Zmap: Hack The Planet!

Baskı: 29 Haziran 2017

Ozan Uçar - İbrahim UÇAR

İÇİNDEKİLER

Yazı Hakkında

Zmap Nedir?

Başarılı Port Tarama için Önerileriler

Kurulum Seçenekleri

- Kaynak Koddan Kurulum
- Zmap Kurulumu ve Derleme
- Zmap Parametreleri

Örneklerle Zmap Kullanımı

Sonuçlar Çıktısı

Çıktı Alanları

Yazı Hakkında

Bu yazı, Ozan UÇAR ve İbrahim UÇAR tarafından hazırlanmıştır ve kaynak göstermek şartıyla yazarlardan izin alınmaksızın paylaşılabilir, kullanılabilir. Katkıda bulunmak için yazarlar ile iletişime geçebilirsiniz.

İnternet dünyasındaki aktif ip adreslerini ve zafiyetleri adreslemek için açık portlarını tespit etmek için başlatılan bir projede, aktif tarama aracı olarak Zmap[1] tercih edilmiştir. Yazının devamında Zmap projesi, kurulum ve konfigürasyon seçenekleri gerçek dünyadan kullanım örnekleri ile yer alacaktır.

Deneysel olarak, ZMap'in tüm dünyadaki IPv4 ağlarını taramada, en agresif Nmap varsayılan ayarlarından 1300 kat daha hızlı olduğu görülmektedir.[2]

Bu dökümanda yer alan detaylar ve örnek kullanım seçenekleri, büyük ağlarda çalışan sızma testi uzmanlarının, güvenlik araştırmacılarının ve akademisyenlerin de işlerini kolaylaştıracaktır.

[1] https://zmap.io/
[2] https://zmap.io/paper.pdf

Zmap Nedir?

ZMap projesi, araştırmacıların halka açık internet'leri oluşturan servislerle ilgili geniş çaplı çalışmalar yapmalarını sağlayan, açık kaynak kodlu araçlardan oluşan bir koleksiyondur. ZMap projesi içerisinde açık kaynak kodlu 12 tane araç bulunmaktadır. ZMap aracı diğer araçlardan sadece bir tanesidir ve bu dökümanda ZMap aracının nasıl kurulduğuna ve kullanımını detayları ile birlikte sizlere anlatıyor olacağız. Güncellenecek yeni versiyonlarda diğer araçlarıve Zmap ile ilişkilendirilmiş örneklerine yer vermeye çalışacağız.

ZMap, internet çapında ağ haritası için tasarlanmış hızlı tek paketli açık kaynak kodlu bir ağ tarayıcısı aracıdır. Gigabit bağlantılı bir bilgisayarda ZMap, internet dünyasındaki tüm IPv4 adresleri 45 dakika içinde tarayabilir! 10gbit bağlantılı ve PF_RING ile beraber çalışan ZMap, internet dünyasının tamamını yaklaşık 5 dakika içinde tarayabilir.

ZMap aracı GNU/Linux, Mac OS ve BSD tabanlı sistemlerde başarılı bir şekilde çalışabilmektedir.

ZMap projesinin web sitesi : zmap.io

Başarılı Port Tarama için Önerileriler

İnternette iyi bir birey olmak ve yaptığınız çalışmalar ile farkında olmadan ağlara zarar vermemeniz için aşağıdaki önerileri dikkate almanızı isteriz;

- Kurumsal ağlarda çalışma yapacaksanız, olası darboğazlar ve ağda olası sorunlara karşı ağ yöneticisi ile yakın çalışmaya özen gösterin.
- Taramanın amacını ve kapsamını açıkca belirtin, hedef odaklı tarama seçenekleri oluşturun.
- Taramalarınızın, amaçladığınız hedeflerinizden ve araştırma konunuzdan uzaklaşmamasına ve amacını aşmamasına dikkat edin.
- İzinsiz taramaların bazı hukuki sorunlar yaratabileceğine göz önünde bulundurarak, kurumsal ağlar için izin talep etmeyi düşünebilirsiniz.
- Başarılı bir tarama için, paket kaybını en aza indirecek ve hedef sistemi gereksiz yormayacak tarama seçenekleri geliştirin;
    - tarama süresini farklı kaynaklara yayma
    - taramanın zamanını geniş tutma
    - eş zamanlı paket gönderimini azaltmak gibi gibi

Kurulum Seçenekleri

Zmap aracının kurulumu gnu/linux tabanlı sistemlerde paket deposundan hızlı ve kolay bir şekilde kurulabilmektedir. Aşağıdaki adımları takip ederek kullandığınız dağıtıma uygun yönergelerle kolaylıkla kurulumu gerçekleştirebilirsiniz.

Debian 8+ ve Ubuntu 14.04+ için;

```
# sudo apt-get install zmap
```

Fedora, CentOS, ve RHEL için;

```
# sudo yum install zmap
```

Gentoo için;

```
# sudo emerge zmap
```

Mac OS (brew) için;

```
# brew install zmap
```

Arch Linux için;

```
# pacman -S zmap
```

Kaynak Koddan Kurulum

Zmap aracını aynı zamanda kaynak koddan kurmak mümkündür. Aşağıdaki adımları takip ederek zmap aracını kaynak koddan kurabilirsiniz.

Zmap aracının kurulması için gereken aşağıdaki gerekli bağımlılıkları indirmeniz gerekmektedir.

- CMake - Cross-platform, open-source build system
- GMP - Free library for arbitrary precision arithmetic
- gengetopt - Command line option parsing for C programs
- libpcap - Famous user-level packet capture library
- flex and byacc - Output filter lexer and parser generator.
- json-c - JSON implementation in C
- libunistring - Unicode string library for C
- libdnet - (Mac Only) Gateway and route detection.

Aşağıda size uygun olan komutları kullanarak bu bağımlılık paketlerini indirebilirsiniz. Bu sadece kaynak koddan kurulum için gerekli olan bağımlılıklardır, paket deposundan kurulumda bu paketler otomatik olarak kurulmaktadır.

Debian tabanlı sistemler için;

```
# sudo apt-get install build-essential cmake libgmp3-dev gengetopt libpcap-dev flex byacc libjson-c-dev pkg-config libunistring-dev
```

RHEL ve Fedora tabanlı sistemler için;

```
# sudo yum install cmake gmp-devel gengetopt libpcap-devel flex byacc json-c-devel libunistring-devel
```

Mac OS sistemler için; (Homebrew)

```
# brew install pkg-config cmake gmp gengetopt json-c byacc libdnet libunistring
```

Zmap Kurulumu ve Derleme

Ön koşullar sisteme yüklendikten sonra, ZMap aşağıdaki komutu çalıştırarak derlenebilir. Bu komutu ZMap dizinin içerisinde çalıştırınız.

```
directory_zmap# cmake .
directory_zmap# make -j4
```

Hemen arından aşağıdaki komut yardımıyla kurulum yapılır.

```
directory_zmap# sudo make install
```

Zmap Parametreleri

Zmap uygulamasını etkin kullanmak için parametrelerine hakim olmamız gerekiyor. Bu parametreler etkili port tarama çalışmaları yapmak için sihirli kelimeleriniz olacaktır.

Temel Parametreler	Açıklama	Örnek
-p, --target-port=port	Taranacak port numarası (tcp ve udp için)	-p 80
-o, --output-file=name	Çıktıyı aktarmak için.	-o output.txt
-b, --blacklist-file=path	CIDR biçimde hariç tutulan alt ağları belirtilebilir. (örn:192.168.0.0/16)	-b blacklist-file.txt
-w, --whitelist-file=path	CIDR biçimde taramayı sınırlandırmak için alt ağların bulunduğu dosya. (örn:192.168.0.0/16)	-w whitelist-file.txt

Tarama Parametreleri	Açıklama	Örnek
-r, --rate=pps	Saniyede gönderilecek paket sayısı.	zmap -r 5
-B, --bandwidth=pps	Saniye/Bit bazında kullanılacak bant genişliği. (G, M ve K bazında destekliyor)	zmap -B 100M
-t, --max-runtime=ses	Gönderilecek paketlerin süre uzunluğunu kısaltın. (saniye bazında)	zmap -t 60
-N, --max-results=n	Geri dönen sonuçların sayısını ayarlamak.	zmap -N 5
-P, --probes=n	Her IP adresi için gönderilecek probe sayısı	-P

-c, --cooldown-time=secs	Bir paket gönderildikten sonra ne kadar süre beklenecek cevap gelmesi için. (default=`8')	-c
--retries=n	Eğer paket başarısız olursa, maksimum deneme sayısı. (varsayılan=’10’)	zmap --retries=3
-d, --dryrun	Paketleri yollama!	zmap -d
--shards=N	Kaç tane ZMap süreci başlatılacak. (default=`1')	
--shards=n	Hangi Zmap sürecinden başlatılacak. (default=’0’)	

Ağ Seçenekleri	**Açıklama**	**Örnek**
-s, --source-port=port|range	Taranacak kaynak port numaraları.	-s 80
-S, --source-ip=ip|range	Taranacak kaynak ip adresleri.	-S 192.168.0.0/16
-G, --gateway-mac=addr	Ağ geçidi adresi belirtmek.	-G 192.168.1.1
--source-mac=addr	Kaynak mac adresi belirtmek.	--source-mac=00:22:34:51:22:33
-i, --interface=name	Kullanılmak üzere ağ arayüzü belirtmek.	-i eth0
-X, --vpn	IP paketlerini ağ arayüzünden yollamak yerine VPN ‘den yolla. (VPNs)	-X vpn_ip

Prob Modülleri	Açıklama	Örnek
-M, --probe-module=name	Farklı bir modül belirtiniz. (varsayılan='tcp_synscan')	zmap -M icmp_echoscan
--probe-args=args	Probe modülünde geçilecek argümanlar.	
--list-probe-modules	Mevcut kullanılabilir modülleri listelemek.	zmap --list-probe-modules

Çıktı Seçenekleri	Açıklama	Örnek
-f, --output-fields=fields	Çıktıdaki oluşacak alanları belirlemek ve ona göre çıktı oluşturmak.	zmap --output-fields=saddr,sport
O, --output-module=name	Çıktıyı aktarmak istediğiniz formatı belirtiniz.	zmap --output-module=json
--output-args=args	Çıktı modülünde geçilecek argümanlar.	
--output-filter=filter	Çıktı modülüne hangi yanıtların gönderileceğini sınırlamak için yanıt alanlarının üzerine bir filtre belirleyin.	zmap --output-filter="success = 1"
--list-output-modules	Mevcut modülleri listelemek.	zmap --list-output-modules
--list-output-fields	Seçilen modül tarafından çıktı alanlarını filtrelemek için tüm mevcut alanları listele.	zmap --list-output-fields

Loglama ve Metadata	Açıklama	Örnek

-v, --verbosity=n	Log çıktı detaylarının seviyesi. (0-5) - (varsayılan='3')	zmap -v 2
-l, --log-file=name	Tüm logları hedef dosyaya yazdır.	zmap -l logfile.txt
-L, --log-directory=directory	Günlük girişlerini bu dizindeki zaman damgalı bir dosyaya yazın.	zmap -L kayitdizini/
-m, --metadata-file=name	Tarama meta verileri için çıktı dosyası.	
-u, --status-updates-file=name	Tarama ilerleme güncellemelerini CVS dosyasına yazın.	zmap -u dosya_ismi
-q, --quiet	Güncelleme durumlarını ekrana basma.	zmap -q
--disable-syslog	Günlük mesajlarını syslog'a devre dışı bırakır.	zmap --disable-syslog

Ek Seçenekler	Açıklama	Örnek
-C, --config=filename	Bu seçeneklerden herhangi birini belirleyebilen bir yapılandırma dosyasını okuyun. (varsayılan=`/etc/zmap/zmap.conf')	zmap -C konfigurasyon.conf
--max-sendto-failures=n	Taramadan önce azami NIC gönderme hatalarına son vermek. (varsayılan='-1')	zmap --max-sendto-failures=10
--min-hitrate=n	Taramanın durdurulmadan önce taramanın vurabileceği en düşük çarpma oranı. (varsayılan='0.0')	zmap --min-hitrate=3
-T, --sender-threads=n	Paket gönderirken kullanılan iş parçacıkları. (varsayılan='1')	
-cores=STRING	Bağlanacak çekirdeklerin virgülle ayrılmış listesi.	
--ignore-invalid-hosts	Whitelist/Blacklist dosyası içerisinde geçersiz kullanıcıları görmezden gel.	
-h, --help	Yardım sayfasını göster ve çık.	zmap -h
-V, --version	Versiyon bilgisini göster ve çık.	zmap -V

Örneklerle Zmap Kullanımı

10Mb/s hız ile 5 HTTP sunucu bul.

```
# zmap -N 5 -B 10M -p 80 -i eth0
```

TCP/80 portu için belirli ağ aralıklarını tara ve çıktıyı **sonuçlar.txt** dosyasına yaz.

```
# zmap -p 80 10.0.0.0/8 192.168.0.0/16 -i eth0 -o sonuçlar.txt
```

TCP/80 portu için 1.2.3.4 10.0.0.3 ip adreslerini tara.

```
# zmap -p 80 1.2.3.4 10.0.0.3 -i eth0
```

Eğer tarama başarılı bir şekilde başladı ise ZMap aracı her 1 dk aralıklarla ekrana aşağıdaki gibi çıtkı basacaktır..

```
0% (1h51m left); send: 28777 562 Kp/s (560 Kp/s avg); recv: 1192 248 p/s (231 p/s avg); hits: 0.04%
0% (1h51m left); send: 34320 554 Kp/s (559 Kp/s avg); recv: 1442 249 p/s (234 p/s avg); hits: 0.04%
0% (1h50m left); send: 39676 535 Kp/s (555 Kp/s avg); recv: 1663 220 p/s (232 p/s avg); hits: 0.04%
0% (1h50m left); send: 45372 570 Kp/s (557 Kp/s avg); recv: 1890 226 p/s (232 p/s avg); hits: 0.04%
```

Bir TCP portu için tüm dünyayı taramak. Aşağıdaki tarama seçenekleri ile Zmap tüm dünyadaki tcp/25 portlarını 700mbit hat ile yaklaşık 79 dakikada taramıştır.

```
# time zmap  -p 25 -B 700M -i eth0 --output-module=json --output-fields=saddr,sport --output-filter="success = 1" -o world25.json

Sonucun çıktısı.
real    78m25.429s
user    33m52.175s
sys     95m41.969s
```

TCP/80 portu için maksimum 10 Mbps hız ile, 10,000 tane rastgele ip adresi tara ve çıktıyı **sonuclar.cs** dosyasına yaz.

```
# zmap --bandwidth=10M --target-port=80 --max-targets=10000 --output-file=sonuclar.cs
```

Veya yukarıdaki komut kısaca şöyle belirtilebilir.

```
# zmap -B 10M -p 80 -n 10000 -o sonuclar.csv
```

ZMap ayrıca taramanın sonucunda sizlere özet yazdırabilme özelliğini de desteklemektedir. Bu özelliği kullanabilmek için **--summary** parametresi kullanılabilir.

```
# zmap --summary

cnf      target-port                 443
cnf      source-port-range-begin     32768
cnf      source-port-range-end       61000
cnf      source-addr-range-begin     1.1.1.4
cnf      source-addr-range-end       1.1.1.8
cnf      maximum-packets             4294967295
cnf      maximum-runtime             0
```

cnf	permutation-seed	0
cnf	cooldown-period	300
cnf	send-interface	eth1
cnf	rate	45000
env	nprocessors	16
exc	send-start-time	Fri Jan 18 01:47:35 2013
exc	send-end-time	Sat Jan 19 00:47:07 2013
exc	recv-start-time	Fri Jan 18 01:47:35 2013
exc	recv-end-time	Sat Jan 19 00:52:07 2013
exc	sent	3722335150
exc	blacklisted	572632145
exc	first-scanned	1318129262
exc	hit-rate	0.874102
exc	synack-received-unique	32537000
exc	synack-received-total	36689941

Sonuçlar Çıktısı

ZMap, modülleri sayesinde çeşitli biçimlerde sonuçlar üretebilmektedir. Varsayılan olarak, ZMap yalnızca CSV çıktısı üretmektedir, ancak redis ve json formatlarınıda destekler. Bu modüllerin ürettiği ve gönderilen çıktılar filtrelenebilir ve bize özel istediğimiz alanları istediğimiz gibi çıktıya aktarabiliriz.

Modüllerin çıktıları kullanıcı tarafından belirtilmelidir. Varsayılan olarak, ZMap aracı sonuçları CSV formatına döndürür ve çıktı dosyası belirtilmezse, ZMap sonuçları ekrana basar ve aynı zamanda belirli sonuçları üretmez.

Çıktı Seçenekleri	**Açıklama**	**Örnek**
-f, --output-fields=fields	Çıktıdaki oluşacak alanları belirlemek ve ona göre çıktı oluşturmak.	zmap --output-fields=”saddr,sport”
-O, --output-module=name	Çıktıyı aktarmak istediğiniz formatı belirtiniz.	zmap --output-module=json
--output-args=args	Arguments to pass to output module.	
--output-filter=filter	Çıktı modülüne hangi yanıtların gönderileceğini sınırlamak için yanıt alanlarının üzerine bir filtre belirleyin.	zmap --output-filter="success = 1"
--list-output-modules	Mevcut modülleri listelemek.	zmap --list-output-modules
--list-output-fields	Seçilen modül tarafından çıktı alanlarını filtrelemek için tüm mevcut alanları listele.	zmap --list-output-fields
-o, --output-file=p	Üretilen çıktıların yazılacağı dosya.	zmap -o output.csv

Çıktı Alanları

ZMap, IP adresinin ötesinde çıktılayabileceği çeşitli alanlara sahiptir. Bu alanlar **--list-output-fields** parametresi kullanarak görüntülenebilir. Özellikle bir modüle ait çıktı alanları listelemek isterseniz aşağıdaki gibi bir komut kullanabilirsiniz ve o modüle ait üretilen çıktıları filtrelemek için kullanılabilir alanları görebilirsiniz.

```
# zmap --probe-module="tcp_synscan" --list-output-fields

saddr          string: source IP address of response
saddr-raw      int: network order integer form of source IP address
daddr          string: destination IP address of response
daddr-raw      int: network order integer form of destination IP address
ipid           int: IP identification number of response
ttl            int: time-to-live of response packet  ort
dport          int: TCP destination port
seqnum         int: TCP sequence number
acknum         int: TCP acknowledgement number
window         int: TCP window
classification  string: packet classification
success        int: is response considered success
repeat         int: is response a rep  eat response from host
cooldown       int: Was response received during the cooldown period
timestamp-str   string: timestamp of when response arrived in ISO8601 format.
timestamp-ts    int: timestamp of when response arrived in seconds since Epoch
timestamp-us    int: microsecond part of timestamp (e.g. microseconds since 'timestamp-ts')
```

Filtrelemek istediğiniz alanları virgül ile ayırarak **--output-field=fields** veya **-f** parametresine verebilirsiniz.

Örneğin, tüm dünyayı TCP/25 portu için tara ve sadece **"saddr,sport"** çıktısını **world25.json** adındaki dosyaya yaz.

```
# zmap  -p 25 -B 700M -i ens192 --output-module=json --output-fields=saddr,sport --output-filter="success = 1" -o world25.json
```

Çıktı şu şekilde görünecektir.

```
# more world25.json
{ "saddr": "89.147.254.133", "sport": 25 }
{ "saddr": "31.220.21.135", "sport": 25 }
{ "saddr": "5.9.108.207", "sport": 25 }
{ "saddr": "77.55.31.219", "sport": 25 }
```

Bir diğer örnek ise çıktıda sadece başarılı yanıtları filtrelemek isterseniz aşağıdaki gibi bir parametre kullanabilirsiniz.

```
# zmap -p 21  -o sonuçlar.csv  --output-filter="success = 1"
```

BGA Bilgi Güvenliği A.Ş. Hakkında

BGA Bilgi Güvenliği A.Ş. 2008 yılından bu yana siber güvenlik alanında faaliyet göstermektedir. Ülkemizdeki bilgi güvenliği sektörüne profesyonel anlamda destek olmak amacı ile kurulan BGA Bilgi Güvenliği, stratejik siber güvenlik danışmanlığı ve güvenlik eğitimleri konularında kurumlara hizmet vermektedir.

Uluslararası geçerliliğe sahip sertifikalı 50 kişilik teknik ekibi ile, faaliyetlerini Ankara ve İstanbul ve USA'da sürdüren BGA Bilgi Güvenliği'nin ilgi alanlarını "*Sızma Testleri, Güvenlik Denetimi, SOME, SOC Danışmanlığı, Açık Kaynak Siber Güvenlik Çözümleri, Büyük Veri Güvenlik Analizi ve Yeni Nesil Güvenlik Çözümleri*" oluşturmaktadır.

Gerçekleştirdiği başarılı danışmanlık projeleri ve eğitimlerle sektörde saygın bir yer edinen BGA Bilgi Güvenliği, kurulduğu günden bugüne alanında lider finans, enerji, telekom ve kamu kuruluşlarına 1.000'den fazla eğitim ve danışmanlık projeleri gerçekleştirmiştir.

BGA Bilgi Güvenliği, kurulduğu 2008 yılından beri ülkemizde bilgi güvenliği konusundaki bilgi ve paylaşımların artması amacı ile güvenlik e-posta listeleri oluşturulması, seminerler, güvenlik etkinlikleri düzenlenmesi, üniversite öğrencilerine kariyer ve bilgi sağlamak için siber güvenlik kampları düzenlenmesi ve sosyal sorumluluk projeleri gibi birçok konuda gönüllü faaliyetlerde bulunmuştur.

BGA Bilgi Güvenliği AKADEMİSİ Hakkında

BGA Bilgi Güvenliği A.Ş.'nin eğitim ve sosyal sorumluluk markası olarak çalışan Bilgi Güvenliği AKADEMİSİ, siber güvenlik konusunda ticari, gönüllü eğitimlerin düzenlenmesi ve siber güvenlik farkındalığını arttırıcı gönüllü faaliyetleri yürütülmesinden sorumludur. Bilgi Güvenliği AKADEMİSİ markasıyla bugüne kadar "Siber Güvenlik Kampları", "Siber Güvenlik Staj Okulu", "Siber Güvenlik Ar-Ge Destek Bursu" , "Ethical Hacking yarışmaları" ve "Siber Güvenlik Kütüphanesi" gibi birçok gönüllü faaliyetin destekleyici olmuştur.