Uçtan Uca Haberleşmede Güvenlik: Tehditler ve Savunma Mekanizmaları

Giriş

◦Uçtan Uca Haberleşme Nedir?

◦ Uçtan uca haberleşme (End-to-End Encryption - E2EE), iletişimdeki verilerin sadece **gönderen** ve **alıcı** tarafında şifrelenip, yalnızca bu iki taraf tarafından çözülebildiği bir güvenlik modelidir.

◦ Aradaki sunucular veya aracılar, verinin içeriğine erişemez.

◦ Bu sayede, araya sızan herhangi biri (örneğin bir hacker, servis sağlayıcı ya da devlet kurumları) veriyi okuyamaz veya değiştiremez.

◦ Örnek uygulamalar: WhatsApp, Signal, ProtonMail gibi gizliliği ön planda tutan mesajlaşma ve e-posta servisleri.

Neden Güvenlik Bu Kadar Önemli?

- Günümüzde veri iletişimi sadece basit mesajlaşma değil, aynı zamanda **finansal işlemler**, **kişisel sağlık bilgileri**, **kurumsal sırlar** ve **kimlik bilgileri** içerir.
- Güvenliği sağlanmayan iletişim kanalları, bu bilgilerin kötü niyetli kişiler tarafından ele geçirilmesine, değiştirilmesine ya da suistimal edilmesine neden olabilir.
- Özellikle kamuya açık Wi-Fi ağları veya yetersiz güvenlikli ağlar, veri dinleme ve değiştirme saldırılarına karşı savunmasızdır.

Bu Sunumda Neler Göreceğiz?

- **Tehditler:** Man-in-the-Middle (MITM) saldırıları, veri kopyalama, injection gibi uçtan uca haberleşmeyi tehlikeye atan saldırı türleri.
- **Savunma Mekanizmaları:** Şifreleme teknikleri, kimlik doğrulama, dijital imza ve ağ güvenliği çözümleri.
- **Güncel Durum ve Örnekler:** Modern protokoller, kullanılan standartlar ve IoT gibi yeni alanlarda ortaya çıkan zorluklar.
- **Proje ve Uygulama Önerileri:** Konu üzerine geliştirilebilecek güvenlik testleri ve uygulamalar.
- 📌 ***Ek Not:***
- ***Bu sunumla amaç, sadece teknik bilgi vermek değil, aynı zamanda veri güvenliğinin neden kritik olduğunu ve bu güvenliği sağlamak için hangi yöntemlerin kullanıldığını anlamanızı sağlamaktır***

Tehdit Türleri

- **1. Man-in-the-Middle (MITM)**
- İki taraf arasındaki iletişime gizlice girerek veri okuma veya değiştirme saldırısı.
- **2. Paket Dinleme (Sniffing)**
- Ağ üzerindeki verilerin pasif olarak yakalanıp analiz edilmesi.
- **3. Replay Attack**
- Önceden yakalanmış geçerli mesajların tekrar gönderilerek yetkilendirme atlatma.
- **4. Injection Saldırıları**
- Veri giriş noktalarına zararlı kod enjekte ederek sistem kontrolü sağlama.
- **5. Spoofing (Kimlik Taklidi)**
- Başka bir cihazın kimlik bilgilerini taklit ederek erişim sağlama.

Man-in-the-Middle (MITM) Saldırısı

- **Tanım:**
  İki iletişim noktası (örneğin kullanıcı ve sunucu) arasında saldırganın gizlice yer alarak iletişimi dinlemesi veya değiştirmesi.
- **Nasıl Çalışır?**
  Saldırgan, iletişimin geçtiği ağda araya girer ve veri paketlerini yakalar. Daha sonra bu paketleri değiştirebilir veya olduğu gibi alıcıya iletebilir. Bu sayede her iki taraf, karşı tarafla doğrudan iletişim kurduğunu sanır ama aslında saldırgan aradadır.
- **Yaygın Teknikler:**
- **ARP Spoofing:** Yerel ağda IP adreslerini kendi MAC adresine yönlendirerek trafiği yakalama.
- **DNS Spoofing:** DNS sorgularını değiştirerek kullanıcıyı sahte bir web sitesine yönlendirme.
- **Wi-Fi Evil Twin:** Sahte Wi-Fi erişim noktası oluşturup kullanıcıların bağlanmasını sağlama.

- **Riskleri:**
- Şifreli olmayan veri ele geçirilebilir.
- Banka bilgileri, kişisel mesajlar, kimlik bilgilerinin çalınması.
- Veri değiştirilerek dolandırıcılık yapılabilir.
- **Korunma Yöntemleri:**
- Uçtan uca şifreleme (E2EE) kullanmak.
- Sertifikaların doğruluğunu kontrol etmek.
- VPN kullanmak.
- Güvenilir ağlara bağlanmak.

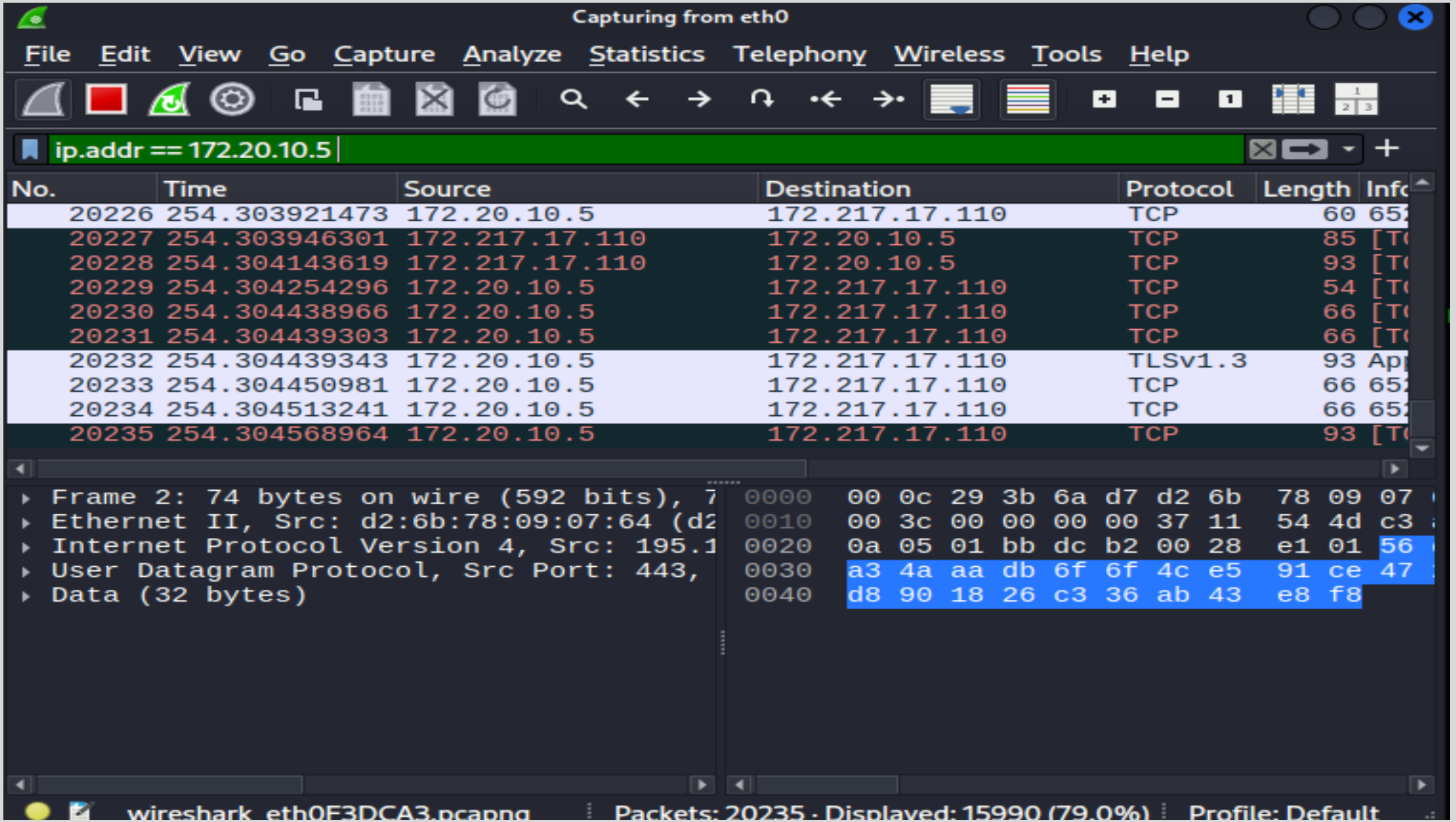
```
172.64.146.98, 104.18.41.158
172.20.10.0/28 > 172.20.10.7  » [09:00:23] [net.sniff.dns] dns gateway > 172.20.10.5 : cdn.oaistatic.com is
172.64.146.98, 104.18.41.158
172.20.10.0/28 > 172.20.10.7  » [09:00:28] [net.sniff.dns] dns gateway > 172.20.10.5 : a.nel.cloudflare.com
is 35.190.80.1
172.20.10.0/28 > 172.20.10.7  » [09:00:28] [net.sniff.dns] dns gateway > 172.20.10.5 : a.nel.cloudflare.com
is 35.190.80.1
172.20.10.0/28 > 172.20.10.7  » [09:00:28] [net.sniff.dns] dns gateway > 172.20.10.5 : a1894.dscb.akamai.net
 is 195.175.180.249, 195.175.180.248, 195.175.181.42, 195.175.180.242, 195.175.180.216
172.20.10.0/28 > 172.20.10.7  » [09:00:28] [net.sniff.dns] dns gateway > 172.20.10.5 : a1894.dscb.akamai.net
 is 195.175.180.249, 195.175.180.248, 195.175.181.42, 195.175.180.242, 195.175.180.216
172.20.10.0/28 > 172.20.10.7  » [09:00:44] [net.sniff.dns] dns gateway > 172.20.10.5 : chat.cdn.whatsapp.net
 is 185.60.218.54
172.20.10.0/28 > 172.20.10.7  » [09:00:44] [net.sniff.dns] dns gateway > 172.20.10.5 : chat.cdn.whatsapp.net
 is 185.60.218.54
172.20.10.0/28 > 172.20.10.7  » [09:00:51] [net.sniff.dns] dns gateway > 172.20.10.5 : browser-intake-datado
ghq.com is 3.233.158.26, 3.233.158.24, 3.233.158.25
172.20.10.0/28 > 172.20.10.7  » [09:00:51] [net.sniff.dns] dns gateway > 172.20.10.5 : browser-intake-datado
ghq.com is 3.233.158.26, 3.233.158.24, 3.233.158.25
172.20.10.0/28 > 172.20.10.7  » [09:01:03] [net.sniff.dns] dns gateway > 172.20.10.5 : beacons-handoff.gcp.g
vt2.com is 142.251.37.227
172.20.10.0/28 > 172.20.10.7  » [09:01:03] [net.sniff.dns] dns gateway > 172.20.10.5 : beacons-handoff.gcp.g
vt2.com is 142.251.37.227
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns fe80::9cea:ff9a:4bd6:4f5a : Unknown query f
or Burki._dosvc._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns 172.20.10.5 : Unknown query for Burki._dosv
c._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns 172.20.10.5 : Unknown query for Burki._dosv
c._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns fe80::9cea:ff9a:4bd6:4f5a : Unknown query f
or Burki._dosvc._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns 172.20.10.5 : Unknown query for Burki._dosv
c._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:04] [net.sniff.mdns] mdns fe80::9cea:ff9a:4bd6:4f5a : Unknown query f
or Burki._dosvc._tcp.local
172.20.10.0/28 > 172.20.10.7  » [09:01:05] [net.sniff.mdns] mdns Burki.local. : Burki.local is 172.20.10.5,
fe80::9cea:ff9a:4bd6:4f5a
172.20.10.0/28 > 172.20.10.7  » [09:01:05] [net.sniff.mdns] mdns fe80::9cea:ff9a:4bd6:4f5a : Burki.local is
172.20.10.5, fe80::9cea:ff9a:4bd6:4f5a
172.20.10.0/28 > 172.20.10.7  » [09:01:05] [net.sniff.mdns] mdns Burki.local. : Burki.local is 172.20.10.5,
fe80::9cea:ff9a:4bd6:4f5a
172.20.10.0/28 > 172.20.10.7  » [09:01:05] [net.sniff.mdns] mdns fe80::9cea:ff9a:4bd6:4f5a : Burki.local is
172.20.10.5, fe80::9cea:ff9a:4bd6:4f5a
172.20.10.0/28 > 172.20.10.7  » [09:01:06] [net.sniff.dns] dns gateway > Burki.local. : e3.whatsapp.net is 3
.33.252.61
172.20.10.0/28 > 172.20.10.7  » [09:01:06] [net.sniff.dns] dns gateway > Burki.local. : e3.whatsapp.net is 3
.33.252.61
[09:01:13] [net.sniff.dns] dns gateway > Burki.local. : wac-0003.wac-msedge.net is 52.108.8.12, 52.108.9.12
172.20.10.0/28 > 172.20.10.7  » [09:01:13] [net.sniff.dns] dns gateway > Burki.local. : wac-0003.wac-msedge.net is 52.108.8.12, 52.108.9.12
172.20.10.0/28 > 172.20.10.7  »
```

- ***"MITM Saldırısı Sonuçları – DNS Trafiğinin Ele Geçirilmesi***
- *Bu ekran, Bettercap aracı ile yapılan bir ARP spoofing tabanlı Man-in-the-Middle (MITM) saldırısı sırasında, hedef cihazın (IP: 172.20.10.5, hostname: Burki.local) DNS sorgularının benim sanal makinem üzerinden geçtiğini göstermektedir.*
- **cdn.oaistatic.com**
- OpenAI statik içerik sunucusu
- **chat.cdn.whatsapp.net**
- WhatsApp mesaj sunucusu
- **a.nel.cloudflare.com**
- Cloudflare ağ izleme servisi
- **browser-intake-datadoghq.com**
- Veri izleme / loglama servisi (Datadog)
- **e3.whatsapp.net**
- WhatsApp ses/video trafiği sunucusu
- **Burki._dosvc._tcp.local (MDNS)**
- Windows Update servis keşfi (yerel ağ içi)

- **1. Servis Tespiti:**
- ARP spoofing + DNS sniffing sayesinde hedef cihazın:
- Kullandığı uygulamalar (WhatsApp, Edge, OpenAI)
- Sistem servisleri (Broadcom update, Datadog, Akamai CDN)
- Hangi altyapılarla (Cloudflare, Akamai, Google, Meta) iletişim kurduğu
  tespit edilir.
- Bu bilgiler, hedef sistemin altyapı bağımlılıklarını ve kullanıcı davranışlarını anlamamıza yardımcı olur.
- **2. Kullanıcı Aktivitesini İzleme:**
- DNS sorguları üzerinden hedefim:
  - Hangi siteleri açtığı
  - Hangi uygulamaların arka planda çalıştığı
  - Ne zaman aktif olduğu
    gibi davranışları analiz edilebilir.
  - Örnek: chat.cdn.whatsapp.net ⇒ WhatsApp açıldı
    cdn.oaistatic.com ⇒ ChatGPT gibi servisler kullanılıyor

- **3.HTTPS'in Kullanılmadığı Durumlarda Risk Katlanır:**
- HTTPS **şifreli veri iletimi** sağlar. MITM saldırganı içeriği göremez.
- Ancak **HTTP kullanan sitelerde**:
  - Girilen formlar, kullanıcı adı/şifreler
  - URL parametreleri
  - Cookie bilgileri
    açık metin olarak okunabilir.

- **Bu tip bir MITM saldırısı sayesinde:**
- Ağ trafiği izlenebilir
- Kimlik bilgileri çalınabilir
- Kullanıcı yanlış yönlendirilebilir
- Güvenlik açıkları tespit edilebilir


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 172.20.10.5
No. Time Source Destination Protocol Length Info
20226 254.303921473 172.20.10.5 172.217.17.110 TCP 60 65
20227 254.303946301 172.217.17.110 172.20.10.5 TCP 85 [T
20228 254.304143619 172.217.17.110 172.20.10.5 TCP 93 [T
20229 254.304254296 172.20.10.5 172.217.17.110 TCP 54 [T
20230 254.304438966 172.20.10.5 172.217.17.110 TCP 66 [T
20231 254.304439303 172.20.10.5 172.217.17.110 TCP 66 [T
20232 254.304439343 172.20.10.5 172.217.17.110 TLSv1.3 93 App
20233 254.304450981 172.20.10.5 172.217.17.110 TCP 66 65
20234 254.304513241 172.20.10.5 172.217.17.110 TCP 66 65
20235 254.304568964 172.20.10.5 172.217.17.110 TCP 93 [T
Frame 2: 74 bytes on wire (592 bits), 7
Ethernet II, Src: d2:6b:78:09:07:64 (d2
Internet Protocol Version 4, Src: 195.1
User Datagram Protocol, Src Port: 443,
Data (32 bytes)
0000 00 0c 29 3b 6a d7 d2 6b 78 09 07
0010 00 3c 00 00 00 00 37 11 54 4d c3
0020 0a 05 01 bb dc b2 00 28 e1 01 56
0030 a3 4a aa db 6f 6f 4c e5 91 ce 47
0040 d8 90 18 26 c3 36 ab 43 e8 f8
wireshark_eth0E3DCA3.pcapng
Packets: 20235 · Displayed: 15990 (79.0%)
Profile: Default

- Bu görüntü, Wireshark üzerinden 172.20.10.5 IP adresine ait trafiği filtreleyerek incelediğimiz bir ağ analiz ekranıdır. Filtre çubuğuna yazılan: ip.addr == 172.20.10.5 ifadesiyle sadece bu cihaza ait gelen/giden trafik gösterilmiştir.

- **Paket Listesi:**

- Orta bölümde, cihazın gerçekleştirdiği bağlantılar görülüyor:

- **Source/Destination (Kaynak/Hedef):** Cihazın iletişim kurduğu IP'ler (örneğin Google, WhatsApp, OpenAI sunucuları)

- **Protocol:** TCP ve TLS 1.3 gibi güvenli protokoller kullanılmış.

- **Info:** Bağlantının kısa özeti (örn. el sıkışma, veri gönderimi vb.)

- **Paket Detayları:**

- Seçilen paketin detayları aşağıdaki gibidir:

- **Ethernet Katmanı:** MAC adresleri

- **IP Katmanı:** IP adresleri (örneğin: 192.168.1.5 → 192.168.1.200)

- **UDP Katmanı:** Taşıma katmanı port bilgileri (örn. 443 → 47291)

- **Veri (Payload):** Ham veri içeriği hem hexadecimal hem ASCII formatında gösterilir.

◦Wireshark Çıktısı – Kısa Teknik Analiz

◦ Bu analiz sayesinde MITM saldırısı sonrası hedef cihazın:

◦ **Eriştiği servisler** tespit edildi (örn. OpenAI, WhatsApp, Akamai).

◦ **Kullandığı protokoller** gözlemlendi (TCP, TLSv1.3).

◦ **DNS sorguları ve IP bilgilerinden**, cihazın internet alışkanlıkları ortaya çıktı.

◦ **Şifrelenmemiş trafikte** veri içeriği açıkça görülebildi.

◦ 🔓 **Sonuç:**
HTTPS dışındaki protokollerde ciddi veri sızıntısı mümkündür. MITM saldırısı, hedefin trafiğini detaylıca izlemeye olanak sağlar

SNİFFİNG

```
172.20.10.0/28 > 172.20.10.7  » [09:38:15] [sys.log] [inf] gateway monitor started ...
172.20.10.0/28 > 172.20.10.7  » net.probe on
172.20.10.0/28 > 172.20.10.7  » [09:38:19] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
172.20.10.0/28 > 172.20.10.7  » [09:38:19] [sys.log] [inf] net.probe probing 16 addresses on 172.20.10.0/28
172.20.10.0/28 > 172.20.10.7  » [09:38:19] [endpoint.new] endpoint 172.20.10.5 detected as 9c:fc:e8:64:bb:d5 (Intel Corpo
rate).
172.20.10.0/28 > 172.20.10.7  » net.sniff on
172.20.10.0/28 > 172.20.10.7  » [09:38:44] [net.sniff.dns] dns gateway > 172.20.10.5 : e2c55.gcp.gvt2.com is 34.176.211.2
4
172.20.10.0/28 > 172.20.10.7  » [09:38:44] [net.sniff.dns] dns gateway > 172.20.10.5 : e2c55.gcp.gvt2.com is 34.176.211.2
4
172.20.10.0/28 > 172.20.10.7  » [09:38:45] [net.sniff.dns] dns gateway > 172.20.10.5 : onedscolprdwus05.westus.cloudapp.a
zure.com is 20.189.173.6
172.20.10.0/28 > 172.20.10.7  » [09:38:45] [net.sniff.dns] dns gateway > 172.20.10.5 : onedscolprdwus05.westus.cloudapp.a
zure.com is 20.189.173.6
172.20.10.0/28 > 172.20.10.7  » [09:38:46] [net.sniff.dns] dns gateway > 172.20.10.5 : play.google.com is 142.250.187.174
172.20.10.0/28 > 172.20.10.7  » [09:38:46] [net.sniff.dns] dns gateway > 172.20.10.5 : play.google.com is 142.250.187.174
172.20.10.0/28 > 172.20.10.7  » [09:38:47] [net.sniff.dns] dns gateway > 172.20.10.5 : optimizationguide-pa.googleapis.co
m is 142.251.140.42, 142.250.187.170, 216.58.212.10, 216.58.212.42, 142.250.184.138, 172.217.17.106, 142.250.187.138, 172
.217.169.138, 142.250.187.106, 216.58.214.138, 172.217.20.74, 142.251.141.42, 172.217.17.138, 172.217.17.234, 216.58.213.
106, 142.251.140.10
172.20.10.0/28 > 172.20.10.7  » [09:38:47] [net.sniff.dns] dns gateway > 172.20.10.5 : gemini.google.com is 216.58.212.14
172.20.10.0/28 > 172.20.10.7  » [09:38:47] [net.sniff.dns] dns gateway > 172.20.10.5 : gemini.google.com is 216.58.212.14
172.20.10.0/28 > 172.20.10.7  » [09:38:47] [net.sniff.dns] dns gateway > 172.20.10.5 : optimizationguide-pa.googleapis.co
m is 142.251.140.42, 142.250.187.170, 216.58.212.10, 216.58.212.42, 142.250.184.138, 172.217.17.106, 142.250.187.138, 172
.217.169.138, 142.250.187.106, 216.58.214.138, 172.217.20.74, 142.251.141.42, 172.217.17.138, 172.217.17.234, 216.58.213.
106, 142.251.140.10
```

◦Bettercap Paket Dinleme Çıktısı Teknik Analizi:

- net.probe on komutu ile 172.20.10.0/28 alt ağında aktif cihazlar tespit edildi; 172.20.10.5 IP adresli ve Intel Corporate MAC vendor'lü cihaz bulundu.
- net.sniff on komutuyla ağ arayüzü üzerinden geçen paketler dinlenmeye başladı.
- Elde edilen DNS sorgularında hedef cihazın Google servisleri (gemini.google.com, play.google.com), Microsoft Azure (onedscolprdwus05.westus.cloudapp.azure.com) ve diğer popüler servisler için DNS talepleri yaptığı görüldü.
- DNS yanıtlarındaki IP adresleri analiz edilerek hedefin hangi sunuculara bağlandığı net olarak izlendi.
- Bu sayede ağ trafiği ve cihazların hangi servislerle iletişim kurduğu tespit edilebilmekte; bu kritik bilgiler MITM saldırılarında hedefin davranışlarını anlamak ve manipülasyon için zemin hazırlamak amacıyla kullanılır.
- Paket dinleme, araya girme saldırıları için bilgi toplama aşamasının temelidir ve HTTPS gibi şifrelenmemiş protokollerde ciddi veri sızıntılarına yol açabilir.

- **Replay Attack Nedir?**
  - Replay Attack, yani "Yeniden Oynatma Saldırısı", daha önce yakalanmış bir veri paketi veya iletişim parçasının tekrar gönderilerek sistemin kandırılmasıdır.
  - Saldırgan, geçerli ve önceden kaydedilmiş bir mesajı veya oturum verisini tekrar sunucuya gönderir.
  - Amaç, karşı tarafı yanıltmak, yetkisiz işlemler yapmak veya kimlik doğrulama süreçlerini aşmaktır.
- **Replay Attack Nasıl Çalışır?**
  - İki taraflı iletişim sırasında saldırgan, paketi veya mesajı yakalar.
  - Bu veriyi değişiklik yapmadan veya ufak modifikasyonlarla yeniden hedefe yollar.
  - Eğer sistem bu "eski" veriyi fark etmezse, saldırgan başarılı olur.
  - Örneğin, bir banka transferinde “1000 TL gönder” komutu yakalanıp tekrar gönderilirse, hesaptan istenmeyen ekstra para çıkabilir.

- **Hangi Protokoller ve Durumlar Risk Altında?**
- **Kimlik doğrulama protokolleri** (özellikle nonce veya timestamp kullanmayanlar)
- **Şifrelenmemiş iletişimler** (örneğin HTTP, eski FTP protokolleri)
- **Kablosuz ağlar** (özellikle WPA/WEP gibi eski güvenlik protokolleri)
- **Token tabanlı sistemler**, nonce/timestamp kontrolü zayıfsa

- **Replay Attack'a Karşı Alınan Önlemler**
- **Nonce (tek kullanımlık sayı)** kullanımı: Her mesajda benzersiz bir sayı yer alır, aynı mesaj tekrar kabul edilmez.
- **Timestamp (zaman damgası)**: Mesajların belirli süre içinde geçerli olması sağlanır.
- **İki yönlü kimlik doğrulama**: Hem istemci hem sunucu kendini kanıtlar.
- **Şifreleme ve MAC (Message Authentication Code)**: Mesaj bütünlüğü sağlanır.
- **Güvenli protokoller kullanmak**: TLS, HTTPS gibi modern standartlar.

Injection Saldırısı Nedir?

- **Injection saldırısı**, kullanıcıdan gelen verinin doğru şekilde filtrelenmemesi sonucu, bu verinin **kodun bir parçası haline gelerek** sistem içinde **istenmeyen komutların çalışmasına** yol açmasıdır.
- En yaygın örnek: Bir web sitesinde giriş formuna girilen veriler, doğrudan veritabanı sorgusunda kullanılırsa, saldırgan bu alana **zararlı kod** yazarak sistemin davranışını değiştirebilir.

Ne Tür Zararlar Verebilir?

- Veritabanındaki tüm kullanıcı bilgilerini çalmak
- Yetki kontrolünü atlayarak giriş yapmak (örneğin şifresiz admin girişi)
- Sistemde komut çalıştırmak
- Web sitesini bozmak, sayfalara zararlı kod enjekte etmek
- Kullanıcılara kötü amaçlı yazılım göndermek

SQL Injection

- **SQL Injection (SQL Enjeksiyonu)**, saldırganın bir web uygulamasının **veritabanıyla olan iletişimini manipüle etmesiyle** oluşan bir siber saldırı türüdür.
- Saldırgan, veri giriş alanlarına (arama kutusu, giriş formu, URL parametresi vb.) özel SQL komutları yazarak uygulamayı kandırır ve veritabanına **yetkisiz komutlar çalıştırır**.

Nasıl Çalışır?

Normal Login Formu:

*SELECT * FROM users WHERE username = 'admin' AND password = '1234';*

Saldırgan Şöyle Girer:

- *username:* `' OR '1'='1`
- *password:* `' OR '1'='1`
- *SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '' OR '1'='1';*
- `'1'='1'` *ifadesi her zaman doğru olduğu için,* ***kimlik doğrulaması atlanır*** *ve saldırgan sisteme giriş yapar.*

SQL Injection Türleri

- **In-band SQL Injection:**
  Saldırgan, aynı kanal üzerinden hem saldırıyı yapar hem de verileri alır. En yaygın türdür.
  - *Error-based:* Veritabanı hatalarından bilgi toplama
  - *Union-based:* UNION SQL operatörü ile veri çekme
  - **Inferential (Blind) SQL Injection:**
    Veritabanı doğrudan veri döndürmez, ancak saldırgan mantıksal sorgularla sonuçları çıkarır.
  - *Boolean-based:* Doğru/yanlış sorgularla veri analizi
  - *Time-based:* Sorgu sonucu gecikme yaratılarak test yapılır
  - **Out-of-band SQL Injection:**
    Veriler, saldırganın kontrolündeki başka bir kanal (DNS, HTTP) üzerinden alınır. Daha nadirdir.

SQL Injection'ın Zararları ve Etkileri

- Yetkisiz veri erişimi (kredi kartı, kişisel bilgiler)
- Veritabanı içeriğinin değiştirilmesi veya silinmesi
- Sistem üzerinde komut çalıştırma (bazı durumlarda)
- Web uygulamasının tamamen kontrolünün ele geçirilmesi

SQL Injection'a Karşı Korunma Yöntemleri

- **Parametrik sorgular (Prepared Statements)** kullanmak
- **Girdi doğrulama ve temizleme** yapmak
- Veritabanı kullanıcılarına **en az yetki** prensibini uygulamak
- Hata mesajlarını kullanıcılara göstermemek
- Web uygulama güvenlik duvarı (WAF) kullanmak
- Güncel yazılım ve kütüphane kullanımı

SQL Injection Tespiti ve Test Araçları

- **Manuel test yöntemleri:**
- Tek tırnak ' ile hata oluşturma
- Mantıksal sorgularla test
- **Otomatik araçlar:**
- sqlmap (en popüler açık kaynaklı araç)
- Burp Suite
- OWASP ZAP

Örnek SQL Injection Payload’ları

- ' OR '1'='1' --
- **Amaç:** Basit bir kimlik doğrulama atlatma saldırısıdır.
- **Nasıl çalışır:**
  SQL sorgusunun WHERE şartını kırar.
  Örneğin:
- SELECT * FROM users WHERE username = '' OR '1'='1' --' AND password = '';
-
- '1'='1' ifadesi her zaman **doğru** olduğu için, WHERE şartı hep sağlanır.
- -- ile geri kalan SQL kodu yorum satırı yapılır, böylece şifre kontrolü devre dışı kalır.
- Sonuç: Saldırgan kullanıcı adı ve şifre olmadan giriş yapar.

- **' UNION SELECT username, password FROM users --**
- **Amaç:** Veritabanından hassas bilgileri çekmek için kullanılır.
- **Nasıl çalışır:**
  UNION SQL operatörü ile saldırgan, kendi sorgusunu uygulamanın sorgusuna ekler.
- Örneğin, uygulama şöyleyse:
- SELECT id, name FROM products WHERE id = 10;
- Saldırgan şu payload'ı ekleyerek:
- 10 UNION SELECT username, password FROM users --
- Sorgu şöyle olur:
- SELECT id, name FROM products WHERE id = 10
- UNION
- SELECT username, password FROM users --;
- Böylece users tablosundaki kullanıcı adı ve şifreler veri olarak çekilir.
- -- ile geri kalan kod yorumlanmaz, hata önlenir.

- **' OR sleep(5) -- (Time-based Blind SQL Injection)**
- **Amaç:** Doğrudan veri çekemediği durumlarda, veritabanının tepki süresi üzerinden bilgi edinmek.
- **Nasıl çalışır:**
  Bu payload sorgunun çalışmasını 5 saniye geciktirir.
- Eğer sorgu cevap verirken 5 saniyeden fazla gecikme oluyorsa, bu noktada sorgu çalışmış demektir.
- Bu şekilde, saldırgan true veya false koşullarını deneyerek veritabanındaki verileri "sorgu süresi" ile tespit eder.
- Çok sinsi bir yöntemdir çünkü doğrudan çıktı vermez.

- **'; DROP TABLE users; -- (Yıkıcı Saldırı Örneği)**
- **Amaç:** Veritabanında veri kaybı yaratmak, sistemi bozmak.
- **Nasıl çalışır:**
  Bu payload, sorguyu erken bitirip ardından zararlı komut çalıştırır.
- Örnek:
- SELECT * FROM users WHERE username = ''; DROP TABLE users; --' AND password = '';
-
- Burada DROP TABLE users; komutu, users tablosunu tamamen siler.
- -- yorum satırı ile geri kalan kod devre dışı kalır.
- Saldırgan veritabanındaki kritik tabloları yok ederek sistemi işlevsiz bırakabilir.

SQL Injection Örnek Payload'ları Özeti

- **' OR '1'='1' --**
  Kimlik doğrulama atlatma: Şifre kontrolünü pas geçer, sorgu her zaman doğru olur ve sisteme giriş sağlanır.
- **' UNION SELECT username, password FROM users --**
  Veri sızdırma: İkinci bir sorgu ekleyerek kullanıcı adı ve şifre gibi hassas bilgileri çeker.
- **' OR sleep(5) --**
  Blind SQL Injection testi: Sorgunun çalışmasını 5 saniye geciktirir. Cevap süresine bakarak veri olup olmadığı anlaşılır.
- **'; DROP TABLE users; --**
  Yıkıcı saldırı: Veritabanındaki kritik bir tabloyu tamamen siler, sistemi çökertir.

- Bu örnekler SQL Injection'ın ne kadar tehlikeli olduğunu ve saldırganların ne tür kötü niyetli sorgular kullanabileceğini gösterir.
  Korunmanın en etkili yolu **parametrik sorgular** ve **girdi doğrulama**dır.

Spoofing Nedir?

- Spoofing, bir saldırganın kendisini başka bir kişi, cihaz veya sistem olarak gösterdiği, kimlik veya kaynak adresi taklidi yaptığı bir saldırı türüdür. Amaç, hedef sistemi veya kullanıcıyı kandırarak yetkisiz erişim sağlamak, veri çalmak veya sistemde manipülasyon yapmaktır.

- ***Spoofing Türleri***

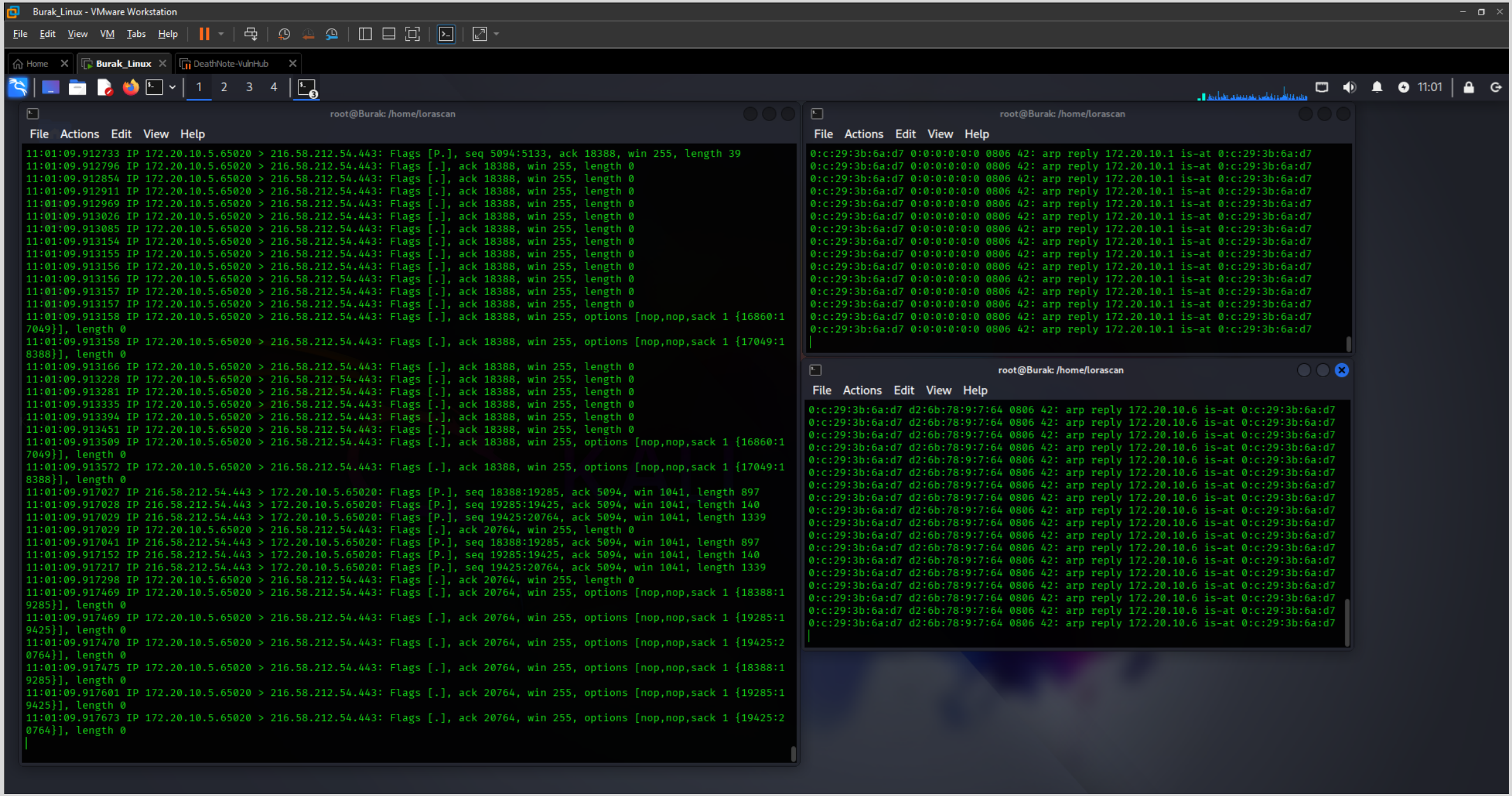
- **IP Spoofing:** Ağdaki paketlerin kaynak IP adresini değiştirerek başka bir cihaz gibi görünmek.
- **Email Spoofing:** E-posta adresini taklit ederek sahte e-posta göndermek (phishing saldırılarında sık kullanılır).
- **ARP Spoofing:** Yerel ağda ARP protokolünü manipüle ederek, iki cihaz arasındaki trafiği kesmek veya yönlendirmek.
- **DNS Spoofing:** DNS sorgularının sahte yanıtlara yönlendirilmesi, kullanıcıyı sahte web sitelerine götürmek.
- **Caller ID Spoofing:** Telefon aramalarında arayan numarayı değiştirme

Spoofing'in Etkileri

- Yetkisiz erişim
- Veri hırsızlığı
- Man-in-the-Middle (MitM) saldırıları
- Hizmet kesintisi (DoS/DDoS için zemin hazırlama)
- Güven kaybı ve itibar zararları

Spoofing'e Karşı Korunma Yöntemleri

- **Ağ filtreleme ve doğrulama:** IP paketlerinin doğruluğunu kontrol etmek için ACL'ler, firewall'lar
- **E-posta güvenliği:** SPF, DKIM ve DMARC gibi protokollerle e-posta doğrulaması
- **Ağ protokollerinin güvenliği:** ARP spoofing için statik ARP girdileri veya dinamik koruma araçları
- **DNS güvenliği:** DNSSEC kullanmak
- **Kullanıcı farkındalığı:** Sosyal mühendislik saldırılarına karşı eğitim


Burak_Linux - VMware Workstation
File Edit View VM Tabs Help
Home
Burak_Linux
DeathNote-VulnHub
1 2 3 4
11:01
root@Burak: /home/lorascan
File Actions Edit View Help
11:01:09.912733 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [P.], seq 5094:5133, ack 18388, win 255, length 39
11:01:09.912796 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.912854 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.912911 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.912969 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913026 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913085 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913154 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913155 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913156 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913156 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913157 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913157 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913158 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, options [nop,nop,sack 1 {16860:17049}], length 0
11:01:09.913158 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, options [nop,nop,sack 1 {17049:18388}], length 0
11:01:09.913166 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913228 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913281 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913335 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913394 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913451 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, length 0
11:01:09.913509 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, options [nop,nop,sack 1 {16860:17049}], length 0
11:01:09.913572 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 18388, win 255, options [nop,nop,sack 1 {17049:18388}], length 0
11:01:09.917027 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 18388:19285, ack 5094, win 1041, length 897
11:01:09.917028 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 19285:19425, ack 5094, win 1041, length 140
11:01:09.917029 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 19425:20764, ack 5094, win 1041, length 1339
11:01:09.917029 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, length 0
11:01:09.917041 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 18388:19285, ack 5094, win 1041, length 897
11:01:09.917152 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 19285:19425, ack 5094, win 1041, length 140
11:01:09.917217 IP 216.58.212.54.443 > 172.20.10.5.65020: Flags [P.], seq 19425:20764, ack 5094, win 1041, length 1339
11:01:09.917298 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, length 0
11:01:09.917469 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {18388:19285}], length 0
11:01:09.917469 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {19285:19425}], length 0
11:01:09.917470 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {19425:20764}], length 0
11:01:09.917475 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {18388:19285}], length 0
11:01:09.917601 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {19285:19425}], length 0
11:01:09.917673 IP 172.20.10.5.65020 > 216.58.212.54.443: Flags [.], ack 20764, win 255, options [nop,nop,sack 1 {19425:20764}], length 0
root@Burak: /home/lorascan
File Actions Edit View Help
0:c:29:3b:6a:d7 0:0:0:0:0:0 0806 42: arp reply 172.20.10.1 is-at 0:c:29:3b:6a:d7
root@Burak: /home/lorascan
File Actions Edit View Help
0:c:29:3b:6a:d7 d2:6b:78:9:7:64 0806 42: arp reply 172.20.10.6 is-at 0:c:29:3b:6a:d7

- ***SAĞ ÜSTTEKİ TERMİNAL EKRANI İÇİN:***
- **Ne Yaptık?**
- **arpspoof**: Bu araç, ARP (Address Resolution Protocol) paketleri göndererek ağdaki cihazların ARP tablolarını kandırmaya yarar.
- **-i eth0**: Bu komutun hangi ağ arayüzü (interface) üzerinden çalışacağını belirtir. Burada eth0 benim ağ kartım.
- **-t 172.20.10.6**: Bu parametreyle hedef cihazın IP adresini belirtiyoruz, yani kandırmaya çalıştığımız cihaz bu IP'de (örnek: 172.20.10.6).
- **172.20.10.1**: Bu ise taklit ettiğimiz adres, yani hedef cihazın ağ geçidi olarak bildiği IP. Biz bu IP'nin MAC adresini kendi cihazımızın MAC adresiymiş gibi göstereceğiz

- Bu komutla hedef cihazın (172.20.10.6) ARP tablosuna şunu söylüyoruz:
- “Hey, 172.20.10.1 (yani ağ geçidi) benim MAC adresim!”
- Normalde hedef cihaz, ağ geçidinin gerçek MAC adresini bilir ve tüm dış dünya trafiğini ona gönderir. Ancak biz bu komutla hedef cihazı kandırarak, ağ geçidinin MAC adresinin bizim cihazımızın MAC adresi olduğunu düşünmesini sağlıyoruz.

- **Sonuç:**
- Hedef cihaz dışarıya (internet veya ağdaki başka cihazlara) çıkarken tüm paketleri doğrudan bize gönderiyor.
- Biz de bu paketleri alıp istediğimiz gibi inceleyebilir, kaydedebilir ya da değiştirebiliriz.
- Böylece “Man-in-the-Middle” saldırısı gerçekleştirmiş oluyoruz.

SAĞ ALTTAKİ TERMİNAL EKRANI İÇİN

- ***Ne Yaptık?***
  - **arpspoof** aracı ile yine ARP spoofing yapıyoruz.
  - **-i eth0**: eth0 ağ arayüzünü kullanıyoruz.
  - **-t 172.20.10.1**: Bu kez hedefimiz ağ geçidi (gateway) cihazı.
  - **172.20.10.6**: Bu IP adresini taklit ediyoruz.

- Bu komutla **ağ geçidi cihazının ARP tablosunu kandırıyoruz.**
  - Ağ geçidine şunu söylüyoruz:
  - “Hey, 172.20.10.6 (hedef cihaz) benim MAC adresim!”
  - Yani, ağ geçidi hedef cihazın MAC adresi yerine bizim MAC adresimizi görüyor

SOLDAKİ BÜYÜK TERMİNAL İÇİN

- ***sudo tcpdump -i eth0 -nn***

- **Ne yaptık bu komutla?**

- Bu komutla ağdaki verileri dinlemeye başladık.
  Yani **kim, kime, ne gönderiyor** onu anlık olarak izliyoruz.

- **Daha basit haliyle:**

- Bizim bilgisayarı bir tür "kulak" gibi açtık.
- Ağda geçen paketleri (verileri) anlık olarak görmeye başladık.
- Mesela biri bir siteye giriyorsa, ping atıyorsa, veri alıp veriyorsa, hepsi burada gözüküyor.

- **Ne işimize yarar?**
  - Eğer ağda birine saldırı yapıyorsak (mesela ARP spoofing ile araya girdiysek),onun internet trafiği bizim üzerimizden geçer. Bu komutla da o geçenleri canlı canlı izliyoruz.

- **Sonuç:**
  - Ağda olup bitenleri dinliyoruz. Kim nereye bağlanıyor, hangi IP, hangi port... Hepsini çıplak gözle görüyoruz.Yani “ortamda ne dönüyor” kontrol bizde.

Sonuç:

- Ağ geçidi, hedef cihaza giden paketleri doğrudan bizim makinamıza yönlendiriyor.
- Böylece, hedef cihaz ile ağ geçidi arasındaki çift yönlü iletişimde her iki tarafın da trafiği saldırganın cihazına (bizim cihazımıza) geliyor.
- Bu da tam bir **Man-in-the-Middle (MITM)** saldırısı için gerekli iki yönlü ARP spoofing'in tamamlanması anlamına gelir.

Özet:

- Bu komutla, **ağ geçidinin hedef cihazın MAC adresi yerine senin cihazının MAC adresini görmesini sağlıyorsun**. Böylece, ağ geçidi de trafiği doğrudan sana yönlendiriyor.

(***)Birinci komut hedef cihazı kandırırken, bu komut ağ geçidini kandırıyor. İkisi birlikte çalışınca, hedef ile ağ geçidi arasındaki tüm trafik benim cihazımdan geçiyor.(***)

SON

- Siber saldırılar günümüzde giderek daha karmaşık ve sinsi hale geliyor.
  MITM, Sniffing, Replay, Injection ve Spoofing gibi saldırılar;
  hem bireyleri hem kurumları ciddi şekilde tehdit ediyor.
- 🔐 **Güvenlik sadece yazılımla değil; bilinçli kullanıcı, doğru yapılandırma ve sürekli takip ile sağlanır.**
- 🛡️ Her sistem saldırıya uğrayabilir,
  önemli olan; hazırlıklı olmak, erken tespit etmek ve hızlı tepki verebilmektir.
- ***Saldırganın bir adım önünde ol, bir adım gerisinde kalma!***

Kullandığım Güvenlik ve Analiz Araçları(LİNUX)

Araç	Açıklama	Kullanım Alanı
Bettercap	MITM saldırıları, ARP spoofing, DNS spoofing gibi aktif saldırılar	Ağ manipülasyonu, trafiğe sızma
Wireshark	Grafiksel ağ trafiği analiz aracı	Paket dinleme, TLS analizi, parola yakalama
sqlmap	SQL Injection açıklarını tespit edip istismar eder	Web uygulamalarında veri sızdırma
airmon-ng / airmon-ng start wlan0	Kablosuz ağ kartını monitör moduna geçirir	Kablosuz ağ dinleme (Wi-Fi analiz)
airodump-ng	Etraftaki Wi-Fi ağlarını ve bağlı cihazları listeler	Ağ keşfi ve analiz
arp-scan	Yerel ağdaki cihazları ve MAC adreslerini bulur	Ağ haritalama, aktif cihaz tespiti
nmap	Port taraması ve servis tespiti	Ağ keşfi, güvenlik açığı ön taraması
tcpdump	Komut satırında paket yakalama	Hızlı ve filtreli trafik analizi
ifconfig / ip a	Ağ arayüzü bilgilerini gösterir	IP ve MAC adresi tespiti
macchanger	MAC adresi değiştirme aracı	Kimlik gizleme, iz bırakmama

KAYNAKÇA

- ***Stallings, William.***
  Cryptography and Network Security: Principles and Practice. Pearson, 2020.
  (Şifreleme, ağ güvenliği ve MITM gibi saldırılardan korunma protokollerini kapsar.)
- ***RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2***
  https://datatracker.ietf.org/doc/html/rfc5246
  (İki taraflı haberleşmede TLS protokolünün nasıl koruma sağladığını açıklar.)
- ***OWASP (Open Web Application Security Project)***
  Injection attacks, XSS, MITM ve güvenlik önlemleri hakkında rehber.
  https://owasp.org
- ***Andress, Jason.***
  Foundations of Information Security. Syngress, 2021.
  (Sniffing, Spoofing, Replay gibi saldırıların tanımı ve korunma yolları.)