa o olur.

Description of zenmap.conf (zenmap.conf Açıklaması)

zenmap.conf Zenmap için kullanıcıya özel yapılandırma dosyasıdır. Kullanıcı başına yapılandırma dizininde bulunan düz bir metin dosyasıdır ("Kullanıcı Başına Yapılandırma Dosyaları" adlı bölüme bakın). Python ConfigParser sınıfı tarafından

tanınan sözdizimi Windows INI dosyalarınıninkine benzer. Bölümler köşeli parantez içindeki başlıklarla sınırlandırılmıştır. Bölümlerin içinde <name>=<value> veya <name>: <değer> çiftlerini içeren satırlar bulunur. Bir zenmap.conf dosyasından bir alıntı gösterilmektedir.

```
[output_highlight]
enable_highlight = True

[paths]
nmap_command_path = nmap
ndiff_command_path = ndiff

[search]
search_db = 1
file_extension = xml
store_results = 1
directory =
save_time = 60;days
```

Bu ayarlardan bazıları, yapılandırma dosyasını doğrudan düzenlemeden Zenmap içinden kontrol edilebilir.

Sections of [zenmap.conf](#) (zenmap.conf'un bölümleri)

Boolean değerleri True, true veya 1'den true'ya veya başka herhangi bir şeyden false'a normalize edilir.

[\[paths\]](#) ⇒ paths] bölümü Zenmap tarafından kullanılan önemli yolları tanımlar.

- [nmap_command_path](#) ⇒ Nmap çalıştırılabilir dosyasının yolu. Zenmap tarafından çalıştırılan bir komut satırındaki ilk kelime ne olursa olsun, bu değişkenin değeri ile değiştirilecektir. Varsayılan değeri olan nmap çoğu sistem için uygundur. Örnekler için "nmap Çalıştırılabilir" başlıklı bölüme bakın.
- [ndiff_command_path](#) ⇒ Ndifff tarama karşılaştırma yardımcı programının yolu. Zenmap tarama karşılaştırmaları yapmak için Ndifff kullanır; "Sonuçları Karşılaştırma" bölümüne bakın.

[search] ⇒ Arama] bölümü, arama aracının ("Kaydedilen Sonuçların Aranması" bölümüne bakın) nasıl davranışacağını tanımlar. Bu bölümdeki isimler, arama iletişim kutusunun "Arama seçenekleri" sekmesindeki seçeneklere karşılık gelir. Aşağıdaki isimler tanımlanmıştır.

- `directory` ⇒ Kayıtlı tarama sonuçları dosyalarının aranacağı dizin.
- `file_extension` ⇒ Aranacak dosya adı uzantılarının noktalı virgülle ayrılmış bir listesi.
- `search_db` ⇒ Son taramalar veritabanında arama yapılp yapılmayacağını kontrol eden bir Boolean.
- `store_results` ⇒ Tarama sonuçlarının son taramalar veritabanında saklanıp saklanmayacağı kontrol eden bir Boolean. "Son Taramalar Veritabanı" adlı bölüme bakın.
- `save_time` ⇒ Tarama sonuçlarının son taramalar veritabanında ne kadar süreyle tutulacağı. Bundan daha eski sonuçlar Zenmap kapatıldığında silinir. Biçim bir sayı ve noktalı virgülle ayrılmış bir zaman aralığıdır, örneğin 60;gün veya 1;yıl.

[diff] ⇒ diff] bölümü karşılaştırma aracının ("Sonuçları Karşılaştırma" bölümüne bakın) nasıl davranışlığını tanımlar. Aşağıdaki isimler tanımlanmıştır.

- `diff_mode` ⇒ Karşılaştırmaların varsayılan olarak grafik modunda mı yoksa metin modunda mı gösterileceğini kontrol eder. Grafik modu veya metin için karşılaştırma olmalıdır.
- `colored_diff` ⇒ Karşılaştırmalarda renk kullanılıp kullanılmayacağını kontrol eden bir Boolean.

[diff_colors] ⇒ diff_colors] bölümü karşılaştırma aracı tarafından kullanılan renkleri tanımlar. Şu isimler tanımlanmıştır: değişmemiş, eklenmiş, mevcut değil ve değiştirilmiş, bunların anımları "Sonuçların Karşılaştırılması" bölümünde tanımlanmıştır. Bunların her birinin değeri, [<kırmızı>, <yeşil>, <mavi>] biçiminde kırmızı, yeşil ve maviyi temsil eden 0-65535 aralığında üç tamsayıdan oluşan bir listedir. Örneğin, [65535, 0, 0] kırmızı rengi belirtir.

[output_highlight] ⇒ output_highlight] bölümü, True olduğunda çıktı vurgulamayı etkinleştirir ve False olduğunda devre dışı bırakır tek bir Boole değişkeni enable_highlight içerir.

[date_highlight] , [hostname_highlight] , [ip_highlight] , [port_list_highlight] , [open_port_highlight] , [closed_port_highlight] , [filtered_port_highlight] , [details_highlight] ⇒ Bu bölümlerin tümü, "Nmap Çıktısı" sekmesi adlı bölümde tartışılan Nmap çıktı vurgulamasının doğasını tanımlar. Bunlar en iyi Zenmap içinden düzenlenir. Bu bölümlerin her birinde aşağıdaki isimler tanımlanmıştır.

- **regex** ⇒ Çıktının ilgili bölümüyle eşleşen düzenli ifade.
- **bold** ⇒ Bu vurgunun kalın yapılip yapılmayacağını kontrol eden bir Boolean.
- **italic** ⇒ Bu vurgunun italik yapılip yapılmayacağını kontrol eden bir Boolean.
- **underline** ⇒ Bu vurgunun altının çizilip çizilmeyeceğini kontrol eden bir Boolean.
- **text** ⇒ Bu vurgulamadaki metnin rengi. Sözdizimi, [<kırmızı>, <yeşil>, <mavi>] biçiminde kırmızı, yeşil ve maviyi temsil eden 0-65535 aralığında üç tamsayıdan oluşan bir listedir. Örneğin, kırmızı vurgu için [65535, 0, 0].
- **highlight** ⇒ Bu vurgulamadaki arka planın rengi. Sözdizimi metin için olanla aynıdır.

Command-line Options (Komut Satırı Seçenekleri)

Grafiksel bir uygulama olan Zenmap'in işlevsellüğünün çoğu grafiksel arayüzü aracılığıyla ortaya çıkar. Zenmap'in komut satırı seçenekleri burada eksiksizlik için ve bazen yararlı oldukları için verilmiştir. Özellikle, zenmap <results file> komutunun Zenmap'i <results file> içindeki sonuçlar zaten açıkken başlattığını bilmek iyidir.

Synopsis (Özet)

`zenmap [<options>] [<results file>]`

Options Summary (Seçenek Özeti)

-f , **--file** <results file> ⇒ Verilen sonuç dosyasını görüntülemek için açar. Sonuç dosyası bir Nmap XML çıktı dosyası (.xml, nmap -oX tarafından üretildiği gibi) veya daha önce Zenmap tarafından kaydedilmiş bir dosya olabilir.

-h , **--help** ⇒ Bir yardım mesajı gösterin ve çıkış.

`--confdir <dir>` ⇒ Kullanıcı başına yapılandırma dizini olarak `<dir>` kullanın.

`-n , --nmap <Nmap command line>` ⇒ Verilen Nmap komutunu Zenmap arayüzü içinde çalıştırın. `n` veya `--nmap`'ten sonra, kalan her komut satırı argümanı yürütülecek komut satırı olarak okunur. Bu, `-n` veya `--nmap`'in diğer seçeneklerden sonra en son verilmesi gereki̇ği anlamına gelir. Komut satırının `nmap` çalıştırılabilir adını içermesi gerektiğini unutmayın: `zenmap -n nmap -sS target`.

`-p , --profile <profile>` ⇒ Verilen profil seçili olarak başlayın. Profil adı sadece bir dizedir: "Düzenli tarama". Eğer `-t` ile birlikte kullanılırsa, belirtilen hedefe karşı verilen profil ile bir tarama başlatır.

`-t , --target <target>` ⇒ Verilen hedefle başlayın. `p` ile birleştirilirse, belirtilen hedefe karşı verilen profille bir tarama başlatır.

`-v , --verbose` ⇒ Ayrıntı düzeyini artırın (Zenmap'in, Nmap'in değil). Bu seçenek, Zenmap'i başlatmak için kullanılan konsol penceresine yazdırılan daha fazla ayrıntı için birden çok kez verilebilir.

Error Output (Hata Çıkışı)

Zenmap çökerse, normalde bir yığın izi ile bir hata raporu göndermenize yardımcı olur. Otomatik çökme raporlamasını devre dışı bırakmak ve hataların konsola yazdırılmasını sağlamak için `ZENMAP_DEVELOPMENT` ortam değişkenini ayarlayın (değer önemli değildir). Kullanışlı bir hata ayıklama çıktısı almak için `ZENMAP_DEVELOPMENT=1 zenmap -v -v -v` Bash kabuk komutunu deneyin.

Windows'ta standart hata konsola yazdırılmak yerine `zenmap.exe` ile aynı dizinde bulunan `zenmap.exe.log` dosyasına yönlendirilir.

History (Geçmiş)

Zenmap ilk olarak 2005 ve 2006 yıllarında Google sponsorluğundaki Nmap Summer of Code sırasında oluşturulan bir Nmap GUI olan Umit'ten türetilmiştir. Umit'in birincil yazarı Adriano Monteiro Marques idi. Umit 2007 yılında modifiye edilip Nmap'e entegre edildi̇nde Zenmap olarak yeniden adlandırıldı.

Chapter 13. Nmap Output Formats (Bölüm 13. Nmap Çıktı Biçimleri)

- Introduction (Giriş)
- Command-line Flags (Komut Satırı Bayrakları)
 - Controlling Output Type (Çıktı Türünü Kontrol Etme)
 - Controlling Verbosity of Output (Çıktının Ayrıntılarını Kontrol Etme)
 - Enabling Debugging Output (Hata Ayıklama Çıktısını Etkinleştirme)
 - Enabling Packet Tracing (Paket İzlemeyi Etkinleştirme)
 - Resuming Aborted Scans (Durdurulan Taramaları Sürdürme)
- Interactive Output (Etkileşimli Çıktı)
- Normal Output (`-oN`) (Normal Çıktı (-oN))
- \$crlpT klldl3 OuTPut (`-oS`) (\$crlpT klldl3 OuTPut (-oS))
- XML Output (`-oX`) (XML Çıktısı (-oX))
 - Using XML Output (XML Çıktısını Kullanma)
- Manipulating XML Output with Perl (XML Çıktısını Perl ile Yönetme)
- Common Platform Enumeration (CPE) (Ortak Platform Numaralandırma (CPE))
 - Structure of a CPE Name (CPE Adının Yapısı)
- Output to a Database (Veritabanına Çıktı)
- Creating HTML Reports (HTML Raporları Oluşturma)
 - Saving a Permanent HTML Report (Kalıcı HTML Raporunu Kaydetme)
- Grepable Output (`-oG`) (Grepable Çıktısı (-oG))
 - Grepable Output Fields (Greplenebilir Çıktı Alanları)
 - `Host` field (Ana Bilgisayar alanı)
 - `Status` field (Durum alanı)
 - `Ports` field (Bağlantı Noktaları alanı)

- `Protocols` field (Protokoller alanı)
 - `Ignored State` field (Yoksayılan Durum alanı)
 - `OS` field (İşletim Sistemi alanı)
 - `Seq Index` field (Seq Dizini alanı)
 - `IP ID Seq` field (IP Kimliği Seq alanı)
- Parsing Grepable Output on the Command Line (Komut Satırında Grepable Çıktısını Ayırıştırma)

Introduction (Giriş)

Açık kaynak kodlu güvenlik araçlarıyla ilgili yaygın bir sorun, kafa karıştırıcı ve düzensiz çıktılardır. Bu araçlar genellikle çok sayıda alakasız hata ayıklama bilgisini satırlara dökmekte ve kullanıcıları önemli sonuçları gürültüden ayırt etmeye çalışırken sayfalarca çıktı aramaya zorlamaktadır. Program yazarları genellikle sonuçları etkili bir şekilde düzenlemek ve sunmak için çok az çaba harcarlar. Çıktı mesajlarının anlaşılması zor ve yetersiz belgelenmiş olabilir. Bu çok şaşırtıcı olmamalı - bazı TCP/IP zayıflıklarından yararlanmak için akıllıca kod yazmak genellikle dokümantasyon veya kullanıcı arayüzü çalışmalarından daha tatmin edicidir. Açık kaynak yazarlarına nadiren ödeme yapıldığından, zevk aldıkları şeyi yaparlar.

Arkadaşım Dan Kaminsky'yi güncendirmek pahasına, Scanrand port tarayıcısını, kullanıcı dostu bir kullanıcı arayüzünden çok daha fazla teknik hilelere vurgu yapılarak geliştirildiği açıkça belli olan bir program örneği olarak adlandıracığım. Örnek 13.1'deki örnek çıktı Scanrand dokümantasyon sayfasından alınmıştır.

Örnek 13.1. Yerel bir ağa karşı Scanrand çıktısı

```
bash-2.05a# scanrand 10.0.1.1-254:quick
UP: 10.0.1.38:80 [01] 0.003s
UP: 10.0.1.110:443 [01] 0.017s
UP: 10.0.1.254:443 [01] 0.021s
UP: 10.0.1.57:445 [01] 0.024s
UP: 10.0.1.59:445 [01] 0.024s
UP: 10.0.1.38:22 [01] 0.047s
UP: 10.0.1.110:22 [01] 0.058s
UP: 10.0.1.110:23 [01] 0.058s
UP: 10.0.1.254:22 [01] 0.077s
UP: 10.0.1.254:23 [01] 0.077s
UP: 10.0.1.25:135 [01] 0.088s
UP: 10.0.1.57:135 [01] 0.089s
UP: 10.0.1.59:135 [01] 0.090s
UP: 10.0.1.25:139 [01] 0.097s
UP: 10.0.1.27:139 [01] 0.098s
UP: 10.0.1.57:139 [01] 0.099s
UP: 10.0.1.59:139 [01] 0.099s
UP: 10.0.1.38:111 [01] 0.127s
UP: 10.0.1.57:1025 [01] 0.147s
UP: 10.0.1.59:1025 [01] 0.147s
UP: 10.0.1.57:5000 [01] 0.156s
UP: 10.0.1.59:5000 [01] 0.157s
UP: 10.0.1.53:111 [01] 0.182s
bash-2.05a#
```

Bu işi yapsa da yorumlaması zordur. Çıktı, bağlantı noktası numaralarını sıralamak veya hatta bir hedef ana bilgisayardaki tüm açık bağlantı noktalarını birlikte grüplamak için herhangi bir seçenek olmaksızın yanıtın alındığı zamana göre yazdırılır. Her satırın başında bir sürü boşluk boşça harcanır ve sonuçların özeti verilmez.

Nmap'in çıktısı da mükemmel olmaktan uzaktır, ancak okunabilir, iyi organize edilmiş ve esnek olması için oldukça çaba sarf ediyorum. Nmap'in insanlar ve diğer yazılımlar tarafından kullanıldığı yolların sayısı göz önüne alındığında, tek bir format herkesi memnun edemez. Bu nedenle Nmap, insanların doğrudan okuması için etkileşimli mod ve yazılım tarafından kolay ayrıştırma için XML dahil olmak üzere çeşitli formatlar sunar.

Farklı çıktı biçimleri sunmanın yanı sıra, Nmap çıktılarının ayrıntı düzeyini ve hata ayıklama mesajlarını kontrol etmek için seçenekler sunar. Çıktı türleri standart çıktıya ya da Nmap'in ekleyebileceği ya da saklayabileceği adlandırılmış dosyalara gönderilebilir. Çıktı dosyaları iptal edilen taramaları devam ettirmek için de

kullanılabilir. Bu bölüm, bu seçenekler ve her çıktı formatı hakkında tüm ayrıntıları içerir.

Command-line Flags (Komut Satırı Bayrakları)

Neredeyse tüm diğer Nmap özelliklerinde olduğu gibi, çıktı davranışını komut satırı bayrakları tarafından kontrol edilir. Bu bayraklar kategorilere göre gruplandırılmış ve aşağıdaki bölümlerde açıklanmıştır.

Controlling Output Type (Çıkış Tipini Kontrol Etme)

En temel çıktı kontrolü, istediğiniz çıktı biçim(ler)ini belirlemektir. Nmap, aşağıdaki listede özetlenen ve daha sonraki bölümlerde tam olarak açıklanan beş tür sunar.

Nmap tarafından desteklenen çıktı biçimleri

Etkileşimli çıktı ⇒ Bu, Nmap'in varsayılan olarak standart çıktı akışına (stdout) gönderdiği çıktıdır. Bu yüzden özel bir komut satırı seçeneği yoktur. Etkileşimli mod, sonuçları doğrudan okuyan insan kullanıcılarla hitap eder ve bu kitap boyunca dzinelerce örnekte gösterilen ilginç bağlantı noktalarından oluşan bir tablo ile karakterize edilir.

Normal çıkış (-oN) ⇒ Bu interaktif çıktıya çok benzer ve seçtiğiniz dosyaya gönderilir. Etkileşimli çıktıdan birkaç yönden farklıdır, bu da bu çıktıının etkileşimli olarak değil tarama tamamlandıktan sonra analiz edileceği bekentisinden kaynaklanır. Dolayısıyla etkileşimli çıktı, tarama tamamlanma süresi tahminleri ve açık bağlantı noktası uyarıları gibi mesajları (-v ile belirtilen ayrıntı düzeyine bağlı olarak) içerir. Normal çıktı, tarama tamamlandığında ve son ilginç bağlantı noktaları tablosu yazdırıldığında bunları gereksiz olarak atlar. Bu çıktı türü, kullanılan nmap komut satırını ve çalışma saatini ve tarihini ilk satırına yazdırır.

XML çıktısı (-oX) ⇒ XML, yazılım tarafından kolayca ayırtılabilen kararlı bir format sunar. C/C++, Perl, Python ve Java dahil olmak üzere tüm önemli bilgisayar dilleri için ücretsiz XML ayırtıcıları mevcuttur. Önemsiz olmayan bir uygulamanın Nmap ile arayüz oluşturduğu neredeyse tüm durumlarda XML tercih edilen formattır. Bu bölümde ayrıca XML sonuçlarının HTML raporları ve veritabanı tabloları gibi diğer biçimlere nasıl dönüştürülebileceği de tartışılmaktadır.

Greplenebilir çıktı (-oG) ⇒ Bu basit format, grep, awk, cut ve diff gibi basit Unix araçlarıyla komut satırında kolayca işlenebilir. Her ana bilgisayar tek bir satırda listelenir ve çıktı alanlarını sınırlamak için sekme, eğik çizgi ve virgül karakterleri kullanılır. Bu, sonuçları hızlı bir şekilde anlamak için kullanışlı olsa da, XML biçimini daha kararlı olduğundan ve daha fazla bilgi içerdiginden daha önemli görevler için tercih edilir.

sCRiPt KiDDi3 OutPU+ (-oS) ⇒ Bu format l33t haXXorZ için sağlanmıştır!

Etkileşimli çıktı varsayılandır ve ilişkili komut satırı seçenekleri yoktur, ancak diğer dört biçim seçenekleri aynı sözdizimini kullanır. Sonuçların saklanacağı dosya adı olan bir bağımsız değişken alırlar. Birden fazla format belirtilebilir, ancak her format yalnızca bir kez belirtilebilir. Örneğin, kendi incelemeniz için normal çıktıyı kaydederken, programatik analiz için aynı taramanın XML'ini kaydetmek isteyebilirsiniz. Bunu -oX myscan.xml -oN myscan.nmap seçenekleri ile yapabilirsiniz. Bu bölümde kısalık için myscan.xml gibi basit isimler kullanılsa da, genellikle daha açıklayıcı isimler önerilir. Seçilen isimler kişisel bir tercih meselesiştir, ancak ben tarama tarihini ve taramayı açıklayan bir iki kelimeyi içeren ve taradığım şirketin adını taşıyan bir dizine yerleştirilen uzun isimler kullanıyorum. Bir kolaylık olarak, tarama sonuçlarını normal, XML ve grepable formatlarında aynı anda saklamak için -oA <basename> belirtebilirsiniz. Bunlar sırasıyla <basename>.nmap, <basename>.xml ve <basename>.gnmap dosyalarında saklanır. Çoğu programda olduğu gibi, dosya adlarının önüne Unix'te ~/nmaplogs/foocorp/ veya Windows'ta c:\hacking\sco gibi bir dizin yolu ekleyebilirsiniz.

Bu seçenekler sonuçları dosyalara kaydederken, Nmap etkileşimli çıktıyı her zamanki gibi stdout'a yazdırılmaya devam eder. Örneğin, nmap -oX myscan.xml target komutu XML'i myscan.xml dosyasına yazdırır ve standart çıktıyı -oX belirtmemiş olsaydı yazdıracağı etkileşimli sonuçlarla doldurur. Biçim türlerinden birine argüman olarak bir tire karakteri geçirerek bunu değiştirebilirsiniz. Bu, Nmap'in etkileşimli çıktıyı devre dışı bırakmasına ve bunun yerine sonuçları standart çıktı akışına belirttiğiniz biçimde yazdırmasına neden olur. Yani nmap -oX -target komutu stdout'a sadece XML çıktısı gönderecektir. Ciddi hatalar yine de normal hata akışı olan stderr'ye yazdırılabilir.

oN gibi bir çıktı biçimini bayrağına bir dosya adı belirttiğinizde, varsayılan olarak bu dosyanın üzerine yazılır. Dosyanın mevcut içeriğini korumayı ve yeni sonuçları

eklemeyi tercih ederseniz, --append-output seçeneğini belirtin. Bu Nmap yürütmesinde belirtilen tüm çıktı dosya adları daha sonra clobbered yerine eklenecektir. Bu, XML (-oX) tarama verileri için iyi çalışmaz, çünkü sonuçta ortaya çıkan dosya, siz elle düzeltene kadar genellikle düzgün bir şekilde ayırtılmasız.

Bazı Nmap argümanlarından farklı olarak, logfile seçenek bayrağı (-oX gibi) ile dosya adı veya kısa çizgi arasındaki boşluk zorunludur. Bayrakları atlar ve -oG- veya -oXscan.xml gibi argümanlar verirseniz, Nmap'in geriye dönük uyumluluk özelliği sırasıyla G- ve Xscan.xml adlı normal formatlı çıktı dosyalarının oluşturulmasına neden olacaktır.

Bu argümanların tümü dosya adında strftime benzeri dönüşümleri destekler. H, %M, %S, %m, %d, %y ve %Y'nin tümü strftime'daki ile tamamen aynıdır. T, %H%M%S ile aynıdır, %R, %H%M ile aynıdır ve %D, %m%d%y ile aynıdır. Herhangi bir karakterin ardından gelen bir % sadece o karakteri verir (%% size bir yüzde sembolü verir). Yani -oX 'scan-%T-%D.xml', scan-144840-121307.xml şeklinde bir adı olan bir XML dosyası kullanacaktır.

Controlling Verbosity of Output (Çıktının Belirginliğini Kontrol Etme)

Sonuçların hangi format(lar)da kaydedilmesini istediğiniz karar verdikten sonra, bu sonuçların ne kadar ayrıntılı olması gereğine karar verebilirsiniz. İlk -v seçeneği bir seviyeli ayrıntı düzeyini etkinleştirir. Biraz daha büyük bir etki için -vv seçeneğini iki kez belirtin. İkiden büyük verbosity seviyeleri kullanışlı değildir. Coğu değişiklik yalnızca etkileşimli çıktıyı etkiler ve bazıları normal ve komut dosyası kiddie çıktısını da etkiler. Diğer çıktı türleri makineler tarafından işlemek üzere tasarlanmıştır, bu nedenle Nmap bir insan kullanıcıyı yormadan bu formatlarda varsayılan olarak önemli ayrıntılar verebilir. Bununla birlikte, diğer modlarda bazı ayrıntıların atlanmasıyla çıktı boyutunun önemli ölçüde azaltılabileceği birkaç değişiklik vardır. Örneğin, grepable çıktısında taranan tüm portların listesini veren bir yorum satırı oldukça uzun olabileceği için yalnızca ayrıntılı modda yazdırılır. Aşağıdaki listede en az bir -v seçeneği ile elde edeceğiniz önemli değişiklikler açıklanmaktadır.

Tarama tamamlanma süresi tahminleri ⇒ Bir ya da iki dakikadan uzun süren taramalarda, etkileşimli çıktı modunda ara sıra bunun gibi güncellemeler göreceksiniz:

```
SYN Stealth Scan Timing: About 30.01% done; ETC: 16:04 (0:01:09 remaining)
```

Tahminler önemli ölçüde değişirse yeni güncellemeler verilir. Boşta tarama ve FTP sıçrama taraması dışındaki tüm bağlantı noktası tarama teknikleri tamamlanma süresi tahminini destekler ve sürüm algılama, komut dosyası tarama ve traceroute gibi diğer aşamalar da öyle.

Keşfedildiğinde bildirilen açık portlar ⇒ Verbosity etkinleştirildiğinde, açık portlar keşfedildikçe etkileşimli modda yazdırılır. Bunlar hala son ilginç portlar tablosunda da rapor edilmektedir. Bu, kullanıcıların Nmap daha tamamlanmadan açık portları araştırmaya başlamasına olanak tanır. Açık port uyarıları şu şekilde görünür:

Discovered open port 53/tcp on 64.13.134.52

Ek uyarılar ⇒ Nmap her zaman bariz hatalar ve kritik sorunlar hakkında uyarılar yazdırır. Verbosity etkinleştirildiğinde bu standart düşürülür ve daha fazla uyarının yazdırılmasına izin verilir. Bu uyarılardan düzinece vardır ve aşırı düşme veya olağanüstü uzun gecikme yaşayan hedeflerden, problara beklenmedik şekillerde yanıt veren portlara kadar konuları kapsar. Hız sınırlaması bu uyarıların ekranı doldurmasını engeller.

Ek notlar ⇒ Nmap verbose modundayken birçok ekstra bilgi notu yazdırır. Örneğin, taranan ana bilgisayar ve bağlantı noktası sayısıyla birlikte her bağlantı noktası taramasının başlatıldığı zamanı yazdırır. Daha sonra, taramanın ne kadar sürdüğünü açıklayan ve sonuçları kısaca özetleyen bir sonuç satırı yazdırır.

Ekstra işletim sistemi algılama bilgileri ⇒ Verbosity ile TCP ISN ve IP ID sıra numarası tahmin edilebilirlik testlerinin sonuçları gösterilir. Bunlar işletim sistemi tespitinin bir yan ürünü olarak yapılır. Verbosity birden büyük olduğunda, gerçek işletim sistemi algılama parmak izi daha fazla durumda gösterilir.

Aşağı ana bilgisayarlar ping taramasında yazdırılır ⇒ Verbosity etkinleştirilmiş bir ping taraması sırasında, sadece yukarıdakiler yerine aşağıdakiler de yazdırılacaktır.

Doğum günü dilekleri ⇒ Nmap, 1 Eylül'de verbose modunda çalıştırıldığında kendisine mutlu yıllar diler.

Genellikle sadece Nmap raporunu bitirip yazdırana kadar faydalı olan değişiklikler sadece etkileşimli çıktı moduna gönderilir. Normal çıktıyı -oN ile bir dosyaya gönderirseniz, bu dosya açık port uyarıları veya tamamlanma süresi tahminleri içermez, ancak bunlar yine de stdout'a yazdırılır. Buradaki varsayımdır, Nmap'in işi bittiğinde dosyayı inceleyeceğiniz ve fazladan bir şey istemeyeceğinizdir, ancak Nmap'in yürütme ilerlemesini standart çıktıda izleyebilir ve çalışma zamanı

ilerlemesini önemseyebilirsiniz. Eğer gerçekten stdout'a yazdırılan her şeyin bir dosyaya gönderilmesini istiyorsanız, kabuğunuz tarafından sağlanan çıktı akışı yönlendirmesini kullanın (örn. nmap -v scanme.nmap.org > scanoutput.nmap).

Ayrıntıya bağlı düzinelere küçük değişiklik (çoğunlukla ekstra mesajlar) burada ele alınamayacak kadar çoktur. Ayrıca her zaman değişime tabidirler. Hepsini görmemin etkili bir yolu, en son Nmap tarball paketini açmak ve grep -A1 o.verbose *.cc gibi bir komutla bunları aramaktır. Çıktıdan temsili alıntılar Örnek 13.2'de gösterilmektedir.

Örnek 13.2. Verbosity koşulları için Grepping

```
output.cc:    if (o.verbose)
output.cc-        log_write(LOG_PLAIN, "Uptime guess: %.3f days (since %s)\n",
-- 
nmap.cc:  if (o.verbose)
nmap.cc-    output_ports_to_machine_parseable_output(&ports, o.TCPScan(),
                                                 o.UDPScan(), o.SCTPScan(), o.ipprotscan);
-- 
portlist.cc: if ((state == PORT_OPEN && o.verbose) || (o.debugging > 1)) {
portlist.cc-    log_write(LOG_STDOUT, "Discovered %s port %hu/%s%s\n",
-- 
scan_engine.cc:   if (o.verbose && hss->sdn.delayms != olddelay)
scan_engine.cc-    log_write(LOG_PLAIN, "Increasing send delay for %s..."
```

Aşağıdaki iki örnek tüm bunları bir araya getirmektedir. Örnek 13.3 -v seçeneği olmadan normal bir taramanın çıktısını göstermektedir.

Örnek 13.3. Verbosity etkinleştirilmeden etkileşimli çıktı

```

# nmap -T4 -A scanme.nmap.org

Starting Nmap ( https://nmap.org )
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.045s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey: 1024 60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)
| 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)
25/tcp    closed smtp
53/tcp    open  domain
70/tcp    closed gopher
80/tcp    open  http   Apache httpd 2.2.3 ((CentOS))
|_html-title: Go ahead and ScanMe!
| http-methods: Potentially risky methods: TRACE
| See https://nmap.org/nsedoc/scripts/http-methods.html
113/tcp   closed auth
31337/tcp closed Elite
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.31, Linux 2.6.18
Network Distance: 13 hops

TRACEROUTE (using port 25/tcp)
HOP RTT      ADDRESS
[Cut first 10 hops for brevity]
1  44.63 ms layer42.car2.sanjose2.level3.net (4.59.4.78)
12  44.33 ms xe6-2.core1.svk.layer42.net (69.36.239.221)
13  44.59 ms scanme.nmap.org (64.13.134.52)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.06 seconds

```

Örnek 13.4 aynı taramanın verbosity etkinleştirilmiş çıktısıdır. Ekstra işletim sistemi tanımlama verileri, tamamlanma süresi tahminleri, açık port uyarıları ve ekstra bilgi mesajları gibi özellikler ikinci çıktıda kolayca tanımlanabilir. Bu ekstra bilgiler genellikle etkileşimli tarama sırasında yardımcı olur, bu nedenle iyi bir nedenim olmadığı sürece tek bir makineyi tararken her zaman -v seçeneğini belirtirim.

Örnek 13.4. Verbosity etkinleştirilmiş etkileşimli çıktı

```

# nmap -v -T4 -A scanme.nmap.org

Starting Nmap ( https://nmap.org )
NSE: Loaded 49 scripts for scanning.
Initiating Ping Scan at 15:08
Scanning scanme.nmap.org (64.13.134.52) [4 ports]
Completed Ping Scan at 15:08, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:08
Completed Parallel DNS resolution of 1 host. at 15:08, 0.00s elapsed
Initiating SYN Stealth Scan at 15:08
Scanning scanme.nmap.org (64.13.134.52) [1000 ports]
Discovered open port 22/tcp on 64.13.134.52
Discovered open port 80/tcp on 64.13.134.52
Discovered open port 53/tcp on 64.13.134.52
Completed SYN Stealth Scan at 15:08, 4.77s elapsed (1000 total ports)
Initiating Service scan at 15:08
Scanning 3 services on scanme.nmap.org (64.13.134.52)
Completed Service scan at 15:08, 11.13s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against scanme.nmap.org (64.13.134.52)
Initiating Traceroute at 15:08
Completed Traceroute at 15:08, 0.06s elapsed
Initiating Parallel DNS resolution of 13 hosts. at 15:08
Completed Parallel DNS resolution of 13 hosts. at 15:08, 0.00s elapsed
NSE: Script scanning 64.13.134.52.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 15:08
Completed NSE at 15:08, 4.11s elapsed
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.044s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh   OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey: 1024 60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)
| 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)
25/tcp    closed  smtp
53/tcp    open   domain
70/tcp    closed  gopher
80/tcp    open   http   Apache httpd 2.2.3 ((CentOS))
| http-methods: GET HEAD POST OPTIONS TRACE
| Potentially risky methods: TRACE
|_ See https://nmap.org/nsedoc/scripts/http-methods.html
|_html-title: Go ahead and ScanMe!
113/tcp   closed auth
31337/tcp closed Elite
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.31, Linux 2.6.18
Uptime guess: 23.640 days (since Thu Jun 24 23:46:34 2010)
Network Distance: 13 hops
TCP Sequence Prediction: Difficulty=206 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
[Cut first 10 hops for brevity]
11  44.09 ms layer42.car2.sanjose2.level3.net (4.59.4.78)
12  43.98 ms xe6-2.core1.svk.layer42.net (69.36.239.221)
13  44.73 ms scanme.nmap.org (64.13.134.52)

Read data files from: .
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.28 seconds
Raw packets sent: 2040 (91.266KB) | Rcvd: 40 (2.278KB)

```

Enabling Debugging Output (Hata Ayıklama Çıktısını Etkinleştirme)

Verbose modu bile sizin için yeterli veri sağlamadığında, hata ayıklama sizi çok daha fazlaıyla doldurmak için kullanılır! Verbosity seçeneğinde (-v) olduğu gibi, hata ayıklama bir komut satırı bayrağı (-d) ile etkinleştirilir ve hata ayıklama seviyesi birden fazla kez belirtilerek artırılabilir. Alternatif olarak, -d'ye bir argüman vererek bir hata ayıklama seviyesi ayarlayabilirsiniz. Örneğin, -d9 dokuzuncu

seviyeyi ayarlar. Bu en yüksek etkili seviyedir ve çok az sayıda bağlantı noktası ve hedefle çok basit bir tarama yapmadığınız sürece binlerce satır üretecektir.

Hata ayıklama çıktısı, Nmap'te bir hatadan şüphelenildiğinde veya Nmap'in ne yaptığı ve neden yaptığı konusunda kafanız karıştıığında kullanılabilir. Bu özellik çoğunlukla geliştiriciler için tasarlandığından, hata ayıklama satırları her zaman kendi kendini açıklayıcı değildir. Bir satırı anlamadıysanız, tek çareniz onu görmezden gelmek, kaynak koduna bakmak veya geliştirme listesinden (nmap-dev) yardım istemektir. Bazı satırlar kendi kendini açıklar, ancak hata ayıklama seviyesi arttıkça mesajlar daha belirsiz hale gelir. Örnek 13.5, Scanme'nin -d5 taramasından kaynaklanan birkaç farklı hata ayıklama satırını göstermektedir.

Örnek 13.5. Bazı temsili hata ayıklama satırları

```
Timeout vals: srtt: 27495 rttvar: 27495 to: 137475 delta -2753
          ==> srtt: 27150 rttvar: 21309 to: 112386
RCVD (15.3330s) TCP 64.13.134.52:25 > 132.239.1.115:50122 RA ttl=52
          id=0 iplen=40 seq=0 win=0 ack=4222318673
**TIMING STATS** (15.3350s): IP, probes active/freshportsleft/retry_stack/
                                outstanding/retranwait/onbench,
                                cwnd/ccthresh/delay, timeout/srtt/rttvar/
Groupstats (1/1 incomplete): 83/*/*/*/* 82.80/75/* 100000/25254/4606
  64.13.134.52: 83/60836/0/777/316/4295 82.80/75/0 100000/26200/4223
Current sending rates: 711.88 packets / s, 31322.57 bytes / s.
Overall sending rates: 618.24 packets / s, 27202.62 bytes / s.
Discovered filtered port 10752/tcp on 64.13.134.52
Packet capture filter (device eth0): dst host 132.239.1.115 and
                                      (icmp or ((tcp or udp) and
                                      (src host 64.13.134.52)))
SCRIPT ENGINE: TCP 132.239.1.115:59045 > 64.13.134.52:53 | CLOSE
```

Hata ayıklama günlükleri çok uzun olduğu için burada tam bir örnek verilmemiştir. Scanme'ye karşı yapılan bir taramada verbosity olmadan 40 satır (Örnek 13.3, "Verbosity etkin değilken etkileşimli çıktı") ve verbosity ile 40 satır (Örnek 13.4, "Verbosity etkinken etkileşimli çıktı") metin kullanılmıştır. Aynı tarama -v yerine -d ile 136 satır sürdü. d2 ile 1,324 satır, -d5 ile ise 6,391 satırda çıkmıştır! Hata ayıklama seçeneği dolaylı olarak ayrıntı düzeyini etkinleştirir, bu nedenle her ikisini de belirtmeye gerek yoktur.

Belirli bir hata ayıklama görevi için en iyi çıkış seviyesini belirlemek bir deneme yanlışına meselesidir. Neler olup bittiğini anlamak için önce düşük bir seviye denerim, ardından gerekiğinde artırırım. Daha fazla şey öğrendikçe, sorunu veya soruyu daha iyi izole edebilirim. Daha sonra, daha yüksek hata ayıklama

seviyesinin bazı artan laf kalabalığını dengelemek için komutu basitleştirmeye çalışıyorum.

Tıpkı grep'in verbosity ile ilişkili değişiklikleri ve seviyeleri tanımlamak için yararlı olabileceği gibi, hata ayıklama çıktısını araştırmaya da yardımcı olur. Bu komutu Nmap kaynak tarball'undaki nmap-<VERSION> dizininden çalıştırmanızı öneririm:

grep -A1 o.debugging *.cc

Enabling Packet Tracing (Paket İzlemeyi Etkinleştirme)

--packet-trace seçeneği Nmap'in gönderdiği ve aldığı her paketin bir özetini yazdırmasına neden olur. Bu, bu kitap boyunca örneklerin gösterdiği gibi, hata ayıklama veya Nmap'in davranışını anlamak için son derece yararlı olabilir. Örnek 13.6, paket izleme etkinleştirilmiş halde Scanme'nin basit bir ping taramasını göstermektedir.

Örnek 13.6. Scanme'nin ping taramasını detaylandırmak için --packet-trace kullanımı

```
# nmap --packet-trace -n -sn scanme.nmap.org
Starting Nmap 5.35DC18 ( https://nmap.org ) at 2010-07-18 15:23 MDT
SENT (0.0130s) ICMP 132.239.1.115 > 64.13.134.52 Echo request (type=8/code=0) ttl=53 id=43882 iplen=28
SENT (0.0130s) TCP 132.239.1.115:39273 > 64.13.134.52:443 S ttl=44 id=18217 iplen=44 seq=215684135 win=1024 <mss 1460>
SENT (0.0130s) TCP 132.239.1.115:39273 > 64.13.134.52:80 A ttl=52 id=37510 iplen=40 seq=0 win=1024
SENT (0.0130s) ICMP 132.239.1.115 > 64.13.134.52 Timestamp request (type=13/code=0) ttl=52 id=54744 iplen=40
RCVD (0.0570s) TCP 64.13.134.52:80 > 132.239.1.115:39273 R ttl=56 id=0 iplen=40 seq=215684135 win=0
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.044s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Burada, "Varsayılan Kombinasyon" adlı bölümdeki varsayılan dört problk ana bilgisayar keşif kombinasyonunu görebilirsiniz. Bu çıktı, paket izlemenin neden olduğu üç beş ekstra satırı göstermektedir (her biri okunabilirlik için sarılmıştır). Her satır birkaç alan içerir. İlkı, SENT ve RCVD olarak kısaltılan bir paketin Nmap tarafından gönderilip gönderilmediği veya alınıp alınmadığıdır. Bir sonraki alan, Nmap'in başlamasından bu yana geçen süreyi sağlayan bir zaman sayacıdır. Zaman saniye cinsindendir ve bu durumda Nmap yalnızca bir saniyenin küçük bir kısmını gerektirmiştir. Bir sonraki alan protokoldür: TCP, UDP veya ICMP. Ardından, yönlü bir okla ayrılmış kaynak ve hedef IP adresleri gelir. TCP veya UDP paketleri için, her IP'yi iki nokta üst üste ve kaynak veya hedef port numarası izler.

Her satırın geri kalanı protokole özeldir. Gördüğünüz gibi ICMP, varsa insan tarafından okunabilir bir tür (bu durumda Echo isteği) ve ardından ICMP türü ve

kod değerleri sağlar. ICMP paket günlükleri IP TTL, ID ve paket uzunluğu alanıyla sona erer. TCP paketleri hedef IP ve port numarasından sonra biraz farklı bir format kullanır. İlk olarak ayarlanmış TCP bayraklarını temsil eden karakterlerin bir listesi gelir. Bayrak karakterleri sırasıyla SYN, ACK, FIN, RST, PSH, URG, ECE ve CWR anlamına gelen SAFRPUEC'dir. Son iki bayrak RFC 3168'de açıklanan TCP açık tıkanıklık bildiriminin bir parçasıdır.

Paket izleme binlerce çıktı satırına yol açabileceğinden, tarama yoğunluğunu hala amacınıza hizmet eden minimum değerle sınırlamak yardımcı olur. Tek bir makinede tek bir bağlantı noktasının taranması sizi veriye boğmazken, tüm bir ağın --packet-izleme taramasının çıktısı bunaltıcı olabilir. Hata ayıklama seviyesi (-d) en az üç olduğunda paket izleme otomatik olarak etkinleştirilir.

Bazen --packet-trace, TTL'ler gibi Nmap'in başka türlü asla göstermediği özel veriler sağlar. Örnek 13.6, "Scanme'nin ping taramasını detaylandırmak için --packet-trace kullanımı" hedef ana bilgisayara gönderilen ICMP ve TCP ping paketlerini gösterir, hedef TCP ACK paketine yanıt verir. Hedef ana bilgisayarın diğer problara da yanıt vermiş olması mümkündür-Nmap, bir ping taramasına bir yanıt aldığında dinlemeyi durdurur, çünkü bir ana bilgisayarın çevrimiçi olduğunu belirlemek için gereken tek şey budur.

Resuming Aborted Scans (İptal Edilen Taramaları Devam Ettirme)

Bazı kapsamlı Nmap çalışmaları çok uzun zaman alır - günlerce. Bu tür taramalar her zaman tamamlanmayabilir. Kısıtlamalar Nmap'in çalışma saatleri içinde çalıştırılmasını engelleyebilir, ağ çökebilir, Nmap'in üzerinde çalıştığı makine planlı ya da plansız bir şekilde yeniden başlatılabilir ya da Nmap'in kendisi çökebilir. Nmap'i çalıştırın yönetici, ctrl-C tuşuna basarak başka herhangi bir nedenle de iptal edebilir. Tüm taramayı baştan başlatmak istenmeyebilir. Neyse ki, normal (-oN) veya grepable (-oG) günlükler tutulduysa, kullanıcı Nmap'ten yürütme durduğunda üzerinde çalıştığı hedefle taramaya devam etmesini isteyebilir. Devam et seçeneğini belirtin ve normal/greplenebilir çıktı dosyasını argüman olarak iletin. Nmap daha önce belirtilenleri kullanmak için çıktı dosyasını ayırtıldığından başka argümanlara izin vermez. Nmap'i nmap --resume <logfilename> şeklinde çağrımanız yeterlidir. Nmap yeni sonuçları önceki yürütmede belirtilen veri dosyalarına ekleyecektir.

Bu özelliğin bazı sınırlamaları vardır. İki çalıştırmayı tek bir geçerli XML dosyasında birleştirmek zor olacağından, Yeniden Başlatma XML çıktı biçimini desteklemez. Bu

özellik yalnızca tüm taramaları tamamlanmış ana bilgisayarları atlar. Nmap durdurulduğunda belirli bir hedefe karşı bir tarama devam ediyorsa, --resume bu konağın taranmasını baştan başlatabilir.

Interactive Output (İnteraktif Çıktı)

Etkileşimli çıktı, Nmap'in stdout akışına yazdırıldığı şeydir ve genellikle Nmap'i çalıştırığınız terminal penceresinde görünür. Diğer durumlarda, stdout'u bir dosyaya yönlendirmiş olabilirsiniz ya da Nessus veya Nmap GUI gibi başka bir uygulama sonuçları okuyor olabilir. Nmap çıktısını doğrudan kullanıcıya yazdırmak yerine daha büyük bir uygulama sonuçları yorumluyorsa, "XML Çıktısı (-oX)" adlı bölümde tartışılan XML çıktısını kullanmak daha uygun olacaktır.

Bu formatın tek bir amacı vardır: bunları okuyan bir insan için değerli olacak sonuçları sunmak. Bunları kolayca makinede ayırtılabilir hale getirmek veya Nmap sürümleri arasında istikrarlı bir format sağlamak için hiçbir çaba gösterilmemiştir. Bunlar için daha iyi formatlar mevcuttur. En zorlu görev, hangi bilginin yazdırılacak kadar değerli olduğuna karar vermektir. Kullanıcının istediği verileri atlamak utanç vericidir, ancak kullanıcıyı çoğunuyla alakasız çıktı sayfalarıyla doldurmak daha da kötü olabilir. Ayrıntı, hata ayıklama ve paket izleme bayrakları, bu dengeyi bireysel kullanıcıların tercihlerine göre değiştirmek için kullanılabilir.

Bu çıktı biçimini, bu kitaptaki çoğu Nmap örneğinde zaten gösterildiği için burada kapsamlı bir açıklamaya gerek yoktur. Belirli bir özellik için Nmap'in etkileşimli çıktısını anlamak için, bu kitabın o özelliğe ayrılmış bölümune bakın. Etkileşimli çıktılarının tipik örnekleri Örnek 13.3, "Verbosity etkin olmadan etkileşimli çıktı" ve Örnek 13.4, "Verbosity etkin olarak etkileşimli çıktı"'da verilmiştir.

Normal Output (-oN) (Normal Çıkış (-oN))

Normal çıktı, -oN seçeneği bir dosya adı argümanıyla belirtildiğinde bir dosyaya yazdırılır. Tarama tamamlandığında geçerliliğini yitiren notların kaldırılması dışında

etkileşimli çıktıya benzer. Dosyanın Nmap tamamlandıktan sonra okunacağı varsayıılır, bu nedenle tahmini tamamlanma süreleri ve yeni açık bağlantı noktası uyarıları gerçek tamamlanma süresi ve sıralı bağlantı noktası tablosu için gereksizdir. Çıktı uzun bir süre kaydedileceğinden ve diğer birçok günlük arasında incelenebileceğinden, Nmap ilk satırda yürütme süresini, komut satırı argümanlarını ve Nmap sürüm numarasını yazdırır. Bir taramanın sonundaki benzer bir satır, son zamanlamayı ve bir ana bilgisayar sayısını açıklar. Bu iki satır, yorum olarak tanımlamak için bir pound karakteri ile başlar. Uygulamanızın XML/greplenebilir biçimler yerine normal çıktıları ayırtılması gerekiyorsa, tanımadığı yorumları hata olarak değerlendirip iptal etmek yerine yok sayıldığından emin olun. Örnek 13.7 tipik bir normal çıktı örneğidir. Etkileşimli çıktıyı önlemek ve normal çıktıyı doğrudan stdout'a göndermek için -oN - kullanıldığına dikkat edin.

Örnek 13.7. Tipik bir normal çıktı örneği

```
# nmap -T4 -A -p- -oN - scanme.nmap.org
# Nmap 5.35DC18 scan initiated Sun Jul 18 15:33:26 2010 as: ./nmap -T4 -A -oN - scanme.nmap.org
Nmap scan report for scanme.nmap.org (64.13.134.52)
Host is up (0.045s latency).
Not shown: 993 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 4.3 (protocol 2.0)
| ssh-hostkey: 1024 60:ac:4d:51:bl:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)
|_ 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)
25/tcp    closed  smtp
53/tcp    open   domain
70/tcp    closed  gopher
80/tcp    open   http   Apache httpd 2.2.3 ((CentOS))
| http-methods: Potentially risky methods: TRACE
| See https://nmap.org/nsedoc/scripts/http-methods.html
|_ html-title: Go ahead and ScanMe!
113/tcp   closed auth
31337/tcp closed Elite
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.31, Linux 2.6.18
Network Distance: 13 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
[Cut first 10 hops for brevity]
11 45.16 ms  layer42.car2.sanjose2.level3.net (4.59.4.78)
12 43.97 ms  xe6-2.core1.svk.layer42.net (69.36.239.221)
13 45.15 ms  scanme.nmap.org (64.13.134.52)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jul 18 15:33:48 2010 -- 1 IP address (1 host up) scanned in 22.47 seconds
```

\$crIpt kIddI3 OuTPut (-oS) (\$crIpt kIddI3 OuTPut (-oS))

Script kiddie çıktısı interaktif çıktı gibidir, ancak 'I33t HaXXorZ'a daha uygun olması için sonradan işlenmiştir! Daha önce tutarlı büyük harf kullanımı ve yazımı nedeniyle Nmap'i küçümsüyorlardı. Örnek 13.8'de verildiği gibi, en iyi örnekle anlaşılabilir.

Örnek 13.8. Tipik bir \$crlpt KiDDi3 OutPut örneği

```
# nmap -T4 -A -oS - scanme.nmap.org

Starting Nmap 5.35dC18 ( http://Nmap.org ) at 2010-07-18 15:36 MDT
Nmap $caN r3p0rT f0R sCAnm3.nMAp.0rg (64.13.134.52)
Host is up (0.044z lat3Ncy).
n0t $h0wn: 993 filter3d p0rtS
PORT      sTATe   $erV|CE v3R$|oN
22/tcp    open     $$h     0pEn$$h 4.3 (pRotocol 2.0)
|_ $H-h0stkEy: 1024 60:ac:4D:51:b1:cD:85:09:12:16:92:76:1d:5D:27:6e (D$4)
|_ 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (R$4)
25/tcp    c10$ed sMTp
53/Tcp    op3n   domain
70/tcp    cl0s3d G0pher
80/tcP    0Pen   httP   ApAchE httpd 2.2.3 ((C3nt0z))
|_ HTml-Titl3: Go aHEad And $canm3!
|_ Http-m3ThodS: P0t3nTiAlly ri$ky m3tHodS: TRACE
|_ seE https://nmap.0rg/nSeD0c/$cr|ptS/http-m3th0Ds.html
113/tcp   c10$ed AUth
31337/Tcp CL0$3d 3l!tE
Dev!Ce tYpE: Gen3rAl Purp0$3
Runn1Ng: L|nUX 2.6.X
Oz d3tAllz: l|nux 2.6.13 - 2.6.31
[Many lines cut for brevity]
Nmap Done: 1 IP addRe$z (1 h0$t up) $cann3d |n 22.50 sec0nd$
```

Bazı mizah özürlü insanlar bu seçeneği çok ciddiye alıyor ve beni script çocuklarına hizmet ettiğim için azarlıyorlar. Bu sadece script çocuklarıyla dalga geçmek için yapılmış bir şakadır - onlar bu modu gerçekten kullanmazlar (umarım).

XML Output (-oX) (XML Çıktısı (-oX))

XML, genişletilebilir işaretleme dili, eleştirmenlerin yanı sıra çok sayıda gayretli savunucuya da sahiptir. Ben uzun süre ilk grupta yer aldım ve işin çoğunu gönüllüler yaptıktan sonra XML'i Nmap'e istemeye dahil ettim. O

zamandan beri XML'in sunduğu gücü ve esnekliği takdir etmeyi öğrendim ve hatta bu kitabı DocBook XML formatında yazdım. Programcıların normal, etkileşimli veya grepable çıktıyı ayırtırmaya çalışmak yerine XML arayüzü üzerinden Nmap ile etkileşime girmelerini şiddetle tavsiye ederim. XML formatı diğerlerinden daha fazla bilgi içerir ve onu kullanan mevcut programları bozmadan yeni özellikler eklenebilecek kadar genişletilebilir. Genellikle ücretsiz olarak tüm popüler programlama dilleri için mevcut olan standart XML ayırtırıcıları tarafından ayırtılabilir. Editörler, doğrulayıcılar, dönüştürme sistemleri ve diğer birçok uygulama bu biçimini nasıl kullanacağını zaten biliyor. Öte yandan normal ve etkileşimli çıktılar Nmap'e özeldir ve son kullanıcılar daha net bir sunum için çabalarken düzenli değişikliklere tabidir. Grepable çıktı da Nmap'e özgüdür ve genişletilmesi XML'den daha zordur. Kullanımdan kaldırılmış olarak kabul edilir ve MAC adresi algılama gibi birçok Nmap özelliği bu çıktı biçiminde sunulmaz.

Nmap XML çıktısının bir örneği Örnek 13.9'da gösterilmektedir. Beyaz boşluklar okunabilirlik için ayarlanmıştır. Bu durumda, XML -oX - yapısı sayesinde stdout'a gönderilmiştir. Nmap çalıştırılan bazı programlar çıktıyu bu şekilde okumayı tercih ederken, diğerleri çıktıının bir dosya adına gönderilmesini ve Nmap tamamlandıktan sonra bu dosyanın okunmasını belirtir.

Örnek 13.9. Nmap XML çıktısına bir örnek

```

# nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org
<?xml version="1.0"?>
<!DOCTYPE nmaprun [
    <!-- Nmap 5.59BETA3 scan initiated Fri Sep  9 18:33:41 2011 as:
        nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org -->
]>
<nmaprun scanner="nmap" args="nmap -T4 -A -p 1-1000 -oX - scanme.nmap.org" start="1315618421"
    starttime="Fri Sep  9 18:33:41 2011" version="5.59BETA3" xmloutputversion="1.83">
<scaninfo type="syn" protocol="tcp" numservices="1000" services="1-1000"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1315618421" endtime="1315618434">
    <status state="up" reason="echo-reply"/>
    <address addr="74.207.244.221" addrtype="ipv4"/>
    <hostnames>
        <hostname name="scanme.nmap.org" type="user"/>
        <hostname name="l180-221.members.linode.com" type="PTR"/>
    </hostnames>
    <ports>
        <extrports state="closed" count="997">
            <extrareasons reason="resets" count="997"/>
        </extrports>
        <port protocol="tcp" portid="22">
            <state state="open" reason="syn-ack" reason_ttl="53"/>
            <service name="ssh" product="OpenSSH" version="5.3pl1 Debian 3ubuntu7"
                extrainfo="protocol 2.0 ostype='Linux' method='probed' conf='10'"
                cpe=cpe:/a:openbsd:openssh:5.3pl1</cpe>
                cpe=cpe:/o:linux:kernel</cpe>
            </service>
            <script id="ssh-hostkey" output="1024 8d:08:f1:7c:c8:b7:3d:8a:d0:67:54:9d:69:d9:dd (DSA)&#xa;
                2048 79:f8:09:ac:d4:e2:32:42:18:49:d3:bd:20:82:85:ec (RSA)"/>
        </port>
        <port protocol="tcp" portid="80">
            <state state="open" reason="syn-ack" reason_ttl="53"/>
            <service name="http" product="Apache httpd" version="2.2.14"
                extrainfo="(Ubuntu)" method="probed" conf='10'
                cpe=cpe:/a:apache:http_server:2.2.14</cpe>
            </service>
            <script id="http-title" output="Go ahead and ScanMe!"/>
        </port>
    </ports>
    <os>
        <portused state="open" proto="tcp" portid="22"/>
        <portused state="closed" proto="tcp" portid="1"/>
        <portused state="closed" proto="udp" portid="31289"/>
        <osclass type="general purpose" vendor="Linux" osfamily="Linux"
            osgen="2.0.X" accuracy="100">
            <cpe:cpe:/o:linux:linux_kernel:2.6.39</cpe>
        </osclass>
        <osmatch name="Linux 2.6.39" accuracy="100" line="39278"/>
    </os>
    <uptime seconds="23458" lastboot="Fri Sep  9 12:03:04 2011"/>
    <distance value="11"/>
    <tcpsequence index="199" difficulty="Good luck!">
        values="49018209,48C3EBED,495A2E7F,493EF38C,48ED43B3,495A9B0C"/>
    <ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
    <tcptssequence class="1000HZ">
        values="105CC09,105CC0E,105CC02,105CD30,105CD9A,105CE48"/>
    <trace port="256" proto="tcp">
        <!-- Several hop elements removed for brevity ...>
        <hop ttl="9" ipaddr="72.52.92.109" rtt="15.09" host="10gigabitethernet1-1.core1.fmt1.he.net"/>
        <hop ttl="10" ipaddr="64.02.250.6" rtt="12.86" host="linode-llc.10gigabitethernet2-3.core1.fmt1.he.net"/>
        <hop ttl="11" ipaddr="74.207.244.221" rtt="10.35" host="l180-221.members.linode.com"/>
    </trace>
    <times srtt="20517" rttdvar="19989" to="186473"/>
</host>
<runstats>
    <finished time="1315618434" timestamp="Fri Sep  9 18:33:54 2011" elapsed="13.66"
        summary="Nmap done at Fri Sep  9 18:33:54 2011; 1 IP address (1 host up)
                    scanned in 13.66 seconds' exit="success"/>
    <hosts up="1" down="0" total="1" />
</runstats>
</nmaprun>

```

XML'in bir diğer avantajı da ayrıntılı yapısının diğer formatlara göre okunmasını ve anlaşılması kolaylaştırmasıdır. Genel olarak Nmap'e aşina olan okuyucular, Örnek 13.9, "Nmap XML çıktısı örneği "ndeki XML çıktısının çogunu daha fazla dokümantasyon olmadan anlayabilirler. Öte yandan, greplenebilir çıktı biçiminin kendi başvuru kılavuzu olmadan deşifre edilmesi zordur.

Örnek XML çıktısının kendi kendini açıklayamayan birkaç yönü vardır. Örneğin, Örnek 13.10'daki iki bağlantı noktası ögesine bakın

Örnek 13.10. Nmap XML bağlantı noktası öğeleri

```
<port protocol="tcp" portid="22">
  <state state="open" reason="syn-ack" reason_ttl="56"/>
  <service name="ssh" product="OpenSSH" version="4.3" extrainfo="protocol 2.0"
    method="probed" conf="10"/>
  <script id="ssh-hostkey">
    output="1024 60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA)&#xa;
           2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)"/>
  </script>
</port>
<port protocol="tcp" portid="113">
  <state state="closed" reason="reset" reason_ttl="56"/>
  <service name="auth" method="table" conf="3"/>
</port>
```

Bağlantı noktası protokolü, kimliği (bağlantı noktası numarası), durumu ve hizmet adı, etkileşimli çıkış bağlantı noktası tablosunda gösterilecek olanlarla aynıdır. Hizmet ürünü, sürüm ve extrainfo öznitelikleri sürüm algılamadan gelir ve etkileşimli çıktı bağlantı noktası tablosunun bir alanında birleştirilir. Method ve conf öznitelikleri diğer çıktı türlerinde mevcut değildir. Yöntem tablo olabilir, yani hizmet adı bağlantı noktası numarası ve protokole dayalı olarak nmap-services'te basitçe aranmıştır veya problanmış olabilir, yani sürüm algılama sistemi aracılığıyla belirlenmiştir. conf özniteliği Nmap'in hizmet adının doğru olduğuna dair güvenini ölçer. Değerler bir (en az emin) ile on arasında değişir. Nmap, tablo aramasıyla belirlenen bağlantı noktaları için yalnızca 3 güven seviyesine sahipken, Örnek 13.10, "Nmap XML bağlantı noktası öğeleri "ndeki 22 numaralı bağlantı noktasının OpenSSH olduğundan son derece emindir (seviye 10), çünkü Nmap bağlantı noktasına bağlanmış ve OpenSSH olarak tanımlanan bir SSH sunucusu bulmuştur.

Bazı kullanıcıların kafa karıştırıcı bulduğu bir diğer husus da /nmaprun/@start ve /nmaprun/runstats/finished/@time niteliklerinin 1 Ocak 1970'ten bu yana geçen saniye sayısı olan Unix zamanıyla verilen zaman damgalarını tutmasıdır. Bu genellikle programların işlemesi için daha kolaydır. İnsan okuyuculara kolaylık sağlamak için, 3.78 ve daha yeni sürümler /nmaprun/@startstr ve /nmaprun/runstats/finished/@endstr niteliklerinde yazılı eşdeğer takvim zamanını içerir.

Orijinal komut satırı (argv dizisi) /nmaprun/@args niteliğinde saklanır. Bağımsız değişkenler boşluk ile ayrılır. Başlangıçta boşluk içeren bağımsız değişkenler çift tırnak içine alınır (XML'de " olarak görünür). Tek tek karakterler de tırnaklı dizeler içinde ters eğik çizgilerle kaçabilir.

Nmap, XML ayıristiricilerinin Nmap XML çıktısını doğrulamasını sağlayan bir belge türü tanımı (DTD) içerir. Öncelikle programatik kullanım için tasarlanmış olsa da, insanların Nmap XML çıktısını yorumlamasına da yardımcı olabilir. DTD, formatın yasal öğelerini tanımlar ve genellikle alabilecekleri nitelikleri ve değerleri sıralar. Ek A, Nmap XML Çıktı DTD'sinde yeniden üretilmiştir.

Using XML Output (XML Çıktısını Kullanma)

Nmap XML formatı birçok güçlü şekilde kullanılabilir, ancak çok az kullanıcı bundan faydalananmaktadır. Bunun nedeninin birçok kullanıcının XML konusundaki deneyimsizliği ve Nmap XML formatının kullanımına ilişkin pratik, çözüm odaklı dokümantasyon eksikliği olduğuna inanıyorum. Bu bölüm, "Perl ile XML Çıktısını Manipüle Etme", "Veritabanına Çıktı" ve "HTML Raporları Oluşturma" bölümleri de dahil olmak üzere çeşitli pratik örnekler sunmaktadır.

XML'in önemli bir avantajı, grepable ve etkileşimli çıktı gibi özel Nmap çıktı türleri için yaptığınız gibi kendi ayıristiricınızı yazmanıza gerek olmamasıdır. Herhangi bir genel XML ayıristirici işinizi görecektir.

Nmap XML çıktısı elbette herhangi bir metin editöründe veya XML editöründe görüntülenebilir. Microsoft Excel de dahil olmak üzere bazı elektronik tablo programları Nmap XML verilerini görüntülemek için doğrudan içe aktarabilir. Bu genel amaçlı XML işlemcileri, Nmap XML'i diğer XML dosyaları gibi genel olarak ele alma sınırlamasını paylaşırlar. Öğelerin göreceli önemini ya da daha kullanışlı bir sunum için verilerin nasıl düzenleneceğini anlamazlar. Nmap XML çıktısını anımlandıran özel XML işlemcilerin kullanımı aşağıdaki bölümlerin konusudur.

Manipulating XML Output with Perl (Perl ile XML Çıktısını Yönetme)

Genel XML ayıristiricileri tüm popüler programlama dilleri için genellikle ücretsiz olarak mevcuttur. Örnekler libxml C kütüphanesi ve Java ve C++ için Apache Xerces ayıristiricisidir (Perl ve COM bağlamaları ile). Bu ayıristiriciler Nmap XML çıktısını işlemek için yeterli olsa da, geliştiriciler Nmap XML ile birlikte çalışma görevini daha da kolaylaştırabilecek çeşitli diller için özel modüller oluşturmuşlardır.

En iyi özel Nmap XML desteğine sahip dil Perl'dür. Max Schubert (sevgiyle Perldork olarak bilinir) Nmap::Scanner adında bir modül yaratırken Anthony Persaud Nmap::Parser'ı yaratmıştır. Bu iki modül birçok benzerliğe sahiptir: Nmap'i kendileri çalıştırabilir veya bir çıktı dosyasından okuyabilirler, iyi belgelenmişlerdir, çok sayıda örnek komut dosyası ile birlikte gelirler, Kapsamlı Perl Arşiv Ağının (CPAN) bir parçasıdır ve kullanıcılar arasında popülerdirler. Her ikisi de Nmap çalışırken verileri yorumlamak için geri arama tabanlı bir ayırtıcıyı ve Nmap yürütmemi bitirdiğinde tamamen ayırtılmış bir belge elde etmek için bir kerede ayırtıcıyı sunar. API'leri biraz farklıdır-Nmap::Scanner tip güvenli sınıflara dayanırken, Nmap::Parser daha hafif yerel Perl dizilerine dayanır. İhtiyaçlarınızı ve tercihlerinizi en iyi hangisinin karşılaşmasına karar vermek için her birine bakmanızı öneririm.

Örnek 13.11 Nmap::Parser'ın basit bir gösterimidir. Modülün dokümantasyonundan alınmıştır (başka birçok örnek de içerir). Hızlı bir tarama gerçekleştirir, ardından genel tarama istatistiklerinin yanı sıra mevcut her hedef ana bilgisayar hakkında bilgi yazdırır. Ayırtırma mantığı ve düzenli ifadelerin hakim olduğu diğer Nmap çıktı biçimlerini kullanan komut dosyalarına kıyasla ne kadar okunabilir olduğuna dikkat edin. Perl becerileri zayıf olan kişiler bile bunu Nmap tarama ihtiyaçlarını otomatikleştirmek için basit programlar oluşturmak için bir başlangıç noktası olarak kullanabilir.

Örnek 13.11. Nmap::Parser örnek kodu

```
use Nmap::Parser;

#PARSING
my $np = new Nmap::Parser;

$nmap_exe = '/usr/bin/nmap';
$np->parsescan($nmap_exe,'-sT -p1-1023', @ips);

#or

$np->parsefile('nmap_output.xml'); #using filenames

#GETTING SCAN INFORMATION
```

```

print "Scan Information:\n";
$si = $np->get_scaninfo();
#get scan information by calling methods
print
'Number of services scanned: '.$si->num_of_services()."\\n",
'Start Time: '.$si->start_time()."\\n",
'Scan Types: ,(join ' ', $si->scan_types()))."\\n";

```

#GETTING HOST INFORMATION

```

print "Hosts scanned:\\n";
for my $host_obj ($np->get_host_objects()){
    print
    'Hostname : '.$host_obj->hostname()."\\n",
    'Address : '.$host_obj->ipv4_addr()."\\n",
    'OS match : '.$host_obj->os_match()."\\n",
    'Open Ports: .(join ',', $host_obj->tcp_ports('open'))."\\n";
    #... you get the idea...
}

```

```

#frees memory--helpful when dealing with memory intensive scripts
$np->clean();

```

Karşılaştırma için, Örnek 13.12, Nmap::Scanner kullanan ve belgelerinden kopyalanan örnek bir Perl betiğidir. Bu betik, bir ana bilgisayar bulunduğuunda ve ana bilgisayarda her açık bağlantı noktası keşfedildiğinde gerçek zamanlı uyarılar yazdırma için scan_started ve port_found işlevlerini kaydederek olay odaklı bir geri arama yaklaşımı kullanır.

Örnek 13.12. Nmap::Scanner örnek kodu

```

my $scanner = new Nmap::Scanner;
$scanner->register_scan_started_event(\&scan_started);
$scanner->register_port_found_event(\&port_found);
$scanner->scan('-sS -p 1-1024 -O --max-rtt-timeout 200ms somehost.org.ne

```

```

t.it');

sub scan_started {
    my $self    = shift;
    my $host    = shift;

    my $hostname = $host->name();
    my $addresses = join(', ', map {$_->address()} $host->addresses());
    my $status = $host->status();

    print "$hostname ($addresses) is $status\n";
}

sub port_found {
    my $self    = shift;
    my $host    = shift;
    my $port    = shift;

    my $name = $host->name();
    my $addresses = join(', ', map {$_->addr()} $host->addresses());

    print "On host $name ($addresses), found ",
        $port->state()," port ",
        join('/',$port->protocol(),$port->portid()),"\n";
}

```

Common Platform Enumeration (CPE) (Ortak Platform Numaralandırma (CPE))

Common Platform Enumeration (CPE), yazılım uygulamalarını, işletim sistemlerini ve donanım platformlarını adlandırmmanın standartlaştırılmış bir yoludur. Nmap, hizmet ve işletim sistemi tespiti için CPE çıktısı içerir.

Structure of a CPE Name (CPE Adının Yapısı)

CPE adı, yedi sıralı alanı kodlayan bir URL'dir:

```
cpe:/<part> : <vendor> : <product> : <version> : <update> : <edition> : <language>
```

Bazı alanlar boş bırakılabilir ve URL'nin sonunda boş alanlar bırakılabilir. CPE adlarının ana bölümü <part> alanındadır; bu yalnızca üç değer alabilir:

a başvurular için,

h donanım platformları için veya

o işletim sistemleri için.

URL'nin başına bakarak cpe:/a:microsoft:sql_server:6.5'in bir uygulamayı, cpe:/h:asus:rt-n16'nın bir donanım türünü ve cpe:/o:freebsd:freebsd:3.5.1'in bir işletim sistemini adlandırdığını kolayca görebilirsiniz.

Nmap üç tür CPE adının da çıktısını alabilir: İşletim sistemi tespiti h ve o yazdırabilir; ve hizmet tespiti potansiyel olarak üçünün de çıktısını alabilir. Örneğin CPE adları normal işletim sistemi ve hizmet çıktısı ile karıştırılır:

Örnek 13.13. CPE vurgulanmış normal çıktı

```
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Network Distance: 10 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
```

Uygulamalar için CPE adları (a kısmı ile) normal çıktıda gösterilmez, ancak XML'de mevcuttur. CPE, service veya osclass'ın bir çocuğu olabilen bir cpe ögesi olarak temsil edilir.

Output to a Database (Veritabanına Çıktı)

Yaygın bir istek, daha kolay sorular ve izleme için Nmap sonuçlarını bir veritabanına çıkarmaktır. Bu, bireysel bir sizma test uzmanından uluslararası bir kuruluşa kadar kullanıcıların tüm tarama sonuçlarını saklamasına ve bunları kolayca karşılaşmasına olanak tanır. İşletme günlük olarak büyük taramalar yapabilir ve yeni açık portları ya da kullanılabilir makineleri yöneticilere postalamak için

sorgular planlayabilir. Sızma testi uzmanı yeni bir güvenlik açığını öğrenebilir ve ilgili müşterileri uyarabilmek için etkilenen uygulama için tüm eski tarama sonuçlarını arayabilir. Araştırmacılar milyonlarca IP adresini tarayabilir ve sonuçları kolay gerçek zamanlı sorgular için bir veritabanında tutabilir.

Bu hedefler övgüye değer olsa da, Nmap doğrudan veritabanı çıktı işlevselligi sunmamaktadır. Sadece hepsini destekleyemeyeceğim kadar çok farklı veritabanı türü olmakla kalmıyor, aynı zamanda kullanıcıların ihtiyaçları tek bir veritabanı şemasının uygun olamayacağı kadar önemli ölçüde değişiyor. İşletmenin, pentester'in ve araştırmacının ihtiyaçlarının hepsi farklı tablo yapıları gerektiriyor.

Bir veritabanı gerektirecek kadar büyük projeler için, önce en uygun DB şemasına karar vermenizi, ardından Nmap XML verilerini uygun şekilde içe aktarmak için basit bir program veya komut dosyası yazmanızı öneririm. XML ayırtıcılarının ve veritabanı erişim modüllerinin geniş kullanılabilirliği sayesinde bu tür komut dosyaları genellikle sadece birkaç dakika sürer. Perl, güçlü bir veritabanı soyutlama katmanı ve ayrıca özel Nmap XML desteği sunduğu için genellikle iyi bir seçimdir. "Perl ile XML Çıktısını Değiştirme" adlı bölüm, Perl komut dosyalarının Nmap XML verilerini ne kadar kolay kullanabileceğini göstermektedir.

Başka bir seçenek de özel bir Nmap veritabanı destek yaması kullanmaktadır. Bir örnek, Nmap'in kendisine MySQL günlük kaydı işlevselligi ekleyen nmap-sql'dir. Dezavantajları, şu anda yalnızca MySQL veritabanını desteklemesi ve sık sık yeni Nmap sürümlerine taşınması gereğidir. Öte yandan, XML tabanlı bir yaklaşımın yeni Nmap sürümleri yayınlandığında kırılma olasılığı daha düşüktür.

Başka bir seçenek de, zaman içinde bir agdaki değişiklikleri izlemek için bir araç paketi olan PBNJ'dir. Çevrimiçi ana bilgisayarlar ve açık bağlantı noktaları gibi tarama verilerini bir veritabanında (SQLite, MySQL veya Postgres) depolar. Bu verilere erişmek veya değişiklikleri görüntülemek için esnek bir sorgulama ve uyarı sistemi sunar.

Creating HTML Reports (HTML Raporları Oluşturma)

Nmap tarama sonuçlarını HTML olarak kaydetmek için bir seçeneğe sahip değildir, ancak XML çıktısını otomatik olarak HTML'ye dönüştürmek mümkündür. Bir Nmap

XML çıktı dosyası genellikle dönüşümün nasıl gerçekleştiğini açıklayan nmap.xsl adlı bir XSL stil sayfasına bir referans içerir.

Stil sayfasının nerede bulunabileceğini belirten XML işleme talimatı aşağıdaki gibi görünecektir

```
<?xmlstylesheet href="/usr/share/nmap/nmap.xsl" type="text/xsl"?>
```

Tam konum, platforma ve Nmap'in nasıl yapılandırıldığına bağlı olarak farklı olabilir.

Böyle bir stil sayfası referansı, taramayı başlatan aynı makinede tarama sonuçlarını görüntülerken iyi çalışacaktır, ancak XML dosyası nmap.xsl dosyasının farklı bir yerde olduğu veya tamamen bulunmadığı başka bir makineye aktarılırsa çalışmayaçaktır. XML stilini taşınabilir hale getirmek için Nmap'e --webxml seçeneğini verin. Bu, işleme talimatını okumak için değiştirecektir

```
<?xmlstylesheet href="https://nmap.org/svn/docs/nmap.xsl" type="text/xsl"?>
```

Sonuçta ortaya çıkan XML çıktı dosyası, web'e bağlı herhangi bir makinede HTML olarak işlenecektir. Ağ konumunu bu şekilde kullanmak genellikle daha kullanışlıdır, ancak nmap.xsl'nin yerel kopyası gizlilik nedeniyle varsayılan olarak kullanılır.

Farklı bir stil sayfası kullanmak için --stylesheet <dosya> seçeneğini kullanın. --webxml seçeneğinin --stylesheet https://nmap.org/svn/docs/nmap.xsl için bir takma ad olduğunu unutmayın. Stil sayfasını tamamen atlamak için --no-stylesheet seçeneğini kullanın.

Saving a Permanent HTML Report (Kalıcı HTML Raporu Kaydetme)

Burada, yaygın XSLT işlemcileri kullanarak bir Nmap XML çıktı dosyasını bir HTML dosyasına dönüştüren komutlar bulunmaktadır. Bir web tarayıcısında görüntülenen örnek çıktı Şekil 13.1'de gösterilmiştir, "Web tarayıcısında XML çıktısından HTML".

xsltproc ⇒ **xsltproc** <nmap-output.xml> -o <nmap-output.html>

Saxon ⇒ Saxon 9:

```
java -jar saxon9.jar -s: <nmap-output.xml> -o: <nmap-output.html>
```

Previous Saxon releases:

```
java -jar saxon.jar -a <nmap-output.xml> -o <nmap-output.html>
```

Xalan ⇒ Using Xalan C++:

Xalan -a <nmap-output.xml> -o <nmap-output.html>

Using Xalan Java:

java -jar xalan.jar -IN <nmap-output.xml> -OUT <nmap-output.html>

Şekil 13.1. Bir web tarayıcısında XML çıktısından HTML

The screenshot shows a web browser window with the following details:

- Address Bar:** file:///home/fyodor/nmap/logs/xml-sample.html
- Title Bar:** File Edit View Go Bookmarks Tools Tabs Help
- Content Area:**
 - ping results**: echo-reply
 - address**: 64.13.134.52 (ipv4)
 - hostnames**: scanme.nmap.org (user), scanme.nmap.org (PTR)
 - ports**:
 - The 95 ports scanned but not shown below are in state: **filtered**.
 - 95 ports replied with: **no-responses**.
 - Table of Open Ports**:

Port	State	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack	OpenSSH	4.3
ssh-hostkey		1024 60:ac:4d:51:b1:cd:85:09:12:16:92:76:1d:5d:27:6e (DSA) 2048 2c:22:75:60:4b:c3:3b:18:a2:97:2c:96:7e:28:dc:dd (RSA)				protocol 2.0

Bu programlar, gömülü stil sayfası referansı sayesinde stil sayfasını nereye yükleyeceklerini otomatik olarak bilirler. Tarihsel bir not olarak, stil sayfasının başlangıçta XML dosyasını bir web tarayıcısında açarak XML çıktısını HTML olarak oluşturması amaçlanmıştı. Bir süre için bu şekilde çalıştı. Ancak web tarayıcıları, XML dosyalarının dar sınırlı konumlar dışında yüklenmesini engelleyen giderek

daha ciddi aynı kaynak kısıtlamaları uygulamaktadır. Örneğin, Mozilla tarafından kullanılan işleme motoru Gecko, stil sayfasının XML dosyasıyla aynı dizinde veya XML dosyasının bir alt dizininde bulunmasını gerektirir.

Grepable Output (-oG) (Greplenebilir Çıktı (-oG))

Bu çıktı biçimini kullanımdan kaldırıldığı için en son ele alınmıştır. XML çıktı biçimini çok daha güçlendir ve deneyimli kullanıcılar için neredeyse aynı derecede kullanışlıdır. XML, düzinece mükemmel ayrıştırıcının mevcut olduğu bir standarttır, grepable çıktı ise benim kendi basit hack'imdır. XML, yeni Nmap özellikleri yayındıkça bunları destekleyecek şekilde genişletilebilirken, ben bu özellikleri koyacak bir yer olmadığı için sık sık grepable çıktısından çıkarmak zorunda kalıyorum.

Bununla birlikte, greplenebilir çıktı hala oldukça popülerdir. Her bir ana bilgisayarı tek bir satırda listeleyen basit bir formattır ve grep, awk, cut, sed, diff ve Perl gibi standart Unix araçları ile kolayca aranabilir ve ayırtılabilir. Ben bile genellikle komut satırında yapılan tek seferlik testler için kullanıyorum. SSH portu açık olan veya Solaris çalıştırın tüm ana bilgisayarları bulmak, ana bilgisayarları tanımlamak için sadece basit bir grep gerektir, istenen alanları yazdırma için bir awk veya cut komutuna borulanır. Bu bölümde katkıda bulunan Lee "MadHat" Heath grep çıktısı meraklılarından biridir.

Örnek 13.14 grepable çıktısının tipik bir örneğini göstermektedir. Normalde her konak sadece bir satır alır, ancak bu girişi sayfaya sığdırmak için yedi satır写了。 Ayrıca hash komut istemi ile başlayan üç satır vardır (Nmap komut satırını saymazsak). Bunlar Nmap'in ne zaman başladığını, kullanılan komut satırı seçeneklerini, tamamlanma süresini ve istatistikleri açıklayan yorumlardır. Yorum satırlarından biri taranan port numaralarını sıralıyor. Düzinece satırı boş harcamamak için kısalttım. Bu özel yorum yalnızca ayrıntılı (-v) modda yazdırılır. Verbosity seviyesini bir -v'nin ötesine yükseltmek grepable çıktısını daha fazla değiştirmeyecektir. Satır uzunluğunu azaltmak için saatler ve tarihler [time] ile değiştirilmiştir。

Örnek 13.14. Tipik bir grepable çıktı örneği

```
# nmap -T4 -A -v -oG - scanme.nmap.org
# Nmap 5.35OC18 scan initiated [time] as: nmap -T4 -A -v -oG - scanme.nmap.org
# Ports scanned: TCP(1000;1,3-4,6-7,...,65389) UDP(0;) SCTP(0;) PROTOCOLS(0;)
Host: 64.13.134.52 (scanme.nmap.org) Status: Up
Host: 64.13.134.52 (scanme.nmap.org) Ports: 22/open/tcp//ssh//OpenSSH 4.3 (protocol 2.0)/, 25/closed/tcp//smtp///, 53/open/tcp//dns///
# Nmap done at [time] -- 1 IP address (1 host up) scanned in 21.90 seconds
```

Buradaki komut satırı, greplenebilir çıktıının -oG'ye - argümanı ile standart çıktıya gönderilmesini talep etmiştir. Agresif zamanlamanın (-T4) yanı sıra işletim sistemi ve sürüm tespiti (-A) talep edildi. Yorum satırları kendi kendini açıklıyor ve greplenebilir çıktıının etini Host satırında bırakıyor. Daha fazla ana bilgisayar taramış olsaydım, mevcut olanların her biri kendi Ana Bilgisayar satırına sahip olacaktı.

Grepable Output Fields (Greplenebilir Çıktı Alanları)

Ana bilgisayar satırı, her biri iki nokta üst üste ve boşlukla takip edilen bir alan adından ve ardından alan içeriğinden oluşan alanlara bölünür. Alanlar sekme karakterleriyle ayrılır (ASCII dokuz numara, '\t'). Örnek 13.14, "Tipik bir grepable çıktı örneği" altı alan göstermektedir: Ana Bilgisayar, Durum, Bağlantı Noktaları, Yoksayılan Durum, İşletim Sistemi, Seq Dizini ve IP Kimliği Seq. IP protokolü (-sO) taramalarına bir Protokoller bölümü dahildir. Tam olarak verilen alanlar kullanılan Nmap seçeneklerine bağlıdır. Örneğin, OS tespiti OS, Seq Index ve IP ID Seq alanlarını tetikler. Bunlar sekmeyle sınırlandırıldığından, alanları aşağıdaki gibi bir Perl satırıyla bölebilirsınız:

```
@fields = split("\t", $host_line);
```

Örnek 13.14, "Tipik bir greplenebilir çıktı örneği" durumunda, @fields dizisi altı üye içerecektir. \$fields[0] "Host: 64.13.134.52 (scanme.nmap.org)" ve \$fields[1] uzun Ports alanını içerecektir. Nmap geliştirmelerini desteklemek için yeni alanlar eklenebileceğinden, grepable çıktısını ayırtıran komut dosyaları tanımadıkları alanları göz ardı etmelidir.

Sekiz olası alan aşağıdaki bölümlerde açıklanmaktadır.

Host field (Ev sahibi alanı)

Örnek: Ana Bilgisayar: 64.13.134.52 (scanme.nmap.org)

Ana Bilgisayar alanı her zaman önce gelir ve hangi Nmap seçenekleri seçilirse seçilsin dahil edilir. İçerik IP adresi (-6 belirtilmişse bir IPv6 adresi), bir boşluk ve ardından parantez içinde ters DNS adıdır. Ters ad mevcut değilse, parantezler boş bırakılır.

Status field (Durum alanı)

Örnek: Durumunuz nedir? Yukarı

Durum alanı hedef ana bilgisayarın Yukarı, Aşağı veya Bilinmiyor olduğunu gösterir. Liste taraması (-sL) herhangi bir test gerçekleştirmediği için hedefleri her zaman Bilinmiyor olarak sınıflandırır. Ping taraması, bir ana bilgisayar en az bir ping probuna yanıt verirse Yukarı, yanıt alınmazsa Aşağı olarak listeler. Eskiden, hedefe gönderilen ping problemleri diğer ana bilgisayarlardan bir veya daha fazla yanıtla sonuçlanırsa Smurf olarak da raporlanırdı, ancak artık bu yapılmamaktadır. Örnek 13.15'te rastgele beş ana bilgisayarın ping taraması gösterilirken, Örnek 13.16'da beş ana bilgisayarın liste taraması gösterilmektedir.

Örnek 13.15. Ping taraması grepable çıktısı

```
# nmap -sn -oG - -iR 5
# Nmap 5.35DC18 scan initiated [time] as: nmap -sn -oG - -iR 5
Host: 93.182.218.153 () Status: Up
Host: 154.223.142.85 () Status: Down
Host: 120.128.8.97 () Status: Down
Host: 47.159.134.149 () Status: Down
Host: 24.172.4.19 () Status: Down
# Nmap done at [time] -- 5 IP addresses (1 host up) scanned in 4.25 seconds
```

Örnek 13.16. Liste taraması greplenebilir çıktı

```
# nmap -sL -oG - -iR 5
# Nmap 5.35DC18 scan initiated [time] as: ./nmap -sL -oG - -iR 5
Host: 91.244.202.129 () Status: Unknown
Host: 216.36.141.91 (cm216036141091.wcgwave.ca) Status: Unknown
Host: 17.130.29.192 () Status: Unknown
Host: 45.89.164.99 () Status: Unknown
Host: 215.22.1.81 () Status: Unknown
# Nmap done at [time] -- 5 IP addresses (0 hosts up) scanned in 13.00 seconds
```

Ports field(Limanlar alanı)

Örnek: Portlar: 111/open/tcp//rpcbind (rpcbind V2)/(rpcbind:100000*2-2)/2 (rpc #100000), 113/closed/tcp//auth//

Örnek 13.14, "Tipik bir grepable çıktı örneği "nde görülebileceği gibi, Portlar alanı açık ara en karmaşık olanıdır. Her ilginç port için girişler içerir (normal Nmap çıktısında port tablosuna dahil edilecek olanlar). Port girişleri virgül ve boşluk

karakteri ile ayrılır. Her bağlantı noktası girişi, ileri eğik çizgi (/) ile ayrılmış yedi alt alandan oluşur. Alt alanlar şunlardır: bağlantı noktası numarası, durum, protokol, sahip, hizmet, SunRPC bilgisi ve sürüm bilgisi. Bazı alt alanlar, özellikle işletim sistemi veya sürüm tespiti olmayan temel bağlantı noktası taramaları için boş olabilir. Örnek 13.14, "Tipik bir grepable çıktı örneği "ndeki ardışık eğik çizgiler boş alt alanları gösterir. Perl'de bunları şu şekilde bölebilirsiniz:

```
($port, $state, $protocol, $owner, $service, $rpc_info, $version) =  
    split('/', $ports);
```

Alternatif olarak, aşağıdaki gibi komutları kullanarak bilgileri komut satırından alabilirsiniz:

```
cut -d/ -f<fieldnumbers>  
awk -F/ '{print $<fieldnumber>}'
```

Bazı alt alanlar diğer çıktı modlarında eğik çizgi içerebilir. Örneğin, SSL özellikli bir web sunucusu ssl/http olarak görünür ve sürüm bilgisi mod_ssl/2.8.12 gibi dizeler içerebilir. Eğik çizgi alt alan sınırlayıcısı olduğundan, bu durum ayırtırmayı bozacaktır. Bu sorunu önlemek için, eğik çizgiler Port alanında herhangi bir yerde göründüklerinde boru karakterine () dönüştürülür.

Ayırtıcılar yediden fazla eğik çizgi ile ayrılmış alt alana izin verecek ve gelecekteki Nmap geliştirmeleri yenilerini gerektirebileceğinden ekstraları göz ardedecek şekilde yazılmalıdır. Aşağıdaki listede şu anda tanımlanmış yedi Port alt alanının her biri açıklanmaktadır.

Port number (Liman numarası) ⇒ Bu basitçe sayısal TCP veya UDP bağlantı noktası numarasıdır.

State ⇒ Normal çıkış portu tablosunda görünecek olan aynı port durumu burada gösterilmektedir.

Protocol ⇒ Bu tcp, udp veya sctp'dir.

Owner ⇒ Bu, hedef ana bilgisayarın bir identd (auth) sunucusunun sorgulanmasından elde edilen sonuçlara göre uzak sunucunun altında çalıştığı kullanıcı adını belirtmek için kullanılır. Ident taraması (-I) artık Nmap ile kullanılamamaktadır, bu nedenle bu alan her zaman boştur. Kimlik verileri hala

auth-owners NSE betiği kullanılarak elde edilebilir, ancak sonuçlar bu alana yerleştirilmez.

Service ⇒ Hizmet adı, nmap-services aramasından veya (daha güvenilir olarak) sürüm algılama (-sV) yoluyla istendiye ve başarılı olduysa elde edilir. Sürüm algılama etkinleştirildiğinde, ssl|http gibi bileşik girişler ve sonda soru işaretleri olan girişler görülebilir. Anlamı, Bölüm 7, Hizmet ve Uygulama Sürüm Tespiti'nde tartışıldığı gibi normal çıktı ile aynıdır.

SunRPC info ⇒ Sürüm tespiti (-sV) istendiye ve bağlantı noktasının SunRPC protokolünü kullandığı tespit edildiyse, RPC program numarası ve kabul edilen sürüm numaraları buraya dahil edilir. Tipik bir örnek (rpcbind:100000*2-2) şeklindedir. Veriler her zaman parantez içinde döndürülür. Program adı ile başlar, ardından iki nokta üst üste ve program numarası, ardından bir yıldız işareteti ve ardından bir tire ile ayrılmış düşük ve yüksek desteklenen sürüm numaraları gelir. Yani bu örnekte, rpcbind (program numarası 100000) rpcbind sürüm 2 istekleri için bağlantı noktasını dinliyor.

Version info ⇒ Sürüm tespiti istenir ve başarılı olursa, sonuçlar burada etkileşimli çıktıda kullanılan aynı biçimde sağlanır. SunRPC portları için, RPC verileri de burada yazdırılır. Bu sütundaki RPC sonuçları için biçim <düşük sürüm numarası>-<yüksek sürüm numarası> (rpc #<rpc program numarası>) şeklindedir. Yalnızca bir sürüm numarası desteklendiğinde, bu numara bir aralık olarak değil tek başına yazdırılır. SunRPC bilgi alt alanında (rpcbind:100000*2-2) gösteren bir bağlantı noktası, sürüm bilgisi alt alanında 2 (rpc #100000) gösterecektir.

Protocols field (Protokoller alanı)

Örnek: Protokoller: 1/açık/icmp/, 2/açık|filtrelenmiş/igmp/

IP protokol taraması (-sO) Portlar yerine Protokoller alanına sahiptir. İçeriği Bağlantı Noktaları alanına oldukça benzer, ancak yedi yerine yalnızca üç alt alanı vardır. Ports alanında olduğu gibi eğik çizgilerle sınırlanırlılar. Bir alt alanda görünecek eğik çizgiler, Bağlantı Noktaları alanında olduğu gibi borulara (|) dönüştürülür. Alt alanlar protokol numarası, durum ve protokol adıdır. Bunlar, bir protokol taraması için etkileşimli çıktıda gösterilen üç alana karşılık gelir. IP protokol taraması grepable çıktısının bir örneği Örnek 13.17'de gösterilmektedir. Normalde tek satır olması gereken Host satırı burada okunabilirlik için sarılmıştır.

Örnek 13.17. IP protokol taraması için Grepable çıktısı

```
# nmap -v -sO -oG - localhost
# Nmap 5.35DC18 scan initiated [time] as: nmap -v -sO -oG - localhost
# Ports scanned: TCP(0;) UDP(0;) SCTP(0;) PROTOCOLS(256;0-255)
Host: 127.0.0.1 (localhost)      Status: Up
Host: 127.0.0.1 (localhost)      Protocols: 1/open/icmp/, 2/open/igmp/, 4/open|filtered/ip/, 6/open/tcp/, 17/open/udp/, 41/open|filtered/ssh/
# Nmap done at [time] -- 1 IP address (1 host up) scanned in 2.36 seconds
```

Ignored State field (Yoksayılan Durum alanı)

Örnek: Yoksayılan Durum: filtrelenmiş (1658)

Yer kazanmak için, Nmap açık olmayan bir durumdaki portları Portlar alanındaki listeden çıkarabilir. Nmap bunu etkileşimli çıktıda da yapar. Düzenli Nmap kullanıcıları aşağıdaki gibi satırlara aşinadır Gösterilmiyor: 993 kapalı bağlantı noktası. Grepable modu için, bu durum Yoksayılan Durum alanında verilir. Durum adından sonra bir boşluk, ardından parantez içinde o durumda bulunan bağlantı noktası sayısını yer alır.

os field (İşletim sistemi alanı)

Örnek: İşletim Sistemi: Linux 2.4.0 - 2.5.20

Tüm mükemmel OS eşleşmeleri burada listelenir. Birden fazla eşleşme varsa, bunlar Örnek 13.14, "Tipik bir grepable çıktı örneği "nde gösterildiği gibi bir boru karakteri ile ayrılır. Yalnızca serbest metin açıklamaları sağlanır. Grepable modu, diğer çıktı modlarında gösterilen satıcı, işletim sistemi ailesi ve cihaz türü sınıflandırmasını sağlamaz.

Seq Index field (Seq Dizin alanı)

Örnek: Sıra Dizini: 3004446

Bu sayı, uzak ana bilgisayara karşı TCP başlangıç sıra numarası tahmin saldırıları gerçekleştirmenin zorluğuna ilişkin bir tadmındır. Bunlar kör sahtekarlık saldırıları olarak da bilinir ve bir saldırının uzaktaki bir ana bilgisayara sanki başka bir IP adresinden gelmiş gibi tam bir TCP bağlantısı kurmasına olanak tanır. Bu, bir saldırının izlerini gizlemesine her zaman yardımcı olabilir ve genellikle güvenilir IP adreslerine ekstra ayrıcalıklar veren rlogin gibi hizmetlere karşı ayrıcalık artışına yol açabilir. Seq Index değeri yalnızca OS algılama (-O) istendiğinde ve bunun için problema başarılı olduğunda kullanılabilir. Verbosity (-v) istendiğinde etkileşimli çıktıda rapor edilir. Bu değerin hesaplanması ve anlamı hakkında daha fazla ayrıntı Bölüm 8, Uzak İşletim Sistemi Algılama'da verilmiştir.

IP ID Seq field (IP Kimliği Sıra alanı)

Örnek: IP Kimliği Sıra: Tüm sıfırlar

Bu basitçe uzak ana bilgisayarın IP kimliği oluşturma algoritmasını açıklar. Yalnızca işletim sistemi algılama (-O) istendiğinde ve bunun için problema başarılı olduğunda kullanılabilir. Etkileşimli mod bunu da bildirir ve Bölüm 8, Uzak İşletim Sistemi Algılama'da tartışılmıştır.

Parsing Grepable Output on the Command Line (Komut Satırında Greplenebilir Çıktıyı Ayırıştırma)

Grepable çıktı, XML çıktısını ayırtırmak için bir komut dosyası yazmaya gerek kalmadan hızlı bir şekilde bilgi toplamak istediğinizde gerçekten parlar. Örnek 13.18 bunun tipik bir örneğini göstermektedir. Amaç, 80 numaralı portu açık olan C sınıfı büyülükteki bir ağdaki tüm ana bilgisayarları bulmaktır. Nmap'e her hostun sadece o portunu taraması (ping aşamasını atlayarak) ve stdout'a grepable bir rapor çıktısı vermesi söylenir. Sonuçlar, /open/ içeren satırları bulan ve eşleşen her satır için iki ve üç numaralı alanları çıktı olarak veren önemsiz bir awk komutuna aktarılır. Bu alanlar IP adresi ve ana bilgisayar adıdır (veya ana bilgisayar adı mevcut değilse boş parantezler).

Örnek 13.18. Komut satırında greplenebilir çıktıyı ayırıştırma

```
> nmap -p80 -Pn -oG - 10.1.1.0/24 | awk '/open/{print $2 " " $3}'  
10.1.1.72 (userA.corp.foocompany.biz)  
10.1.1.73 (userB.corp.foocompany.biz)  
10.1.1.75 (userC.corp.foocompany.biz)  
10.1.1.149 (admin.corp.foocompany.biz)  
10.1.1.152 (printer.corp.foocompany.biz)  
10.1.1.160 (10-1-1-160.foocompany.biz)  
10.1.1.161 (10-1-1-161.foocompany.biz)  
10.1.1.201 (10-1-1-201.foocompany.biz)  
10.1.1.254 (10-1-1-254.foocompany.biz)
```

Chapter 14. Understanding and Customizing Nmap Data Files (Bölüm 14. Nmap Veri Dosyalarını Anlama ve Özelleştirme)

- Introduction (Giriş)
- Well Known Port List: `nmap-services` (İyi Bilinen Bağlantı Noktası Listesi: nmap-services)
- Version Scanning DB: `nmap-service-probes` (Sürüm Tarama DB: nmap-service-probes)
- SunRPC Numbers: `nmap-rpc` (SunRPC Numaraları: nmap-rpc)
- Nmap OS Detection DB: `nmap-os-db` (Nmap İşletim Sistemi Algılama DB: nmap-os-db)
- MAC Address Vendor Prefixes: `nmap-mac-prefixes` (MAC Adresi Satıcı Önekleri: nmap-mac-prefixes)
- IP Protocol Number List: `nmap-protocols` (IP Protokol Numarası Listesi: nmap-protocols)
- Files Related to Scripting (Scripting ile İlgili Dosyalar)
- Using Customized Data Files (Özelleştirilmiş Veri Dosyalarını Kullanma)

Introduction (Giriş)

Nmap, bağlantı noktası taraması ve diğer işlemler için yedi veri dosyasına dayanır ve bunların tümü nmap- ile başlayan adlara sahiptir. Örneklerden biri olan nmap-services, bağlantı noktası adlarının karşılık gelen bağlantı noktası numarasına ve protokole göre kaydedilmesidir. Bu bölümde tek tek açıklanan diğerleri nmap-service-probes (sürüm algılama prob veritabanı), nmap-rpc (doğrudan RPC taraması için SunRPC program adından numaraya veritabanı), nmap-os-db (OS algılama veritabanı), nmap-mac-prefixes (ethernet MAC adresi önekinden (OUI) satıcı arama tablosuna) ve nmap-protocols (protokol taraması için IP protokoller listesi). Ayrıca bu bölüm Nmap Scripting Engine ile komut dosyası oluşturma ile ilgili belirli dosyaları kapsamaktadır. Kaynak dağıtım bu dosyaları /usr/local/share/nmap/ içine yükler ve resmi Linux RPM'leri bunları /usr/share/nmap/ içine koyar. Diğer dağıtımlar bunları başka bir yere yükleyebilir.

Bu dosyaların en son sürümleri <https://nmap.org/svn/> adresinde tutulmaktadır, ancak kullanıcıların daha yeni veri dosyalarını à la carte olarak almak yerine en son Nmap sürümüne yükseltmeleri şiddetle tavsiye edilir. Yeni dosyaların Nmap'in eski sürümleriyle çalışacağının garantisini yoktur (ancak neredeyse her zaman çalışırlar) ve sonuçta ortaya çıkan Nmap'in Frankenstein sürümleri işletim sistemini ve hizmet parmak izi gönderme sürecini karıştırabilir.

Çoğu kullanıcı veri dosyalarını asla değiştirmez, ancak şirketlerinde çalışan özel bir daemon için bir sürüm parmak izi veya port ataması eklemek isteyebilecek ileri düzey kullanıcılar için kullanışlı olabilir. Bu bölümde her bir dosyanın açıklaması ve yaygın olarak nasıl değiştirildikleri anlatılmaktadır. Daha sonra Nmap veri dosyalarını özel sürümlerle değiştirmek için genel mekanizma tartışılmaktadır. Birkaç dosya doğrudan port taramasıyla ilgili değildir, ancak kolaylık sağlamak için hepsi burada tartışılmıştır.

Well Known Port List: [nmap-services](#) (İyi Bilinen Bağlantı Noktası Listesi: [nmap-services](#))

nmap-services dosyası, bağlantı noktası adlarının karşılık gelen numara ve protokole göre bir kayıt defteridir. Her giriş, o portun açık bulunma olasılığını temsil eden bir sayıya sahiptir. Çoğu satırda bir de yorum vardır. Nmap yorumları göz ardı eder, ancak kullanıcılar bazen Nmap kullanıcının tanımadığı bir türde açık bir hizmet bildirdiğinde dosyada bunları ararlar. Örnek 14.1 dosyadan tipik bir alıntıyı göstermektedir. Okunabilirlik için bazı dolgu boşlukları eklenmiştir.

Örnek 14.1. nmap-services'ten alıntı

```

qotd      17/tcp    0.002346 # Quote of the Day
qotd      17/udp    0.009209 # Quote of the Day
msp       18/udp    0.000610 # Message Send Protocol
chargen   19/tcp    0.002559 # ttyst source Character Generator
chargen   19/udp    0.015865 # ttyst source Character Generator
ftp-data  20/tcp    0.001079 # File Transfer [Default Data]
ftp-data  20/udp    0.001878 # File Transfer [Default Data]
ftp       21/tcp    0.197667 # File Transfer [Control]
ftp       21/udp    0.004844 # File Transfer [Control]
ssh       22/tcp    0.182286 # Secure Shell Login
ssh       22/udp    0.003905 # Secure Shell Login
telnet   23/tcp    0.221265
telnet   23/udp    0.006211
priv-mail 24/tcp    0.001154 # any private mail system
priv-mail 24/udp    0.000329 # any private mail system
smtp     25/tcp    0.131314 # Simple Mail Transfer
smtp     25/udp    0.001285 # Simple Mail Transfer

```

Bu dosya ilk olarak <http://www.iana.org/assignments/port-numbers> adresindeki IANA tarafından atanmış portlar listesine dayanmaktadır, ancak yıllar içinde başka birçok port eklenmiştir. IANA truva atları, solucanlar ve benzerlerini takip etmez, ancak bunları keşfetmek birçok Nmap kullanıcısı için önemlidir.

Bu dosyanın dilbilgisi oldukça basittir. Beyaz boşluklarla ayrılmış üç sütun vardır. Birincisi, Nmap çıktısının SERVICE sütununda görüldüğü gibi servis adı veya kısaltmasıdır. İkinci sütun, eğik çizgi ile ayrılmış bağlantı noktası numarasını ve protokolü verir. Bu sözdizimi Nmap çıktısının PORT sütununda görülür. Üçüncü sütun "port frekansı" olup, internet taramaları sırasında portun ne sıklıkta açık bulunduğu bir ölçüsündür. Eğer atlanırsa, frekans sıfır olur. Nmap üçüncü sütunun ötesindeki her şeyi dikkate almaz, ancak çoğu satır beyaz boşluk ve ardından bir pound ('#') karakteri ve ardından bir yorum ile devam eder. Satırlar boş olabilir veya sadece bir pound karakteri ve ardından yorum içerebilir.

Dikkatli okuyucular nmap-services ve /etc/services (Windows'ta genellikle C:\windows\system32\drivers\etc\services adresinde bulunur) arasındaki yapı benzerliğini fark ederler. Bu bir tesadüf değildir. Bu format, sistem yöneticilerinin kendi /etc/services dosyalarındaki özel girdileri kopyalamalarına ve hatta bu dosyanın kendi versiyonunu tamamen değiştirmelerine izin vermek için korunmuştur. etc/services biçimi, bir hizmet için takma adlar sağlayan üçüncü bir sütuna izin verir. Bu, bağlantı noktası frekansı için kullanılan üçüncü sütunla çakışacaktır, bu nedenle bu sütunun içeriği sayısal değilse göz ardı edilir.

Örnek 14.1, UDP bağlantı noktalarının genellikle SSH ve FTP gibi yalnızca TCP hizmetleri için kaydedildiğini göstermektedir. Bu durum, her zaman her iki protokol için de hizmet kaydetme eğiliminde olan IANA'dan miras kalmıştır. Ekstra girişlere sahip olmak zarar vermez, çünkü varsayılan olarak Nmap en yüksek frekanslı portları tarar ve düşük frekanslı portlar basitçe atlanır. Ve beklenmedik olsa da, alıntı bazen popüler TCP portlarının UDP karşılıklarının açık bulunduğu göstermektedir.

Yöneticiler bazen ağlarında çalışan özel hizmetleri yansıtmak için bu dosyayı değiştirirler. Örneğin, bir zamanlar danışmanlığını yaptığım bir çevrimiçi hizmetler şirketinin yüksek numaralı portlarda çalışan düzinelere farklı özel daemon'u vardı. Bunu yapmak Nmap'in bu portlar için sonuçları bilinmeyen yerine gerçek adlarını kullanarak görüntülemesini sağlar. Port frekans rakamı olmayan girişler eklerseniz, frekansın sıfır olarak alınacağını, bu nedenle portun varsayılan olarak taranmayacağıını unutmayın. Tüm adlandırılmış portların tarandığından emin olmak için -p [1-65535] gibi bir seçenek kullanın.

Benzer şekilde, belirli bir kayıtlı bağlantı noktası belirli bir kuruluş için sıklıkla yanlış olabilir. nmap-services, bağlantı noktası numarası ve protokol kombinasyonu başına yalnızca bir hizmet adını işleyebilir, ancak bazen birkaç farklı uygulama türü aynı varsayılan bağlantı noktasını numarasını kullanır. Bu durumda, nmap-services için en popüler olanı seçmeye çalışıyorum. Böyle bir port numarası üzerinde yaygın olarak başka bir hizmet kullanan kuruluşlar dosyayı buna göre değiştirebilir.

Tek bir kuruluşla özgür hizmetler genellikle kendi nmap-hizmetlerinde kalmalıdır, ancak diğer port kayıtları herkese fayda sağlayabilir. Eğer önemli bir solucan, truva atı, dosya paylaşım uygulaması veya başka bir hizmet için varsayılan bağlantı noktasının en son nmap-services'te eksik olduğunu fark ederseniz, lütfen bir sonraki sürümde eklenmesi için bana (fyodor@nmap.org) gönderin. Bu, tüm kullanıcılaraya yardımcı olurken, nmap-services'in kendi özel sürümünüzü korumak ve güncellemek zorunda kalmanızı önler.

Bir başka yaygın özelleştirme de nmap-services'i yalnızca bir kuruluş için en yaygın, temel hizmetlere indirmektedir. Bir bağlantı noktası belirtimi olmadan, Nmap hizmetler dosyasında listelenmeyen herhangi bir bağlantı noktasını taramayacaktır, bu nedenle bu, -p seçeneğine uzun bir argüman kullanmadan taranan bağlantı noktalarının sayısını sınırlamanın bir yoludur. Soyulmuş dosya normalde Nmap'in varsayılan olarak kullanacağı yer yerine --datadir veya --

servicedb seçeneği ile erişilebilen özel bir konuma yerleştirilmelidir. Nmap yükseltmelerinin değiştirilmiş sürümlerini silmesini önlemenin yolları da dahil olmak üzere bu dosyaları özelleştirmek için tavsiyeler "Özelleştirilmiş Veri Dosyalarını Kullanma" adlı bölümde bulunabilir.

Version Scanning DB: [nmap-service-probes](#) (Sürüm Tarama DB: [nmap-service-probes](#))

Bu dosya, Nmap servis/sürüm tespit sisteminin (-sV veya -A seçenekleri) bir portta hangi programın dinlendiğini belirlemek için port sorgulaması sırasında kullandığı problemleri içerir. Örnek 14.2 tipik bir alıntı sunmaktadır.

Örnek 14.2. nmap-service-probes'dan alıntı

```
#####
# DNS Server status request: http://www.rfc-editor.org/rfc/rfc1035.txt
Probe UDP DNSStatusRequest q|\0\0\x10\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0|
ports 53,135
match domain m|^@\0\0\x90\x04\0\0\0\0\0\0\0\0|
# This one below came from 2 tested Windows XP boxes
match msrpc m|^x04\x06\0\0\x10\0\0\0\0\0\0\0|
[...]
#####
Probe UDP Help q|help\r\n\r\n|
ports 7,13,37
match chargen m|@ABCDEFGHIJKLMNPQRSTUVWXYZ|
match echo m|^help\r\n\r\n$|
match time m|^[\xc0-\xc5]...$|
```

Bu dosyanın grameri Bölüm 7, Hizmet ve Uygulama Sürüm Tespiti'nde tam olarak açıklanmıştır. nmap-service-probes, nmap-services'den daha karmaşık olsa da, onu geliştirmenin faydaları da daha büyük olabilir. Nmap'e, nmap-services bağlantı noktası kaydına dayanarak tahmin etmek yerine, bir şirketin özel hizmetlerini gerçekten tanımaması öğretilebilir.

Bu dosyadaki problemler, bazı UDP problemleriyle birlikte gönderilen protokole özgü yükler olarak UDP port taramasında da kullanılır. UDP taraması zordur çünkü çoğu hizmet boş bir sondaya yanıt göndermez, bu da açık ve filtrelenmiş portları ayırt etmeyi imkansız hale getirir. Buradaki problemler, gönderilmesi güvenli olacak ve olumlu yanıt verecek şekilde tasarlanmıştır.

Ayrıca, bazı yöneticiler sürüm tespitini asıl amacının çok ötesinde görevler için kullanmaktadır. Kısa bir sonda, Nmap'in web sayfalarının başlığını yazdırmasına, solucan bulaşmış makineleri tanımamasına, açık proxy'leri bulmasına ve daha fazlasına neden olabilir. Bunun pratik bir örneği "ÇÖZÜM: Sürüm Tespitini Açık Proxy Tespiti gibi Özel İhtiyaçlara Uyacak Şekilde Hackleme" adlı bölümde verilmiştir.

SunRPC Numbers: [nmap-rpc](#) (SunRPC Numaraları: [nmap-rpc](#))

nmap-services'de olduğu gibi, nmap-rpc basitçe numaraları isimlerle eşler. Bu durumda, SunRPC program numaraları, onları kullanan program adıyla eşlenir. Örnek 14.3 tipik bir alıntı sunmaktadır.

Örnek 14.3. nmap-rpc'den alıntı

```
rpcbind      100000  portmap sunrpc rpcbind
rstatd       100001  rstat rup perfmeter rstat_svc
rusersd      100002  rusers
nfs          100003  nfsprog nfsd
ypserv        100004  ypprog
mountd       100005  mount showmount
rpc.operd    100080  opermsg      # Sun Online-Backup
# DMFE/DAWS (Defense Automated Warning System)
#
Gqsvr        200034  gqsvr
Ppt           200035  ppt
Pmt           200036  pmt
```

Nmap yalnızca ilk iki boşluk bırakılarak ayrılmış sütunla (program adı ve numarası) ilgilendir. Bunun ötesinde görünebilecek takma adlara veya yorumlara bakmaz. Boş satırlara ve pound yorumlarıyla başlayanlara izin verilir. Bu biçim Unix'te /etc/rpc tarafından kullanılanla aynıdır, bu nedenle yöneticiler isterlerse bunun yerine bu dosyayı kullanabilirler.

nmap-rpc yalnızca Nmap sürüm açıklamalarının RPC öğütme özelliği tarafından kullanılır. Bu özellik "RPC Taşlama" adlı bölümde ele alınmıştır.

Kullanıcılar nmap-rpc'yi nadiren değiştirirler. Bunu yaptıklarında, genellikle özel bir hizmet veya en son nmap-rpc'de eksik olan genel bir hizmet eklemek içindir. İkinci

durumda, lütfen fyodor@nmap.org adresinden bana bir not gönderin, böylece bir sonraki sürüme ekleyebilirim. nmap-services'de olduğu gibi, bazı yöneticiler tarama süresinden tasarruf etmek için belirsiz RPC programlarını kaldırarak dosyayı soyarlar. Aynı uyarı geçerlidir: soyulmuş nmap-rpc'nizi --datadir seçeneği ile belirtin, dolaylı olarak kullanılacağı yere yüklemek yerine.

Nmap OS Detection DB: [nmap-os-db](#) (Nmap İşletim Sistemi Algılama DB: nmap-os-db)

nmap-os-db veri dosyası, farklı işletim sistemlerinin Nmap'in özel işletim sistemi algılama problemlerine nasıl yanıt verdiğine dair yüzlerce örnek içerir. Parmak izi olarak bilinen bloklara bölünmüştür ve her parmak izi bir işletim sisteminin adını, genel sınıflandırmasını ve yanıt verilerini içerir. Örnek 14.4, birkaç tipik parmak izini gösteren dosyadan bir alıntıdır.

Örnek 14.4. nmap-os-db'den alıntı

```
Fingerprint FreeBSD 7.0
Class FreeBSD | FreeBSD | 7.X | general purpose
SEQ(SP=100-10A%GCD=1-6%ISR=108-112%TI=I%II=I%SS=S%TS=21|22)
OPS(01=M5B4NW8NNT11%02=M578NW8NNT11%03=M280NW8NNT11%04=M5B4NW8NNT11%05=M218NW8NNT11%06=M109NNT11)
WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)
ECN(R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%0=M5B4NW8%CC=N%Q=)
T1(R=Y%DF=Y%T=3B-45%TG=40%S=0%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%S=0%A=S+%F=AS%0=M109NW8NNT11%RD=0%Q=)
T4(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)
T6(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)
T7(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S%F=AR%0=%RD=0%Q=)
U1(DF=N%T=3B-45%TG=40%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=S%T=3B-45%TG=40%CD=S)

Fingerprint Linux 2.6.17 - 2.6.24
Class Linux | Linux | 2.6.X | general purpose
SEQ(SP=A5-D5%GCD=1-6%ISR=A7-D7%TI=Z%II=I%TS=U)
OPS(01=M400C%02=M400C%03=M400C%04=M400C%05=M400C%06=M400C)
WIN(W1=8018%W2=8018%W3=8018%W4=8018%W5=8018%W6=8018)
ECN(R=Y%DF=Y%T=3B-45%TG=40%W=8018%0=M400C%CC=N%Q=)
T1(R=Y%DF=Y%T=3B-45%TG=40%S=0%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=3B-45%TG=40%W=8018%S=0%A=S+%F=AS%0=M400C%RD=0%Q=)
T4(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)
T6(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%0=%RD=0%Q=)
T7(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%0=%RD=0%Q=)
U1(DF=N%T=3B-45%TG=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=3B-45%TG=40%CD=S)
```

O seçeneği ile uzaktan işletim sistemi tespiti istendiğinde nmap-os-db işletim sistemi veritabanına başvurulur. Kısacası, Nmap bir hedef sisteme özel problemler gönderir ve yanıtları OS veritabanındaki girdilerle karşılaştırır. Eğer bir eşleşme varsa, veritabanı girdisi muhtemelen hedef sistemi tanımlamaktadır. İşletim sistemi algılama süreci Bölüm 8, Uzak İşletim Sistemi Algılama'da tam olarak açıklanmıştır. Referans parmak izi formatının ayrıntılı bir açıklaması için "Konu Parmak İzi Formatının Kodunun Çözülmesi" bölümüne bakın.

nmap-os-db kullanıcılar tarafından nadiren değiştirilir. Bir parmak izi eklemek veya değiştirmek oldukça karmaşık bir işlemdir ve genellikle bir parmak izini kaldırmak için hiçbir neden yoktur. İşletim sistemi veritabanının güncel bir sürümünü edinmenin en iyi yolu Nmap'in en son sürümünü edinmektir.

OS veritabanı (henüz) şimdije kadar yapılmış tüm ağ bağlantılı işletim sistemleri hakkında bilgi içermemektedir. Veritabanı Nmap kullanıcılarının katkılarıyla büyümektedir. Nmap bir işletim sistemini tahmin edemiyorsa ancak ne olduğunu biliyorsanız, lütfen "Nmap Bir Eşleşme Bulamadığında ve Bir Parmak İzi Yazdırdığında" adlı bölümdeki talimatları izleyerek parmak izini gönderin. Bazen parmak izlerinde hatalar olabilir veya güncelliğini yitirebilir. Bunu görürseniz, "Nmap Yanlış Tahmin Ettiğinde" adlı bölümde açıklandığı gibi bir düzeltme göndermeyi düşünün. Veritabanı iyileştirildiğinde herkes bundan faydalanan ve iyileştirmelerinizi göndermek sizi dosyanın kendi catalını korumak zorunda bırakmaz.

MAC Address Vendor Prefixes: [nmap-mac-prefixes](#) (MAC Adresi Satıcı Önekleri: [nmap-mac-prefixes](#))

Kullanıcılar MAC adresi örneklerini satıcı adlarıyla eşlestiren bu dosyayı nadiren değiştirirler. Tedavinin tamamı için okumaya devam edin.

Baskın ağ arayüzü türü haline gelen Ethernet cihazlarının her biri MAC adresi olarak bilinen 48 bitlik benzersiz bir tanımlayıcı ile programlanmıştır. Bu adres, yerel ağdaki hangi makinenin bir paket gönderdiğini ve paketin hangi makineye gideceğini belirlemek için ethernet başlıklarına yerleştirilir. İnsanlar bunu genellikle 00:60:1D:38:32:90 gibi onaltılk bir dize olarak gösterir.

Binlerce satıcının bulunduğu bir dünyada MAC adreslerinin benzersiz olmasını sağlamak için IEEE, ethernet cihazları üreten her şirkete bir Organizasyonel Benzersiz Tanımlayıcı (OUI) atar. Şirket, ürettiği ekipman için MAC adreslerinin ilk üç baytı için kendi OUI'sini kullanmalıdır. Örneğin, 00:60:1D:38:32:90'ın OUI'si 00601D'dir. Kalan üç baytı benzersiz oldukları sürece istediği gibi seçebilir. Bir sayaç basit bir yaklaşımdır. Tüm 16,8 milyon olası değeri atayan şirketler daha fazla OUI elde edebilir. nmap-mac-prefixes atanan her OUI'yi bunları satan satıcının adıyla eşler. Örnek 14.5 tipik bir alıntıdır.

Örnek 14.5. nmap-mac-prefixes'ten alıntı

```
006017 Tokimec
006018 Stellar ONE
006019 Roche Diagnostics
00601A Keithley Instruments
00601B Mesa Electronics
00601C Telxon
00601D Lucent Technologies
00601E Softlab
00601F Stallion Technologies
006020 Pivotal Networking
006021 DSC
006022 Vicom Systems
006023 Pericom Semiconductor
006024 Gradient Technologies
006025 Active Imaging PLC
006026 Viking Modular Solutions
```

İlk değer, 6 hex hanesi olarak üç baylıklı OUI'dir. Bunu şirket adı takip eder. Bu dosya <http://standards.ieee.org/regauth/oui/oui.txt> adresindeki tam listeden basit bir Perl betiği ile dönüştürülrerek oluşturulmuştur. IEEE ayrıca <http://standards.ieee.org/faqs/OUI.html> adresinde bir OUI SSS sunmaktadır.

Nmap, yerel bir ethernet LAN'ındaki ana bilgisayarların MAC adresini kablodaki başlıklarını okuyarak belirleyebilir. OUI'ye dayalı olarak üretici adını aramak ve raporlamak için bu tabloyu kullanır. Bu, ugraşığınız makinenin türünü kabaca tanımlamak için yararlı olabilir. Cisco, Hewlett Packard ya da Sun OUI'si olan bir cihaz muhtemelen sırasıyla bir yönlendirici, yazıcı ya da SPARCstation'ı tanımlar. Örnek 14.5, "nmap-mac-prefixes'ten alıntı" 00:60:1D:38:32:90 adresindeki aygıtın Lucent tarafından üretildiğini göstermektedir. Bu aslında dizüstü bilgisayarındaki Lucent Orinoco kablosuz kartı.

IP Protocol Number List: [nmap-protocols](#) (IP Protokol Numarası Listesi: nmap-protocols)

Bu dosya, IP başlığındaki tek baytlık IP protokol numarasını ilgili protokol adına eşler. Örnek 14.6 tipik bir alıntıdır.

Örnek 14.6. nmap-protocols'den alıntı

hopopt	0	HOPOPT	# IPv6 Hop-by-Hop Option
icmp	1	ICMP	# Internet Control Message
igmp	2	IGMP	# Internet Group Management
ggp	3	GGP	# Gateway-to-Gateway
ip	4	IP	# IP in IP (encapsulation)
st	5	ST	# Stream
tcp	6	TCP	# Transmission Control
cbt	7	CBT	# CBT
egp	8	EGP	# Exterior Gateway Protocol
[...]			
chaos	16	CHAOS	# Chaos
udp	17	UDP	# User Datagram

İlk iki alan protokol adı veya kısaltması ve ondalık biçimdeki sayıdır. Nmap protokol numarasından sonraki hiçbir şeyle ilgilenmez. "IP Protokol Taraması (-sO)" adlı bölümde açıklandığı gibi IP protokol taraması için kullanılır. 140'tan az protokol tanımlanmıştır ve kullanıcılar bu dosyayı neredeyse hiç değiştirmezler. Ham veriler IANA tarafından <http://www.iana.org/assignments/protocol-numbers> adresinde kullanılmıştır.

Files Related to Scripting (Scripting ile İlgili Dosyalar)

Nmap Scripting Engine tarafından kullanılan komut dosyaları başka bir tür veri dosyası olarak düşünülebilir. Komut dosyaları "Özelleştirilmiş Veri Dosyalarının Kullanımı" bölümünde listelenen dizinlerden birinin komut dosyaları alt dizininde saklanır. Her komut dosyasının adı .nse ile biter. Komut dosyalarıyla ilgili tüm ayrıntılar için Bölüm 9, Nmap Komut Dosyası Motoru'na bakın.

Script dizinindeki tüm dosyalar çalıştırılabilir scriptlerdir, biri hariç: script.db. Bu dosya, her bir betiğin hangi kategorilere ait olduğunu gösteren düz metin bir önbellektir. Doğrudan düzenlenmemelidir; bunun yerine --script-updatedb seçeneğini kullanın.

NSE'nin uzantı modülleri ("NSE Kütüphaneleri" bölümüne bakın) Nmap veri dizininin nselib alt dizininde saklanır, normalde scripts'in bulunduğu dizinle aynıdır. Shortport ve stdnse gibi modüller burada, adları .lua ile biten dosyalarda tutulur.

Using Customized Data Files (Özelleştirilmiş Veri Dosyalarını Kullanma)

Nmap veri dosyalarının herhangi biri veya tamamı, kullanıcının beğenisine göre özelleştirilmiş sürümlerle değiştirilebilir. Yalnızca bütün olarak değiştirilebilirler; çalışma zamanında orijinal dosyalarla birleştirilecek değişiklikleri belirleyemezsiniz. Nmap her bir dosyayı ararken, birçok dizinde isme göre arama yapar ve ilk bulduğunu seçer. Bu, Unix kabuğunuzun PATH'inizdeki dizinleri sırayla arayarak çalıştırılmasını istediğiniz programları bulmasına benzer. Aşağıdaki liste Nmap dizin arama sırasını vermektedir. Burada --datadir ile belirtilen dizinde bulunan bir nmap-services'in ~/.nmap/ dizininde bulunan bir nmap-services'e tercihen kullanılacağı gösterilmektedir, çünkü ilk önce bu dizin aranmaktadır.

Nmap veri dosyası dizin arama sırası

1. Eğer --datadir seçeneği belirtilmişse, argüman olarak verilen dizini kontrol eder.
2. NMAPDIR çevresel değişkeni ayarlanmışsa, bu dizini kontrol edin.
3. Eğer Nmap Windows üzerinde çalışmayıorsa, Nmap'i çalıştırın kullanıcının ~/.nmap dizininde arama yapın. Gerçek kullanıcı kimliğinin ev dizinini ve ardından farklılarsa etkin UID'leri dener.
4. Nmap ikili dosyasının bulunduğu dizini kontrol edin. Windows dışı platformlarda, ek olarak aynı dizini ../share/nmap eklenmiş olarak kontrol edin.
5. Derlenmiş NMAPDATADIR dizinini kontrol edin. Bu değer Windows üzerinde c:\nmap ve Unix üzerinde <\$prefix>/share/nmap olarak tanımlanır. <\$prefix>

varsayılan kaynak derlemesi için /usr/local ve Linux RPM'leri için /usr'dır. <\$prefix>, kaynak derlenirken ./configure dosyasına --prefix seçeneği verilerek değiştirilebilir.

Nmap, geçerli çalışma dizinindeki(.) dosyaları, kabuk çalışma PATH'inizde ilk olarak . görünmemesi gerektiği gibi aynı güvenlik nedenleriyle kontrol etmez. Paylaşılan bir sistemde, kötü niyetli bir kullanıcı /tmp gibi paylaşılan bir dizine sahte veri dosyaları yerleştirebilir. Bu dosyalar hatalı biçimlendirilmiş olabilir ve Nmap'in şikayet edip çıkışmasına neden olabilir ya da Nmap'in önemli bağlantı noktalarını atlamasına neden olabilir. Nmap bunu denerse, Nmap'i bu paylaşılan dizinde çalıştırın diğer kullanıcılar sahte sürümleri alırlar. Bu durum, Nmap'i yanlışlıkla nmap-services (ya da diğerlerinden biri) adlı bir dosyanın bulunduğu bir dizinde çalıştırıldığınızda da meydana gelebilir. Nmap'in geçerli dizini erken denemesini gerçekten isteyen kullanıcılar NMAPDIR ortam değişkenini . olarak ayarlayabilirler.

Bu liste, kullanıcıların bir dosyayı kendi özelleştirilmiş sürümleriyle nasıl değiştireceklerine karar verirken sahip oldukları birçok seçeneği göstermektedir. Genellikle tavsiye ettiğim seçenek, özelleştirilmiş dosyaları değişiklik için uygun şekilde adlandırılmış özel bir dizine yerleştirmektir. Örneğin, sadece en yaygın yüz portu içerecek şekilde soyulmuş bir nmap-services ~/nmap-fewports dizinine yerleştirilebilir. Daha sonra bu dizini --datadir seçeneği ile belirtin. Bu, özelleştirilmiş dosyaların yalnızca kasıtlı olarak kullanılmasını sağlar. Nmap çıktı-dosya formatları kullanılan Nmap komut satırını içerdiginden, daha sonra günlükleri incelerken hangi dosyaların kullanıldığını bileceksiniz.

Diğer bir seçenek de NMAPDATADIR içindeki orijinal dosyayı düzenlemektir. Bu nadiren önerilir, çünkü düzenlenen dosya muhtemelen Nmap'in bir sonraki yükseltilmesinde üzerine yazılacaktır. Ek olarak, bu, değiştirmelerinizin bir soruna neden olduğundan şüpheleniyorsanız orijinal dosyaları kullanmayı zorlaştırır. Bu aynı zamanda neyi değiştirdiğinizi hatırlamak için sürümünüzü orijinaliyle karşılaştırmayı da zorlaştırır.

Üçüncü bir seçenek ise özelleştirilmiş dosyaları Unix ~/.nmap dizininize yerleştirmektir. Tabii ki sadece değiştirdiğiniz dosyaları eklemelisiniz. Diğerleri her zamanki gibi NMAPDATADIR'den alınmaya devam edecktir. Bu çok kullanışlıdır, çünkü Nmap her çalıştırıldığınızda özelleştirilmiş dosyaları dolaylı olarak kullanacaktır. Bu aynı zamanda bir dezavantaj da olabilir. Kullanıcılar bazen dosyaların varlığını unuturlar. Nmap'i daha yeni veri dosyalarına sahip bir sürümme

yükselttiklerinde, `~/.nmap` içindeki eski kopyalar kullanılmaya devam edecek ve sonuçların kalitesini düşürecektir.

NMAPDIR ortam değişkenini dosyaların bulunduğu dizine ayarlamak başka bir alternatifdir. Bu, Nmap'in yeni bir sürümünü test ederken faydalı olabilir. Diyelim ki Nmap 5.21 sürümünü edindiniz, büyük değişiklik listesini fark ettiniz ve mevcut bilinen çalışma sürümünüzü değiştirmeden önce test etmeye karar verdiniz. Bu sürümü `~/src/nmap-5.21` dosyasında derleyebilirsiniz, ancak orada çalıştırıldığınızda Nmap veri dosyalarını `/usr/local/share/nmap` dosyasından okumaya çalışır. Nmap 5.21 henüz kurulmadığı için bunlar eski sürümlerdir. NMAPDIR'i `~/src/nmap-5.21` olarak ayarlayın, gönlünüzce test edin ve ardından `make install`'ı gerçekleştirin. NMAPDIR'i düzenli olarak kullanmanın bir dezavantajı, dizin adının `--datadir` kullanıldığından olduğu gibi Nmap çıktı dosyalarına kaydedilmemesidir.

Chapter 15. Nmap Reference Guide (Bölüm 15. Nmap Referans Kılavuzu)

- Description (Açıklama)
- Options Summary (Seçenekler Özeti)
- Target Specification (Hedef Belirleme)
- Host Discovery (Ana Bilgisayar Bulma)
- Port Scanning Basics (Bağlantı Noktası Tarama Temelleri)
- Port Scanning Techniques (Bağlantı Noktası Tarama Teknikleri)
- Port Specification and Scan Order (Bağlantı Noktası Belirleme ve Tarama Sırası)
- Service and Version Detection (Hizmet ve Sürüm Algılama)
- OS Detection (İşletim Sistemi Algılama)
- Nmap Scripting Engine (NSE) (Nmap Scripting Engine (NSE))
- Timing and Performance (Zamanlama ve Performans)

- Firewall/IDS Evasion and Spoofing (Güvenlik Duvarı/IDS Kaçırma ve Aldatma)
- Output (Çıktı)
- Miscellaneous Options (Çeşitli Seçenekler)
- Runtime Interaction (Çalışma Zamanı)
- Examples (Etkileşim Örnekleri)
- Nmap Book (Nmap Kitabı)
- Bugs (Hatalar)
- Authors (Yazarlar)
- Legal Notices (Yasal Bildirimler)
 - Nmap Copyright and Licensing (Nmap Telif Hakkı ve Lisanslama)
 - Creative Commons License for this Nmap Guide (Bu Nmap Kılavuzu için Creative Commons Lisansı)
 - Source Code Availability and Community Contributions (Kaynak Kodu Kullanılabilirliği ve Topluluk Katkıları)
 - No Warranty (Garanti Yok)
 - Inappropriate Usage (Uygunsuz Kullanım)
 - Third-Party Software and Funding Notices (Üçüncü Taraf Yazılım ve Finansman Bildirimleri)
 - United States Export Control (Amerika Birleşik Devletleri İhracat Kontrolü)

Description (Açıklama)

Name

nmap - Ağ keşif aracı ve güvenlik / bağlantı noktası tarayıcısı

Synopsis

```
nmap [ <Scan Type> ...] [ <Options> ] { <target specification> }
```

Not : Bu belge, Nmap'in <https://nmap.org/download.html> veya <https://nmap.org/dist/?C=M&O=D> adreslerinde bulunan en son sürümünü açıklamaktadır. Bir özelliğin açıklanıldığı gibi çalışmadığını bildirmeden önce lütfen en son sürümü kullandığınızdan emin olun.

Nmap ("Network Mapper") ağ keşfi ve güvenlik denetimi için açık kaynaklı bir araçtır. Tek ana bilgisayarlara karşı iyi çalışmasına rağmen, büyük ağları hızla taramak için tasarlanmıştır. Nmap, ağda hangi ana bilgisayarların bulunduğu, bu ana bilgisayarların hangi hizmetleri (uygulama adı ve sürümü) sunduğunu, hangi işletim sistemlerini (ve işletim sistemi sürümlerini) çalıştırdıklarını, ne tür paket filtrelerinin / güvenlik duvarlarının kullanıldığını ve düzinecece başka özelliği belirlemek için ham IP paketlerini yeni yollarla kullanır. Nmap genellikle güvenlik denetimleri için kullanılsa da, birçok sistem ve ağ yöneticisi, ağ envanteri, hizmet yükseltme programlarını yönetme ve ana bilgisayar veya hizmet çalışma süresini izleme gibi rutin görevler için yararlı bulmaktadır.

Nmap'in çıktısı, kullanılan seçeneklere bağlı olarak her biri hakkında ek bilgiler içeren taranan hedeflerin bir listesidir. Bu bilgilerin başında "ilginç portlar tablosu" gelir. Bu tabloda bağlantı noktası numarası ve protokolü, hizmet adı ve durumu listelenir. Durum açık, filtrelenmiş, kapalı veya filtrelenmemiş şeklidir. Açık, hedef makinedeki bir uygulamanın o bağlantı noktasındaki bağlantıları/paketleri dinlediği anlamına gelir. Filtrelenmiş, bir güvenlik duvari, filtre veya başka bir ağ engelinin bağlantı noktasını engellediği anlamına gelir, böylece Nmap açık veya kapalı olup olmadığını anlayamaz. Kapalı bağlantı noktalarında onları dinleyen hiçbir uygulama yoktur, ancak herhangi bir zamanda açılabilirler. Portlar Nmap'in probleme yanıt verdiğiinde filtrelenmemiş olarak sınıflandırılır, ancak Nmap bunların açık mı yoksa kapalı mı olduğunu belirleyemez. Nmap, iki durumdan hangisinin bir bağlantı noktasını tanımladığını belirleyemediğinde açık|filtreli ve kapalı|filtreli durum kombinasyonlarını raporlar. Port tablosu, versiyon tespiti istendiğinde yazılım versiyon detaylarını da içerebilir. Bir IP protokol taraması istendiğinde (-sO), Nmap dinleme portları yerine desteklenen IP protokollerini hakkında bilgi sağlar.

Nmap, ilginç bağlantı noktaları tablosuna ek olarak, ters DNS adları, işletim sistemi tahminleri, cihaz türleri ve MAC adresleri dahil olmak üzere hedefler hakkında daha fazla bilgi sağlayabilir.

Tipik bir Nmap taraması Örnek 15.1'de gösterilmektedir. Bu örnekte kullanılan tek Nmap argümanı işletim sistemi ve sürüm tespiti, komut dosyası taraması ve traceroute'u etkinleştirmek için -A; daha hızlı yürütme için -T4; ve ardından ana bilgisayar adıdır.

Örnek 15.1. Temsili bir Nmap taraması

```
# nmap -A -T4 scanme.nmap.org

Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE    SERVICE      VERSION
22/tcp    open     ssh          OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
| ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)
|_ 2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp    open     http         Apache httpd 2.2.14 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
646/tcp   filtered ldp
1720/tcp  filtered H.323/Q.931
9929/tcp  open     nping-echo  Nping echo
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Network Distance: 11 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel

TRACEROUTE (using port 53/tcp)
HOP RTT      ADDRESS
[Cut first 10 hops for brevity]
11  17.65 ms  li86-221.members.linode.com (74.207.244.221)

Nmap done: 1 IP address (1 host up) scanned in 14.40 seconds
```

Nmap'in en yeni sürümü <https://nmap.org> adresinden edinilebilir. Bu man sayfasının en yeni sürümü <https://nmap.org/book/man.html> adresinde mevcuttur. Ayrıca Nmap Ağ Tarama kitabımda bir bölüm olarak yer almaktadır: Ağ Keşfi ve Güvenlik Taraması için Resmi Nmap Projesi Kılavuzu.

Options Summary (Seçenekler Özeti)

Bu seçenek özet, Nmap argüman olmadan çalıştırıldığında yazdırılır ve en son sürüm her zaman <https://svn.nmap.org/nmap/docs/nmap.usage.txt> adresinde mevcuttur. İnsanların en yaygın seçenekleri hatırlamasına yardımcı olur, ancak bu kılavuzun geri kalanındaki derinlemesine belgelerin yerini tutmaz. Bazı belirsiz seçenekler buraya dahil bile edilmemiştir.

Nmap 7.93SVN (<https://nmap.org>)

Usage: nmap [Scan Type(s)] [Options] {target specification}

TARGET SPECIFICATION:

Can pass hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iL <inputfilename>: Input from list of hosts/networks

-iR <num hosts>: Choose random targets

--exclude <host1[,host2][,host3],...>: Exclude hosts/networks

--excludefile <exclude_file>: Exclude list from file

HOST DISCOVERY:

-sL: List Scan - simply list targets to scan

-sn: Ping Scan - disable port scan

-Pn: Treat all hosts as online -- skip host discovery

-PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given port

s

-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes

-PO[protocol list]: IP Protocol Ping

-n/-R: Never do DNS resolution/Always resolve [default: sometimes]

--dns-servers <serv1[,serv2],...>: Specify custom DNS servers

--system-dns: Use OS's DNS resolver

--traceroute: Trace hop path to each host

SCAN TECHNIQUES:

-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans

-sU: UDP Scan

-sN/sF/sX: TCP Null, FIN, and Xmas scans

--scanflags <flags>: Customize TCP scan flags

-sI <zombie host[:probeport]>: Idle scan

-sY/sZ: SCTP INIT/COOKIE-ECHO scans

-sO: IP protocol scan

-b <FTP relay host>: FTP bounce scan

POR T SPECIFICATION AND SCAN ORDER:

- p <port ranges>: Only scan specified ports
Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
- exclude-ports <port ranges>: Exclude the specified ports from scanning
- F: Fast mode - Scan fewer ports than the default scan
- r: Scan ports sequentially - don't randomize
- top-ports <number>: Scan <number> most common ports
- port-ratio <ratio>: Scan ports more common than <ratio>

SERVICE/VERSION DETECTION:

- sV: Probe open ports to determine service/version info
- version-intensity <level>: Set from 0 (light) to 9 (try all probes)
- version-light: Limit to most likely probes (intensity 2)
- version-all: Try every single probe (intensity 9)
- version-trace: Show detailed version scan activity (for debugging)

SCRIPT SCAN:

- sC: equivalent to --script=default
- script=<Lua scripts>: <Lua scripts> is a comma separated list of directories, script-files or script-categories
- script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
- script-args-file=filename: provide NSE script args in a file
- script-trace: Show all data sent and received
- script-updatedb: Update the script database.
- script-help=<Lua scripts>: Show help about scripts.

 <Lua scripts> is a comma-separated list of script-files or script-categories.

OS DETECTION:

- O: Enable OS detection
- osscan-limit: Limit OS detection to promising targets
- osscan-guess: Guess OS more aggressively

TIMING AND PERFORMANCE:

- Options which take <time> are in seconds, or append 'ms' (milliseconds), 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
- T<0-5>: Set timing template (higher is faster)
- min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
- min-parallelism/max-parallelism <numprobes>: Probe parallelization
- min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies

probe round trip time.

--max-retries <tries>: Caps number of port scan probe retransmissions.

--host-timeout <time>: Give up on target after this long

--scan-delay/--max-scan-delay <time>: Adjust delay between probes

--min-rate <number>: Send packets no slower than <number> per second

--max-rate <number>: Send packets no faster than <number> per second

FIREWALL/IDS EVASION AND SPOOFING:

-f; --mtu <val>: fragment packets (optionally w/given MTU)

-D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys

-S <IP_Address>: Spoof source address

-e <iface>: Use specified interface

-g/--source-port <portnum>: Use given port number

--proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies

--data <hex string>: Append a custom payload to sent packets

--data-string <string>: Append a custom ASCII string to sent packets

--data-length <num>: Append random data to sent packets

--ip-options <options>: Send packets with specified ip options

--ttl <val>: Set IP time-to-live field

--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address

--badsum: Send packets with a bogus TCP/UDP/SCTP checksum

OUTPUT:

-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rlpt klddi3, and Grepable format, respectively, to the given filename.

-oA <basename>: Output in the three major formats at once

-v: Increase verbosity level (use -vv or more for greater effect)

-d: Increase debugging level (use -dd or more for greater effect)

--reason: Display the reason a port is in a particular state

--open: Only show open (or possibly open) ports

--packet-trace: Show all packets sent and received

--iflist: Print host interfaces and routes (for debugging)

--append-output: Append to rather than clobber specified output files

--resume <filename>: Resume an aborted scan

--noninteractive: Disable runtime interactions via keyboard

--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML

--webxml: Reference stylesheet from Nmap.Org for more portable XML

--no-stylesheet: Prevent associating of XSL stylesheet w/XML output

MISC:

- 6: Enable IPv6 scanning
- A: Enable OS detection, version detection, script scanning, and traceroute
- datadir <dirname>: Specify custom Nmap data file location
- send-eth/--send-ip: Send using raw ethernet frames or IP packets
- privileged: Assume that the user is fully privileged
- unprivileged: Assume the user lacks raw socket privileges
- V: Print version number
- h: Print this help summary page.

EXAMPLES:

```
nmap -v -A scanme.nmap.org  
nmap -v -sn 192.168.0.0/16 10.0.0.0/8  
nmap -v -iR 10000 -Pn -p 80
```

SEE THE MAN PAGE (<https://nmap.org/book/man.html>) FOR MORE OPTIONS AND EXAMPLES

Target Specification (Hedef Belirleme)

Nmap komut satırında bir seçenek (veya seçenek argümanı) olmayan her şey bir hedef ana bilgisayar belirtimi olarak değerlendirilir. En basit durum, tarama için bir hedef IP adresi veya ana bilgisayar adı belirtmektir.

Bir ana bilgisayar adı hedef olarak verildiğinde, taranacak IP adresini belirlemek için Alan Adı Sistemi (DNS) aracılığıyla çözümlenir. İsim birden fazla IP adresine çözümlenirse, yalnızca ilki taranacaktır. Nmap'in yalnızca ilk adres yerine çözümlenen tüm adresleri taramasını sağlamak için --resolve-all seçeneğini kullanın.

Bazen bitişik ana bilgisayarlardan oluşan bir ağın tamamını taramak istersiniz. Bunun için Nmap CIDR tarzı adreslemeyi destekler. Bir IP adresine veya ana bilgisayar adına /<numbits> ekleyebilirsiniz ve Nmap, ilk <numbits>'in verilen referans IP veya ana bilgisayar adıyla aynı olduğu her IP adresini tarayacaktır. Örneğin, 192.168.10.0/24, 192.168.10.0 (ikili: 11000000 10101000 00001010 00000000) ile 192.168.10.255 (ikili: 11000000 10101000 00001010 11111111)

arasındaki 256 ana bilgisayarı tarayacaktır. 192.168.10.40/24 tam olarak aynı hedefleri tarayacaktır. Scanme.nmap.org ana bilgisayarının 64.13.134.52 IP adresinde olduğu düşünüldüğünde, scanme.nmap.org/16 belirtimi 64.13.0.0 ile 64.13.255.255 arasındaki 65.536 IP adresini tarayacaktır. İzin verilen en küçük değer, tüm Internet'i hedefleyen /0'dır. IPv4 için en büyük değer /32'dir ve tüm adres bitleri sabit olduğu için yalnızca adlandırılmış ana bilgisayarı veya IP adresini tarar. IPv6 için en büyük değer /128'dir ve aynı şeyi yapar.

CIDR notasyonu kısadır ancak her zaman yeterince esnek değildir. Örneğin, 192.168.0.0/16 adresini taramak isteyebilirsiniz ancak .0 veya .255 ile biten IP'leri atlayabilirsiniz çünkü bunlar alt ağ ve yayın adresleri olarak kullanılabilir. Nmap bunu oktet aralığı adresleme yoluyla destekler. Normal bir IP adresi belirtmek yerine, her oktet için virgülle ayrılmış bir sayı veya aralık listesi belirtebilirsiniz. Örneğin, 192.168.0-255.1-254, aralıktaki .0 veya .255 ile biten tüm adresleri atlar ve 192.168.3-5,7.1, 192.168.3.1, 192.168.4.1, 192.168.5.1 ve 192.168.7.1 adreslerini tarar. Bir aralığın her iki tarafı da atlanabilir; varsayılan değerler solda 0 ve sağda 255'tir. Tek başına - kullanmak 0-255 ile aynıdır, ancak hedef belirtiminin bir komut satırı seçeneği gibi görünmemesi için ilk sekizlide 0- kullanmayı unutmayın. Aralıkların son oktetlerle sınırlı olması gerekmek: 0-255.0-255.13.37 belirteci, 13.37 ile biten tüm IP adresleri için Internet çapında bir tarama gerçekleştirecektir. Bu tür geniş örnekleme Internet anketleri ve araştırmaları için yararlı olabilir.

IPv6 adresleri, tam nitelikli IPv6 adresleri veya ana bilgisayar adlarıyla veya alt ağlar için CIDR gösterimiyle belirtilebilir. Oktet aralıkları henüz IPv6 için desteklenmemektedir.

Küresel olmayan kapsama sahip IPv6 adreslerinin bir bölge kimliği sonekine sahip olması gereklidir. Unix sistemlerinde bu, bir arayüz adının ardından gelen yüzde işaretidir; tam bir adres fe80::a8bb:ccff:fedd:eff%eth0 olabilir. Windows'ta, arabirim adı yerine arabirim dizin numarası kullanın: fe80::a8bb:ccff:fedd:eff%1. Arayüz dizinlerinin bir listesini netsh.exe interface ipv6 show interface komutunu çalıştırarak görebilirsiniz.

Nmap, komut satırında birden fazla ana bilgisayar belirtimini kabul eder ve bunların aynı türde olması gerekmekz. nmap scanme.nmap.org 192.168.0.0/8 10.0.0,1,3-7.- komutu beklediğiniz şeyi yapar.

Hedefler genellikle komut satırlarında belirtilmekle birlikte, hedef seçimini kontrol etmek için aşağıdaki seçenekler de mevcuttur:

-iL <inputfilename> (Listeden girdi) ⇒ Hedef özelliklerini <inputfilename>'den okur.

Büyük bir ana bilgisayar listesi iletmek komut satırında genellikle gariptir, ancak bu yaygın bir istektir. Örneğin, DHCP sunucunuz taramak istediğiniz 10.000 güncel kiralama listesini dışa aktarabilir. Ya da yetkisiz statik IP adresleri kullanan ana bilgisayarları bulmak için bunlar dışındaki tüm IP adreslerini taramak isteyebilirsiniz. Basitçe taranacak ana bilgisayarların listesini oluşturun ve bu dosya adını -iL seçeneğine argüman olarak Nmap'e iletin. Girişler, Nmap tarafından komut satırında kabul edilen biçimlerden herhangi birinde olabilir (IP adresi, ana bilgisayar adı, CIDR, IPv6 veya sekizli aralıklar). Her girdi bir veya daha fazla boşluk, sekme veya satırsonu ile ayrılmalıdır. Nmap'in ana bilgisayarları gerçek bir dosya yerine standart girdiden okumasını istiyorsanız dosya adı olarak bir kısa çizgi (-) belirtebilirsiniz.

Girdi dosyası # ile başlayan ve satır sonuna kadar uzanan yorumlar içerebilir.

-iR <num hosts> (Rastgele hedefler seçin) ⇒ İnternet çapında anketler ve diğer araştırmalar için hedefleri rastgele seçmek isteyebilirsiniz. <num hosts> argümanı Nmap'e kaç tane IP üreteceğini söyler. Belirli özel, çok noktaya yayın veya tahsis edilmemiş adres aralıklarında bulunanlar gibi istenmeyen IP'ler otomatik olarak atlanır. Hiç bitmeyen bir tarama için 0 argümanı belirtilebilir. Bazı ağ yöneticilerinin ağlarının izinsiz taranmasından rahatsız olabileceğini ve şikayet edebileceğini unutmayın. Bu seçeneği kendi sorumluluğunuza kullanın! Yağmurlu bir öğleden sonra canınız gerçekten sıkılırsa, nmap -Pn -sS -p 80 -iR 0 --open komutunu deneyerek rastgele web sunucularına göz atabilirsiniz.

--exclude <host1> [, <host2> [...]] (Exclude hosts/networks) (Ana bilgisayarları/ağları hariç tut) ⇒ Belirttiğiniz genel ağ aralığının bir parçası olsalar bile taramanın dışında tutulacak hedeflerin virgülle ayrılmış bir listesini belirtir. İlettiğiniz liste normal Nmap sözdizimini kullanır, bu nedenle ana bilgisayar adları, CIDR net blokları, sekizli aralıkları vb. içerebilir. Bu, taramak istediğiniz ağ dokunulmaz görev açısından kritik sunucular, bağlantı noktası taramalarına olumsuz tepki verdiği bilinen sistemler veya başkaları tarafından yönetilen alt ağlar içinde yararlı olabilir.

--excludefile <exclude_file> (Listeyi dosyadan hariç tut) ⇒ Bu, --exclude seçeneği ile aynı işlevselligi sunar, ancak hariç tutulan hedefler komut satırı yerine satırsonu, boşluk veya sekmeyle ayrılmış bir <exclude_file> içinde sağlanır.

Dışlama dosyası # ile başlayan ve satır sonuna kadar uzanan yorumlar içerebilir.

-n (No DNS resolution) (DNS çözümünürlüğü yok) ⇒ Nmap'e bulduğu etkin IP adresleri üzerinde asla ters DNS çözümlemesi yapmamasını söyler. DNS, Nmap'in yerleşik paralel saplama çözümleyicisi ile bile yavaş olabileceğinden, bu seçenek tarama sürelerini kısaltabilir.

-R (DNS resolution for all targets) (tüm hedefler için DNS çözümünürlüğü) ⇒ Nmap'e hedef IP adreslerinde her zaman ters DNS çözümlemesi yapmasını söyler. Normalde ters DNS yalnızca duyarlı (çevrimiçi) ana bilgisayarlara karşı gerçekleştirilir.

--resolve-all (Scan each resolved address) (Çözülen her adresi tarayın) ⇒ Bir ana bilgisayar adı hedefi birden fazla adrese çözümlenirse, hepsini tarayın. Varsayılan davranış, yalnızca ilk çözümlenen adresi taramaktır. Ne olursa olsun, yalnızca uygun adres ailesindeki adresler taranacaktır: Varsayılan olarak IPv4, -6 ile IPv6.

--unique (Scan each address only once) (Her adresi yalnızca bir kez tarayın) ⇒ Her IP adresini yalnızca bir kez tarayın. Varsayılan davranış, ağ aralıkları çakıştığında veya farklı ana bilgisayar adları aynı adrese çözümlendiğinde olduğu gibi, her adresi hedef listesinde belirtildiği kadar taramaktır.

--system-dns (Use system DNS resolver) (Sistem DNS çözümleyicisini kullanın) ⇒ Varsayılan olarak Nmap, sorguları doğrudan ana bilgisayarınızda yapılandırılmış ad sunucularına göndererek ve ardından yanıtları dinleyerek IP adreslerini tersine çözer. Performansı artırmak için birçok istek (genellikle düzinelere) paralel olarak gerçekleştirilir. Bunun yerine sistem çözümleyicinizi kullanmak için bu seçeneği belirtin (getnameinfo çağrıları yoluyla her seferinde bir IP). Bu daha yavaştır ve Nmap paralel çözümleyicisinde bir hata bulmadığınız sürece nadiren kullanışlıdır (bulursanız lütfen bize bildirin). Sistem çözümleyicisi her zaman ileri aramalar için kullanılır (bir ana bilgisayar adından bir IP adresi almak).

--dns-servers <server1> [, <server2> [...]] (Servers to use for reverse DNS queries) ((Ters DNS sorguları için kullanılacak sunucular)) ⇒ Varsayılan olarak, Nmap DNS sunucularınızı (rDNS çözümlemesi için) resolv.conf dosyanızdan (Unix) veya Kayıt Defterinden (Win32) belirler. Alternatif olarak, alternatif sunucuları belirtmek için bu seçeneği kullanabilirsiniz. Eğer --system-dns kullanıyorsanız bu seçenek geçerli değildir. Birden fazla DNS sunucusu kullanmak, özellikle hedef IP alanınız için yetkili sunucular seçerseniz, genellikle daha hızlıdır. Bu seçenek aynı zamanda gizliliği de artırabilir, çünkü talepleriniz Internet üzerindeki hemen hemen tüm özyinelemeli DNS sunucularından geri dönebilir.

Bu seçenek özel ağları tararken de kullanışlıdır. Bazen sadece birkaç isim sunucusu uygun rDNS bilgisi sağlar ve nerede olduklarını bile bilmiyor olabilirsiniz. Ağ 53 numaralı bağlantı noktası için tarayabilir (belki de sürüm algılama ile), ardından çalışan bir tane bulana kadar --dns-servers ile her bir sunucusunu birer birer belirterek Nmap liste taramalarını (-sL) deneyebilirsiniz.

DNS yanıtı bir UDP paketinin boyutunu aşarsa bu seçenek kullanılamayabilir. Böyle bir durumda DNS çözümleyicimiz kesilmiş paketten bir yanıt çıkarmak için en iyi çabayı gösterecek ve başarılı olamazsa sistem çözümleyicisini kullanmaya geri dönecektir. Ayrıca, CNAME takma adları içeren yanıtlar sistem çözümleyicisine geri dönecektir.

Host Discovery (Ana Bilgisayar Bulma)

Herhangi bir ağ keşif görevinin ilk adımlarından biri, (bazen çok büyük) bir IP aralığı kümesini aktif veya ilginç ana bilgisayarların bir listesine indirmektir. Her bir IP adresinin her bir portunu taramak yavaş ve genellikle gereksizdir. Elbette bir ana bilgisayarı neyin ilginç kıldığı büyük ölçüde tarama amaçlarına bağlıdır. Ağ yöneticileri yalnızca belirli bir hizmeti çalıştırın ana bilgisayarlarla ilgilenebilirken, güvenlik denetçileri IP adresi olan her bir cihazla ilgilenebilir. Bir yönetici iç ağındaki ana bilgisayarları bulmak için sadece ICMP ping kullanarak rahat edebilirken, harici bir sızma test uzmanı güvenlik kısıtlamalarından kaçmak için dzinelerce probdan oluşan çeşitli bir set kullanabilir.

Ana bilgisayar bulma ihtiyaçları çok çeşitli olduğundan, Nmap kullanılan teknikleri özelleştirmek için çok çeşitli seçenekler sunar. Ana bilgisayar keşfi bazen ping taraması olarak adlandırılır, ancak her yerde bulunan ping aracıyla ilişkili basit ICMP yanıt istek paketlerinin çok ötesine geçer. Kullanıcılar bir liste taramasıyla (-sL) veya ana bilgisayar bulmayı devre dışı bırakarak (-Pn) bulma adımını tamamen atlayabilir veya çok portlu TCP SYN/ACK, UDP, SCTP INIT ve ICMP problemlerinin rastgele kombinasyonlarıyla ağ meşgul edebilir. Bu problemlerin amacı, bir IP adresinin gerçekten aktif olduğunu (bir ana bilgisayar veya ağ cihazı tarafından kullanıldığını) gösteren yanıtlar istemektir. Birçok ağda, IP adreslerinin yalnızca küçük bir yüzdesi herhangi bir zamanda aktiftir. Bu durum özellikle 10.0.0.0/8 gibi özel adres alanlarında yaygındır. Bu ağda 16 milyon IP var, ancak binden az

makineye sahip şirketler tarafından kullanıldığını gördüm. Ana bilgisayar keşfi, bu makineleri seyrek olarak tahsis edilmiş IP adresleri denizinde bulabilir.

Herhangi bir ana bilgisayar bulma seçeneği verilmezse, Nmap bir ICMP yankı isteği, 443 numaralı bağlantı noktasına bir TCP SYN paketi, 80 numaralı bağlantı noktasına bir TCP ACK paketi ve bir ICMP zaman damgası isteği gönderir. (IPv6 için ICMP zaman damgası isteği ICMPv6'nın bir parçası olmadığından atlanmıştır). Bu varsayılanlar -PE -PS443 -PA80 -PP seçeneklerine eşdeğerdir. Bunun istisnaları, yerel bir ethernet ağındaki herhangi bir hedef için kullanılan ARP (IPv4 için) ve Komşu Bulma (IPv6 için) taramalarıdır. Ayrıcalıksız Unix kabuk kullanıcıları için varsayılan problemler, connect sistem çağrısını kullanarak 80 ve 443 numaralı portlara gönderilen bir SYN paketidir. Bu ana bilgisayar keşfi yerel ağları tararken genellikle yeterlidir, ancak güvenlik denetimi için daha kapsamlı bir keşif sondası seti önerilir.

P* seçenekleri (ping türlerini seçer) birleştirilebilir. Farklı TCP portları/bayrakları ve ICMP kodları kullanarak birçok prob türü göndererek katı güvenlik duvarlarını aşma şansınızı artırabilirsiniz. Ayrıca, diğer -P* seçeneklerini belirtseniz bile ARP/Komşu Bulma işleminin varsayılan olarak yerel Ethernet ağındaki hedeflere karşı yapıldığını unutmayın, çünkü bu neredeyse her zaman daha hızlı ve daha etkilidir.

Varsayılan olarak, Nmap ana bilgisayar keşfi yapar ve ardından çevrimiçi olduğunu belirlediği her ana bilgisayara karşı bir bağlantı noktası taraması gerçekleştirir. Bu, UDP problemleri (-PU) gibi varsayılan olmayan ana bilgisayar bulma türlerini belirtseniz bile geçerlidir. Yalnızca ana bilgisayar bulma işleminin nasıl gerçekleştirileceğini öğrenmek için -sn seçeneği hakkında bilgi edinin veya ana bilgisayar bulmayı atlayıp tüm hedef adresleri port taraması yapmak için -Pn seçeneğini kullanın. Aşağıdaki seçenekler ana bilgisayar bulmayı kontrol eder:

-sL (List Scan) ⇒ Liste taraması, hedef ana bilgisayarlara herhangi bir paket göndermeden, belirtilen ağ(lar)daki her bir ana bilgisayarı listeleyen dejenere bir ana bilgisayar bulma biçimidir. Varsayılan olarak, Nmap hala ana bilgisayarların isimlerini öğrenmek için ters-DNS çözümlemesi yapar. Basit ana bilgisayar adlarının ne kadar yararlı bilgiler verdiği çoğu zaman şartlıdır. Örneğin, fw.chi bir şirketin Chicago güvenlik duvarının adıdır. Nmap ayrıca sonunda toplam IP adresi sayısını da bildirir. Liste taraması, hedefleriniz için doğru IP adreslerine sahip olduğunuzdan emin olmak için iyi bir güvenlik kontrolüdür. Ana bilgisayarlar tanımadığınız alan adlarını kullanıyorsa, yanlış şirketin ağını taramayı önlemek için

daha fazla araştırmaya değer.

Amaç sadece hedef ana bilgisayarların bir listesini yazdırmak olduğundan, bağlantı noktası tarama, işletim sistemi algılama veya ana bilgisayar bulma gibi daha üst düzey işlevler için seçenekler bununla birleştirilemez. Bu tür üst düzey işlevleri gerçekleştirmeye devam ederken ana bilgisayar bulmayı devre dışı bırakmak istiyorsanız, -Pn (ana bilgisayar bulmayı atla) seçeneğini okuyun.

-sn (No port scan) ⇒ Bu seçenek Nmap'e ana bilgisayar bulduktan sonra port taraması yapmamasını ve yalnızca ana bilgisayar bulma problemlerine yanıt veren mevcut ana bilgisayarları yazdırmasını söyler. Bu genellikle "ping taraması" olarak bilinir, ancak traceroute ve NSE ana bilgisayar komut dosyalarının çalıştırılmasını da isteyebilirsiniz. Bu, varsayılan olarak liste taramasından bir adım daha müdahalecidir ve genellikle aynı amaçlar için kullanılabilir. Fazla dikkat çekmeden hedef ağıda hafif bir keşif yapılmasını sağlar. Saldırganlar için kaç ana bilgisayarın açık olduğunu bilmek, her bir IP ve ana bilgisayar adının liste taramasıyla sağlanan listeden daha değerlidir.

Sistem yöneticileri de genellikle bu seçeneği değerli bulurlar. Bir agdaki mevcut makineleri saymak veya sunucu kullanılabilirliğini izlemek için kolayca kullanılabilir. Bu genellikle ping taraması olarak adlandırılır ve yayın adresine ping atmaktan daha güvenilirdir çünkü birçok ana bilgisayar yayın sorgularına yanıt vermez.

sn ile yapılan varsayılan ana bilgisayar keşfi, varsayılan olarak bir ICMP eko isteği, 443 numaralı bağlantı noktasına TCP SYN, 80 numaralı bağlantı noktasına TCP ACK ve bir ICMP zaman damgası isteğinden oluşur. Ayrıcalıksız bir kullanıcı tarafından çalıştırıldığında, hedef üzerindeki 80 ve 443 numaralı bağlantı noktalarına yalnızca SYN paketleri gönderilir (bir bağlantı çağrısı kullanılarak). Ayrıcalıklı bir kullanıcı yerel bir ethernet ağı üzerindeki hedefleri taramaya çalıştığında, --send-ip belirtildiği sürece ARP istekleri kullanılır. Daha fazla esneklik için -sn seçeneği keşif probu türlerinden herhangi biriyle (-P* seçenekleri) birleştirilebilir. Bu prob tipi ve port numarası seçeneklerinden herhangi biri kullanılırsa, varsayılan problemler geçersiz kılınır. Nmap çalıştırılan kaynak ana bilgisayar ile hedef ağ arasında sıkı güvenlik duvarları varsa, bu gelişmiş tekniklerin kullanılması önerilir. Aksi takdirde, güvenlik duvarı problemleri veya yanıtlarını düşürdüğünde ana bilgisayarlar gözden kaçabilir.

Nmap'in önceki sürümlerinde -sn, -sP olarak biliniyordu.

-Pn (No ping) ⇒ Bu seçenek ana bilgisayar keşif aşamasını tamamen atlar.

Normalde, Nmap bu aşamayı daha ağır tarama için aktif makineleri belirlemek ve ağın hızını ölçmek için kullanır. Varsayılan olarak, Nmap yalnızca açık olduğu tespit edilen ana bilgisayarlara karşı bağlantı noktası taramaları, sürüm algılama veya işletim sistemi algılama gibi ağır problema gerçekleştirir. Ana bilgisayar keşfinin -Pn ile devre dışı bırakılması, Nmap'in belirtilen her hedef IP adresine karşı istenen tarama işlevlerini denemesine neden olur. Dolayısıyla, komut satırında /16 boyutunda bir ağ belirtilirse, 65.536 IP adresinin tümü taranır. Liste taramasında olduğu gibi uygun ana bilgisayar keşfi atlanır, ancak hedef listesini durdurmak ve yazdırma yerine, Nmap her hedef IP aktifmiş gibi istenen işlevleri yerine getirmeye devam eder. Varsayılan zamanlama parametreleri kullanılır, bu da daha yavaş taramalara neden olabilir. NSE'nin çalışmasına izin verirken ana bilgisayar bulma ve bağlantı noktası taramasını atlamak için -Pn -sn seçeneklerini birlikte kullanın.

Yerel bir ethernet ağındaki makineler için ARP taraması yapılmaya devam edecektir (--disable-arp-ping veya --send-ip belirtildiği sürece) çünkü Nmap hedef ana bilgisayarı daha fazla taramak için MAC adreslerine ihtiyaç duyar. Nmap'in önceki sürümlerinde -Pn, -P0 ve -PN idi.

-PS <port list> (TCP SYN Ping) ⇒ Bu seçenek SYN bayrağı ayarlanmış boş bir TCP paketi gönderir. Varsayılan hedef bağlantı noktası 80'dir (nmap.h dosyasında DEFAULT_TCP_PROBE_PORT_SPEC değiştirilerek derleme zamanında yapılandırılabilir). Alternatif portlar parametre olarak belirtilebilir. Söz dizimi -p ile aynıdır, ancak T: gibi bağlantı noktası türü belirleyicilerine izin vermez. Örnekler -PS22 ve -PS22-25,80,113,1050,35000'dir. PS ile port listesi arasında boşluk olamayacağına dikkat edin. Birden fazla prob belirtilirse, bunlar paralel olarak gönderilecektir.

SYN bayrağı, uzak sisteme bir bağlantı kurmaya çalıştığını gösterir. Normalde hedef port kapatılır ve bir RST (reset) paketi geri gönderilir. Eğer port açıksa, hedef bir SYN/ACK TCP paketi ile yanıt vererek TCP üç yönlü el sıkışmasının ikinci adımını atacaktır. Nmap çalıştırılan makine daha sonra üç yönlü el sıkışmayı tamamlayacak ve tam bir bağlantı kuracak bir ACK paketi göndermek yerine bir RST ile yanıt vererek yeni oluşan bağlantıyı koparır. RST paketi Nmap'in kendisi tarafından değil, beklenmedik SYN/ACK'ye yanıt olarak Nmap'i çalıştırılan makinenin çekirdeği tarafından gönderilir.

Nmap portun açık ya da kapalı olmasına ilgilenmez. Daha önce tartışılan RST veya SYN/ACK yanıtı Nmap'e ana bilgisayarın kullanılabilir ve yanıt verebilir olduğunu söyler.

Unix kutularında, yalnızca ayrıcalıklı kullanıcı root genellikle ham TCP paketleri gönderebilir ve alabilir. Ayrıcalıksız kullanıcılar için, her hedef porta karşı connect sistem çağrısının başlatıldığı bir geçici çözüm otomatik olarak kullanılır. Bu, bir bağlantı kurma girişimi olarak hedef ana bilgisayara bir SYN paketi gönderme etkisine sahiptir. Eğer connect hızlı bir başarı veya ECONNREFUSED hatası ile dönerse, temel TCP yiğini bir SYN/ACK veya RST almış olmalıdır ve ana bilgisayar kullanılabilir olarak işaretlenir. Bağlantı girişimi bir zaman aşımına ulaşılana kadar aşağıda kalırsa, ana bilgisayar kapalı olarak işaretlenir.

-PA <port list> (TCP ACK Ping) ⇒ TCP ACK pingi, az önce tartışılan SYN pingine oldukça benzer. Aradaki fark, tahmin edebileceğiniz gibi, SYN bayrağı yerine TCP ACK bayrağının ayarlanmış olmasıdır. Böyle bir ACK paketi, kurulmuş bir TCP bağlantısı üzerinden veriyi onayladığını iddia eder, ancak böyle bir bağlantı yoktur. Bu nedenle, uzak ana bilgisayarlar her zaman bir RST paketi ile yanıt vermelii ve bu süreçte varlıklarını ifşa etmeliidir.

PA seçeneği SYN probu ile aynı varsayılan portu (80) kullanır ve aynı formatta bir hedef port listesi de alabilir. Ayrıcalıksız bir kullanıcı bunu denerse, daha önce tartışılan connect geçici çözümü kullanılır. Bu geçici çözüm kusurludur çünkü connect aslında bir ACK yerine bir SYN paketi göndermektedir.

Hem SYN hem de ACK ping problemleri sunmanın nedeni, güvenlik duvarlarını aşma şansını en üst düzeye çıkarmaktır. Birçok yönetici yönlendiricileri ve diğer basit güvenlik duvarlarını, şirket web sitesi veya posta sunucusu gibi genel hizmetlere yönelik olanlar dışında gelen SYN paketlerini engelleyecek şekilde yapılandırır. Bu, kuruluşu gelen diğer bağlantıları engellerken, kullanıcıların Internet'e engelsiz giden bağlantılar yapmasına izin verir. Bu durum bilgisi içermeyen yaklaşım güvenlik duvarı/yönlendirici üzerinde çok az kaynak kaplar ve donanım ve yazılım filtreleri tarafından yaygın olarak desteklenir. Linux Netfilter/iptables güvenlik duvarı yazılımı, bu durum bilgisi olmayan yaklaşımı uygulamak için --syn kolaylık seçeneğini sunar. Bunun gibi durumsuz güvenlik duvari kuralları mevcut olduğunda, SYN ping problemleri (-PS) kapalı hedef portlara gönderildiğinde muhtemelen engellenecektir. Bu gibi durumlarda, ACK probu bu kuralları doğrudan aştığı için parlar.

Bir başka yaygın güvenlik duvarı türü de beklenmedik paketleri düşüren durum bilgisi kurallarını kullanır. Bu özellik başlangıçta çoğunlukla üst düzey güvenlik duvarlarında bulunmaktaydı, ancak yıllar içinde çok daha yaygın hale geldi. Linux Netfilter/iptables sistemi bunu, paketleri bağlantı durumuna göre kategorize eden -state seçeneği ile destekler. Beklenmedik ACK paketleri genellikle sahte olarak algılanıp düşürüldüğünden, bir SYN probunun böyle bir sisteme karşı çalışması daha olasıdır. Bu ikileme bir çözüm, -PS ve -PA belirterek hem SYN hem de ACK problemleri göndermektir.

-PU <port list> (UDP Ping) ⇒ Bir başka ana bilgisayar bulma seçeneği de, verilen bağlantı noktalarına bir UDP paketi gönderen UDP ping'dir. Çoğu port için paket boş olacaktır, ancak bazıları yanıt alma olasılığı daha yüksek olan protokole özgü bir yük kullanır. Yükler, hizmet ve sürüm tespitinde kullanılan problarla aynıdır ve nmap-service-probes dosyasında tanımlanmıştır. Paket içeriği --data, --data-string ve --data-length seçenekleri ile de etkilenebilir.

Bağlantı noktası listesi, daha önce tartışılan -PS ve -PA seçenekleriyle aynı biçimde alınır. Herhangi bir port belirtilmezse, varsayılan 40125'tir. Bu varsayılan, nmap.h dosyasında DEFAULT_UDP_PROBE_PORT_SPEC değiştirilerek derleme zamanında yapılandırılabilir. Varsayılan olarak çok yaygın olmayan bir bağlantı noktası kullanılır, çünkü açık bağlantı noktalarına göndermek bu özel tarama türü için genellikle istenmeyen bir durumdur.

Hedef makinede kapalı bir bağlantı noktasına ulaşıldığında, UDP probu karşılığında bir ICMP bağlantı noktası ulaşılamaz paketi ortaya çıkarmalıdır. Bu, Nmap'e makinenin açık ve kullanılabilir olduğunu gösterir. Ana bilgisayar/ağ ulaşılamıyor veya TTL aşındı gibi diğer birçok ICMP hatası türü, kapalı veya ulaşılamayan bir ana bilgisayarın göstergesidir. Yanıt alınamaması da bu şekilde yorumlanır. Açık bir bağlantı noktasına ulaşılırsa, çoğu hizmet boş paketi yok sayar ve herhangi bir yanıt döndürmez. Bu nedenle varsayılan prob portu 40125'tir ve bu portun kullanımda olma ihtimali oldukça düşüktür. Karakter Oluşturucu (chargen) protokolü gibi birkaç hizmet boş bir UDP paketine yanıt verir ve böylece Nmap'e makinenin kullanılabilir olduğunu bildirir.

Bu tarama türünün birincil avantajı, yalnızca TCP'yi tarayan güvenlik duvarlarını ve filtreleri atlamasıdır. Örneğin, bir zamanlar Linksys BEFW11S4 kablosuz geniş bant yönlendiricim vardı. Bu cihazın harici arayüzü varsayılan olarak tüm TCP portlarını

filtreliyordu, ancak UDP problemleri yine de porta ulaşılamıyor mesajlarını ortaya çıkarıyor ve böylece cihazı ele veriyordu.

-PY <port list> (SCTP INIT Ping) ⇒ Bu seçenek, minimal bir INIT yiğini içeren bir SCTP paketi gönderir. Varsayılan hedef bağlantı noktası 80'dir (nmap.h dosyasında DEFAULT_SCTP_PROBE_PORT_SPEC değiştirilerek derleme zamanında yapılandırılabilir). Alternatif portlar parametre olarak belirtilebilir. Sözdizimi -p ile aynıdır, ancak S: gibi bağlantı noktası türü belirleyicilerine izin vermez. Örnekler -PY22 ve -PY22,80,179,5060'dır. PY ile port listesi arasında boşluk olamayacağına dikkat edin. Birden fazla prob belirttilirse, bunlar paralel olarak gönderilecektir.

INIT yiğini, uzak sisteme bir ilişki kurmaya çalıştığını gösterir. Normalde hedef port kapalı olacaktır ve bir ABORT yiğini geri gönderilecektir. Eğer port açıksa, hedef bir INIT-ACK yiğiniyla yanıt vererek SCTP dört yönlü el sıkışmasının ikinci adımını atacaktır. Nmap çalıştırılan makine işlevsel bir SCTP yiğinına sahipse, dört yönlü el sıkışmanın bir sonraki adımı olan COOKIE-ECHO yiğinini göndermek yerine bir ABORT yiğinıyla yanıt vererek yeni oluşan ilişkiyi koparır. ABORT paketi Nmap'in kendisi tarafından değil, beklenmedik INIT-ACK'e yanıt olarak Nmap'i çalıştırılan makinenin çekirdeği tarafından gönderilir.

Nmap, portun açık ya da kapalı olmasına ilgilenmez. Daha önce tartışılan ABORT veya INIT-ACK yanıtı Nmap'e ana bilgisayarın kullanılabilir ve yanıt verebilir olduğunu söyler.

Unix kutularında, yalnızca ayrıcalıklı kullanıcı root genellikle ham SCTP paketleri gönderebilir ve alabilir. SCTP INIT Ping'lerini kullanmak şu anda ayrıcalıksız kullanıcılar için mümkün değildir.

-PE ; -PP ; -PM (ICMP Ping Types) ⇒ Daha önce tartışılan alışılmadık TCP, UDP ve SCTP ana bilgisayar keşif türlerine ek olarak, Nmap her yerde bulunan ping programı tarafından gönderilen standart paketleri gönderebilir. Nmap hedef IP adreslerine bir ICMP tip 8 (yankı isteği) paketi gönderir ve karşılığında mevcut ana bilgisayarlardan bir tip 0 (yankı yanıtı) bekler. Ne yazık ki ağ kaşifleri için, birçok ana bilgisayar ve güvenlik duvarı artık RFC 1122'nin gerektirdiği şekilde yanıt vermek yerine bu paketleri engelliyor. Bu nedenle, yalnızca ICMP taramaları Internet üzerinden bilinmeyen hedeflere karşı nadiren yeterince güvenilirdir. Ancak dahili bir ağı izleyen sistem yöneticileri için pratik ve verimli bir yaklaşım olabilir. Bu yanıt isteği davranışını etkinleştirmek için -PE seçeneğini kullanın.

Eko isteği standart ICMP ping sorgusu olsa da, Nmap bununla yetinmez. ICMP standartları (RFC 792 ve RFC 950) ayrıca zaman damgası isteği, bilgi isteği ve adres maskesi isteği paketlerini sırasıyla 13, 15 ve 17 kodları olarak belirtir. Bu sorguların görünürdeki amacı adres maskeleri ve geçerli saatler gibi bilgileri öğrenmek olsa da, ana bilgisayar keşfi için kolayca kullanılabilirler. Yanıt veren bir sistem çalışır ve kullanılabilir durumdadır. Nmap, yaygın olarak desteklenmedikleri için şu anda bilgi istek paketlerini uygulamamaktadır. RFC 1122, "bir ana bilgisayarın bu mesajları uygulamaması gerektiği" konusunda ısrar etmektedir. Zaman damgası ve adres maskesi sorguları sırasıyla -PP ve -PM seçenekleriyle gönderilebilir. Bir zaman damgası yanıtı (ICMP kodu 14) veya adres maskesi yanıtı (kod 18) ana bilgisayarın kullanılabilir olduğunu açıklar. Bu iki soru, yöneticiler yanıt isteği paketlerini özellikle engellerken diğer ICMP sorgularının aynı amaçla kullanabileceğini unuttuklarında değerli olabilir.

-PO <protocol list> (IP Protocol Ping) ⇒ Daha yeni ana bilgisayar bulma seçeneklerinden biri, IP başlıklarında belirtilen protokol numarasıyla IP paketleri gönderen IP protokolü ping'dir. Protokol listesi, daha önce tartışılan TCP, UDP ve SCTP ana bilgisayar bulma seçeneklerindeki bağlantı noktası listeleriyle aynı biçimde alır. Hiçbir protokol belirtilmezse, varsayılan olarak ICMP (protokol 1), IGMP (protokol 2) ve IP-in-IP (protokol 4) için birden fazla IP paketi gönderilir. Varsayılan protokoller, nmap.h dosyasında DEFAULT_PROTO_PROBE_PORT_SPEC değiştirilerek derleme zamanında yapılandırılabilir. ICMP, IGMP, TCP (protokol 6), UDP (protokol 17) ve SCTP (protokol 132) için paketlerin uygun protokol başlıklarıyla gönderildiğini, diğer protokollerin ise IP başlığının ötesinde hiçbir ek veri olmadan gönderildiğini unutmayın (--data, --data-string veya --data-length seçeneklerinden herhangi biri belirtilmediği sürece).

Bu ana bilgisayar bulma yöntemi, ya bir sonda ile aynı protokolü kullanan yanıtları ya da verilen protokolün hedef ana bilgisayarda desteklenmediğini belirten ICMP protokolüne ulaşamıyor iletilerini arar. Her iki yanıt türü de hedef ana bilgisayarın canlı olduğunu gösterir.

--disable-arp-ping (No ARP or ND Ping) ⇒ Nmap normalde, -Pn veya -PE gibi diğer ana bilgisayar bulma seçenekleri kullanılsa bile, yerel olarak bağlı ethernet ana bilgisayarlarının ARP veya IPv6 Komşu Bulma (ND) keşfini yapar. Bu örtük davranışını devre dışı bırakmak için --disable-arp-ping seçeneğini kullanın.

Varsayılan davranış normalde daha hızlıdır, ancak bu seçenek, bir yönlendiricinin tüm ARP isteklerine spekulatif olarak yanıt verdiği ve ARP taramasına göre her hedefin açık görünmesini sağlayan proxy ARP kullanan ağlarda kullanışlıdır.

`--discovery-ignore-rst` ⇒ Bazı durumlarda, güvenlik duvarları boş veya izin verilmeyen adreslere yapılan problara yanıt olarak TCP sıfırlama (RST) yanıtlarını taklit edebilir. Nmap normalde RST yanıtlarını hedefin açık olduğunu kanıtlamak için kabul ettiğinden, bu durum orada olmayan hedefleri taramak için boş zaman harcanmasına neden olabilir. `--discovery-ignore-rst` seçeneğini kullanmak, Nmap'in ana bilgisayar keşfi sırasında bu yanıtları dikkate almasını engelleyecektir. Bu durumda hedefleri kaçırmadığınızdan emin olmak için ekstra ana bilgisayar bulma seçenekleri seçmeniz gerekebilir.

`--traceroute` (Trace path to host) ⇒ Tracerout'lar, hedefe ulaşması en muhtemel bağlantı noktasını ve protokolü belirlemek için tarama sonuçlarındaki bilgiler kullanılarak tarama sonrası gerçekleştirilir. Bağlantı taramaları (-sT) ve boşta taramalar (-sI) hariç tüm tarama türleriyle çalışır. Tüm izler Nmap'in dinamik zamanlama modelini kullanır ve paralel olarak gerçekleştirilir.

Traceroute, tarayıcı ile hedef ana bilgisayar arasındaki ara atlamlardan ICMP Zaman Aşındır mesajlarını ortaya çıkarmak amacıyla düşük TTL'li (yaşam süresi) paketler göndererek çalışır. Standart traceroute uygulamaları 1 TTL ile başlar ve hedef ana bilgisayara ulaşılana kadar TTL'yi artırır. Nmap'in traceroute'u yüksek bir TTL ile başlar ve daha sonra sıfıra ulaşana kadar TTL'yi azaltır. Bunu geriye doğru yapmak, Nmap'in birden fazla ana bilgisayar üzerinden izleri hızlandırmak için akıllı önbellek algoritmaları kullanmasını sağlar. Ortalama olarak Nmap, ağ koşullarına bağlı olarak ana bilgisayar başına 5-10 daha az paket gönderir. Tek bir alt ağ taranıyorsa (örn. 192.168.0.0/24) Nmap'in çoğu ana bilgisayara yalnızca iki paket göndermesi gerekebilir.

Port Scanning Basics (Port Tarama Temelleri)

Nmap yıllar içinde işlevsellik açısından büyümüş olsa da, etkili bir port tarayıcı olarak başladı ve bu temel işlevi olmaya devam ediyor. Basit nmap <hedef> komutu, <hedef> ana bilgisayardaki 1.000 TCP bağlantı noktasını tarar. Birçok port tarayıcı geleneksel olarak tüm portları açık ya da kapalı olarak sınırlandırırken,

Nmap çok daha ayrıntılıdır. Portları altı duruma ayırır: açık, kapalı, filtrelenmiş, filtrelenmemiş, açık|filtrelenmiş veya kapalı|filtrelenmiş.

Bu durumlar portun kendine özgü özelliklerini değildir, ancak Nmap'in onları nasıl gördüğünü açıklar. Örneğin, hedefle aynı ağdan yapılan bir Nmap taraması 135/tcp bağlantı noktasını açık olarak gösterebilirken, aynı anda Internet üzerinden aynı seçeneklerle yapılan bir tarama bu bağlantı noktasını filtrelenmiş olarak gösterebilir.

Nmap tarafından tanınan altı bağlantı noktası durumu

open (Açık) ⇒ Bir uygulama bu bağlantı noktasında TCP bağlantılarını, UDP datagramlarını veya SCTP ilişkilerini aktif olarak kabul etmektedir. Bunları bulmak genellikle port taramasının birincil hedefidir. Güvenliği düşünen insanlar her açık portun bir saldırısı yolu olduğunu bilirler. Saldırganlar ve pen-testerler açık portlardan faydalananmak isterken, yöneticiler meşru kullanıcıları engellemeden bunları kapatmaya veya güvenlik duvarlarıyla korumaya çalışır. Açık portlar güvenlik dışı taramalar için de ilgi çekicidir çünkü ağ üzerinde kullanılabilecek hizmetleri gösterirler.

closed (Kapalı) ⇒ Kapalı bir port erişilebilirdir (Nmap prob paketlerini alır ve yanıtlar), ancak onu dinleyen bir uygulama yoktur. Bir ana bilgisayarın bir IP adresinde olduğunu göstermede (ana bilgisayar bulma veya ping taraması) ve işletim sistemi algılamanın bir parçası olarak yardımcı olabilirler. Kapalı bağlantı noktalarına erişilebildiğinden, bazılarının açılması durumunda daha sonra taramaya değer olabilir. Yöneticiler bu tür bağlantı noktalarını bir güvenlik duvarı ile engelleme amacıyla düşünürebilirler. Bu durumda, daha sonra ele alınacak olan filtrelenmiş durumda görüneceklərdir.

filtered (filtrelenmiş) ⇒ Nmap portun açık olup olmadığını belirleyemez çünkü paket filtreleme problemlerinin porta ulaşmasını engeller. Filtreleme özel bir güvenlik duvarı cihazından, yönlendirici kurallarından veya ana bilgisayar tabanlı güvenlik duvarı yazılımından kaynaklanıyor olabilir. Bu portlar çok az bilgi sağladıkları için saldırıcıları hayal kırıklığına uğratır. Bazen tip 3 kod 13 (hedefe ulaşılamıyor: iletişim idari olarak yasaklandı) gibi ICMP hata mesajlarıyla yanıt verirler, ancak yanıt vermeden problemleri bırakın filtreler çok daha yaygındır. Bu, Nmap'i, probun filtreleme yerine ağ tikanıklığı nedeniyle düşürülmesi ihtimaline karşı birkaç kez yeniden denemeye zorlar. Bu da taramayı önemli ölçüde yavaşlatır.

unfiltered (filtrelenmemiş) ⇒ Filtrelenmemiş durum, bir bağlantı noktasının erişilebilir olduğu, ancak Nmap'in açık mı yoksa kapalı mı olduğunu belirleyemediği anlamına gelir. Yalnızca güvenlik duvarı kural kümelerini eşlemek için kullanılan ACK taraması bağlantı noktalarını bu duruma sınıflandırır. Filtrelenmemiş bağlantı noktalarını Pencere taraması, SYN taraması veya FIN taraması gibi diğer tarama türleriyle taramak, bağlantı noktasının açık olup olmadığını çözmeye yardımcı olabilir.

open|filtered (açık|filtreli) ⇒ Nmap, bir portun açık mı yoksafiltrelenmiş mi olduğunu belirleyemediğinde portları bu duruma yerleştirir. Bu, açık portların yanıt vermediği tarama türleri için gerçekleşir. Yanıtın olmaması, bir paket filtresinin probu veya ortaya çıkardığı herhangi bir yanıtı düşündüğü anlamına da gelebilir. Bu yüzden Nmap portun açık mı yoksafiltrelenmiş mi olduğunu kesin olarak bilemez. UDP, IP protokolü, FIN, NULL ve Xmas taramaları portları bu şekilde sınıflandırır.

closed|filtered (kapalı|filtreli) ⇒ Bu durum, Nmap bir bağlantı noktasının kapalı mı yoksafiltrelenmiş mi olduğunu belirleyemediğinde kullanılır. Sadece IP ID boşta taraması için kullanılır.

Port Scanning Techniques (Liman Tarama Teknikleri)

Otomotiv tamiri yapan bir acemi olarak, ilkel aletlerimi (çekiç, koli bandı, anahtar vb.) elimdeki işe uydurmak için saatlerce uğraşabilirim. Başarısız olduğumda ve külüstürümü gerçek bir tamirciye götürdüğümde, her zaman işi zahmetsiz hale getiren mükemmel aleti çıkarana kadar büyük bir alet sandığını karıştırır. Port tarama sanatı da benzerdir. Uzmanlar dzinelerce tarama tekniğini anlar ve belirli bir görev için uygun olanı (veya kombinasyonu) seçerler. Deneyimsiz kullanıcılar ve script çocukları ise her sorunu varsayılan SYN taraması ile çözmeye çalışırlar. Nmap ücretsiz olduğu için, port tarama ustalığının önündeki tek engel bilgidir. Bu kesinlikle otomotiv dünyasını geride bırakıyor; burada bir dikme yayı kompresörüne ihtiyacınız olduğunu belirlemek büyük bir beceri gerektirebilir, ancak yine de bunun için binlerce dolar ödemek zorunda kalırsınız.

Tarama türlerinin çoğu yalnızca ayrıcalıklı kullanıcılar tarafından kullanılabilir. Bunun nedeni, Unix sistemlerinde root erişimi gerektiren ham paketleri gönderip almalarıdır. Windows'ta bir yönetici hesabı kullanılması önerilir, ancak Npcap

işletim sistemine zaten yüklendiğinde Nmap bazen bu platformdaki ayrıcalıksız kullanıcılar için çalışır. Nmap 1997'de piyasaya sürüldüğünde kök ayrıcalıkları gerektirmek ciddi bir sınırlamaydı, çünkü birçok kullanıcının yalnızca paylaşılan kabuk hesaplarına erişimi vardı. Şimdi ise dünya farklı. Bilgisayarlar daha ucuz, çok daha fazla insan her zaman doğrudan Internet erişimine sahip ve masaüstü Unix sistemleri (Linux ve Mac OS X dahil) yaygın. Nmap'in Windows versiyonu artık mevcut ve bu sayede daha da fazla masaüstü bilgisayarda çalışabiliyor. Tüm bu nedenlerden dolayı, kullanıcıların Nmap'i sınırlı paylaşımı kabuk hesaplarından çalıştırılmaya daha az ihtiyacı vardır. Ayrıcalıklı seçenekler Nmap'i çok daha güçlü ve esnek hale getirdiği için bu bir şans.

Nmap doğru sonuçlar üretmeye çalışsa da, tüm öngörülerinin hedef makineler (veya bunların önündeki güvenlik duvarları) tarafından döndürülen paketlere dayandığını unutmayın. Bu tür ana bilgisayarlar güvenilmez olabilir ve Nmap'i karıştırmak veya yaniltmak için yanıtlar gönderebilir. Çok daha yaygın olanı, Nmap probleme gerekliği gibi yanıt vermeyen RFC uyumlu olmayan ana bilgisayarlardır. FIN, NULL ve Xmas taramaları bu soruna özellikle duyarlıdır. Bu tür sorunlar belirli tarama türlerine özgüdür ve bu nedenle ayrı tarama türü girişlerinde tartışılmıştır.

Bu bölüm Nmap tarafından desteklenen bir düzine kadar port tarama tekniğini belgelemektedir. UDP taraması (-sU) ve SCTP tarama türlerinden herhangi birinin (-sY, -sZ) TCP tarama türlerinden herhangi biriyle birleştirilebilmesi dışında, aynı anda yalnızca bir yöntem kullanılabilir. Hafıza yardımı olarak, port tarama tipi seçenekleri -s<C> şeklindedir, burada <C> tarama adında öne çıkan bir karakterdir, genellikle ilk karakterdir. Bunun tek istisnası, kullanımdan kaldırılan FTP sıçrama taramasıdır (-b). Varsayılan olarak, Nmap bir SYN Taraması gerçekleştirir, ancak kullanıcının ham paketler göndermek için uygun ayrıcalıkları yoksa (Unix'te root erişimi gerektirir) bir bağlantı taraması yerine geçer. Bu bölümde listelenen taramalardan ayrıcalıksız kullanıcılar yalnızca connect ve FTP bounce taramalarını yürütebilir.

-sS (TCP SYN scan) ⇒ SYN taraması iyi nedenlerden dolayı varsayılan ve en popüler tarama seçeneğidir. Kısıtlayıcı güvenlik duvarları tarafından engellenmeyen hızlı bir ağda saniyede binlerce bağlantı noktasını tarayarak hızlı bir şekilde gerçekleştirilebilir. Ayrıca TCP bağlantılarını asla tamamlamadığı için nispeten göze batmaz ve gizlidir. SYN taraması, Nmap'in FIN(NULL/Xmas, Maimon ve boşta taramalarının yaptığı gibi belirli platformların özelliklerine bağlı olmak

yerine herhangi bir uyumlu TCP yiğinına karşı çalışır. Ayrıca açık, kapalı ve filtrelenmiş durumlar arasında net ve güvenilir bir ayrim yapılmasını sağlar.

- Bu teknik genellikle yarı açık tarama olarak adlandırılır, çünkü tam bir TCP bağlantısı açmazsınız. Sanki gerçek bir bağlantı açacaksınız gibi bir SYN paketi gönderir ve ardından bir yanıt beklersiniz. Bir SYN/ACK portun dinlediğini (açık) gösterirken, bir RST (sıfırlama) dinlemediğini gösterir. Birkaç yeniden iletimden sonra yanıt alınmazsa, bağlantı noktası filtrelenmiş olarak işaretlenir. ICMP ulaşılamıyor hatası (tip 3, kod 0, 1, 2, 3, 9, 10 veya 13) alınırsa da bağlantı noktası filtrelenmiş olarak işaretlenir. Yanıt olarak bir SYN paketi (ACK bayrağı olmadan) alınırsa da bağlantı noktası açık olarak kabul edilir. Bunun nedeni, eşzamanlı açık veya bölünmüş el sıkışma bağlantısı olarak bilinen son derece nadir bir TCP özelliği olabilir (bkz. <https://nmap.org/misc/split-handshake.pdf>).

-sT (TCP connect scan) ⇒ TCP bağlantı taraması, SYN taraması bir seçenek olmadığındaysa varsayılan TCP tarama türündür. Bu, bir kullanıcının ham paket ayrıcalıklarına sahip olmadığı durumdur. Diğer tarama türlerinin çoğunu yaptığı gibi ham paketler yazmak yerine, Nmap alitta yatan işletim sisteminden connect sistem çağrısını yayınlayarak hedef makine ve bağlantı noktası ile bir bağlantı kurmasını ister. Bu, web tarayıcılarının, P2P istemcilerinin ve diğer ağ özellikli uygulamaların çoğunu bağlantı kurmak için kullandığı aynı üst düzey sistem çağrısidir. Berkeley Sockets API olarak bilinen bir programlama arayüzünün parçasıdır. Nmap, kablodan ham paket yanıtlarını okumak yerine, her bağlantı denemesinde durum bilgisi almak için bu API'yi kullanır.

SYN taraması mevcut olduğunda, genellikle daha iyi bir seçimdir. Nmap'in yüksek seviye bağlantı çağrıları üzerinde ham paketlere göre daha az kontrolü vardır, bu da onu daha az verimli hale getirir. Sistem çağrıları, SYN taramasının yaptığı yarı açık sıfırlamayı gerçekleştirmek yerine açık hedef portlara bağlantıları tamamlar. Bu sadece daha uzun sürmez ve aynı bilgiyi elde etmek için daha fazla paket gerektir, aynı zamanda hedef makinelerin bağlantıyı günlüğe kaydetme olasılığı daha yüksektir. İyi bir IDS her ikisini de yakalayacaktır, ancak çoğu makinede böyle bir alarm sistemi yoktur. Ortalama Unix sisteminizdeki birçok hizmet, Nmap bağlandığında ve ardından veri göndermeden bağlantıyı kaptığında syslog'a bir not ve bazen şifreli bir hata mesajı ekleyecektir. Bu gerçekleştiğinde gerçekten

acınası hizmetler çöker, ancak bu nadirdir. Günlüklerinde tek bir sistemden bir sürü bağlantı girişimi gören bir yönetici, bağlantı taraması yapıldığını bilmelidir.

-sU (UDP scans) ⇒ İnternet üzerindeki en popüler hizmetler TCP protokolü üzerinden çalışsa da, UDP hizmetleri yaygın olarak kullanılmaktadır. DNS, SNMP ve DHCP (kayıtlı 53, 161/162 ve 67/68 numaralı portlar) en yaygın olanlardan üçdür. UDP taraması genellikle TCP'den daha yavaş ve daha zor olduğundan, bazı güvenlik denetçileri bu portları görmezden gelir. Bu bir hatadır, çünkü istismar edilebilir UDP hizmetleri oldukça yaygındır ve saldırganlar kesinlikle tüm protokolü görmezden gelmezler. Neyse ki Nmap UDP portlarının envanterini çıkarmaya yardımcı olabilir.

UDP taraması -sU seçeneği ile etkinleştirilir. Aynı çalışma sırasında her iki protokolü de kontrol etmek için SYN taraması (-sS) gibi bir TCP tarama türü ile birleştirilebilir.

UDP taraması hedeflenen her porta bir UDP paketi göndererek çalışır. Yanıt oranını artırmak için 53 ve 161 gibi bazı yaygın bağlantı noktaları için protokole özgü bir yük gönderilir, ancak çoğu bağlantı noktası için --data, --data-string veya --data-length seçenekleri belirtildiğinde paket boştur. Bir ICMP bağlantı noktası ulaşılamıyor hatası (tip 3, kod 3) döndürülürse, bağlantı noktası kapatılır. Diğer ICMP ulaşılamaz hataları (tip 3, kod 0, 1, 2, 9, 10 veya 13) bağlantı noktasını filtrelenmiş olarak işaretler. Bazen bir hizmet, açık olduğunu kanıtlayan bir UDP paketiyle yanıt verir. Yeniden iletimlerden sonra yanıt alınmazsa, bağlantı noktası açık|filtreli olarak sınıflandırılır. Bu, portun açık olabileceği veya paket filtrelerinin iletişimini engellediği anlamına gelir. Sürüm algılama (-sV), gerçekten açık olan portları filtrelenmiş olanlardan ayırt etmeye yardımcı olmak için kullanılabilir.

UDP taraması ile ilgili en büyük zorluk bunu hızlı bir şekilde yapmaktadır. Açık ve filtrelenmiş portlar nadiren yanıt gönderir, bu da Nmap'ı zaman aşımına uğratır ve ardından probun veya yanıtın kaybolması durumunda yeniden iletimler gerçekleştirir. Kapalı portlar genellikle daha da büyük bir sorundur. Genellikle bir ICMP portuna ulaşılamıyor hatası gönderirler. Ancak kapalı TCP portları tarafından bir SYN veya bağlantı taramasına yanıt olarak gönderilen RST paketlerinin aksine, birçok ana bilgisayar ICMP port ulaşılamaz mesajlarını varsayılan olarak sınırlar. Linux ve Solaris bu konuda özellikle katıdır. Örneğin, Linux 2.4.20 çekirdeği hedefe ulaşılamıyor mesajlarını saniyede bir ile sınırlar (net/ipv4/icmp.c'de).

Nmap hız sınırlamasını algılar ve hedef makinenin düşüreceği gereksiz paketlerle ağı doldurmaktan kaçınmak için buna göre yavaşlar. Ne yazık ki, Linux tarzı saniyede bir paket sınırı, 65.536 portluk bir taramanın 18 saatten fazla sürmesine neden olur. UDP taramalarınızı hızlandırmak için daha fazla ana bilgisayarı paralel olarak taramak, önce sadece popüler bağlantı noktalarını hızlı bir şekilde taramak, güvenlik duvarının arkasından tarama yapmak ve yavaş ana bilgisayarları atlamak için --host-timeout kullanmak gibi fikirler vardır.

-sY (SCTP INIT scan) ⇒ SCTP, TCP ve UDP protokollerine nispeten yeni bir alternatifdir, TCP ve UDP'nin çoğu özelliğini birleştirir ve ayrıca çoklu geçiş ve çoklu akış gibi yeni özellikler ekler. Çoğunlukla SS7/SIGTRAN ile ilgili hizmetler için kullanılmaktadır ancak diğer uygulamalar için de kullanılma potansiyeline sahiptir. SCTP INIT taraması, TCP SYN taramasının SCTP eşdeğeridir. Kısıtlayıcı güvenlik duvarları tarafından engellenmeyen hızlı bir ağda saniyede binlerce bağlantı noktasını tarayarak hızlı bir şekilde gerçekleştirilebilir. SYN taraması gibi, INIT taraması da SCTP ilişkilendirmelerini asla tamamlamadığı için nispeten göze batmaz ve gizlidir. Ayrıca açık, kapalı ve filtrelenmiş durumlar arasında net ve güvenilir bir ayırım yapılmasını sağlar.

- Bu teknik genellikle yarı açık tarama olarak adlandırılır, çünkü tam bir SCTP ilişkisi açmazsınız. Sanki gerçek bir ilişki açacakmısınız gibi bir INIT yiğini gönderir ve ardından yanıt beklersiniz. Bir INIT-ACK yiğini bağlantı noktasının dinlediğini (açık) gösterirken, bir ABORT yiğini dinleyici olmadığını gösterir. Birkaç yeniden iletimden sonra yanıt alınamazsa, bağlantı noktası filtrelenmiş olarak işaretlenir. Bir ICMP ulaşılamıyor hatası (tip 3, kod 0, 1, 2, 3, 9, 10 veya 13) alındığında da bağlantı noktası filtrelenmiş olarak işaretlenir.

-sN ; -sF ; -sX (TCP NULL, FIN, and Xmas scans) ⇒ Bu üç tarama türü (bir sonraki bölümde açıklanan --scanflags seçeneği ile daha da fazlası mümkündür) açık ve kapalı portlar arasında ayırmak için TCP RFC'deki ince bir boşluktan yararlanır. RFC 793'ün 65. sayfasında "[hedef] port durumu KAPALI ise bir RST içermeyen gelen bir segment yanıt olarak bir RST gönderilmesine neden olur" denmektedir. Bir sonraki sayfada, SYN, RST veya ACK bitleri ayarlanmadan açık portlara gönderilen paketler tartışılmakta ve şöyle denmektedir: "buraya ulaşmanız pek olası değil, ancak ulaşırsanız, segmenti bırakın ve geri dönün."

- Bu RFC metniyle uyumlu sistemleri tararken, SYN, RST veya ACK bitlerini içermeyen herhangi bir paket, bağlantı noktası kapalıysa RST döndürülmesine

ve bağlantı noktası açıksa hiçbir yanıt alınmamasına neden olur. Bu üç bitten hiçbirini dahil edilmediği sürece, diğer üçünün (FIN, PSH ve URG) herhangi bir kombinasyonu tamamdır. Nmap bunu üç tarama türü ile kullanır:

- Null scan (`-sN`) ⇒ Hiçbir biti ayarlamaz (TCP bayrak başlığı 0'dır)
- FIN scan (`-sF`) ⇒ Sadece TCP FIN bitini ayarlar.
- Xmas scan (`-sX`) ⇒ FIN, PSH ve URG bayraklarını ayarlayarak paketi bir Noel ağacı gibi aydınlatır.
- Bu üç tarama türü, prob paketlerinde ayarlanan TCP bayrakları dışında davranış olarak tamamen aynıdır. Bir RST paketi alınırsa, bağlantı noktası kapalı olarak kabul edilirken, yanıt alınmaması açık|filtrelenmiş olduğu anlamına gelir. Bir ICMP ulaşılamıyor hatası (tip 3, kod 0, 1, 2, 3, 9, 10 veya 13) alınırsa bağlantı noktası滤relenmiş olarak işaretlenir.
- Bu tarama türlerinin en önemli avantajı, bazı devletli olmayan güvenlik duvarlarından ve paket filtreleme yönlendiricilerinden gizlice geçebilmeleridir. Diğer bir avantaj ise bu tarama türlerinin SYN taramasından bile biraz daha gizli olmasıdır. Yine de buna güvenmeyin - çoğu modern IDS ürünü bunları tespit edecek şekilde yapılandırılabılır. En büyük dezavantajı, tüm sistemlerin RFC 793'ü harfiyen takip etmemesidir. Bazı sistemler, portun açık olup olmadığına bakmaksızın problara RST yanıtları gönderir. Bu da tüm portların kapalı olarak etiketlenmesine neden olur. Bunu yapan başlıca işletim sistemleri Microsoft Windows, birçok Cisco cihazı, BSDI ve IBM OS/400'dür. Ancak bu tarama Unix tabanlı sistemlerin çoğuna karşı çalışmaktadır. Bu taramaların bir diğer dezavantajı da açık portları滤relenmiş olanlardan ayırt edememeleri ve sizi açık|filtrelenmiş yanıtıyla baş başa bırakmalarıdır.

`-sA` (TCP ACK scan) ⇒ Bu tarama, açık (hatta açık|filtrelenmiş) portları asla belirlemediği için şimdije kadar tartışılan diğerlerinden farklıdır. Güvenlik duvari kural kümelerinin harmasını çıkarmak, durum bilgisi olup olmadıklarını ve hangi bağlantı noktalarının滤relenendiğini belirlemek için kullanılır.

- ACK tarama prob paketinde yalnızca ACK bayrağı ayarlanmıştır (`--scanflags` kullanmadığınız sürece). Filterenmemiş sistemleri tararken, açık ve kapalı portların her ikisi de bir RST paketi döndürecektil. Nmap daha sonra bunları filterenmemiş olarak etiketler, yani ACK paketi tarafından erişilebilirler, ancak açık mı yoksa kapalı mı oldukları belirlenemez. Yanıt vermeyen veya belirli

ICMP hata mesajlarını (tip 3, kod 0, 1, 2, 3, 9, 10 veya 13) geri gönderen bağlantı noktaları filtrelenmiş olarak etiketlenir.

-sW (TCP Window scan) ⇒ Pencere taraması ACK taramasıyla tamamen aynıdır, ancak bir RST döndürüldüğünde her zaman filtrelenmemiş olarak yazdırınak yerine, açık bağlantı noktalarını kapalı olanlardan ayırmak için belirli sistemlerin bir uygulama ayrıntısından yararlanır. Bunu, döndürülen RST paketlerinin TCP Pencere alanını inceleyerek yapar. Bazı sistemlerde, açık portlar pozitif pencere boyutu kullanırken (RST paketleri için bile), kapalı olanlar sıfır pencereye sahiptir. Bu nedenle, bir RST geri aldığımda bir bağlantı noktasını her zaman filtrelenmemiş olarak listelemek yerine, Pencere taraması, bu sıfırlamadaki TCP Pencere değeri sırasıyla pozitif veya sıfır ise bağlantı noktasını açık veya kapalı olarak listeler.

- Bu tarama internetteki az sayıda sistemin uygulama detayına dayanır, bu nedenle her zaman güvenemezsiniz. Bunu desteklemeyen sistemler genellikle tüm portları kapalı olarak döndüreceklerdir. Elbette, makinenin gerçekten hiç açık portu olmaması mümkündür. Taranan portların çoğu kapalıysa ancak birkaç yaygın port numarası (22, 25, 53 gibi) filtrelenmişse, sistem büyük olasılıkla hassastır. Bazen sistemler tam tersi bir davranış bile gösterebilir. Eğer taramanız 1.000 açık port ve üç kapalı ya da filtrelenmiş port gösteriyorsa, bu üç port gerçekten açık olanlar olabilir.

-sM (TCP Maimon scan) ⇒ Maimon taraması, adını keşfeden Uriel Maimon'dan almıştır. Bu teknigi Phrack Magazine'in 49. sayısında (Kasım 1996) tanımlamıştır. Bu teknigi içeren Nmap iki sayı sonra yayınlandı. Bu teknik NULL, FIN ve Xmas taramalarıyla tamamen aynıdır, tek farkı probun FIN/ACK olmasıdır. RFC 793'e (TCP) göre, port açık ya da kapalı olsun, böyle bir proba yanıt olarak bir RST paketi oluşturulmalıdır. Ancak Uriel, BSD türevi birçok sistemin port açıkça paketi düşürdüğünü fark etmiştir.

--scanflags (Custom TCP scan) ⇒ Gerçekten gelişmiş Nmap kullanıcılarının kendilerini sunulan konserve tarama türleriyle sınırlamalarına gerek yoktur. --scanflags seçeneği, keyfi TCP bayrakları belirterek kendi taramanızı tasarlamanıza olanak tanır. Satıcıları sadece Nmap man sayfasını tarayarak belirli kurallar ekleyen saldırı tespit sistemlerinden kaçarken, yaratıcılığınızı konuşurun!

- --scanflags argümanı 9 (PSH ve FIN) gibi sayısal bir bayrak değeri olabilir, ancak sembolik isimler kullanmak daha kolaydır. URG, ACK, PSH, RST, SYN ve FIN'in herhangi bir kombinasyonunu bir araya getirin. Örneğin, --scanflags

URGACKPSHRSTSYNFIN her şeyi ayarlar, ancak tarama için pek kullanışlı değildir. Bunların hangi sırada belirtildiği önemsizdir.

- İstediğiniz bayrakları belirtmenin yanı sıra, bir TCP tarama türü de belirtebilirisiniz (-sA veya -sF gibi). Bu temel tür Nmap'e yanıtları nasıl yorumlayacağını söyler. Örneğin, bir SYN taraması filtrelenmiş bir bağlantı noktasını belirtmek için yanıt yok olarak değerlendirirken, bir FIN taraması aynı şeyi açık|filtrelenmiş olarak değerlendirir. Nmap, temel tarama türü için yaptığı gibi davranışacaktır, ancak bunun yerine belirttiğiniz TCP bayraklarını kullanacaktır. Bir temel tür belirtmezseniz, SYN taraması kullanılır.

-sz (SCTP COOKIE ECHO scan) ⇒ SCTP COOKIE ECHO taraması daha gelişmiş bir SCTP taramasıdır. SCTP uygulamalarının açık portlarda COOKIE ECHO parçaları içeren paketleri sessizce düşürmesi, ancak port kapalıysa bir ABORT göndermesi gerektiği gereğinden yararlanır. Bu tarama türünün avantajı, bir INIT taramasından daha belirgin bir port taraması olmamasıdır. Ayrıca, INIT parçalarını engelleyen ancak COOKIE ECHO parçalarını engellemeyen durum bilgisi olmayan güvenlik duvarı kural kümeleri olabilir. Bunun bir port taramasını görünmez kılacağını düşünerek aldanmayın; iyi bir IDS SCTP COOKIE ECHO taramalarını da tespit edebilecektir. Dezavantajı, SCTP COOKIE ECHO taramalarının açık ve filtrelenmiş portlar arasında ayırmayı yapamaması ve sizi her iki durumda da açık|filtrelenmiş durumuyla baş başa bırakmasıdır.

-sl <zombie host> [: <probeport>] (idle scan) ⇒ Bu gelişmiş tarama yöntemi, hedefin gerçekten kör bir TCP bağlantı noktasını taramasına olanak tanır (yani hedefe gerçek IP adresinizden hiçbir paket gönderilmez). Bunun yerine, benzersiz bir yan kanal saldırısı, hedef üzerindeki açık portlar hakkında bilgi toplamak için zombi ana bilgisayarda öngörlülebilir IP parçalanma kimliği dizisi oluşturmayı kullanır. IDS sistemleri taramayı sizin belirttiğiniz zombi makineden geliyormuş gibi gösterecektir (bu makinenin çalışır durumda olması ve belirli kriterleri karşılaması gereklidir). Bu büyüleyici tarama türünün tüm ayrıntıları "TCP Idle Scan (-sl)" adlı bölümde yer almaktadır.

- Bu tarama türü olağanüstü derecede gizli olmasının yanı sıra (kör doğası nedeniyle) makineler arasındaki IP tabanlı güven ilişkilerinin haritalanmasına izin verir. Bağlantı noktası listesi, zombi ana bilgisayarın bakış açısından açık bağlantı noktalarını gösterir. Böylece, güvenilir olabileceğini düşündüğünüz

çeşitli zombileri kullanarak (yönlendirici/paket filtresi kuralları aracılığıyla) bir hedefi taramayı deneyebilirsiniz.

- IP ID değişiklikleri için zombi üzerindeki belirli bir portu araştırmak istiyorsanız, zombi ana bilgisayarına iki nokta üst üste ve ardından bir port numarası ekleyebilirsiniz. Aksi takdirde Nmap TCP pingleri için varsayılan olarak kullandığı portu (80) kullanacaktır.

-so (IP protocol scan) ⇒ IP protokol taraması, hedef makineler tarafından hangi IP protokollerinin (TCP, ICMP, IGMP, vb.) desteklendiğini belirlemenizi sağlar. Bu teknik olarak bir port taraması değildir, çünkü TCP veya UDP port numaraları yerine IP protokol numaraları arasında geçiş yapar. Yine de taranan protokol numaralarını seçmek için -p seçeneğini kullanır, sonuçlarını normal port tablosu biçiminde raporlar ve hatta gerçek port tarama yöntemleriyle aynı temel tarama motorunu kullanır. Yani buraya ait olması için bir port taramasına yeterince yakındır.

- Protokol taraması kendi başına faydalı olmasının yanı sıra açık kaynaklı yazılımın gücünü de gösteriyor. Temel fikir oldukça basit olsa da, bunu eklemeyi düşünmemiştüm ya da böyle bir işlevsellik için herhangi bir talep almamıştım. Daha sonra 2000 yazında Gerhard Rieger bu fikri tasarladı, bunu uygulayan mükemmel bir yama yazdı ve duyuru posta listesine (o zamanki adı nmap-hackers) gönderdi. Bu yamayı Nmap ağaçına dahil ettim ve ertesi gün yeni bir sürüm yayınladım. Çok az ticari yazılım parçası kendi geliştirmelerini tasarılayacak ve katkıda bulunacak kadar hevesli kullanıcılarla sahiptir!
- Protokol taraması UDP taramasına benzer bir şekilde çalışır. Bir UDP paketinin bağlantı noktası numarası alanını yinelemek yerine, IP paket başlıklarını gönderir ve sekiz bitlik IP protokolü alanını yineler. Başlıklar genellikle boştur, veri içermez ve talep edilen protokol için uygun başlık bile yoktur. İstisnalar TCP, UDP, ICMP, SCTP ve IGMP'dir. Bazı sistemler bunları başka türlü göndermeyeceğinden ve Nmap bunları oluşturmak için zaten işlevlere sahip olduğundan, bunlar için uygun bir protokol başlığı dahil edilmiştir. ICMP port ulaşılamaz mesajlarını izlemek yerine, protokol taraması ICMP protokol ulaşılamaz mesajlarını arar. Nmap hedef ana bilgisayardan herhangi bir protokolde herhangi bir yanıt alırsa, Nmap o protokolü açık olarak işaretler. Bir ICMP protokolüne ulaşılamıyor hatası (tip 3, kod 2) protokolün kapalı olarak işaretlenmesine neden olurken, porta ulaşılamıyor (tip 3, kod 3) protokolü açık olarak işaretler. Diğer ICMP ulaşılamaz hataları (tip 3, kod 0, 1, 9, 10 veya 13)

protokolün filtrelenmiş olarak işaretlenmesine neden olur (aynı zamanda ICMP'nin açık olduğunu kanıtlarlar). Yeniden iletimlerden sonra yanıt alınmazsa, protokol açık|filtrelenmiş olarak işaretlenir

-b <FTP relay host> (FTP bounce scan) ⇒ FTP protokolünün (RFC 959) ilginç bir özelliği de proxy FTP bağlantılarını desteklemesidir. Bu, bir kullanıcının bir FTP sunucusuna bağlanması ve ardından dosyaların üçüncü taraf bir sunucuya gönderilmesini istemesine olanak tanır. Böyle bir özellik birçok düzeyde kötüye kullanıma açıktır, bu nedenle çoğu sunucu bunu desteklemeyi bırakmıştır. Bu özelliğin izin verdiği kötüye kullanımlardan biri, FTP sunucusunun diğer ana bilgisayarları port taramasına neden olmasıdır. FTP sunucusundan sırayla hedef ana bilgisayarın her bir ilginç portuna bir dosya göndermesini isteyin. Hata mesajı portun açık olup olmadığını açıklayacaktır. Bu, güvenlik duvarlarını atlamak için iyi bir yoldur çünkü kurumsal FTP sunucuları genellikle diğer dahili ana bilgisayarlara herhangi bir eski Internet ana bilgisayarıdan daha fazla erişime sahip oldukları yerlere yerleştirilir. Nmap, -b seçeneği ile FTP sıçrama taramasını destekler. <kullanıcı adı>:<parola>@<sunucu>:<port> şeklinde bir argüman alır. <Server> savunmasız bir FTP sunucusunun adı veya IP adresidir. Normal bir URL'de olduğu gibi <kullanıcı adı>:<şifre> atlanabilir, bu durumda anonim oturum açma bilgileri (kullanıcı: anonim şifre:-wwwuser@) kullanılır. Bağlantı noktası numarası (ve önceki iki nokta üst üste) da atlanabilir, bu durumda <sunucu> üzerindeki varsayılan FTP bağlantı noktası (21) kullanılır.

- Bu güvenlik açığı 1997 yılında Nmap yayınlandığında yaygındı, ancak büyük ölçüde düzeltildi. Savunmasız sunucular hala etrafta, bu yüzden her şey başarısız olduğunda denemeye değer. Amacınız bir güvenlik duvarını aşmaksa, hedef ağı 21 numaralı bağlantı noktası için tarayın (hatta sürüm algılama ile tüm bağlantı noktalarını tararsanız herhangi bir FTP hizmeti için) ve ftp-bounce NSE betiğini kullanın. Nmap size ana bilgisayarın savunmasız olup olmadığını söyleyecektir. Eğer sadece izinizi kaybettirmeye çalışıyorsanız, kendinizi hedef ağdaki ana bilgisayarlarla sınırlamanıza gerek yoktur (ve aslında olmamalıdır). Savunmasız FTP sunucuları için rastgele Internet adreslerini taramaya başlamadan önce, sistem yöneticilerinin sunucularını bu şekilde kötüye kullanmanızdan hoşlanmayabileceğini göz önünde bulundurun.

Port Specification and Scan Order (Bağlantı Noktası Özellikleri ve Tarama Sırası)

Daha önce tartışılan tüm tarama yöntemlerine ek olarak, Nmap hangi bağlantı noktalarının taranacağını ve tarama sırasının rastgele mi yoksa sıralı mı olacağını belirleme seçenekleri sunar. Varsayılan olarak, Nmap her protokol için en yaygın 1.000 portu tarar.

-p <port ranges> (Only scan specified ports) ⇒ Bu seçenek hangi portları taramak istediğinizizi belirtir ve varsayılanı geçersiz kılar. Tek tek bağlantı noktası numaraları ve kısa çizgi ile ayrılmış aralıklar (örn. 1-1023) uygundur. Bir aralığın başlangıç ve/veya bitiş değerleri atlanabilir, bu da Nmap'in sırasıyla 1 ve 65535'i kullanmasına neden olur. Böylece 1'den 65535'e kadar olan portları taramak için -p- belirtebilirsiniz. Açıkça belirtmeniz halinde sıfır numaralı portun taramasına izin verilir. IP protokol taraması (-sO) için, bu seçenek taramak istediğiniz protokol numaralarını belirtir (0-255).

- Bir protokol kombinasyonunu tararken (örneğin TCP ve UDP), bağlantı noktası numaralarının önüne TCP için T:, UDP için U:, SCTP için S: veya IP Protokolü için P: ekleyerek belirli bir protokol belirtebilirsiniz. Niteleyici, siz başka bir niteleyici belirleyene kadar sürer. Örneğin, -p U:53,111,137,T:21-25,80,139,8080 argümanı 53, 111 ve 137 numaralı UDP portlarının yanı sıra listelenen TCP portlarını da tarayacaktır. Hem UDP hem de TCP'yi taramak için -sU ve en az bir TCP tarama türü (-sS, -sF veya -sT gibi) belirtmeniz gerektiğini unutmayın. Herhangi bir protokol niteleyicisi verilmemezse, bağlantı noktası numaraları tüm protokol listelerine eklenir.
- Bir protokol kombinasyonunu tararken (örneğin TCP ve UDP), bağlantı noktası numaralarının önüne TCP için T:, UDP için U:, SCTP için S: veya IP Protokolü için P: ekleyerek belirli bir protokol belirtebilirsiniz. Niteleyici, siz başka bir niteleyici belirleyene kadar sürer. Örneğin, -p U:53,111,137,T:21-25,80,139,8080 argümanı 53, 111 ve 137 numaralı UDP portlarının yanı sıra listelenen TCP portlarını da tarayacaktır. Hem UDP hem de TCP'yi taramak için -sU ve en az bir TCP tarama türü (-sS, -sF veya -sT gibi) belirtmeniz gerektiğini unutmayın. Herhangi bir protokol niteleyicisi verilmemezse, bağlantı noktası numaraları tüm protokol listelerine eklenir.

- Portlar, portun nmap-services'de ne olarak adlandırıldığına göre isimle de belirtilebilir. Hatta isimlerle birlikte * ve ? joker karakterlerini de kullanabilirsiniz. Örneğin, FTP'yi ve adı "http" ile başlayan tüm portları taramak için -p ftp,http* kullanın. Kabuk genişletmeleri konusunda dikkatli olun ve emin değilseniz -p argümanını alıntılayın.
- Port aralıkları, nmap-services içinde görünen bu aralıktaki portları belirtmek için köşeli parantezlerle çevrelenebilir. Örneğin, aşağıdakiler nmap-services içindeki 1024'e eşit veya altındaki tüm portları tarayacaktır: -p [-1024]. Kabuk genişletmelerine dikkat edin ve emin değilseniz -p argümanını alıntılayın.

--exclude-ports <port ranges> (Exclude the specified ports from scanning) ⇒ Bu seçenek, Nmap'in hangi portları taramadan hariç tutmasını istediğiniz belirtir. <port aralıkları> -p'ye benzer şekilde belirtilir. IP protokol taraması (-sO) için, bu seçenek hariç tutmak istediğiniz protokol numaralarını belirtir (0-255).

- Bağlantı noktalarının hariç tutulması istendiğinde, bu bağlantı noktaları her türlü taramanın dışında tutulur (yani hiçbir koşulda taramazlar). Buna keşif aşaması da dahildir.

-F (Fast (limited port) scan) ⇒ Varsayılandan daha az bağlantı noktası taramak istediğiniz belirtir. Normalde Nmap taranan her protokol için en yaygın 1.000 portu tarar. F ile bu sayı 100'e düşürülür.

- Nmap, hangi portların en yaygın olduğunu bilmek için frekans bilgisine sahip bir nmap-services dosyasına ihtiyaç duyar (port frekansları hakkında daha fazla bilgi için "İyi Bilinen Port Listesi: nmap-services" bölümünü bakın). Eğer port frekans bilgisi mevcut değilse, belki de özel bir nmap-services dosyası kullanıldığından, Nmap tüm adlandırılmış portları ve 1-1024 portlarını tarar. Bu durumda, -F yalnızca hizmetler dosyasında adı geçen bağlantı noktalarını taramak anlamına gelir.

-r (Don't randomize ports) ⇒ Varsayılan olarak, Nmap taranan bağlantı noktası sırasını rastgele yapar (yaygın olarak erişilebilen belirli bağlantı noktalarının verimlilik nedenleriyle başlangıça yakın taşınması dışında). Bu rastgele sıralama normalde tercih edilir, ancak bunun yerine sıralı (en düşükten en yükseğe doğru sıralanmış) port taraması için -r belirtebilirsiniz.

--port-ratio <ratio> <decimal number between 0 and 1> ⇒ nmap-services dosyasındaki tüm portları verilen orandan daha büyük bir oranla tarar. <oran> 0.0 ile 1.0 arasında

olmalıdır.

`--top-ports <n>` ⇒ nmap-services dosyasında bulunan `<n>` en yüksek oranlı portları --exclude-ports ile belirtilen tüm portları hariç tuttuktan sonra tarar. `<n>` 1 veya daha büyük olmalıdır.

Service and Version Detection (Servis ve Sürüm Algılama)

Nmap'i uzaktaki bir makineye yönlendirdiğinizde size 25/tcp, 80/tcp ve 53/udp bağlantı noktalarının açık olduğunu söyleyebilir. Nmap, yaklaşık 2.200 iyi bilinen hizmetten oluşan nmap-services veritabanını kullanarak, bu bağlantı noktalarının muhtemelen sırasıyla bir posta sunucusuna (SMTP), web sunucusuna (HTTP) ve ad sunucusuna (DNS) karşılık geldiğini bildirecektir. Bu arama genellikle doğrudur - TCP bağlantı noktası 25'i dinleyen daemonların büyük çoğunluğu aslında posta sunucularıdır. Ancak, güvenliğinizı buna bağılamamalısınız! İnsanlar garip bağlantı noktalarında hizmet çalıştırabilir ve çalıştırmaktadır.

Nmap haklı olsa ve yukarıdaki varsayımsal sunucu SMTP, HTTP ve DNS sunucularını çalıştırıyor olsa bile, bu çok fazla bilgi değildir. Şirketlerinizin veya müşterilerinizin güvenlik açığı değerlendirmelerini (hatta basit ağ envanterlerini) yaparken, hangi posta ve DNS sunucularının ve sürümlerinin çalıştığını gerçekten bilmek istersiniz. Doğru bir sürüm numarasına sahip olmak, bir sunucunun hangi açıklara karşı savunmasız olduğunu belirlemeye önemli ölçüde yardımcı olur. Sürüm tespiti bu bilgiyi elde etmenize yardımcı olur.

TCP ve/veya UDP portları diğer tarama yöntemlerinden biri kullanılarak keşfedildikten sonra, sürüm tespiti gerçekte neyin çalıştığı hakkında daha fazla bilgi edinmek için bu portları sorgular. nmap-service-probes veritabanı, çeşitli hizmetleri sorgulamak için propler ve yanıtları tanımak ve ayırtmak için eşleşme ifadeleri içerir. Nmap, hizmet protokolünü (örn. FTP, SSH, Telnet, HTTP), uygulama adını (örn. ISC BIND, Apache httpd, Solaris telnetd), sürüm numarasını, ana bilgisayar adını, aygit türünü (örn. yazıcı, yönlendirici), işletim sistemi ailesini (örn. Windows, Linux) belirlemeye çalışır. Mümkün olduğunda, Nmap bu bilgilerin Ortak Platform Numaralandırma (CPE) gösterimini de alır. Bazen bir X sunucusunun bağlantılarına açık olup olmadığı, SSH protokolü sürümü veya KaZaA kullanıcı adı gibi çeşitli ayrıntılar mevcuttur. Elbette, çoğu hizmet bu bilgilerin tümünü sağlamaz.

Eğer Nmap OpenSSL desteği ile derlenmişse, SSL sunucularına bağlanarak bu şifreleme katmanının arkasında dinlenen hizmeti çıkaracaktır. Bazı UDP portları, bir UDP port taraması portun açık mı yoksa filtrelenmiş mi olduğunu belirleyemedikten sonra açık|filtrelenmiş durumda bırakılır. Sürüm tespiti bu portlardan bir yanıt almaya çalışır (tipki açık portlarda yaptığı gibi) ve başarılı olursa durumu açık olarak değiştirir. açık|filtreli TCP portları da aynı şekilde ele alınır. Nmap -A seçeneğinin diğer şeylerin yanı sıra sürüm algılamayı etkinleştirdiğini unutmayın. Sürüm tespiti Bölüm 7, Hizmet ve Uygulama Sürüm Tespiti'nde ayrıntılı olarak açıklanmaktadır.

RPC hizmetleri keşfedildiğinde, Nmap RPC öğütücü otomatik olarak RPC programını ve sürüm numaralarını belirlemek için kullanılır. RPC olarak algılanan tüm TCP/UDP portlarını alır ve RPC portları olup olmadıklarını ve eğer öyleyse, hangi program ve sürüm numarasını sunduklarını belirlemek için SunRPC program NULL komutları ile doldurur. Böylece, hedefin port eşleyicisi bir güvenlik duvarının arkasında olsa bile (veya TCP sarmalayıcıları tarafından korunsa bile) rpcinfo -p ile aynı bilgileri etkili bir şekilde elde edebilirsiniz. Tuzaklar şu anda RPC taraması ile çalışmamaktadır.

Nmap bir hizmetten yanıt aldığında ancak bunları veritabanıyla eşleştiremediğinde, özel bir parmak izi ve bağlantı noktasında neyin çalıştığını eminseniz göndermeniz için bir URL yazdırır. Bulduklarınızın herkese faydalı olabilmesi için lütfen birkaç dakikanızı ayırarak gönderimde bulunun. Bu gönderimler sayesinde Nmap, SMTP, FTP, HTTP vb. gibi 650'den fazla protokol için yaklaşık 6.500 desen eşleşmesine sahiptir.

Sürüm algılama aşağıdaki seçeneklerle etkinleştirilir ve kontrol edilir:

-sV (Version detection) ⇒ Yukarıda tartışıldığı gibi sürüm algılamayı etkinleştirir. Alternatif olarak, diğer şeylerin yanı sıra sürüm algılamayı etkinleştirme -A kullanabilirsiniz.

- -sR, -sV için bir takma addır. Mart 2011'den önce, RPC öğütücüyı sürüm algılamadan ayrı olarak etkinleştirmek için kullanılıyordu, ancak şimdi bu seçenekler her zaman birleştiriliyor.

--allports (Don't exclude any ports from version detection) ⇒ Varsayılan olarak, Nmap sürüm tespiti TCP bağlantı noktası 9100'ü atlar çünkü bazı yazıcılar bu bağlantı noktasına gönderilen her şeyi yazdırır, bu da dzinelerce sayfa HTTP GET

isteğine, ikili SSL oturum isteğine vb. yol açar. Bu davranış, nmap-service-probes içindeki Exclude yönergesini değiştirerek veya kaldırarak değiştirilebilir ya da herhangi bir Exclude yönergesine bakılmaksızın tüm bağlantı noktalarını taramak için --allports belirtebilirsiniz.

`--version-intensity <intensity>` (Set version scan intensity) ⇒ Bir sürüm taraması (-sV) gerçekleştirirken, Nmap her birine bir ile dokuz arasında bir nadirlik değeri atanmış bir dizi sonda gönderir. Düşük numaralı problemler çok çeşitli yaygın hizmetlere karşı etkiliyken, yüksek numaralı olanlar nadiren yararlıdır. Yoğunluk seviyesi hangi problemlerin uygulanması gerektiğini belirtir. Sayı ne kadar yüksek olursa, hizmetin doğru şekilde tanımlanma olasılığı o kadar yüksek olur. Ancak, yüksek yoğunluklu taramalar daha uzun sürer. Yoğunluk 0 ile 9 arasında olmalıdır. Varsayılan değer 7'dir. nmap-service-probes ports yönergesi aracılığıyla hedef porta bir sonda kaydedildiğinde, yoğunluk seviyesine bakılmaksızın bu sonda denenir. Bu, DNS problemlerinin her zaman herhangi bir açık port 53'e karşı denenmesini, SSL probunun 443'e karşı yapılmasını vb. sağlar.

`--version-light` (Enable light mode) ⇒ Bu, --version-intensity 2 için kullanışlı bir takma addır. Bu hafif mod sürüm taramasını çok daha hızlı hale getirir, ancak hizmetleri tanımlama olasılığı biraz daha düşüktür.

`--version-all` (Try every single probe) ⇒ Her bir porta karşı her bir probun denenmesini sağlayan --version-intensity 9 için bir takma ad.

`--version-trace` (Trace version scan activity) ⇒ Bu, Nmap'in sürüm taramasının ne yaptığı hakkında kapsamlı hata ayıklama bilgisi yazdırmasına neden olur. Bu, --packet-trace ile elde ettiğinizin bir alt kümesidir.

OS Detection (İşletim Sistemi Algılama)

Nmap'in en iyi bilinen özelliklerinden biri TCP/IP yiğini parmak izi kullanarak uzak işletim sistemi tespitiidir. Nmap, uzak ana bilgisayara bir dizi TCP ve UDP paketi gönderir ve yanıtlardaki hemen hemen her biti inceler. TCP ISN örneklemesi, TCP seçenekleri desteği ve sıralaması, IP ID örneklemesi ve ilk pencere boyutu kontrolü gibi düzinelere test gerçekleştirdikten sonra Nmap, sonuçları 2.600'den fazla bilinen işletim sistemi parmak izinden oluşan nmap-os-db veritabanıyla karşılaşır

ve bir eşleşme varsa işletim sistemi ayrıntılarını yazdırır. Her parmak izi, işletim sisteminin serbest biçimli bir metin açıklamasını ve satıcı adını (örn. Sun), temel işletim sistemini (örn. Solaris), işletim sistemi neslini (örn. 10) ve cihaz türünü (genel amaçlı, yönlendirici, anahtar, oyun konsolu vb.) sağlayan bir sınıflandırma içerir.Çoğu parmak izi ayrıca cpe:/o:linux:linux_kernel:2.6 gibi bir Ortak Platform Numaralandırma (CPE) gösterimine sahiptir.

Nmap bir makinenin işletim sistemini tahmin edemiyorsa ve koşullar iyiye (örneğin, en az bir açık bağlantı noktası ve bir kapalı bağlantı noktası bulunduğu), Nmap, makinede çalışan işletim sistemini biliyorsanız (kesin olarak) parmak izini göndermek için kullanabileceğiniz bir URL sağlayacaktır. Bunu yaparak Nmap tarafından bilinen işletim sistemleri havuzuna katkıda bulunursunuz ve böylece herkes için daha doğru olur.

İşletim sistemi tespiti, yine de süreç sırasında toplanan bilgileri kullanan diğer bazı testleri mümkün kılar. Bunlardan biri TCP Sıra Tahmin Edilebilirlik Sınıflandırmasıdır. Bu, uzak ana bilgisayara karşı sahte bir TCP bağlantısı kurmanın yaklaşık olarak ne kadar zor olduğunu ölçer. Kaynak-IP tabanlı güven ilişkilerini (rlogin, güvenlik duvarı filtreleri vb.) istismar etmek veya bir saldırının kaynağını gizlemek için kullanışlıdır. Bu tür bir sahtekarlık artık nadiren yapılmaktadır, ancak birçok makine hala buna karşı savunmasızdır. Gerçek zorluk sayısı istatistiksel örneklemeye dayanır ve dalgalanma gösterebilir. Genellikle "değerli meydan okuma" veya "önemsiz şaka" gibi İngilizce sınıflandırmayı kullanmak daha iyidir. Bu yalnızca ayrıntılı (-v) modda normal çıktıda raporlanır. Verbose modu -O ile birlikte etkinleştirildiğinde, IP ID sıra üretimi de rapor edilir.Çoğu makine "artımlı" sınıfındadır, yani gönderdikleri her paket için IP başlığındaki kimlik alanını artırırlar. Bu da onları çeşitli gelişmiş bilgi toplama ve sahtekarlık saldırılarına karşı savunmasız hale getirir.

İşletim sistemi algılaması tarafından etkinleştirilen bir başka ekstra bilgi de hedefin çalışma süresine ilişkin bir tahmindir. Bu, bir makinenin en son ne zaman yeniden başlatıldığını tahmin etmek için TCP zaman damgası seçeneğini (RFC 1323) kullanır. Tahmin, zaman damgası sayacının sıfıra başlatılmaması veya sayacın taşması ve etrafından dolanması nedeniyle yanlış olabilir, bu nedenle yalnızca ayrıntılı modda yazdırılır.

İşletim sistemi algılama, Bölüm 8, Uzak İşletim Sistemi Algılama'da ele alınmıştır.
İşletim sistemi algılama aşağıdaki seçeneklerle etkinleştirilir ve kontrol edilir:

`-O` (Enable OS detection) ⇒ Yukarıda tartışıldığı gibi işletim sistemi algılamayı etkinleştirir. Alternatif olarak, diğer şeyle birlikte işletim sistemi algılamayı etkinleştirmek için `-A` kullanabilirsiniz.

`--osscan-limit` (Limit OS detection to promising targets) ⇒ En az bir açık ve bir kapalı TCP portu bulunursa OS tespiti çok daha etkili olur. Bu seçeneği ayarladığınızda Nmap, bu kriterleri karşılamayan ana bilgisayarlarla karşı işletim sistemi algılamayı bile denemeyecektir. Bu, özellikle birçok ana bilgisayara karşı `-Pn` taramalarında önemli ölçüde zaman kazandırabilir. Yalnızca `-O` veya `-A` ile işletim sistemi tespiti istediğinizde önemlidir.

`--osscan-guess` ; `--fuzzy` (Guess OS detection results) ⇒ Nmap mükemmel bir işletim sistemi eşleşmesi tespit edemediğinde, bazen yakın eşleşmeleri olasılık olarak sunar. Nmap'in varsayılan olarak bunu yapması için eşleşmenin çok yakın olması gereklidir. Bu (esdeğer) seçeneklerden herhangi biri Nmap'in daha agresif tahmin yapmasını sağlar. Nmap hala kusurlu bir eşleşme yazdırıldığında size söyleyecek ve her tahmin için güven seviyesini (yüzde) gösterecektir.

`--max-os-tries` (Set the maximum number of OS detection tries against a target) ⇒ Nmap bir hedefe karşı işletim sistemi tespiti gerçekleştirdiğinde ve mükemmel bir eşleşme bulamadığında, genellikle denemeyi tekrarlar. Varsayılan olarak, Nmap koşullar işletim sistemi parmak izi gönderimi için uygunsa beş kez, koşullar o kadar iyi değilse iki kez dener. Daha düşük bir `--max-os-tries` değeri belirtmek (1 gibi) Nmap'i hızlandırır, ancak işletim sistemini potansiyel olarak tanımlayabilecek yeniden denemeleri kaçırmazsınız. Alternatif olarak, koşullar uygun olduğunda daha fazla yeniden denemeye izin vermek için yüksek bir değer ayarlanabilir. Bu, Nmap OS veritabanına gönderme ve entegrasyon için daha iyi parmak izleri oluşturmak dışında nadiren yapılır.

Nmap Scripting Engine (NSE) (Nmap Komut Dosyası Motoru (NSE))

Nmap Scripting Engine (NSE), Nmap'in en güçlü ve esnek özelliklerinden biridir. Kullanıcıların çok çeşitli ağ görevlerini otomatikleştirmek için basit komut dosyaları (Lua programlama dilini kullanarak) yazmasına (ve paylaşmasına) olanak tanır. Bu komut dosyaları, Nmap'ten beklediğiniz hız ve verimlilikle paralel olarak yürütülür.

Kullanıcılar Nmap ile birlikte dağıtılan ve giderek büyüyen komut dosyalarına güvenebilir veya özel ihtiyaçlarını karşılamak için kendi komut dosyalarını yazabilirler.

Sistemi oluştururken aklımızda olan görevler arasında ağ keşfi, daha sofistike sürümler tespiti, güvenlik açığı tespiti yer alıyor. NSE güvenlik açığı istismarı için bile kullanılabilir.

Bu farklı kullanımları yansıtmak ve hangi komut dosyalarının çalıştırılacağı seçiminin basitleştirmek için, her komut dosyası bir veya daha fazla kategori ile ilişkilendiren bir alan içerir. Şu anda tanımlı kategoriler auth, broadcast, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version ve vuln'dur. Bunların hepsi "Komut Dosyası Kategorileri" adlı bölümde açıklanmıştır.

Komut dosyaları bir korumalı alanda çalıştırılmaz ve bu nedenle yanlışlıkla veya kötü niyetle sisteminize zarar verebilir veya gizliliğinizi ihlal edebilir. Yazarlarına güvenmediğiniz veya komut dosyalarını kendiniz dikkatlice denetlemediğiniz sürece üçüncü taraflardan gelen komut dosyalarını asla çalıştmayın.

Nmap Scripting Engine Bölüm 9, Nmap Scripting Engine'de ayrıntılı olarak açıklanmıştır ve aşağıdaki seçeneklerle kontrol edilir:

`-sc` ⇒ Varsayılan betik kümesini kullanarak bir betik taraması gerçekleştirir. Bu --script=default ile eşdeğerdir. Bu kategorideki bazı komut dosyaları müdahaleci olarak kabul edilir ve izinsiz olarak hedef ağıda çalıştırılmamalıdır.

`--script <filename> | <category> | <directory> /| <expression> [...]` ⇒ Virgülle ayrılmış dosya adları, komut dosyası kategorileri ve dizinler listesini kullanarak bir komut dosyası taraması çalıştırır. Listedeki her öğe, daha karmaşık bir komut dosyası kümesini tanımlayan bir Boolean ifadesi de olabilir. Her öğe önce bir ifade, sonra bir kategori ve son olarak da bir dosya veya dizin adı olarak yorumlanır.

- Yalnızca ileri düzey kullanıcılar için iki özel özellik vardır. Birincisi, normalde çalışmaya olsalar bile (örneğin, ilgili hizmet hedef bağlantı noktasında algılanmamışsa) onları çalışmaya zorlamak için komut dosyası adlarının ve ifadelerinin önüne + eklemektir. Diğer ise all argümanın Nmap'in veritabanındaki her betiği belirtmek için kullanılabilmesidir. Bu konuda dikkatli olun çünkü NSE istismarlar, kaba kuvvet kimlik doğrulama kırcıları ve hizmet reddi saldırıları gibi tehlikeli betikler içerir.

- Dosya ve dizin adları görelî veya mutlak olabilir. Mutlak isimler doğrudan kullanılır. Göreceli yollar, bulunana kadar aşağıdaki yerlerin her birinin komut dosyalarında aranır:

```
--datadir
$NMAPDIR
~/nmap (not searched on Windows)
<APPDATA>\nmap (only on Windows)
the directory containing the nmap executable
the directory containing the nmap executable, followed by ../share/nmap (not searched on Windows)
NMAPDATADIR (not searched on Windows)
the current directory.
```

ile biten bir dizin adı verildiğinde, Nmap dizindeki adı .nse ile biten her dosyayı yükler. Diğer tüm dosyalar göz ardı edilir ve dizinler özyinelemeli olarak aranmaz. Bir dosya adı verildiğinde, .nse uzantısına sahip olmak zorunda değildir; gerekirse otomatik olarak eklenecektir.

Nmap komut dosyaları varsayılan olarak Nmap veri dizininin scripts alt dizininde saklanır (bkz. Bölüm 14, Nmap Veri Dosyalarını Anlama ve Özelleştirme). Verimlilik için, betikler scripts/script.db'de depolanan ve her betığın ait olduğu kategori veya kategorileri listeleyen bir veritabanında dizinlenir.

script.db'deki komut dosyalarına adıyla atıfta bulunurken, kabuk tarzı bir '*' joker karakteri kullanabilirsiniz.

nmap --script "http-*" ⇒ Adı http- ile başlayan http-auth ve http-open-proxy gibi tüm betikleri yükler. Joker karakteri kabuktan korumak için --script argümanının tırnak içinde olması gerekiyordu.

- Boolean ifadeleri oluşturmak için and, or ve not operatörleri kullanılarak daha karmaşık kod seçimi yapılabilir. İşleçler Lua'da olduğu gibi aynı önceliğe sahiptir: not en yüksektir, ardından and ve sonra or gelir. Parantez kullanarak önceliği değiştirebilirsiniz. İfadeler boşluk karakterleri içerdiginden, bunları tırnak içine almak gereklidir.

nmap --script "not intrusive" ⇒ Müdahaleci kategorisindekiler hariç tüm komut dosyalarını yükler.

nmap --script "default or safe" ⇒ Bu işlevsel olarak nmap --script "default,safe" ile eşdeğerdir. Varsayılan kategorideki veya güvenli kategorideki ya da her ikisindeki tüm betikleri yükler.

nmap --script "default and safe" ⇒ Hem varsayılan hem de güvenli kategorilerde bulunan komut dosyalarını yükler.

nmap --script "(default or safe or intrusive) and not http-*" ⇒ Adları http- ile başlayanlar hariç, varsayılan, güvenli veya müdahaleci kategorilerdeki betikleri yükler.

--script-args <n1> = <v1>, <n2> ={ <n3> = <v3> }, <n4> ={ <v4> , <v5> } ⇒ NSE komut dosyalarına bağımsız değişkenler sağlamanıza olanak tanır. Bağımsız değişkenler, ad=değer çiftlerinin virgülle ayrılmış bir listesidir. Adlar ve değerler, boşluk veya '{', '}', '=' veya '' karakterlerini içermeyen dizeler olabilir. Bu karakterlerden birini bir dizeye dahil etmek için, dizeyi tek veya çift tırnak içine alın. Tırnak içine alınmış bir dize içinde, '\' bir tırnak işaretinden kaçar. Ters eğik çizgi yalnızca bu özel durumda tırnak işaretlerinden kaçmak için kullanılır; diğer tüm durumlarda ters eğik çizgi tam anlamıyla yorumlanır. Değerler, Lua'da olduğu gibi {} içine alınmış tablolar da olabilir. Bir tablo basit dize değerleri veya iç içe geçmiş tablolar da dahil olmak üzere daha fazla isim-değer çifti içerebilir. Birçok betik, argümanlarını xmpp-info.server_name'de olduğu gibi betik adıyla nitelendirir. Bu tam nitelikli sürümü yalnızca belirtilen betiği etkilemek için kullanabilir veya bu bağımsız değişken adını kullanan tüm betikleri etkilemek için niteliksiz sürümü (bu durumda sunucu_adı) iletibilsiniz. Bir komut dosyası, nitelendirilmemiş bir bağımsız değişken adını kabul etmeden önce tam olarak nitelendirilmiş bağımsız değişken adını (belgelerinde belirtilen ad) kontrol eder. Kod bağımsız değişkenlerinin karmaşık bir örneği --script-args 'user=foo,pass=",{}=bar",whois={whodb=nofollow+ripe},xmpp-info.server_name=localhost' şeklidir. <https://nmap.org/nsedoc/> adresindeki çevrimiçi NSE Dokümantasyon Portalı her bir betığın kabul ettiği argümanları listeler.

--script-args-file <filename> ⇒ NSE komut dosyalarına bir dosyadan bağımsız değişkenler yüklemenizi sağlar. Komut satırındaki tüm bağımsız değişkenler dosyadakilerin yerine geçer. Dosya mutlak bir yol veya Nmap'in olağan arama yoluna (NMAPDIR, vb.) göreli bir yol olabilir. Bağımsız değişkenler virgülle ayrılmış veya satırsonuyla ayrılmış olabilir, ancak kabuk tarafından ayırtılmalıdır için özel alıntı ve kaçış gerektirmeden --script-args ile aynı kuralları izlerler.

--script-help <filename> | <category> | <directory> | <expression> | all[...] ⇒ Komut dosyaları hakkında yardım gösterir. Verilen belirtme uyan her betik için Nmap betik adını, kategorilerini ve açıklamasını yazdırır. Belirtimler --script tarafından kabul

edilenlerle aynıdır; örneğin ftp-anon betiği hakkında yardım istiyorsanız, nmap --script-help ftp-anon komutunu çalıştırırsınız. Tek tek betikler için yardım almanın yanı sıra, bunu bir belirtim için hangi betiklerin çalıştırılacağının bir önizlemesi olarak da kullanabilirsiniz, örneğin nmap --script-help default ile.

--script-trace ⇒ Bu seçenek --packet-trace seçeneğinin yaptığını yapar, sadece bir ISO katmanı daha yüksektir. Bu seçenek belirtilirse, bir kod tarafından gerçekleştirilen tüm gelen ve giden iletişim yazdırılır. Görüntülenen bilgiler iletişim protokolünü, kaynağı, hedefi ve iletilen verileri içerir. İletilen tüm verilerin %5'inden fazlası yazdırılamıyorsa, izleme çıktısı hex dökümü biçimindedir. --packet-trace belirtilmesi komut dosyası izlemeyi de etkinleştirir.

--script-updatedb ⇒ Bu seçenek, Nmap tarafından mevcut varsayılan komut dosyalarını ve kategorileri belirlemek için kullanılan scripts/script.db dosyasında bulunan komut dosyası veritabanını günceller. Veritabanını güncellemek yalnızca varsayılan betikler dizinine NSE betikleri eklediğinizde veya dizinden kaldırıldığınızda ya da herhangi bir betiğin kategorilerini değiştirdiğinizde gereklidir. Bu seçenek genellikle tek başına kullanılır: nmap --script-updatedb.

Timing and Performance (Zamanlama ve Performans)

En yüksek Nmap geliştirme önceliklerimden biri her zaman performans olmuştur. Yerel ağimdaki bir ana bilgisayarın varsayılan taraması (nmap <ana bilgisayar adresi>) saniyenin beşte biri kadar sürüyor. Bu, göz kırmak için ancak yeterli bir süre, ancak yüzlerce veya binlerce ana bilgisayarı taradığınızda eklenir. Dahası, UDP taraması ve sürüm tespiti gibi belirli tarama seçenekleri tarama sürelerini önemli ölçüde artırabilir. Belirli güvenlik duvarı yapılandırmaları, özellikle de yanıt hızı sınırlaması da öyle. Nmap bu taramaları hızlandırmak için paralellik ve birçok gelişmiş algoritma kullanırken, kullanıcı Nmap'in nasıl çalışacağı üzerinde nihai kontrole sahiptir. Uzman kullanıcılar, zaman kısıtlamalarını karşılaşarken yalnızca önemsedikleri bilgileri elde etmek için Nmap komutlarını dikkatlice oluştururlar.

Tarama sürelerini iyileştirme teknikleri arasında kritik olmayan testleri atlamak ve Nmap'in en son sürümüne yükseltmek yer alır (performans geliştirmeleri sık sık yapılır). Zamanlama parametrelerini optimize etmek de önemli bir fark yaratabilir. Bu seçenekler aşağıda listelenmiştir.

Bazı seçenekler bir zaman parametresi kabul eder. Bu, varsayılan olarak saniye cinsinden belirtilir, ancak milisaniye, saniye, dakika veya saat belirtmek için değere 'ms', 's', 'm' veya 'h' ekleyebilirsiniz. Yani --host-timeout argümanları 900000ms, 900, 900s ve 15m hepsi aynı şeyi yapar.

`--min-hostgroup <numhosts>; --max-hostgroup <numhosts>` (Adjust parallel scan group sizes)
⇒ Nmap, paralel olarak birden fazla ana bilgisayarı port taraması veya sürümler taraması yapma yeteneğine sahiptir. Nmap bunu hedef IP alanını gruptara bölgerek ve ardından her seferinde bir grubu tarayarak yapar. Genel olarak, daha büyük gruplar daha verimlidir. Dezavantajı, tüm grup tamamlanana kadar ana bilgisayar sonuçlarının sağlanamamasıdır. Bu nedenle, Nmap 50 grup boyutuyla başlıya, kullanıcı ilk 50 ana bilgisayar tamamlanana kadar herhangi bir rapor almayacaktır (ayrıntılı modda sunulan güncellemeler hariç).

- Varsayılan olarak, Nmap bu çatışmaya uzlaşmacı bir yaklaşım benimser. İlk sonuçların hızlı bir şekilde gelmesi için beş gibi düşük bir grup boyutuyla başlar ve daha sonra grup boyutunu 1024'e kadar yükseltir. Tam varsayılan sayılar verilen seçeneklere bağlıdır. Verimlilik nedenleriyle, Nmap UDP veya az portlu TCP taramaları için daha büyük grup boyutları kullanır.
- Maksimum grup boyutu --max-hostgroup ile belirtildiğinde, Nmap bu boyutu asla aşmayıacaktır. -min-hostgroup ile minimum bir boyut belirttiğinizde Nmap grup boyutlarını bu seviyenin üzerinde tutmaya çalışacaktır. Belirli bir arayüzde belirtilen minimum değeri karşılayacak kadar hedef ana bilgisayar kalmamışsa Nmap belirttiğinizden daha küçük gruplar kullanmak zorunda kalabilir. Her ikisi de grup boyutunu belirli bir aralıkta tutmak için ayarlanabilir, ancak bu nadiren istenir.
- Bu seçeneklerin taramanın ana bilgisayar bulma aşamasında bir etkisi yoktur. Buna düz ping taramaları da dahildir (-sn). Ana bilgisayar keşfi, hızı ve doğruluğu artırmak için her zaman büyük ana bilgisayar gruplarında çalışır.
- Bu seçeneklerin birincil kullanımı, tam taramanın daha hızlı çalışması için büyük bir minimum grup boyutu belirtmektir. Bir ağı /24 boyutlu parçalar halinde taramak için yaygın bir seçim 256'dır. Çok sayıda bağlantı noktası içeren bir tarama için, bu sayıyı aşmanın pek yardımcı olması olası değildir. Yalnızca birkaç bağlantı noktası numarasının taranması için 2048 veya daha fazla ana bilgisayar grubu boyutu yararlı olabilir.

--min-parallelism <numprobes> ; --max-parallelism numprobes (Adjust probe parallelization)

⇒ Bu seçenekler, bir ana bilgisayar grubu için bekleyen toplam prob sayısını kontrol eder. Port taraması ve host keşfi için kullanılırlar. Varsayılan olarak, Nmap ağ performansına dayalı olarak sürekli değişen ideal bir paralellik hesaplar. Eğer paketler düşüyorsa, Nmap yavaşlar ve daha az sayıda sondaya izin verir. Ağ kendini kanitladıkça ideal prob sayısı yavaşça artar. Bu seçenekler bu değişkene minimum veya maksimum sınırlar koyar. Varsayılan olarak, ideal paralellik, ağın güvenilmez olduğu kanıtlanırsa bire düşebilir ve mükemmel koşullarda birkaç yüze yükselebilir.

- En yaygın kullanım --min-parallelism değerini birden büyük bir sayıya ayarlayarak düşük performans gösteren ana bilgisayarların veya ağların taranmasını hızlandırmaktır. Çok yüksek ayarlamak doğruluğu etkileyebileceğinden, bu oynamak için riskli bir seçenekdir. Bunu ayarlamak ayrıca Nmap'in ağ koşullarına bağlı olarak paralelliği dinamik olarak kontrol etme yeteneğini de azaltır. 10 değeri makul olabilir, ancak ben bu değeri yalnızca son olarak ayarlıyorum.
- Nmap'in ana bilgisayarlara aynı anda birden fazla sonda göndermesini önlemek için --max-parallelism seçeneği bazen bir olarak ayarlanır. Daha sonra tartışılacak olan --scan-delay seçeneği bunu yapmanın başka bir yoludur.

--min-rtt-timeout <time> , --max-rtt-timeout <time> , --initial-rtt-timeout <time> (Adjust probe timeouts) ⇒ Nmap, vazgeçmeden veya probu yeniden iletmeden önce bir prob yanıtı için ne kadar bekleyeceğini belirlemek için çalışan bir zaman aşımı değeri tutar. Bu, önceki problemlerin yanıt sürelerine göre hesaplanır. Tam formül "Tarama Kodu ve Algoritmalar" adlı bölümde verilmiştir. Ağ gecikmesinin önemli ve değişken olması durumunda, bu zaman aşımı birkaç saniyeye kadar çıkabilir. Ayrıca muhafazakar (yüksek) bir seviyede başlar ve Nmap yanıt vermeyen ana bilgisayarı tararken bir süre bu şekilde kalabilir.

- Varsayılan değerlerden daha düşük bir --max-rtt-timeout ve --initial-rtt-timeout belirtmek tarama sürelerini önemli ölçüde kısaltabilir. Bu özellikle iğnesiz (-Pn) taramalar ve yoğun şekilde filtrelenmiş ağlara karşı yapılan taramalar için geçerlidir. Yine de çok agresif olmayın. Yanıt aktarılırken birçok probun zaman aşımına uğrayacağı ve yeniden iletileceği kadar düşük bir değer belirlerseniz tarama daha uzun sürebilir.

- Tüm ana bilgisayarlar yerel bir ağ üzerindeyse, 100 milisaniye (--max-rtt-timeout 100ms) makul bir agresif değerdir. Yönlendirme söz konusuysa, agdaki bir ana bilgisayara önce ICMP ping yardımcı programıyla veya güvenlik duvarını aşma olasılığı daha yüksek olan Nping gibi özel bir paket oluşturucu ile ping atın. On paketten maksimum gidiş dönüş süresine bakın. Bunu --initial-rtt-timeout için iki katına ve --max-rtt-timeout için üç veya dört katına çıkarmak isteyebilirsiniz. Ping süreleri ne olursa olsun, genellikle maksimum RTT'yi 100 ms'nin altına ayarlamıyorum. Ayrıca 1000 ms'yi de aşmıyorum.
- --min-rtt-timeout, bir ağ Nmap'in varsayılanının bile çok agresif olduğu kadar güvenilmez olduğunda yararlı olabilecek nadiren kullanılan bir seçenekir. Nmap yalnızca ağ güvenilir göründüğünde zaman aşımını minimuma indirdiğinden, bu ihtiyaç olağandıṣıdır ve nmap-dev posta listesine bir hata olarak bildirilmelidir.

`--max-retries <numtries>` (Specify the maximum number of port scan probe retransmissions)

⇒ Nmap bir port tarama probuna yanıt almadığında, bu portunfiltrelendiği anlamına gelebilir. Ya da prob veya yanıt ağ üzerinde kaybolmuş olabilir. Hedef ana bilgisayarın yanıtı geçici olarak engelleyen hız sınırlamasını etkinleştirmiş olması da mümkündür. Bu yüzden Nmap ilk probu yeniden ileterek tekrar dener. Nmap zayıf ağ güvenilirliği tespit ederse, bir bağlantı noktasından vazgeçmeden önce birçok kez daha deneyebilir. Bu doğruluğa fayda sağlarken, tarama sürelerini de uzatır. Performans kritik olduğunda, izin verilen yeniden iletim sayısı sınırlanırlararak taramalar hızlandırılabilir. Herhangi bir yeniden iletimi önlemek için --max-retries 0 bile belirtebilirsiniz, ancak bu yalnızca ara sıra kaçırılan bağlantı noktalarının ve ana bilgisayarların kabul edilebilir olduğu gayri resmi anketler gibi durumlar için önerilir.

- Varsayılan (-T şablonu olmadan) on yeniden iletme izin vermektedir. Bir ağ güvenilir görünüyorsa ve hedef ana bilgisayarlar hız sınırlaması yapmıyorsa, Nmap genellikle yalnızca bir yeniden iletim yapar. Bu yüzden çoğu hedef taraması --max-retries değerinin üç gibi düşük bir değere düşürülmüşinden bile etkilenmez. Bu tür değerler yavaş (hız sınırlı) ana bilgisayarların taramasını önemli ölçüde hızlandırabilir. Nmap bağlantı noktalarından erken vazgeçtiğinde genellikle bazı bilgileri kaybedersiniz, ancak bu --host-

`timeout`'un süresinin dolmasına izin vermek ve hedeflarındaki tüm bilgileri kaybetmekten daha tercih edilebilir olabilir.

`--host-timeout <time>` (Give up on slow target hosts) ⇒ Bazı ana bilgisayarların taranması uzun zaman alır. Bunun nedeni kötü performans gösteren veya güvenilir olmayan ağ donanımı veya yazılımı, paket hızı sınırlaması veya kısıtlayıcı bir güvenlik duvarı olabilir. Taranan ana bilgisayarların en yavaş yüzde birkaç tarama süresinin büyük bir kısmını tüketebilir. Bazen kayıplarınızı azaltmak ve başlangıçta bu ana bilgisayarları atlamak en iyisidir. Beklemek istediğiniz maksimum süreyi `--host-timeout` ile belirtin. Örneğin, Nmap'in tek bir ana bilgisayarda yarı saat fazla zaman kaybetmemesini sağlamak için 30m belirtin. Nmap'in bu yarı saat boyunca aynı anda diğer ana bilgisayarları tarıyor olabileceğini unutmayın, bu nedenle bu tam bir kayıp değildir. Zaman aşımına uğrayan bir ana bilgisayar atlanır. Bu ana bilgisayar için hiçbir bağlantı noktası tablosu, işletim sistemi algılama veya sürüm algılama sonucu yazdırılmaz.

- Özel 0 değeri "zaman aşımı yok" anlamında kullanılabilir ve bu da ana bilgisayar zaman aşımını 15 dakikaya ayarlayan T5 zamanlama şablonunu geçersiz kılmak için kullanılabilir.

`--script-timeout <time>` ⇒ Bazı komut dosyaları saniyenin kesirlerinde tamamlanırken, diğerleri komut dosyasının niteliğine, aktarılan bağımsız değişkenlere, ağ ve uygulama koşullarına ve daha fazlasına bağlı olarak saatler veya daha uzun sürebilir. `--script-timeout` seçeneği, komut dosyası yürütme süresi için bir tavan belirler. Bu süreyi aşan herhangi bir betik örneği sonlandırılacak ve hiçbir çıktı gösterilmeyecektir. Hata ayıklama (-d) etkinleştirilmişse, Nmap her zaman aşımında rapor verecektir. Ana bilgisayar ve hizmet komut dosyaları için, bir komut dosyası örneği yalnızca tek bir hedef ana bilgisayarı veya bağlantı noktasını tarar ve zaman aşımı süresi bir sonraki örnek için sıfırlanır.

- "Zaman aşımı yok" anlamına gelen 0 özel değeri, kod zaman aşımını 10 dakikaya ayarlayan T5 zamanlama şablonunu geçersiz kılmak için kullanılabilir.

`--scan-delay <time>; --max-scan-delay <time>` (Adjust delay between probes) ⇒ Bu seçenek, Nmap'in belirli bir ana bilgisayara gönderdiği her sonda arasında en az verilen süre kadar beklemesine neden olur. Bu özellikle hız sınırlaması durumunda kullanışlıdır. Solaris makineleri (diğerlerinin yanı sıra) genellikle UDP tarama prob paketlerine saniyede yalnızca bir ICMP mesajı ile yanıt verir. Nmap tarafından gönderilen bundan daha fazlası israf olacaktır. 1s'lik bir `--scan-delay` Nmap'i bu

yavaş hızda tutacaktır. Nmap hız sınırlamasını tespit etmeye ve tarama gecikmesini buna göre ayarlamaya çalışır, ancak hangi hızın en iyi şekilde çalıştığını zaten biliyorsanız, bunu açıkça belirtmenin bir zararı olmaz.

- Nmap hız sınırlamasıyla başa çıkmak için tarama gecikmesini yukarı doğru ayarladığında, tarama önemli ölçüde yavaşlar. max-scan-delay seçeneği Nmap'in izin vereceği en büyük gecikmeyi belirtir. Düşük bir --max-scan-delay Nmap'i hızlandırabilir, ancak risklidir. Bu değerin çok düşük ayarlanması, hedef katı hız sınırlaması uyguladığında boş paket yeniden iletimlerine ve olası kaçırılan bağlantı noktalarına yol açabilir.
- --scan-delay'in bir başka kullanımı da eşik tabanlı saldırısı tespit ve önleme sistemlerini (IDS/IPS) atlatmaktadır. Bu teknik "Pratik bir örnek: varsayılan Snort 2.2.0 kurallarını atlama" adlı bölümde Snort IDS'deki varsayılan bağlantı noktası tarama algılayıcısını yenmek için kullanılmıştır. Diğer saldırısı tespit sistemlerinin çoğu da aynı şekilde atlatılabilir.

`--min-rate <number> ; --max-rate <number>` (Directly control the scanning rate) ⇒ Nmap'in dinamik zamanlaması, tarama yapmak için uygun bir hız bulma konusunda iyi bir iş çıkarır. Ancak bazen, bir ağ için uygun bir tarama hızı biliyor olabilirsiniz veya bir taramanın belirli bir zamanda biteceğini garanti etmeniz gerekebilir. Ya da Nmap'in çok hızlı tarama yapmasını engellemeniz gerekebilir. min-rate ve --max-rate seçenekleri bu durumlar için tasarlanmıştır.

- --min-rate seçeneği verildiğinde Nmap, paketleri verilen oran kadar hızlı veya daha hızlı göndermek için elinden geleni yapacaktır. Argüman, saniyede paket cinsinden bir paket oranını temsil eden pozitif bir gerçek sayıdır. Örneğin, --min-rate 300 belirtmesi, Nmap'in gönderme hızını saniyede 300 paket veya üzerinde tutmaya çalışacağı anlamına gelir. Minimum hız belirtmek, koşullar gerektiriyorsa Nmap'in daha hızlı gitmesini engelmez.
- Benzer şekilde, --max-rate bir taramanın gönderme hızını belirli bir maksimum değerle sınırlar. Örneğin, hızlı bir ağa gönderimi saniyede 100 paketle sınırlamak için --max-rate 100 kullanın. Her on saniyede bir paketlik yavaş bir tarama için --max-rate 0.1 kullanın. Hızı belirli bir aralıkta tutmak için --min-rate ve --max-rate değerlerini birlikte kullanın.
- Bu iki seçenek geneldir, tek tek ana bilgisayarları değil tüm taramayı etkiler. Yalnızca bağlantı noktası taramalarını ve ana bilgisayar keşif taramalarını

etkilerler. İşletim sistemi algılama gibi diğer özellikler kendi zamanlamalarını uygular.

- Gerçek tarama hızının istenen minimum hızın altına düşebilecegi iki durum vardır. Birincisi, minimum hızın Nmap'in gönderebileceği en yüksek hızdan daha hızlı olması durumudur ki bu da donanıma bağlıdır. Bu durumda Nmap paketleri mümkün olduğunda hızlı gönderecektir, ancak bu kadar yüksek hızların doğruluk kaybına neden olabileceğini unutmayın. İkinci durum, Nmap'in gönderecek bir şeyi olmadığı durumdur, örneğin son problemin gönderildiği ve Nmap'in bunların zaman aşımına uğramasını veya yanıtlanmasını beklediği bir taramanın sonunda veya ana bilgisayar grupları arasında tarama hızının düştüğünü görmek normaldir. Gönderme hızı, öngörülemeyen gecikmeleri telafi etmek için geçici olarak maksimum değeri aşabilir, ancak ortalama olarak hız maksimum değerde veya altında kalacaktır.
- Minimum hızın belirlenmesi dikkatli bir şekilde yapılmalıdır. Bir ağın destekleyebileceğinden daha hızlı tarama yapmak doğruluk kaybına yol açabilir. Bazı durumlarda, daha hızlı bir oran kullanmak, bir taramanın daha yavaş bir oranla olduğundan daha uzun sürmesine neden olabilir. Bunun nedeni, Nmap'in uyarlanabilir yeniden iletim algoritmalarının aşırı tarama hızının neden olduğu ağ tıkanıklığını tespit etmesi ve doğruluğu artırmak için yeniden iletim sayısını artırmasıdır. Bu nedenle, paketler daha yüksek bir hızda gönderilse de, genel olarak daha fazla paket gönderilir. Toplam tarama süresine bir üst sınır koymaz gerekiyorsa --max-retries seçeneği ile yeniden iletim sayısını sınırlayın.

`--defeat-rst-ratelimit` ⇒ Birçok ana bilgisayar, gönderdikleri ICMP hata mesajlarının (port-unreachable hataları gibi) sayısını azaltmak için uzun süredir hız sınırlaması kullanmaktadır. Bazı sistemler artık ürettikleri RST (reset) paketlerine de benzer hız limitleri uygulamaktadır. Bu, zamanlamasını bu hız sınırlarını yansıtacak şekilde ayarladığı için Nmap'i önemli ölçüde yavaşlatabilir. Nmap'e --defeat-rst-ratelimit komutunu vererek bu hız sınırlarını (SYN taraması gibi yanıt vermeyen portları açık olarak değerlendirmeyen port taramaları için) göz ardı etmesini söyleyebilirsiniz.

Bu seçeneği kullanmak doğruluğu azaltabilir, çünkü bazı bağlantı noktaları yanıt vermiyor görünecektir çünkü Nmap hız sınırlı bir RST yanıtı için yeterince uzun süre beklememiştir. Bir SYN taramasıyla, yanıt vermeme, RST paketleri alındığında

gördüğümüz kapalı durum yerine bağlantı noktasının filtrelenmiş olarak etiketlenmesine neden olur. Bu seçenek, yalnızca açık bağlantı noktalarını önemseyinizde ve kapalı ve filtrelenmiş bağlantı noktaları arasında ayrim yapmak fazladan zaman harcamaya deðmediðinde kullanışlıdır.

--defeat-icmp-ratelimit → defeat-rst-ratelimit seçeneğine benzer şekilde, --defeat-icmp-ratelimit seçeneği de ICMP hata mesajlarını hız sınırlamasına tabi tutan ana bilgisayarlara karşı UDP tarama hızını artırarak doğruluðu hız ile takas eder. Bu seçenek Nmap'in ulaþılamayan bağlantı noktası mesajlarını almak için gecikmemesine neden olduğundan, yanıt vermeyen bir bağlantı noktası varsayılan open|filtered yerine closed|filtered olarak etiketlenecektir. Bu, yalnızca UDP üzerinden gerçekten yanıt veren bağlantı noktalarını açık olarak ele alma etkisine sahiptir. Birçok UDP hizmeti bu şekilde yanıt vermediðinden, bu seçenekle yanlışlık ihtiyimali --defeat-rst-ratelimit seçeneğine göre daha yüksektir.

--nsock-engine iocp|epoll|kqueue|poll|select → Belirli bir nsock IO çoklama motorunun kullanımını zorunlu kılar. Yalnızca select(2)tabanlı yedek motorun sisteminizde kullanılabilir olduğu garanti edilir. Motorlar, yararlandıkları IO yönetim tesisinin adıyla adlandırılır. Şu anda uygulanan motorlar epoll, kqueue, poll ve select'tir, ancak hepsi herhangi bir platformda mevcut olmayacağındır. Varsayılan olarak, Nmap "en iyi" motoru, yani bu listede desteklenen ilk motoru kullanacaktır.
Platformunuzda hangi motorların desteklendiðini görmek için nmap -V kullanın.

-T paranoid|sneaky|polite|normal|aggressive|insane (Set a timing template) → Önceki bölümde tartışılan ince taneli zamanlama kontrolleri güçlü ve etkili olsa da, bazı insanlar bunları kafa karıştırıcı bulmaktadır. Dahası, uygun değerleri seçmek bazen optimize etmeye çalışığınız taramadan daha fazla zaman alabilir. Neyse ki, Nmap altı zamanlama şablonu ile daha basit bir yaklaşım sunar. Bunları -T seçeneği ve numaraları (0-5) ya da adları ile belirtebilirsiniz. Şablon adları paranoid (0), sinsi (1), kibar (2), normal (3), agresif (4) ve çılgın (5) şeklindedir. İlk ikisi IDS'den kaçmak içindir. Kibar mod, daha az bant genişliği ve hedef makine kaynaðı kullanmak için taramayı yavaşlatır. Normal mod varsayılandır ve bu nedenle -T3 hiçbir şey yapmaz. Agresif mod, oldukça hızlı ve güvenilir bir ağda olduğunuzu varsayıarak taramaları hızlandırır. Son olarak çılgın mod, olaðanüstü hızlı bir ağda olduğunuzu veya hız için bir miktar doğruluðu feda etmeye istekli olduğunuzu varsayar.

- Bu şablonlar, kullanıcının ne kadar agresif olmak istediğini belirtmesine izin verirken, Nmap'i tam zamanlama değerlerini seçmeye bırakır. Şablonlar ayrıca, şu anda ince taneli kontrol seçeneklerinin mevcut olmadığı bazı küçük hız ayarlamaları da yapar. Örneğin, -T4 TCP bağlantı noktaları için dinamik tarama gecikmesinin 10 ms'yi aşmasını yasaklar ve -T5 bu değeri 5 ms ile sınırlar. Şablonlar ince ayarlı kontrollerle birlikte kullanılabilir ve belirttiğiniz ince ayarlı kontroller o parametre için zamanlama şablonu varsayılanına göre öncelikli olacaktır. Oldukça modern ve güvenilir ağları tararken -T4 kullanmanızı öneririm. Ince taneli kontroller eklediğinizde bile bu seçeneği koruyun, böylece etkinleştirdiği ekstra küçük optimizasyonlardan yararlanabilirsiniz.
- Eğer iyi bir geniş bant ya da ethernet bağlantınız varsa, her zaman -T4 kullanmanızı tavsiye ederim. Benim zevkime göre çok agresif olsa da bazı insanlar -T5'i seviyor. İnsanlar bazen ana bilgisayarları çökertme olasılığının daha düşük olduğunu düşündükleri için veya kendilerini genel olarak kibar gördükleri için -T2'yi belirtirler. Genellikle -T kibarlığının gerçekte ne kadar yavaş olduğunun farkında değildir. Taramaları varsayılan taramadan on kat daha uzun sürebilir. Varsayılan zamanlama seçeneklerinde (-T3) makine çökmeleri ve bant genişliği sorunları nadirdir ve bu nedenle normalde dikkatli tarayıcılar için bunu öneririm. Sürüm algılamayı atlamak, bu sorunları azaltmada zamanlama değerleriyle oynamaktan çok daha etkilidir.
- T0 ve -T1 IDS uyarılarından kaçınmak için yararlı olsa da, binlerce makineyi veya bağlantı noktasını taramak için olağanüstü uzun bir zaman alacaktır. Bu kadar uzun bir tarama için, hazır -T0 ve -T1 değerlerine güvenmek yerine ihtiyacınız olan tam zamanlama değerlerini ayarlamayı tercih edebilirsiniz.
- T0'ın ana etkileri taramayı seri hale getirerek bir seferde sadece bir portun taranmasını sağlamak ve her bir probun gönderilmesi arasında beş dakika beklemektir. T1 ve T2 benzerdir, ancak probalar arasında sırasıyla yalnızca 15 saniye ve 0,4 saniye beklerler. T3, Nmap'in paralelleştirmeyi içeren varsayılan davranışıdır. -T4, --max-rtt-timeout 1250ms --min-rtt-timeout 100ms --initial-rtt-timeout 500ms --max-retries 6 eşdeğerini yapar ve maksimum TCP ve SCTP tarama gecikmesini 10ms olarak ayarlar. T5, --max-rtt-timeout 300ms --min-rtt-timeout 50ms --initial-rtt-timeout 250ms --max-retries 2 --host-timeout 15m --script-timeout 10m eşdeğerini yapar ve maksimum TCP ve SCTP tarama gecikmesini 5ms olarak ayarlar. Maksimum UDP tarama

gecikmesi T4 veya T5 tarafından ayarlanmaz, ancak --max-scan-delay seçeneği ile ayarlanabilir.

Firewall/IDS Evasion and Spoofing (Güvenlik Duvarı/IDS Kaçırma ve Spoofing)

Birçok İnternet öncüsü, herhangi iki düğüm arasında sanal bağlantılarla izin veren evrensel bir IP adres alanına sahip küresel bir açık ağ öngörmüştür. Bu sayede ana bilgisayarlar gerçek eşler olarak hareket edebilecek, birbirlerine bilgi sunabilecek ve birbirlerinden bilgi alabileceklerdi. İnsanlar iş yerlerinden tüm ev sistemlerine erişebilir, klima kontrol ayarlarını değiştirebilir veya erken gelen misafirler için kapıların kilidini açabilir. Bu evrensel bağlantı vizyonu, adres alanı sıkıntısı ve güvenlik endişeleri nedeniyle engellendi. 1990'ların başında kuruluşlar, bağlanabilirliği azaltmak amacıyla güvenlik duvarları kullanmaya başladı. Büyük ağlar, uygulama proxy'leri, ağ adresi çevirisi ve paket filtreleri ile filtrelenmemiş Internet'ten kordon altına alındı. Sınırsız bilgi akışı yerini onaylı iletişim kanallarının ve bu kanallar üzerinden geçen içeriğin sıkı bir şekilde düzenlenmesine bıraktı.

Güvenlik duvarları gibi ağ engelleri bir ağın haritasını çıkarmayı son derece zorlaştırabilir. Sıradan keşifleri bastırmak genellikle cihazların uygulanmasının temel bir amacı olduğundan, daha da kolaylaşmayacaktır. Bununla birlikte, Nmap bu karmaşık ağları anlamaya yardımcı olmak ve filtrelerin amaçlandığı gibi çalıştığını doğrulamak için birçok özellik sunar. Hatta kötü uygulanan savunmaları atlatmak için mekanizmaları bile destekler. Ağ güvenlik duruşunu anlamanın en iyi yöntemlerinden biri onu yenmeye çalışmaktır. Kendinizi bir saldırının zihniyetine yerleştirin ve bu bölümdeki teknikleri ağlarınıza karşı kullanın. Bir FTP sığrama taraması, boşta tarama, parçalanma saldırısı başlatın veya kendi proxy'lerinizden biri üzerinden tünel açmayı deneyin.

Ağ faaliyetlerini kısıtlamanın yanı sıra, şirketler saldırısı tespit sistemleri (IDS) ile trafiği giderek daha fazla izlemektedir. Tüm büyük IDS'ler, Nmap taramalarını tespit etmek için tasarlanmış kurallarla birlikte gönderilir, çünkü taramalar bazen saldırıların öncüsüdür. Bu ürünlerin birçoğu son zamanlarda kötü niyetli olduğu düşünülen trafiği aktif olarak engelleyen saldırısı önleme sistemlerine (IPS) dönüştürülmüştür. Ne yazık ki ağ yöneticileri ve IDS satıcıları için, paket verilerini analiz

ederek kötü niyetleri güvenilir bir şekilde tespit etmek zor bir sorundur. Sabır, beceri ve belirli Nmap seçeneklerinin yardımıyla saldırganlar genellikle IDS'leri tespit edilmeden geçebilirler. Bu arada, yöneticiler masum faaliyetlerin yanlış teşhis edildiği ve uyarıldığı veya engellendiği çok sayıda yanlış pozitif sonuçla başa çıkmak zorundadır.

Zaman zaman insanlar Nmap'in güvenlik duvarı kurallarından kaçmak veya IDS'leri gizlice geçmek için özellikler sunmaması gerektiğini öne sürüyorlar. Bu özelliklerin saldırganlar tarafından kötüye kullanılma olasılığının, yöneticiler tarafından güvenliği artırmak için kullanılma olasılığı kadar yüksek olduğunu savunurlar. Bu mantıkla ilgili sorun, bu yöntemlerin saldırganlar tarafından kullanılmaya devam edeceğii ve saldırganların başka araçlar bulacağı ya da işlevselligi Nmap'e yamalayacağidir. Bu arada, yöneticilerin işlerini yapmaları çok daha zor olacaktır. Yalnızca modern, yamalı FTP sunucuları dağıtmak, FTP sıçrama saldırısını uygulayan araçların dağıtımını engellemeye çalışmaktan çok daha güçlü bir savunmadır.

Güvenlik duvarlarını ve IDS sistemlerini tespit etmek ve yıkmak için sihirli bir mermi (veya Nmap seçeneği) yoktur. Bu beceri ve deneyim gerektirir. Bir öğretici, yalnızca ilgili seçenekleri listeleyen ve ne yaptıklarını açıklayan bu referans kılavuzunun kapsamı dışındadır.

`-f` (fragment packets); `--mtu` (using the specified MTU) ⇒ f seçeneği, istenen taramanın (ana bilgisayar keşif taramaları dahil) küçük parçalara ayrılmış IP paketleri kullanmasına neden olur. Buradaki fikir, TCP başlığını birkaç pakete bölgerek paket filtrelerinin, izinsiz giriş tespit sistemlerinin ve diğer rahatsızlıkların ne yaptığınızı tespit etmesini zorlaştırmaktır. Bu konuda dikkatli olun! Bazı programlar bu küçük paketleri işlemekte sorun yaşarlar. Sniffit segmentation adlı eski tip sniffer ilk parçayı alır almadı hata verdi. Bu seçeneği bir kez belirttiğinizde Nmap paketleri IP başlığından sonra sekiz ya da daha az bayta böler. Yani 20 baytlık bir TCP başlığı üç pakete bölünecektir. İkisi TCP başlığının sekiz baytını ve biri de son dört baytı içerir. Elbette her parçanın bir IP başlığı da vardır. Belirtmek Parça başına 16 bayt kullanmak için -f seçeneğini tekrar belirtin (parça sayısını azaltır). Ya da --mtu seçeneği ile kendi offset boyutunuza belirtebilirsiniz. Eğer --mtu kullanıyorsanız -f seçeneğini de belirtmeyin. Ofset sekizin katı olmalıdır. Parçalanmış paketler, Linux çekirdeğindeki CONFIG_IP_ALWAYS_DEFRAG seçeneği gibi tüm IP parçalarını sıraya koyan paket filtrelerinden ve güvenlik duvarlarından geçemeyecek olsa da, bazı ağlar bunun neden olduğu performans

düşüşünü göze alamaz ve bu nedenle devre dışı bırakır. Diğerleri bunu etkinleştiremez çünkü parçalar kendi ağlarına farklı yollardan girebilir. Bazı kaynak sistemler giden paketleri çekirdekte birleştirir. İptables bağlantı izleme modülüne sahip Linux buna bir örnektir. Gönderilen paketlerin parçalandığından emin olmak için Wireshark gibi bir sniffer çalışırken bir tarama yapın. Ana bilgisayar işletim sisteminiz sorunlara neden oluyorsa, IP katmanını atlamanak ve ham ethernet çerçeveleri göndermek için --send-eth seçeneğini deneyin.

- Parçalama yalnızca Nmap'in TCP ve UDP bağlantı noktası taramalarını (bağlantı taraması ve FTP sıçrama taraması hariç) ve işletim sistemi algılamasını içeren ham paket özellikleri için desteklenir. Sürüm algılama ve Nmap Scripting Engine gibi özellikler genellikle parçalanmayı desteklemez çünkü hedef hizmetlerle iletişim kurmak için ana bilgisayarınızın TCP yiğinına güvenirler.

`-D <decoy1> [, <decoy2>][,ME][,...]` (Cloak a scan with decoys) ⇒ Bir yem taramasının gerçekleştirilemesine neden olur, bu da uzak ana bilgisayara yem olarak belirlediğiniz ana bilgisayar(lar)ın da hedef ağı tariyormuş gibi görünmesini sağlar. Böylece IDS'leri benzersiz IP adreslerinden 5-10 port taraması rapor edebilir, ancak hangi IP'nin onları taradığını ve hangilerinin masum yem olduğunu bilemezler. Bu durum yönlendirici yolu izleme, yanıt düşürme ve diğer aktif mekanizmalarla aşılabilse de, genellikle IP adresinizi gizlemek için etkili bir tekniktir.

- Her tuzak ana bilgisayarı virgülle ayırin ve istege bağlı olarak ME'yi gerçek IP adresinizin konumunu temsil etmek için tuzaklardan biri olarak kullanabilirsiniz. ME'yi altıncı veya daha sonraki bir konuma koyarsanız, bazı yaygın port tarama dedektörlerinin (Solar Designer'in mükemmel Scanlogd'u gibi) IP adresinizi göstermesi pek olası değildir. ME kullanmazsanız, Nmap siz rastgele bir konuma yerlestirecektir. Ayrıca rastgele, rezerve edilmemiş bir IP adresi oluşturmak için RND veya <sayı> adresleri oluşturmak için RND:<sayı> kullanabilirsiniz.
- Tuzak olarak kullandığınız ana bilgisayarların açık olması gerektiğini unutmayın, aksi takdirde yanlışlıkla hedeflerinizi SYN seline maruz bırakabilirsiniz. Ayrıca, ağda yalnızca bir ana bilgisayar açıksa hangi ana bilgisayarın tarama yaptığıni belirlemek oldukça kolay olacaktır. Adlar yerine IP adreslerini kullanmak

isteyebilirsiniz (böylece sahte ağlar siz ad sunucusu günlüklerinde görmez). Şu anda rastgele IP adresi üretimi yalnızca IPv4 ile desteklenmektedir

- Tuzaklar hem ilk ana bilgisayar keşif taramasında (ICMP, SYN, ACK veya her neyse kullanılarak) hem de gerçek port tarama aşamasında kullanılır. Tuzaklar ayrıca uzak işletim sistemi tespiti (-O) sırasında da kullanılır. Tuzaklar sürüm algılama veya TCP bağlantı taraması ile çalışmaz. Bir tarama gecikmesi yürürlükte olduğunda, gecikme her bir prob arasında değil, her bir sahte prob grubu arasında uygulanır. Aldatıcılar bir kerede toplu olarak gönderildiğinden, tıkanıklık kontrol sınırlarını geçici olarak ihlal edebilirler.
- Çok fazla tuzak kullanmanın taramanızı yavaşlatabileceğini ve hatta potansiyel olarak daha az doğru yapabileceğini belirtmek gereklidir. Ayrıca, bazı İSS'ler sahte paketlerinizi filtreleyecektir, ancak çoğu sahte IP paketlerini hiç kısıtlamaz.

-S <IP_Address> (Spoof source address) ⇒ Bazı durumlarda, Nmap kaynak adresinizi belirleyemeyebilir (Nmap bu durumda size bilgi verecektir). Bu durumda, paketleri göndermek istediğiniz arayüzün IP adresiyle birlikte -S kullanın.

- Bu bayrağın bir başka olası kullanımı da hedeflerin başka birinin onları taradığını düşünmesini sağlamak için taramayı taklit etmektir. Bir şirketin bir rakibi tarafından defalarca port taramasından geçirildiğini düşünün! Bu tür bir kullanım için genellikle -e seçeneği ve -Pn gereklidir. Genellikle yanıt paketlerini geri almayacağınızı unutmayın (bunlar sahtecilik yaptığınız IP'ye yönlendirilecektir), bu nedenle Nmap yararlı raporlar üretmeyecektir.

-e <interface> (Use specified interface) ⇒ Nmap'e paketlerin hangi arayüzden gönderileceğini ve alınacağını söyler. Nmap bunu otomatik olarak algılayabilmelidir, ancak algılayamazsa size söyleyecektir.

--source-port <portnumber> ;-g <portnumber> (Spoof source port number) ⇒ Şaşırıcı derecede yaygın bir yanlış yapılandırma, trafiğe yalnızca kaynak bağlantı noktası numarasına göre güvenmektir. Bunun nasıl ortaya çıktığını anlamak kolaydır. Bir yönetici pırıl pırıl yeni bir güvenlik duvarı kuracak, ancak uygulamaları çalışmayı durdurun nankör kullanıcılarından gelen şikayetlerle dolup taşacaktır. Özellikle DNS bozulabilir çünkü harici sunuculardan gelen UDP DNS yanıtları artık ağa giremez. FTP başka bir yaygın örnektir. Aktif FTP aktarımlarında, uzak sunucu istenen dosyayı aktarmak için istemciye geri bağlantı kurmaya çalışır.

- Bu sorumlara genellikle uygulama düzeyinde proxy'ler veya protokol ayırtıran güvenlik duvarı modülleri şeklinde güvenli çözümler mevcuttur. Ne yazık ki daha kolay, güvensiz çözümler de vardır. DNS yanıtlarının 53 numaralı bağlantı noktasından ve aktif FTP'nin 20 numaralı bağlantı noktasından geldiğine dikkat çeken birçok yönetici, bu bağlantı noktalarından gelen trafiğe izin verme tuzağına düşmüştür. Genellikle hiçbir saldırganın bu tür güvenlik duvari açıklarını fark etmeyeceğini ve istismar etmeyeceğini varsayarlar. Diğer durumlarda, yöneticiler bunu daha güvenli bir çözüm uygulayana kadar kısa vadeli bir geçici önlem olarak görürler. Sonra da güvenlik yükseltmesini unuturlar.
- Bu tuzağa düşenler sadece fazla çalışan ağ yöneticileri değildir. Çok sayıda ürün bu güvensiz kurallarla birlikte gönderilmiştir. Microsoft bile suçludur. Windows 2000 ve Windows XP ile birlikte gelen IPsec filtreleri, 88 numaralı bağlantı noktasından (Kerberos) gelen tüm TCP veya UDP trafiğine izin veren örtük bir kural içermektedir. İyi bilinen bir başka durumda, Zone Alarm kişisel güvenlik duvarının 2.1.25'e kadar olan sürümleri, 53 (DNS) veya 67 (DHCP) kaynak bağlantı noktasına sahip tüm gelen UDP paketlerine izin veriyordu.
- Nmap bu zayıflıklardan yararlanmak için -g ve --source-port seçeneklerini (esdeğerdirler) sunar. Basitçe bir port numarası verin ve Nmap mümkün olduğunda bu porttan paketler gönderecektir. SYN ve UDP taramaları da dahil olmak üzere ham soket kullanan çoğu tarama işlemi bu seçeneği tamamen destekler. Bu seçenek, DNS istekleri, TCP bağlantı taraması, sürüm algılama ve komut dosyası taraması dahil olmak üzere normal işletim sistemi soketlerini kullanan hiçbir işlem için etkili değildir. Kaynak bağlantı noktasını ayarlamak işletim sistemi algılaması için de işe yaramaz, çünkü Nmap'in belirli işletim sistemi algılama testlerinin düzgün çalışması için farklı bağlantı noktası numaraları kullanması gereklidir.

`--data <hex string>` (Append custom binary data to sent packets) ⇒ Bu seçenek, gönderilen paketlere yük olarak ikili veri eklemenizi sağlar. `<hex string>` aşağıdaki formatlardan herhangi birinde belirtilebilir: `0xAABBCCDDEEFF<...>`, `AABBCCDDEEFF<...>` veya `\xAA\xBB\xCC\xDD\xEE\xFF<...>`. Kullanım örnekleri `--data 0xdeadbeef` ve `--data \xCA\xFE\x09`'dur. `0x00ff` gibi bir sayı belirtirseniz bayt sırası dönüşümü yapılmayacağını unutmayın. Bilgileri alıcının beklediği bayt sırasına göre belirttiğinizden emin olun.

`--data-string <string>` (Append custom string to sent packets) ⇒ Bu seçenek, gönderilen paketlere yük olarak normal bir dize eklemenizi sağlar. `<string>` herhangi bir dize içerebilir. Ancak, bazı karakterlerin sisteminizin yerel ayarına bağlı olabileceğini ve alıcının aynı bilgiyi görmeyebileceğini unutmayın. Ayrıca, dizeyi çift tırnak içine aldığından ve kabuktaki tüm özel karakterlerden kaçtığınızdan emin olun. Örnekler: `--data-string "Scan conducted by Security Ops, extension 7192"` veya `--data-string "Ph34r my l33t skills"`. Bir sniffer veya özel IDS kuralları ile ağı dikkatlice izlemediği sürece, hiç kimse bu seçenek tarafından bırakılan herhangi bir yorumu gerçekten görmeyeceğini unutmayın.

`--data-length <number>` (Append random data to sent packets) ⇒ Normalde Nmap sadece bir başlık içeren minimalist paketler gönderir. Bu nedenle TCP paketleri genellikle 40 bayttır ve ICMP yankı istekleri sadece 28'dir. Bazı UDP portları ve IP protokolleri varsayılan olarak özel bir yük alır. Bu seçenek Nmap'e gönderdiği paketlerin çoguna verilen sayıda rastgele bayt eklemesini ve herhangi bir protokole özgü yük kullanmamasını söyler. (Rastgele veya protokole özel yükler kullanmamak için `--data-length 0` kullanın. İşletim sistemi algılama (-O) paketleri etkilenmez, çünkü buradaki doğruluk prob tutarlılığı gerektirir, ancak çoğu ping ve portscan paketi bunu destekler. İşleri biraz yavaşlatır, ancak bir taramayı biraz daha az dikkat çekici hale getirebilir.

`--ip-options <R/S [route]/L [route]/T/U ... > ;--ip-options <hex string>` (Send packets with specified ip options) ⇒ IP protokolü, paket başlıklarına yerleştirilebilecek çeşitli seçenekler sunar. Her yerde bulunan TCP seçeneklerinin aksine, IP seçenekleri pratiklik ve güvenlik kaygıları nedeniyle nadiren görülür. Aslında, birçok İnternet yönlendiricisi kaynak yönlendirme gibi en tehlikeli seçenekleri engeller. Yine de seçenekler bazı durumlarda hedef makinelere giden ağ rotasını belirlemek ve değiştirmek için yararlı olabilir. Örneğin, daha geleneksel traceroute tarzı yaklaşımalar başarısız olduğunda bile bir hedefe giden yolu belirlemek için record route seçeneğini kullanabilirsiniz. Ya da paketleriniz belirli bir güvenlik duvarı tarafından düşürülüyorsa, sıkı veya gevşek kaynak yönlendirme seçenekleriyle farklı bir rota belirleyebilirsiniz.

- IP seçeneklerini belirtmenin en güçlü yolu, `--ip-options`'a argüman olarak değerleri basitçe iletmektir. Her onaltılik sayının önüne \x ve ardından iki rakam ekleyin. Belirli karakterleri bir yıldız işaretи ve ardından tekrarlanması

istediğiniz sayı ile takip ederek tekrarlayabilirsiniz. Örneğin,
\\x01\\x07\\x04\\x00*36\\x01 36 NUL baytı içeren bir hex dizesidir.

- Nmap ayrıca seçenekleri belirtmek için bir kısayol mekanizması sunar. Sırasıyla record-route, record-timestamp veya her iki seçeneği birlikte istemek için R, T veya U harfini geçirmeniz yeterlidir. Gevşek veya katı kaynak yönlendirme, bir L veya S ve ardından bir boşluk ve ardından boşluk bırakılarak ayrılmış bir IP adresleri listesi ile belirtilebilir.
- Gönderilen ve alınan paketlerdeki seçenekleri görmek isterseniz --packet-trace seçeneğini belirtin. IP seçeneklerinin Nmap ile kullanımı hakkında daha fazla bilgi ve örnekler için bkz. <https://seclists.org/nmap-dev/2006/q3/52>.

--ttl <value> (Set IP time-to-live field) ⇒ Gönderilen paketlerdeki IPv4 yaşam süresi alanını verilen değere ayarlar.

--randomize-hosts (Randomize target host order) ⇒ Nmap'e 16384 ana bilgisayara kadar olan her grubu taramadan önce karıştırmasını söyler. Bu, özellikle yavaş zamanlama seçenekleriyle birleştirdiğinizde, taramaları çeşitli ağ izleme sistemleri için daha az belirgin hale getirebilir. Daha büyük grup boyutları üzerinde rastgele tarama yapmak istiyorsanız, nmap.h dosyasında PING_GROUP_SZ değerini artırın ve yeniden derleyin. Alternatif bir çözüm, hedef IP listesini bir liste taramasıyla (-sL -n -oN <dosya adı>) oluşturmak, bir Perl betiği ile rastgele hale getirmek ve ardından tüm listeyi -iL ile Nmap'e sağlamaktır.

--spoof-mac <MAC address, prefix, or vendor name> (Spoof MAC address) ⇒ Nmap'in gönderdiği tüm ham ethernet çerçeveleri için verilen MAC adresini kullanmasını ister. Bu seçenek, Nmap'in gerçekten ethernet düzeyinde paketler gönderdiğinde emin olmak için --send-eth'i ima eder. Verilen MAC çeşitli biçimlerde olabilir. Eğer sadece 0 sayısı ise, Nmap oturum için tamamen rastgele bir MAC adresi seçer. Verilen dize çift sayıda onaltılık basamaktan oluşuyorsa (çiftler isteğe bağlı olarak iki nokta üst üste ile ayrılır), Nmap MAC olarak bunları kullanır. Eğer 12'den az hex hanesi verilmişse, Nmap altı baytin geri kalanını rastgele değerlerle doldurur. Eğer argüman sıfır veya onaltılık bir dize değilse, Nmap verilen dizeyi içeren bir satıcı adı bulmak için nmap-mac-prefixes'e bakar (büyük/küçük harfe duyarlı değildir). Bir eşleşme bulunursa, Nmap satıcının OUI'sini (üç baylıklı önek) kullanır ve kalan üç baytı rastgele doldurur. Geçerli --spoof-mac bağımsız değişken örnekleri Apple, 0, 01:02:03:04:05:06, deadbeefcafe, 0020F2 ve Cisco'dur. Bu seçenek

yalnızca SYN taraması veya OS tespiti gibi ham paket taramalarını etkiler, sürüm tespiti veya Nmap Scripting Engine gibi bağlantı odaklı özellikleri etkilemez.

--proxies <Comma-separated list of proxy URLs> (Relay TCP connections through a chain of proxies) ⇒ Nmap'ten bir veya daha fazla HTTP veya SOCKS4 proxy zinciri aracılığıyla son hedefle TCP bağlantıları kurmasını ister. Proxy'ler bir taramanın gerçek kaynağını gizlemeye veya belirli güvenlik duvarı kısıtlamalarından kaçınmaya yardımcı olabilir, ancak gecikmeyi artırarak tarama performansını engelleyebilirler. Kullanıcıların Nmap zaman aşımılarını ve diğer tarama parametrelerini buna göre ayarlamaları gerekebilir. Özellikle, daha düşük bir --max-parallelism yardımcı olabilir, çünkü bazı proxy'ler Nmap'in varsayılan olarak açtığı kadar çok eşzamanlı bağlantıyı işlemeyi reddeder.

- Bu seçenek, proto://host:port biçiminde URL'ler olarak ifade edilen bir proxy listesini argüman olarak alır. Bir zincirdeki düğüm URL'lerini ayırmak için virgül kullanın. Henüz kimlik doğrulama desteklenmemektedir. Geçerli protokoller HTTP ve SOCKS4'tür.
- Uyarı: Bu özellik hala geliştirilme aşamasındadır ve sınırlamaları vardır. Nsock kütüphanesi içinde uygulanmaktadır ve bu nedenle taramanın ping, port tarama ve işletim sistemi keşif aşamaları üzerinde hiçbir etkisi yoktur. Şimdilik sadece NSE ve sürüm taraması bu seçenekten faydalananmaktadır; diğer özellikler gerçek adresinizi ifşa edebilir. SSL bağlantıları henüz desteklenmemektedir ve proxy tarafı DNS çözümlemesi de yapılmamaktadır (ana bilgisayar adları her zaman Nmap tarafından çözümlenir).

--badsum (Send packets with bogus TCP/UDP checksums) ⇒ Nmap'ten hedef ana bilgisayarlara gönderilen paketler için geçersiz bir TCP, UDP veya SCTP sağlama toplamı kullanmasını ister. Neredeyse tüm ana IP yiğinları bu paketleri düzgün bir şekilde düşürdüğünden, alınan yanıtlar büyük olasılıkla sağlama toplamını doğrulama zahmetine girmeyen bir güvenlik duvarı veya IDS'den geliyordur. Bu teknik hakkında daha fazla ayrıntı için bkz. <https://nmap.org/p60-12.html>

--adler32 (Use deprecated Adler32 instead of CRC32C for SCTP checksums) Nmap'in SCTP sağlama toplamını hesaplamak için kullanılmış kaldırılmış Adler32 algoritmasını kullanmasını ister. Eğer --adler32 verilmezse, CRC-32C (Castagnoli) kullanılır. RFC 2960 başlangıçta Adler32'yi SCTP için sağlama toplamı algoritması olarak tanımladı; RFC 4960 daha sonra SCTP sağlama toplamlarını CRC-32C kullanmak üzere yeniden tanımladı. Mevcut SCTP uygulamaları CRC-32C

kullanmalıdır, ancak eski, eski SCTP uygulamalarından yanıt almak için Adler32 kullanmak tercih edilebilir.

Output (Çıkış)

Herhangi bir güvenlik aracı ancak ürettiği çıktı kadar faydalıdır. Karmaşık testler ve algoritmalar, düzenli ve anlaşılır bir şekilde sunulmazlarsa çok az değer taşırlar. Nmap'in insanlar ve diğer yazılımlar tarafından kullanıldığı yolların sayısı göz önüne alındığında, tek bir format herkesi memnun edemez. Bu nedenle Nmap, insanların doğrudan okuması için etkileşimli mod ve yazılım tarafından kolay ayırtırma için XML dahil olmak üzere çeşitli formatlar sunar.

Farklı çıktı biçimleri sunmanın yanı sıra, Nmap çıktıının ayrıntı düzeyini ve hata ayıklama mesajlarını kontrol etmek için seçenekler sunar. Çıktı türleri standart çıktıya ya da Nmap'in ekleylebileceği ya da saklayabileceği adlandırılmış dosyalara gönderilebilir. Çıktı dosyaları iptal edilen taramaları devam ettirmek için de kullanılabilir.

Nmap çıktıyı beş farklı biçimde kullanıma sunar. Varsayılan çıktı etkileşimli çıktı olarak adlandırılır ve standart çıktıya (stdout) gönderilir. Bir de normal çıktı vardır; bu da etkileşimli çıktıya benzer, ancak etkileşimli çıktı yerine tarama tamamlandıktan sonra analiz edilmesi beklenigidinden daha az çalışma zamanı bilgisi ve uyarı görüntüleri.

XML çıktısı, HTML'ye dönüştürülebildiği, Nmap grafik kullanıcı arayüzleri gibi programlar tarafından kolayca ayırtırılabildiği veya veritabanlarına aktarılabilenliği için en önemli çıktı türlerinden biridir.

Geriye kalan iki çıktı türü, bir hedef ana bilgisayar için çoğu bilgiyi tek bir satırda içeren basit grepable çıktısı ve kendilerini |←r4d olarak gören kullanıcılar için sCRiPt KiDDi3 OutPUt'tur.

Etkileşimli çıktı varsayılandır ve ilişkili komut satırı seçenekleri yoktur, ancak diğer dört biçim seçeneği aynı sözdizimini kullanır. Sonuçların saklanacağı dosya adı olan bir bağımsız değişken alırlar. Birden fazla format belirtilebilir, ancak her format yalnızca bir kez belirtilebilir. Örneğin, kendi incelemeniz için normal çıktıyı kaydederken, programatik analiz için aynı taramanın XML'ini kaydetmek

isteyebilirsiniz. Bunu -oX myscan.xml -oN myscan.nmap seçenekleri ile yapabilirsiniz. Bu bölümde kısalık için myscan.xml gibi basit isimler kullanılsa da, genellikle daha açıklayıcı isimler önerilir. Seçilen isimler kişisel bir tercih meselesiştir, ancak ben tarama tarihini ve taramayı açıklayan bir iki kelimeyi içeren ve taradığım şirketin adını taşıyan bir dizine yerleştirilen uzun isimler kullanıyorum. Bu seçenekler sonuçları dosyalara kaydederken, Nmap etkileşimli çıktıyı her zamanki gibi stdout'a yazdırır ve devam eder. Örneğin, nmap -oX myscan.xml -target komutu XML'i myscan.xml dosyasına yazdırır ve standart çıktıyı -oX belirtmemiş olsaydı yazdıracağı etkileşimli sonuçlarla doldurur. Biçim türlerinden birine argüman olarak bir tire karakteri geçirerek bunu değiştirebilirsiniz. Bu, Nmap'in etkileşimli çıktıyı devre dışı bırakmasına ve bunun yerine sonuçları standart çıktı akışına belirttiğiniz biçimde yazdırmasına neden olur. Yani nmap -oX -target komutu stdout'a sadece XML çıktısı gönderecektir. Ciddi hatalar yine de normal hata akışı olan stderr'ye yazdırılabilir.

Bazı Nmap argümanlarından farklı olarak, logfile seçenek bayrağı (-oX gibi) ile dosya adı veya kısa çizgi arasındaki boşluk zorunludur. Bayrakları atlar ve -oG- veya -oXscan.xml gibi argümanlar verirseniz, Nmap'in geriye dönük uyumluluk özelliği sırasıyla G- ve Xscan.xml adlı normal formatlı çıktı dosyalarının oluşturulmasına neden olacaktır.

Bu argümanların tümü dosya adında strftime benzeri dönüşümleri destekler. H, %M, %S, %m, %d, %y ve %Y'nin tümü strftime'daki ile tamamen aynıdır. T, %H%M%S ile aynıdır, %R, %H%M ile aynıdır ve %D, %m%d%y ile aynıdır. Herhangi bir karakterin ardından gelen bir % sadece o karakteri verir (% % size bir yüzde sembolü verir). Yani -oX 'scan-%T-%D.xml', scan-144840-121307.xml şeklinde bir adı olan bir XML dosyası kullanacaktır.

Nmap ayrıca tarama ayrıntı düzeyini kontrol etmek ve çıktı dosyalarına ekleme yapmak için seçenekler sunar. Tüm bu seçenekler aşağıda açıklanmıştır.

Nmap Çıktı Biçimleri

-oN <filespec> (normal output) ⇒ Normal çıktıının verilen dosya adına yönlendirilmesini ister. Yukarıda tartışıldığı gibi, bu etkileşimli çıktıdan biraz farklıdır.

-oX <filespec> (XML output) ⇒ XML çıktısının verilen dosya adına yönlendirilmesini ister. Nmap, XML ayırtıcılarının Nmap XML çıktısını doğrulamasını sağlayan bir

belge türü tanımı (DTD) içerir. Öncelikle programatik kullanım için tasarlanmış olsa da, insanların Nmap XML çıktısını yorumlamasına da yardımcı olabilir. DTD, formatın yasal öğelerini tanımlar ve genellikle alabilecekleri nitelikleri ve değerleri sıralar. En son sürüm her zaman <https://svn.nmap.org/nmap/docs/nmap.dtd> adresinden edinilebilir.

- XML, yazılım tarafından kolayca ayırtılabilen kararlı bir format sunar. C/C++, Perl, Python ve Java dahil olmak üzere tüm büyük bilgisayar dilleri için ücretsiz XML ayırtıcıları mevcuttur. Hatta insanlar bu dillerin çoğu için Nmap çıktısını ve yürütmesini özel olarak işlemek üzere bağlayıcılar bile yazmışlardır. Örnek olarak Perl CPAN'deki Nmap::Scanner ve Nmap::Parser verilebilir. Önemsiz olmayan bir uygulamanın Nmap ile arayüz oluşturduğu neredeyse tüm durumlarda, XML tercih edilen biçimdir.
- XML çıktısı, sonuçları HTML olarak biçimlendirmek için kullanılabilen bir XSL stil sayfasına başvurur. Bunu kullanmanın en kolay yolu XML çıktısını Firefox ya da IE gibi bir web tarayıcısına yüklemektir. Varsayılan olarak bu, sabit kodlanmış nmap.xsl dosya sistemi yolu nedeniyle yalnızca Nmap'i çalıştırıldığınız makinede (veya benzer şekilde yapılandırılmış bir makinede) çalışacaktır. Web'e bağlı herhangi bir makinede HTML olarak işlenen taşınabilir XML dosyaları oluşturmak için --webxml veya --stylesheet seçeneklerini kullanın.

-oS <filespec> (ScRipT Klđđ|3 oUTpuT) ⇒ Script kiddie çıktısı interaktif çıktı gibidir, ancak daha önce tutarlı büyük harf kullanımı ve yazımı nedeniyle Nmap'i küçümseyen I33t HaXXorZ'a daha iyi uyması için sonradan işlenmiştir. Mizah özürlü insanlar, sözde "onlara yardım ettiğim" için beni suçlamadan önce bu seçeneğin betik çocuklarıyla dalga geçtiğini unutmamalıdır.

-oG <filespec> (grepable output) ⇒ Bu çıktı biçimini kullanımdan kaldırıldığı için en son ele alınmıştır. XML çıktı biçimi çok daha güçlündür ve deneyimli kullanıcılar için neredeyse aynı derecede kullanışlıdır. XML, düzinelere mükemmel ayırtıcının mevcut olduğu bir standarttır, grepable çıktı ise benim kendi basit hack'imdir. XML, yeni Nmap özellikleri yayınlandıça bunları destekleyecek şekilde genişletilebilirken, ben bu özellikleri koyacak bir yer olmadığı için sık sık grepable çıktısından çıkarmak zorunda kalıyorum.

- Bununla birlikte, greplenebilir çıktı hala oldukça popülerdir. Her bir ana bilgisayarı tek bir satırda listeleyen basit bir formattır ve grep, awk, cut, sed,

diff ve Perl gibi standart Unix araçları ile kolayca aranabilir ve ayırtılabilir. Ben bile genellikle komut satırında yapılan tek seferlik testler için kullanıyorum. SSH portu açık olan veya Solaris çalıştırın tüm ana bilgisayarları bulmak, ana bilgisayarları tanımlamak için sadece basit bir grep gerektirir, istenen alanları yazdırma için bir awk veya cut komutuna borulanır.

- Grepable çıktısı yorumlardan (pound (#) ile başlayan satırlar) ve hedef satırlardan oluşur. Bir hedef satırı, sekmelerle ayrılmış ve iki nokta üst üste ile takip edilen altı etiketli alanın bir kombinasyonunu içerir. Bu alanlar Ana Bilgisayar, Bağlantı Noktaları, Protokoller, Yoksayıılma Durumu, İşletim Sistemi, Seq Dizini, IP Kimliği ve Durum'dur.
- Bu alanlardan en önemlisi, genellikle her ilginç bağlantı noktası hakkında ayrıntılar veren Bağlantı Noktalarıdır. Virgülle ayrılmış bir port girişleri listesidir. Her port girişi bir ilginç portu temsil eder ve yedi eğik çizgi (/) ile ayrılmış alt alan şeklini alır. Bu alt alanlar şunlardır: Port numarası, Durum, Protokol, Sahip, Hizmet, SunRPC bilgisi ve Sürüm bilgisi.
- XML çıktısında olduğu gibi, bu kılavuz sayfası tüm biçimin belgelenmesine izin vermez. Nmap grepable çıktı formatına daha detaylı bir bakış "Grepable Output (-oG)" adlı bölümde mevcuttur.

`-oA <basename>` (Output to all formats) ⇒ Kolaylık olması açısından, tarama sonuçlarını normal, XML ve grepable formatlarında bir kerede saklamak için `-oA <basename>` belirtebilirsiniz. Bunlar sırasıyla `<basename>.nmap`, `<basename>.xml` ve `<basename>.gnmap` dosyalarında saklanır. Çoğu programda olduğu gibi, dosya adlarının önüne Unix'te `~/nmaplogs/foocorp/` veya Windows'ta `c:\hacking\sco` gibi bir dizin yolu ekleyebilirsiniz.

Verbosity ve hata ayıklama seçenekleri

`-v` (Increase verbosity level), `-V <level>` (Set verbosity level) ⇒ Ayrıntı düzeyini artırarak Nmap'in devam etmekte olan tarama hakkında daha fazla bilgi yazdırmasına neden olur. Açık portlar bulundukça gösterilir ve Nmap bir taramanın birkaç dakikadan fazla süreceğini düşündüğünde tamamlanma süresi tahminleri sağlanır. Daha fazla ayrıntı için iki veya daha fazla kez kullanın: `-vv` veya doğrudan bir ayrıntı seviyesi verin, örneğin `-v3`.

- Çoğu değişiklik yalnızca etkileşimli çıktıyı etkiler ve bazıları normal ve komut dosyası kiddie çıktısını da etkiler. Diğer çıktı türleri makineler tarafından

işlenmek üzere tasarlanmıştır, bu nedenle Nmap bir insan kullanıcıyı yormadan bu formatlarda varsayılan olarak önemli ayrıntılar verebilir. Bununla birlikte, diğer modlarda bazı ayrıntıların atlanmasıyla çıktı boyutunun önemli ölçüde azaltılabilcegi birkaç değişiklik vardır. Örneğin, grepable çıktısında taranan tüm portların bir listesini sağlayan bir yorum satırı oldukça uzun olabileceğinden yalnızca ayrıntılı modda yazdırılır.

`-d` (Increase debugging level), `-d <level>` (Set debugging level) ⇒ Verbose modu sızın için yeterli veri sağlamadığında, hata ayıklama sizi çok daha fazlasıyla doldurmak için kullanılabilir! Verbosity seçeneğinde (-v) olduğu gibi, hata ayıklama bir komut satırı bayrağı (-d) ile etkinleştirilir ve hata ayıklama seviyesi -dd'de olduğu gibi birden fazla kez belirtilerek veya doğrudan bir seviye ayarlanarak artırılabilir. Örneğin, -d9 dokuzuncu seviyeyi ayarlar. Bu en yüksek etkili seviyedir ve çok az sayıda bağlantı noktası ve hedefle çok basit bir tarama yapmadığınız sürece binlerce satır üretecektir.

- Hata ayıklama çıktısı, Nmap'te bir hatadan şüphelenildiğinde veya Nmap'in ne yaptığı ve neden yaptığı konusunda kafanız karıştıığında kullanışlıdır. Bu özellik çoğunlukla geliştiriciler için tasarlandığından, hata ayıklama satırları her zaman kendini açıklayıcı değildir. Şöyledir bir şeyle karşılaşabilirsiniz: Zaman aşımı vals: srtt: -1 rttvar: -1 to: 1000000 delta 14987 ⇒ srtt: 14987 rttvar: 14987 to: 100000. Eğer bir satırı anlamadıysanız, tek çareniz onu görmezden gelmek, kaynak koduna bakmak ya da geliştirme listesinden (nmap-dev) yardım istemektir. Bazı satırlar kendi kendini açıklar, ancak hata ayıklama seviyesi arttıkça mesajlar daha belirsiz hale gelir.

`--reason` (Host and port state reasons) ⇒ Her bağlantı noktasının belirli bir duruma ayarlanma nedenini ve her ana bilgisayarın yukarı veya aşağı olma nedenini gösterir. Bu seçenek, bir bağlantı noktasını veya ana bilgisayar durumunu belirleyen paketin türünü görüntüler. Örneğin, kapalı bir bağlantı noktasından gelen bir RST paketi veya canlı bir ana bilgisayardan gelen bir yanıt yanıtı. Nmap'in sağlayabileceği bilgiler tarama veya ping türüne göre belirlenir. SYN taraması ve SYN ping (-sS ve -PS) çok ayrıntılıdır, ancak TCP bağlantı taraması (-sT) connect sistem çağrısının uygulanmasıyla sınırlıdır. Bu özellik hata ayıklama seçeneği (-d) tarafından otomatik olarak etkinleştirilir ve bu seçenek belirtilmese bile sonuçlar XML günlük dosyalarında saklanır.

--stats-every <time> (Print periodic timing stats) ⇒ Her **<time>** aralığından sonra periyodik olarak bir zamanlama durum mesajı yazdırır. Zaman, "Zamanlama ve Performans" başlıklı bölümde açıklanan türden bir belirtimdir; örneğin, her 10 saniyede bir durum güncellemesi almak için --stats-every 10s kullanın.

Güncellemeler etkileşimli çıktıya (ekran) ve XML çıktısına yazdırılır.

--packet-trace (Trace packets and data sent and received) ⇒ Nmap'in gönderilen veya alınan her paketin bir özetini yazdırmasına neden olur. Bu genellikle hata ayıklama için kullanılır, ancak aynı zamanda yeni kullanıcıların Nmap'in örtülerin altında tam olarak ne yaptığını anlamaları için değerli bir yoldur. Binlerce satır yazdırılmaktan kaçınmak için, -p20-30 gibi taranacak sınırlı sayıda bağlantı noktası belirtmek isteyebilirsiniz. Yalnızca sürüm algılama alt sisteminin gidişatıyla ilgileniyorsanız, bunun yerine --version-trace kullanın. Yalnızca komut dosyası izlemeyi önemsiyorsanız, --script-trace seçeneğini belirtin. -packet-trace ile yukarıdakilerin tümünü elde edersiniz.

--open (Show only open (or possibly open) ports) ⇒ Bazen sadece gerçekten bağlanabileceğiniz portlarla (açık olanlar) ilgilenirsiniz ve sonuçların kapalı, filtrelenmiş ve kapalı|filtrelenmiş portlarla karışmasını istemezsınız. Çıktı özelleştirme normalde grep, awk ve Perl gibi araçlar kullanılarak taramadan sonra yapılır, ancak bu özellik yoğun istekler nedeniyle eklenmiştir. Yalnızca en az bir açık, açık|filtrelenmiş veya filtrelenmemiş bağlantı noktası olan ana bilgisayarları görmek ve yalnızca bu durumlardaki bağlantı noktalarını görmek için --open belirtin. Bu üç durum normalde olduğu gibi ele alınır, bu da open|filtered ve unfiltered durumlarının çok fazla sayıda olması halinde sayımlarda yoğunlaştırılabileceği anlamına gelir.

- Nmap 7.40 ile başlayarak, --open seçeneği --defeat-rst-ratelimit anlamına gelir, çünkü bu seçenek yalnızca --open tarafından gizlenen kapalı ve filtrelenmiş bağlantı noktalarını etkiler.

--iflist (List interfaces and routes) ⇒ Nmap tarafından tespit edilen arayüz listesini ve sistem rotalarını yazdırır ve çıkar. Bu, yönlendirme sorunlarını veya aygit yanlış tanımlamalarını (Nmap'in bir PPP bağlantısını ethernet olarak ele alması gibi) açıklamak için kullanışlıdır.

Çeşitli çıktı seçenekleri

--append-output (Append to rather than clobber output files) ⇒ oX veya -oN gibi bir çıktı biçimini bayrağına bir dosya adı belirttiğinizde, varsayılan olarak bu dosyanın üzerine yazılır. Dosyanın mevcut içeriğini korumayı ve yeni sonuçları eklemeyi tercih ederseniz, --append-output seçeneğini belirtin. Bu Nmap yürütmesinde belirtilen tüm çıktı dosya adları daha sonra clobbered yerine eklenecektir. Bu, XML (-oX) tarama verileri için iyi çalışmaz, çünkü sonuçta ortaya çıkan dosya, siz elle düzeltene kadar genellikle düzgün bir şekilde ayırtılmalıdır.

--resume <filename> (Resume aborted scan) ⇒ Bazı kapsamlı Nmap çalışmaları çok uzun zaman alır - günlerce. Bu tür taramalar her zaman tamamlanmayabilir. Kısıtlamalar Nmap'in çalışma saatleri içinde çalıştırılmasını engelleyebilir, ağ çökebilir, Nmap'in üzerinde çalıştığı makine planlı ya da plansız bir şekilde yeniden başlatılabilir ya da Nmap'in kendisi çökebilir. Nmap'i çalıştırın yönetici, ctrl-C tuşuna basarak başka herhangi bir nedenle de iptal edebilir. Tüm taramanın baştan başlatılması istenmeyebilir. Neyse ki, tarama çıktı dosyaları saklandıysa, kullanıcı Nmap'ten yürütme durduğunda üzerinde çalıştığı hedefle taramaya devam etmesini isteyebilir. Basitçe --resume seçeneğini belirtin ve çıktı dosyasını argüman olarak iletin. Nmap çıktı dosyasını daha önce belirtilenlerle aynı şekilde kullanmak üzere ayırtıldığından başka argümanlara izin vermez. Nmap'i nmap --resume <logfilename> şeklinde çağrımanız yeterlidir. Nmap yeni sonuçları önceki yürütmede belirtilen veri dosyalarına ekleyecektir. Taramalar 3 ana çıktı formatından herhangi birinden devam ettirilebilir: Normal, Grepable veya XML

--noninteractive (Disable runtime interactions) ⇒ Nmap'i bir kabuk arka planında çalıştırırmak gibi zamanlarda, Nmap'in çalışırken kullanıcı klavye girdisini izlemesi ve yanıt vermesi istenmeyebilir. (Tarama sırasında Nmap'in nasıl kontrol edileceği hakkında "Çalışma Zamanı Etkileşimi" bölümüne bakın). Nmap'in terminalin kontrolünü ele geçirmesini önlemek için --noninteractive seçeneğini kullanın.

--stylesheet <path or URL> (Set XSL stylesheet to transform XML output) ⇒ Nmap, XML çıktısını görüntülemek veya HTML'ye çevirmek için nmap.xsl adlı bir XSL stil sayfası ile birlikte gönderilir. XML çıktısı, Nmap tarafından başlangıçta yüklentiği nmap.xml dosyasına işaret eden bir xmlstylesheet yönergesi içerir. Bir HTML dosyası üretmek için XML dosyasını xsltproc gibi bir XSLT işlemcisinden geçirin. XML dosyasını bir tarayıcıda doğrudan açmak artık iyi çalışmıyor çünkü modern tarayıcılar bir stil sayfasının yüklenebileceği konumları sınırlıyor. Farklı bir stil sayfası kullanmak isterseniz, bunu --stylesheet argümanı olarak belirtin. Tam yol adını veya URL'yi geçmelisiniz. Yaygın bir çağrıma --stylesheet

<https://nmap.org svn/docs/nmap.xsl> şeklindedir. Bu, bir XSLT işlemcisine Nmap.Org'dan stil sayfasının en son sürümünü yüklemesini söyler. --webxml seçeneği de aynı şeyi daha az yazarak ve ezberleyerek yapar. XSL'yi Nmap.Org'dan yüklemek, Nmap (ve dolayısıyla nmap.xsl) yüklü olmayan bir makinede sonuçları görüntülemeyi kolaylaştırır. Bu nedenle URL genellikle daha kullanışlıdır, ancak nmap.xsl'nin yerel dosya sistemi konumu gizlilik nedeniyle varsayılan olarak kullanılır.

--webxml (Load stylesheet from Nmap.Org) ⇒ Bu bir kolaylık seçeneğidir, --stylesheet <https://nmap.org svn/docs/nmap.xsl> için bir takma addan başka bir şey değildir.

--no-stylesheet (Omit XSL stylesheet declaration from XML) ⇒ Nmap'in XML çıktısıyla herhangi bir XSL stil sayfasını ilişkilendirmesini önlemek için bu seçeneği belirtin. xml-stylesheet yönergesi atlanır.

Miscellaneous Options (Çeşitli Seçenekler)

Bu bölümde, başka hiçbir yeresgiyan bazı önemli (ve o kadar da önemli olmayan) seçenekler açıklanmaktadır.

-6 (Enable IPv6 scanning) ⇒ Nmap en popüler özellikleri için IPv6 desteğine sahiptir. Ping taraması, port taraması, sürüm tespiti ve Nmap Scripting Engine'in tümü IPv6'yi destekler. Komut sözdizimi, -6 seçeneğini de eklemeniz dışında her zamanki gibidir. Elbette, bir ana bilgisayar adı yerine bir adres belirtirseniz IPv6 sözdizimini kullanmanız gereklidir. Bir adres 3ffe:7501:4819:2000:210:f3ff:fe03:14d0 gibi görünebilir, bu nedenle ana bilgisayar adları önerilir. Çıktı her zamanki gibi aynı görünür, "ilginç bağlantı noktaları" satırındaki IPv6 adresi tek IPv6 eşantyonudur.

- IPv6 dünyayı tam olarak kasıp kavurmamış olsa da, bazı (genellikle Asya) ülkelerde ölçüde önemli ölçüde kullanılmaktadır ve çoğu modern işletim sistemi bunu desteklemektedir. Nmap'i IPv6 ile kullanmak için, taramanızın hem kaynağı hem de hedefi IPv6 için yapılandırılmış olmalıdır. İSS'niz (çoğu gibi) size IPv6 adresleri tahsis etmiyorsa, ücretsiz tünel aracları yaygın olarak mevcuttur ve Nmap ile iyi çalışır. Ben <http://www.tunnelbroker.net> adresindeki ücretsiz IPv6

tünel aracı hizmetini kullanıyorum. Diğer tünel aracları Wikipedia'da listelenmiştir. 6to4 tünelleri bir başka popüler, ücretsiz yaklaşımındır.

- Windows'ta, ham soket IPv6 taramaları yalnızca ethernet cihazlarında (tünelde değil) ve yalnızca Windows Vista ve sonraki sürümlerde desteklenir. Diğer durumlarda --unprivileged seçeneğini kullanın.

-A (Aggressive scan options) ⇒ Bu seçenek ek gelişmiş ve agresif seçenekleri etkinleştirir. Şu anda işletim sistemi algılama (-O), sürüm tarama (-sV), komut dosyası tarama (-sC) ve traceroute (--traceroute) özelliklerini etkinleştirmektedir. Gelecekte daha fazla özellik eklenebilir. Buradaki amaç, insanların çok sayıda bayrağı hatırlamak zorunda kalmadan kapsamlı bir tarama seçenekleri kümesini etkinleştirmektir. Ancak, varsayılan ayarla komut dosyası taraması müdahaleci olarak kabul edildiğinden, -A seçeneğini hedef ağlara karşı izinsiz kullanmamalısınız. Bu seçenek yalnızca özellikleri etkinleştirir, zamanlama seçeneklerini (-T4 gibi) veya isteyebileceğiniz ayrıntı seçeneklerini (-v) etkinleştirmez. İşletim sistemi algılama ve traceroute gibi ayrıcalıklar (örneğin root erişimi) gerektiren seçenekler yalnızca bu ayrıcalıklar mevcutsa etkinleştirilecektir.

--datadir <directoryname> (Specify custom Nmap data file location) ⇒ Nmap çalışma zamanında nmap-service-probes, nmap-services, nmap-protocols, nmap-rpc, nmap-mac-prefixes ve nmap-os-db adlı dosyalarda bazı özel veriler elde eder. Bu dosyalardan herhangi birinin konumu belirtilmişse (--servicedb veya --versiondb seçenekleri kullanılarak), o dosya için o konum kullanılır. Bundan sonra, Nmap bu dosyaları --datadir seçeneği ile belirtilen dizinde arar (eğer varsa). Burada bulunamayan dosyalar NMAPDIR ortam değişkeni tarafından belirtilen dizinde aranır. Ardından gerçek ve etkin UID'ler için ~/.nmap gelir; veya Windows'ta <HOME>\AppData\Roaming\nmap (burada <HOME> kullanıcının ev dizinidir, C:\Users\user gibi). Bunu nmap çalıştırılabilir dosyasının konumu ve ..\share/nmap eklenmiş aynı konum takip eder. Ardından /usr/local/share/nmap veya /usr/share/nmap gibi derlenmiş bir konum.

--servicedb <services file> (Specify custom services file) ⇒ Nmap'ten, Nmap ile birlikte gelen nmap-services veri dosyası yerine belirtilen hizmetler dosyasını kullanmasını ister. Bu seçeneğin kullanılması aynı zamanda hızlı taramanın (-F) kullanılmasına neden olur. Nmap'in veri dosyaları hakkında daha fazla bilgi için --datadir açıklamasına bakın.

`--versiondb <service probes file>` (Specify custom service probes file) ⇒ Nmap'ten, Nmap ile birlikte gelen nmap-service-probes veri dosyası yerine belirtilen hizmet problemleri dosyasını kullanmasını ister. Nmap'in veri dosyaları hakkında daha fazla bilgi için --datadir açıklamasına bakın.

`--send-eth` (Use raw ethernet sending) ⇒ Nmap'in paketleri daha yüksek IP (ağ) katmanı yerine ham ethernet (veri bağlantısı) katmanında göndermesini ister. Varsayılan olarak, Nmap genellikle üzerinde çalıştığı platform için en iyi olanı seçer. Ham soketler (IP katmanı) genellikle Unix makineler için en verimliken, Microsoft ham soket desteğini devre dışı bıraktığından Windows çalışması için ethernet çerçeveleri gereklidir. Nmap, başka bir seçenek olmadığından (ethernet olmayan bağlantılar gibi) bu seçenek'e rağmen Unix'te hala ham IP paketlerini kullanır.

`--send-ip` (Send at raw IP level) ⇒ Nmap'in daha düşük seviyeli ethernet çerçeveleri göndermek yerine ham IP soketleri aracılığıyla paket göndermesini ister. Daha önce tartışılan --send-eth seçenekinin tamamlayıcısıdır.

`--privileged` (Assume that the user is fully privileged) ⇒ Nmap'e, ham soket gönderimleri, paket koklama ve genellikle Unix sistemlerinde kök ayrıcalıkları gerektiren benzer işlemleri gerçekleştirmek için yeterince ayrıcalıklı olduğunu varsaymasını söyler. Varsayılan olarak, bu tür işlemler talep edilirse ancak geteuid sıfır değilse Nmap çıkar. --privileged, Linux çekirdek yetenekleri ve ayrıcalıksız kullanıcıların ham paket taramaları yapmasına izin verecek şekilde yapılandırılabilen benzer sistemlerde kullanışlıdır. Bu seçenek bayrağını, ayrıcalık gerektiren seçenekler (SYN taraması, OS tespiti, vb.) için herhangi bir bayraktan önce sağladığınızdan emin olun. NMAP_PRIVILEGED ortam değişkeni --privileged seçeneğine eşdeğer bir alternatif olarak ayarlanabilir.

`--unprivileged` (Assume that the user lacks raw socket privileges) ⇒ Bu seçenek --privileged seçeneğinin tersidir. Nmap'e kullanıcıya ağ ham soketi ve koklama ayrıcalıkları yokmuş gibi davranışını söyler. Bu, test, hata ayıklama veya işletim sisteminizin ham ağ işlevselligi bir şekilde bozulduğunda kullanışlıdır. NMAP_UNPRIVILEGED ortam değişkeni --unprivileged seçeneğine eşdeğer bir alternatif olarak ayarlanabilir.

`--release-memory` (Release memory before quitting) ⇒ Bu seçenek yalnızca bellek sızıntısı hata ayıklama için kullanışlıdır. Nmap'in ayrılmadan hemen önce ayrılan belleği serbest bırakmasına neden olur, böylece gerçek bellek sızıntılarını tespit

etmek daha kolaydır. Normalde Nmap bunu atlar çünkü işletim sistemi bunu işlem sonlandırıldığında zaten yapar.

`-V` ; `--version` (Print version number) ⇒ Nmap sürüm numarasını yazdırır ve çıkar.

`-h` ; `--help` (Print help summary page) ⇒ En yaygın komut bayraklarını içeren kısa bir yardım ekranı yazdırır. Nmap'i herhangi bir argüman olmadan çalıştırırmak da aynı şeyi yapar.

Runtime Interaction (Çalışma Zamanı Etkileşimi)

Nmap'in yürütülmesi sırasında tüm tuş basışları yakalanır. Bu, programı iptal etmeden ve yeniden başlatmadan programla etkileşim kurmanızı olanak tanır. Bazı özel tuşlar seçenekleri değiştirirken, diğer tuşlar size tarama hakkında bilgi veren bir durum mesajı yazdırır. Kural olarak küçük harfler yazdırma miktarını artırır ve büyük harfler yazdırmayı azaltır. Yardım için '?' tuşuna da basabilirsiniz.

`v` / `V` ⇒ Ayrıntı düzeyini artırma / azaltma

`d` / `D` ⇒ Hata ayıklama seviyesini artırma/azaltma

`p` / `P` ⇒ Paket izlemeyi açma / kapatma

`?` ⇒ Çalışma zamanı etkileşimi yardım ekranı yazdırma

Başka bir şey

Bunun gibi bir durum mesajı yazdırın:

```
Stats: 0:00:07 elapsed; 20 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 20:57 (0:00:12 remaining)
```

Examples (Örnekler)

İşte basit ve rutinden biraz daha karmaşık ve ezoterik olana kadar bazı Nmap kullanım örnekleri. Bazı gerçek IP adresleri ve alan adları, işleri daha somut hale getirmek için kullanılmıştır. Bunların yerine kendi ağınızdaki adresleri/adları

kullanmalısınız. Diğer ağların port taramasının yasadışı olduğunu ya da olması gerektiğini düşünmesem de, bazı ağ yöneticileri ağlarının istenmeyen bir şekilde taranmasından hoşlanmazlar ve şikayet edebilirler. Önce izin almak en iyi yaklaşımındır.

Test amacıyla scanme.nmap.org ana bilgisayarını tarama iznine sahipsiniz. Bu izin yalnızca Nmap aracılığıyla taramayı içerir ve istismarları veya hizmet redi saldırınızı test etmeyi içermez. Bant genişliğini korumak için, lütfen bu ana bilgisayara karşı günde bir düzineden fazla tarama başlatmayın. Bu ücretsiz tarama hedefi kötüye kullanılrsa, kaldırılacak ve Nmap verilen ana bilgisayar adı/IP'nin çözümlenemediğini bildirecektir: scanme.nmap.org. Bu izinler scanme2.nmap.org, scanme3.nmap.org ve benzeri ana bilgisayarlar için de geçerlidir, ancak bu ana bilgisayarlar şu anda mevcut değildir.

nmap -v scanme.nmap.org ⇒ Bu seçenek scanme.nmap.org makinesindeki tüm ayrılmış TCP portlarını tarar. v seçeneği ayrıntılı modu etkinleştirir.

nmap -sS -O scanme.nmap.org/24 ⇒ Scanme'nin bulunduğu /24 boyutlu ağdaki 256 IP'den açık olan her makineye karşı gizli bir SYN taraması başlatır. Ayrıca, çalışır durumda olan her ana bilgisayarda hangi işletim sisteminin çalıştığını belirlemeye çalışır. Bu, SYN taraması ve işletim sistemi tespiti nedeniyle kök ayrıcalıkları gerektirir.

nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127 ⇒ 198.116.0.0/16 adres alanındaki 255 olası sekiz bitlik alt ağın her birinin ilk yarısında ana bilgisayar numaralandırma ve TCP taraması başlatır. Bu, sistemlerin standart bağlantı noktalarında SSH, DNS, POP3 veya IMAP ya da 4564 numaralı bağlantı noktasında herhangi bir şey çalıştırıp çalışmadığını test eder. Açık bulunan bu bağlantı noktalarından herhangi biri için, hangi uygulamanın çalıştığını belirlemek için sürüm algılama kullanılır.

nmap -v -iR 100000 -Pn -p 80 ⇒ Nmap'ten rastgele 100.000 ana bilgisayar seçmesini ve bunları web sunucuları (port 80) için taramasını ister. Ana bilgisayar numaralandırma -Pn ile devre dışı bırakılır, çünkü bir ana bilgisayarın çalışıp çalışmadığını belirlemek için önce birkaç prob göndermek, zaten her hedef ana bilgisayarda yalnızca bir bağlantı noktasını araştırırken boş harcanır.

nmap -Pn -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20 ⇒ Bu, herhangi bir web sunucusu için 4096 IP'yi tarar (onlara ping atmadan) ve çıktıyı grepable ve XML formatlarında kaydeder.

Nmap Book (Nmap Kitap)

Bu referans kılavuzu tüm önemli Nmap seçeneklerini detaylandırsa da, gerçek dünyadaki görevleri hızlı bir şekilde çözmek için bu özelliklerin nasıl uygulanacağını tam olarak gösteremez. Bunun için Nmap Network Scanning'i yayınladık: Ağ Keşfi ve Güvenlik Taraması için Resmi Nmap Projesi Kılavuzu. Konular arasında güvenlik duvarlarının ve saldırı tespit sistemlerinin alt edilmesi, Nmap performansının optimize edilmesi ve Nmap Scripting Engine ile ortak ağ görevlerinin otomatikleştirilmesi yer almaktadır. Ağ envanteri çıkarma, sizma testi, sahte kablosuz erişim noktalarını tespit etme ve ağ solucanı salgınlarını bastırma gibi yaygın Nmap görevleri için ipuçları ve talimatlar verilmektedir. Örnekler ve diyagramlar kablo üzerindeki gerçek iletişimini göstermektedir. Kitabın yarısından fazlası online olarak ücretsiz temin edilebilir. Daha fazla bilgi için <https://nmap.org/book> adresine bakınız.

Bugs (Hatalar)

Yazarı gibi, Nmap de mükemmel değildir. Ancak hata raporları göndererek ve hatta yamalar yazarak daha iyi hale getirilmesine yardımcı olabilirsiniz. Nmap beklediğiniz gibi davranışmazsa, önce <https://nmap.org> adresinden temin edilebilen en son sürümü yükseltin. Sorun devam ederse, daha önce keşfedilmiş ve ele alınmış olup olmadığını belirlemek için biraz araştırma yapın. Çok sayıda forumu bir araya getirdiği için sorunu veya hata mesajını Google'da aramayı deneyin. Bundan bir şey çıkmazsa, izleyicimizde bir Sorun oluşturun (<http://issues.nmap.org>) ve/veya dev@nmap.org adresine bir hata raporu gönderin. Göndermeden önce nmap-dev listesine abone olursanız, mesajınız moderasyonu atlayacak ve daha hızlı ulaşacaktır. <https://nmap.org/mailman/listinfo/dev> adresinden abone olun. Lütfen sorun hakkında öğrendiğiniz her şeyin yanı sıra hangi Nmap sürümünü kullandığınızı ve hangi işletim sistemi sürümünde çalıştığını da ekleyin. Nmap'i geliştirmek için diğer öneriler de Nmap dev posta listesine gönderilebilir.

Nmap'i geliştiren veya bir hatayı düzeltten bir yama yazabiliyorsanız, bu daha da iyi! Yamaları veya git çekme isteklerini göndermek için talimatlara <https://github.com/nmap/nmap/blob/master/CONTRIBUTING.md> adresinden ulaşabilirsiniz.

Güvenlik raporları gibi özellikle hassas konular doğrudan Nmap'in yazarı Fyodor'a fyodor@nmap.org adresinden gönderilebilir. Diğer tüm raporlar ve yorumlar bunun yerine geliştirici listesini veya sorun izleyiciyi kullanmalıdır, çünkü daha fazla kişi bunları okur, takip eder ve yanıtlar.

Authors (Yazarlar)

Gordon "Fyodor" Lyon fyodor@nmap.org Nmap'i 1997 yılında yazdı ve yayınladı. O zamandan bu yana, Nmap ile birlikte dağıtılan ve <https://nmap.org/changelog.html> adresinden de erişilebilen CHANGELOG dosyasında ayrıntılı olarak açıklandığı üzere, yüzlerce kişi değerli katkılarında bulundu. David Fifield ve Daniel Miller, yıllarca süren muazzam katkıları için özel bir takdiri hak ediyor!

Legal Notices (Yasal Bildirimler)

Nmap Telif Hakkı ve Lisanslama

Nmap Güvenlik Tarayıcısı (C) 1996-2022 Nmap Software LLC'dir ("The Nmap Project"). Nmap aynı zamanda Nmap Projesi'nin tescilli ticari markasıdır. Nmap Kamu Kaynak Lisansı altında yayınlanmaktadır. Bu genellikle son kullanıcıların Nmap'i ücretsiz olarak indirmesine ve kullanmasına izin verir. Nmap'in ticari yazılım veya donanım ürünlerinde (cihazlar, sanal makineler ve geleneksel uygulamalar dahil) kullanılmasına ve yeniden dağıtımasına izin vermez. Bu amaçla <https://nmap.org/oem> adresinde açıklandığı gibi özel bir Nmap OEM Sürümü satarak projeyi finanse ediyoruz. Yüzlerce büyük ve küçük yazılım satıcısı, ana bilgisayar keşfi, bağlantı noktası taraması, işletim sistemi algılama, sürüm algılama ve Nmap Komut Dosyası Motoru gibi Nmap teknolojisini ürünlerine yerleştirmek için OEM lisansları satın almıştır.

Nmap Projesi, Microsoft Windows platformu için bir paket yakalama sürücüsü ve kütüphanesi olan Npcap'i yeniden dağıtma iznine sahiptir. Npcap, bu Nmap lisansından ziyade kendi lisansına sahip ayrı bir çalışmadır. Npcap lisansı özel izin olmadan yeniden dağıtıma izin vermediğinden, Npcap içeren Nmap Windows ikili paketlerimiz özel izin olmadan yeniden dağıtılamaz.

NPSL, GPLv2'yi temel alsa da farklı hükümler içerir ve doğrudan uyumlu değildir. Diğer bazı açık kaynak lisansları ile de uyumsuzdur. Bazı durumlarda Nmap'in bazı bölümlerini yeniden lisanslayabilir veya diğer açık kaynaklı yazılımlarda kullanmak için özel izinler verebiliriz. Lütfen bu tür talepleriniz için fyodor@nmap.org ile iletişime geçin. Benzer şekilde, telif hakkı sahiplerinden özel izin olmadan uyumsuz açık kaynak yazılımları Nmap'e dahil etmiyoruz.

Nmap için bunlardan farklı şartları belirten yazılı bir lisans anlaşması veya sözleşmesi (Nmap OEM lisansı gibi) aldıysanız, Nmap'i bu şartlar altında kullanmayı ve yeniden dağıtmayı seçebilirsiniz.

Bu Nmap Kılavuzu İçin Creative Commons Lisansı

Bu Nmap Referans Kılavuzu (C) 2005-2022 Nmap Software LLC'ye aittir. Creative Commons Attribution License'ın 3.0 sürümü altında yer almaktadır. Bu, orijinal kaynağa atıfta bulunduğuuz sürece çalışmayı istediğiniz gibi yeniden dağıtmansa ve değiştirmenize izin verir. Alternatif olarak, bu belgeyi Nmap'in kendisi ile aynı lisans altında değerlendirmeyi seçebilirsiniz (daha önce tartışılmıştır).

Kaynak Kodunun Kullanılabilirliği ve Topluluk Katkıları

Kullanıcıların bir programı çalıştırmadan önce tam olarak ne yapacağını bilmeye hakları olduğuna inandığımız için bu yazılıma kaynak sağlanmıştır. Bu aynı zamanda yazılımı güvenlik açıklarına karşı denetlemenize de olanak tanır.

Kaynak kodu ayrıca Nmap'i yeni platformlara taşmanızı, hataları düzeltmenizi ve yeni özellikler eklemenizi olanak tanır. Değişikliklerinizi Github Çekme İstekleri (PR) olarak göndermeniz veya ana dağıtıma olası dahil edilmeleri için dev@nmap.org adresine göndermeniz önemle tavsiye edilir. Bu tür değişiklikleri göndererek, Nmap Projesi'ne kodu yeniden kullanma, değiştirme ve yeniden lisanslama için sınırsız, münhasır olmayan bir hak sunduğunuz varsayılmaktadır. Bu önemlidir çünkü kodun yeniden lisanslanamaması diğer Özgür Yazılım projeleri (KDE ve NASM gibi) için yıkıcı sorumlara neden olmuştur. Ayrıca Nmap OEM için

ticari lisanslar da satıyoruz. Katkılarınız için özel lisans koşulları belirtmek isterseniz, bunları gönderirken belirtmeniz yeterlidir.

Garanti Verilmemesi

Bu program, yararlı olacağı umuduyla, ancak HİÇBİR GARANTİ VERİLMESEN; hatta ZİMNİ TİCARİ ELVERİŞLİLİK veya BELİRLİ BİR AMACA UYGUNLUK garantisini bile verilmeden dağıtılmaktadır.

Nmap'in zaman zaman kötü yazılmış uygulamaları, TCP/IP yiğinlarını ve hatta işletim sistemlerini çökerttiği de bilinmektedir. Bu son derece nadir olsa da, akılda tutulması önemlidir. Kesinti süresine katlanmaya hazır değilseniz, Nmap asla kritik görev sistemlerine karşı çalıştırılmamalıdır. Burada Nmap'in sistemlerinizi veya ağlarınızı çökertebileceğini kabul ediyoruz ve Nmap'in neden olabileceği herhangi bir hasar veya sorun için tüm sorumluluğu reddediyoruz.

Uygunsuz Kullanım

Hafif çökme riski ve bazı siyah şapkalıların sistemlere saldırmadan önce keşif için Nmap kullanmayı sevmesi nedeniyle, sistemleri tarandığında üzülen ve şikayet edebilen yöneticiler vardır. Bu nedenle, bir ağun hafif bir taramasını bile yapmadan önce izin istemek genellikle tavsiye edilir.

Nmap asla özel ayrıcalıklarla (örn. suid root) kurulmamalıdır. Bu, sistemdeki diğer kullanıcılar (veya saldırganlar) ayrıcalık yükseltmek için kullanabileceğinden büyük bir güvenlik açığına yol açacaktır.

Nmap, yazılımın arızalanmasının doğrudan ölüme, kişisel yaralanmaya veya önemli fiziksel ya da çevresel hasara yol açabileceği, güvenli performans gerektiren tehlikeli ortamlarda kullanılmak üzere tasarlanmamış, üretilmemiş veya tasarlanmamıştır.

Üçüncü Taraf Yazılım ve Finansman Bildirimleri

Bu ürün Apache Software Foundation tarafından geliştirilen yazılımları içerir. Libpcap taşınabilir paket yakalama kütüphanesinin değiştirilmiş bir sürümü Nmap ile birlikte dağıtılır. Nmap'in Windows sürümü bunun yerine Libpcap türevi Ncap kütüphanesini kullanır. Düzenli ifade desteği, Philip Hazel tarafından yazılan açık kaynaklı bir yazılım olan PCRE kütüphanesi tarafından sağlanmaktadır. Bazı ham ağ işlevleri, Dug Song tarafından yazılan Libdnet ağ kütüphanesini kullanır. Değiştirilmiş bir sürümü Nmap ile birlikte dağıtılmaktadır. Nmap isteğe bağlı olarak SSL sürüm algılama desteği için OpenSSL kriptografi araç seti ile bağlantı kurabilir.

Nmap Scripting Engine, Lua programlama dilinin gömülü bir versiyonunu kullanır. Liblinear doğrusal sınıflandırma kütüphanesi IPv6 işletim sistemi algılama makine öğrenme tekniklerimiz için kullanılmaktadır ("IPv6 eşleştirme" adlı bölüme bakınız). Bu paragrafta açıklanan tüm üçüncü taraf yazılımlar BSD tarzı yazılım lisansları altında serbestçe yeniden dağıtılabılır.

Windows ve Mac OS X için ikili paketler, Zenmap ve Ndiff'i Python ve PyGTK ile çalıştırmak için gerekli destek kütüphanelerini içerir. (Unix platformları genellikle bu kütüphanelerin kurulumunu kolaylaştırır, bu nedenle paketlerin bir parçası değildirler). Bu destek kütüphanelerinin ve lisanslarının bir listesi LICENSES dosyalarında yer almaktadır.

Bu yazılım kısmen Google Summer of Code ve DARPA CINDER programı (DARPA-BAA-10-84) aracılığıyla desteklenmiştir.

Amerika Birleşik Devletleri İhracat Kontrolü

Nmap yalnızca istege bağlı OpenSSL desteği ile derlendiğinde ve OpenSSL ile bağlandığında şifreleme kullanır. OpenSSL desteği olmadan derlendiğinde, Nmap Projesi Nmap'in ABD İhracat İdaresi Düzenlemeleri (EAR) ihracat kontrolüne tabi olmadığına inanmaktadır. Bu nedenle, geçerli bir ECCN (ihracat kontrol sınıflandırma numarası) yoktur ve ihracat herhangi bir özel lisans, izin veya başka bir hükümet yetkisi gerektirmez.

OpenSSL desteği ile derlendiğinde veya kaynak kodu olarak dağıtıldığında, Nmap Projesi Nmap'in ABD ECCN 5D002 ("Bilgi Güvenliği Yazılımı") kapsamına girdiğine inanmaktadır. Nmap'i EAR 740.13(e)'de tanımlanan kamuya açık şifreleme yazılımı için TSU istisnası kapsamında dağıtıyoruz.

Next Next Next