OFFENSIVE SECURITY

SİBER GÜVENLİK

OFFENSIVE SECURITY

SİBER GÜVENLİK

AHMET GÜREL - DR. MEHMET ALİ YALÇINKAYA

Ahmet Gürel

1994 Sinop-Ayancık doğumlu olan Ahmet GÜREL, Süleyman Demirel Üniversitesi Bilgisayar mühendisliği lisans mezunu olup şu an aynı üniversitede Bilgisayar mühendisliği alanında yüksek lisans eğitimine devam etmektedir. 2016 yılından beri siber güvenlik sektöründe yer almaktadır. Özel bir danışmanlık firmasında Sızma Testi Uzmanı, Ekip lideri gibi farklı pozisyonlarda Türkiye'nin önde gelen Finans, Telekomünikasyon, Kamu kurumları başta olmak üzere ülke genelinde birçok sızma testi projesinde aktif olarak görev almıştır.

Görev yaptığı yıllar içerisinde uygulamalı ağ sızma testleri, web ve mobil uygulama güvenliği eğitimleri ağırlıklı olmak üzere birçok kuruma kurumsal eğitimler vermiştir. Şu an bir bankada Kıdemli Güvenlik Mühendisi/Sızma Testi Uzmanı olarak çalışmaktadır. Ülke içinde ve dışında bulduğu güvenlik açıkları ile firmalara, yazılım üreticilerine güvenlik zafiyetleri bildirerek güvenlik önlemleri alınmasında yardımcı olmuştur. Bunun dışında gönüllü etkinliklerde öğrencilere yönelik siber güvenlik ve sızma testi eğitimleri vererek, bu alanda meraklı öğrencileri sektöre kazandırmayı hedeflemektedir.

Ödül Avcılığı (Bug Bounty) kapsamında Synack Red Team ekibinde yer almaktadır. Ayrıca Bugcrowd platformu aracılığıyla ve birçok uluslararası firmanın kendi ödül programları kapsamında güvenlik açığı bildirerek Hall of Fame listelerinde yer almaktadır.

CVE-2018-11538, CVE-2018-11586, CVE-2018-9147, CVE-2018-9163 ve CVE-2019-11021 numaraları Ahmet GÜREL'in bulmuş olduğu güvenlik zafiyetleri için sahip olduğu **MITRE CVE** referans numaralarıdır.

Teşekkür

Öncelikle tanıştığımız günden beri tüm çalışmalarıma ve yoğun tempoma katlanan ve desteklerini esirgemeyen sevgili eşim Simge GÜNGÖR GÜREL'e ve tüm hayatım boyunca desteklerini benden esirgemeyen ilk öğretmenim ve babam Mehmet GÜREL'e, annem Fatma GÜREL'e, kardeşlerim Müberra ve Fatih GÜREL'e sonsuz teşekkürlerimi sunarım.

Ayrıca eğitim ve öğretim hayatım boyunca üzerim de emeği olan tüm öğretmenlerime ve iş hayatına başladığım günden itibaren birlikte çalışma fırsatı bulduğum ve bana çok şey katan tüm meslektaşlarıma teşekkürü borç bilirim.

Dr. Mehmet Ali Yalçınkaya

Dr. Mehmet Ali Yalçınkaya, 1990 yılında Isparta'da doğdu. Süleyman Demirel Üniversitesi Bilgisayar Sistemleri Öğretmenliği bölümünden 2012 yılında mezun oldu. 2013 yılında ilk görev yeri olan Kırşehir Ahi Evran Üniversitesi Bilgisayar Mühendisliği Bölümü'ne araştırma görevlisi olarak atandı. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı'nda 2015 yılında yüksek lisans, 2020 yılında ise doktora eğitimini tamamladı.

Yalçınkaya, Kırşehir Ahi Evran Üniversitesi Bilgisayar Mühendisliği Bölümü'nde doktor öğretim üyesi olarak görev yapmaktadır. Ağ güvenliği, web uygulama güvenliği, sızma testleri ve makine öğrenmesi konuları araştırma alanları içerisinde yer almaktadır.

Teşekkür

Bu çalışmanın her aşamasında gösterdiği anlayış ve desteğinden dolayı değerli eşime, maddi ve manevi desteklerini hiçbir zaman esirgemeyen anneme ve babama sonsuz sevgi ve saygılarımı sunarım.

Ahmet Gürel

gurelahmet.com 

twitter.com/ahmettgurell 

instagram.com/ahmettgurel 

facebook.com/gurelahmet 

github.com/ahmetgurel 

lnkedin.com/in/ahmetgurell 

bit.ly/3ysSMt2 

udemy.com/user/ahmetgrel 

bugcrowd.com/ahmet 

ahmetgurel.yazilim@gmail.com 

Dr. Mehmet Ali Yalçınkaya

bit.ly/3dOBcYo 

twitter.com/myalcinkaya 

instagram.com/myalcinkaya 

facebook.com/ maliyalcinkaya32 

mehmetyalcinkaya@ahievran.edu.tr 

Önsöz

Günümüzde teknolojinin gelişmesi ve hayatımızın her noktasının dijitalleşmesiyle birlikte yazılım uygulamaları ve söz konusu uygulamaların güvenliği büyük önem taşımaktadır. Siber güvenlik çok fazla uzmanlık alanının içinde barındıran bir çatı kavramdır. Temel olarak offensive (saldırgan) ve defensive (savunma) olarak iki alt dalda incelenebilir. Bu kitap siber güvenliğin offensive alanına odaklanarak, bu alanı kariyer hedefi olarak belirlemiş kimseler için farklı düzeylerde uygulamalı örnekler içermektedir.

Artan siber saldılar, veri sızıntıları ve hacking olaylarıyla birlikte dünyada farklı alanlarda hizmet veren birçok kurum ve kuruluş için en önemli gündem maddelerinden birisi 'Siber Güvenlik' olmuştur. Bunu sağlamak için dünyada görülen en büyük eksik hem offensive hem de defensive alanda yetişmiş insan kaynağıdır. Bu kitap ile siber güvenliğin offensive security alanı için kapsamlı ve uygulamalı bir Türkçe kaynak oluşturmak amaçlanmıştır.

Siber güvenlik alanı çok geniş bir kavram olduğundan ve içerisinde birçok uzmanlık alanını barındırdığından dolayı 6 ay 1 yıl gibi kısa sürelerde bu alanda uzman yetiştirmek veya uzman olmak pek mümkün değildir. Bu alanda uzmanlaşmak isteyen kişilerin takip ettikleri kurs, kitap gibi materyallerin yanında; bilgisayar ağları, haberleşme teknolojileri gibi alanlara da çalışmaları, çeşitli programlama dillerini deneyerek ve öğrenmeleri ve kendilerini geliştirmeleri gerekmektedir. Bu kitapta ilk bölümlerde Linux, ağ temelleri gibi bilgi teknolojilerin temel konularına kısaca değinilmiştir. Eğer bu bölümlerde kısa olarak değinilen bu başlıkları ilk defa duyuyorsanız, diğer bölümlere geçiş yapmadan önce her bir başlığı derinlemesine araştırılmalı ve öğrenmelisiniz. Söz konusu temel konular özümsendikten sonra siber güvenlikle ilgili diğer bölümlere geçilmesini önermekteyiz. Kitapta sızma testlerinin birçok alanına değinilmiş ve uygulamalı örnekler ile desteklenmiştir. Bu örneklerin gerçekleştirildiği sanal laboratuvarları kendi bilgisayarınıza kurarak kitabı okurken bir yandan da uygulayarak öğrenmeniz konuların özümsenmesi adına oldukça önemlidir.

Offensive Security Siber Güvenlik kitabı, bilişim teknolojileri alanında eğitim gören öğrenciler, bu alanda çalışan network, sistem, yazılım vb. uzmanları ve siber güvenliğin farklı alanlarında çalışıp offensive security alanını merak eden herkes için temel seviyeden başlayarak ilerleyen bir akışa sahiptir.

Kitabın sizlere katkı sağlaması dileğiyle…

Ahmet GÜREL ve Mehmet Ali YALÇINKAYA, Eylül 2021

Son olarak, kitap içerisinde kullandığımız kodları aşağıdaki linkte bulabilirsiniz.

https://github.com/ahmetgurel

İçindekiler

1

GNU/Linux Temelleri

Bu Bölümde

Bu bölümde, GNU/Linux işletim sistemi ve terminal komutları üzerine değinilmiştir. Debian temelli Ubuntu dağıtımı üzerinden temel linux komutları işlenmiş, sızma testleri için özelleştirilmiş pentest araçlarının kurulu olduğu Kali Linux, BlackArch Linux, Parrot OS vb. dağıtımlar bulunmaktadır.

Sızma testi uzmanları bu dağıtımları ve içerisinde yüklü olan araçları sık sık kullanmaktadır. Bu yüzden GNU/Linux temelleri ve terminal komutları oldukça önemlidir.

GİRİŞ

Linux, açık kaynak kodlu ve ücretsiz bir işletim sistemi çekirdeğidir. Linux çekirdeği kaynak kodları, GNU lisansı kapsamında özgürce dağıtılabilmekte, geliştirilebilmekte ve kullanılabilmektedir. Bu özgürlük, günümüzde birçok farklı Linux tabanlı işletim sisteminin ortaya çıkmasını sağlamıştır. Kullanıcılar tarafından yaygın olarak kullanılan Linux dağıtımlarına, Ubuntu, Debian, Centos Arch örnek gösterilebilmektedir.

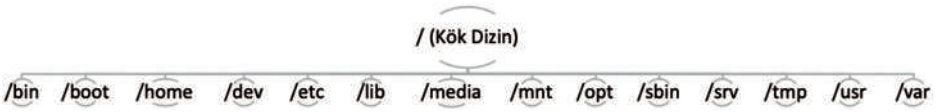
Günümüzde farklı alanlarda çalışan kullanıcılar için geliştirilmiş, içerisinde farklı araçlar barındıran Linux dağıtımları bulunmakta olup, Siber Güvenlik alanında çalışmalar gerçekleştiren kullanıcılar için de geçmişten günümüze çeşitli dağıtımlar geliştirilmiş ve geliştirilmeye devam etmektedir. Söz konusu dağıtımlara Backtrack, Black Arch ve günümüzde en yaygın kullanıma sahip **Kali** örnek olarak gösterilebilmekte.

Kali Linux, içerisinde çeşitli güvenlik testlerinde kullanılabilecek birçok aracı barındıran bir Linux dağıtımıdır. Kali Linux ve içerisinde yer alan araçların güvenlik testlerinde kullanımına ilerleyen bölümlerde detaylı olarak değinilecektir. Fakat ilk olarak siber güvenlik alanında çalışmak isteyen herkes için olmazsa Linux dizin yapısı ve komutlarının bilinmesidir.

LINUX'TA DOSYA VE DİZİN YAPISI

Dosya ve dizin yapısı, bir işletim sisteminin hızlı ve kararlı çalışmasında büyük öneme sahiptir. Bunun yanında bir işletim sistemini etkin olarak kullanabilmek için dosya ve dizin yapısını tam olarak bilmek gerekmektedir. Bu konu başlığımızda Linux tabanlı işletim sistemlerinde dosya sistemini oluşturan dizinler, kapsadıkları dosyalar ve görevlerine değinilecektir.

Linux tabanlı işletim sistemlerinde, Unix' de olduğu gibi **Tekil Hiyerarşik Klasör Yapısı** kullanılmaktadır. Bu yapıda tüm dizinler / (**slash**) işareti ile gösterilen kök dizin altında dallanmaktadır. Kök dizin altında yer alan klasörler, belirli bir standarda göre oluşturulduğu için birçok Linux dağıtımında benzerdir. Şekil 1.1'de Linux tabanlı bir işletim sisteminde kök dizin altında yer alan temel dizinler gösterilmektedir.



Şekil 1.1 Linux Dizin Yapısı

Şekil 1.1' de gösterilmekte olan dizinlerin temel görevleri ve içerdikleri elemanlara ait bilgiler aşağıda listelenmiştir.

» **/bin:** Olması zorunlu temel komut dosyalarını içerir.

» **/boot:** Başlangıç için gerekli dosyaları bulundurur.

» **/home:** Ev dizinidir. İçinde kullanıcı dosyaları masaüstü, resimler, indirilenler gibi dosyalar bulunur.

» **/dev:** Donanım dosyaları vardır.

» **/etc:** Sistem ayarlarını barındırır.

» **/lib:** Kütüphane dosyaları ve çekirdek modülleri bulunur.

» **/media:** Kaldırılabilir aygıtların (CD-ROM, USB bellek vb.) sisteme eklendiği klasördür.

» **/mnt:** Sistem açılışında otomatik olarak bağlanan sabit disk bölümleri bu dizin altında eklenir.

» **/opt:** Üçüncü parti kullanıcı programlarının kurulması içindir.

» **/sbin:** Sistemi yöneticisiyle ilgili çalıştırabilir dosyaları tutar.

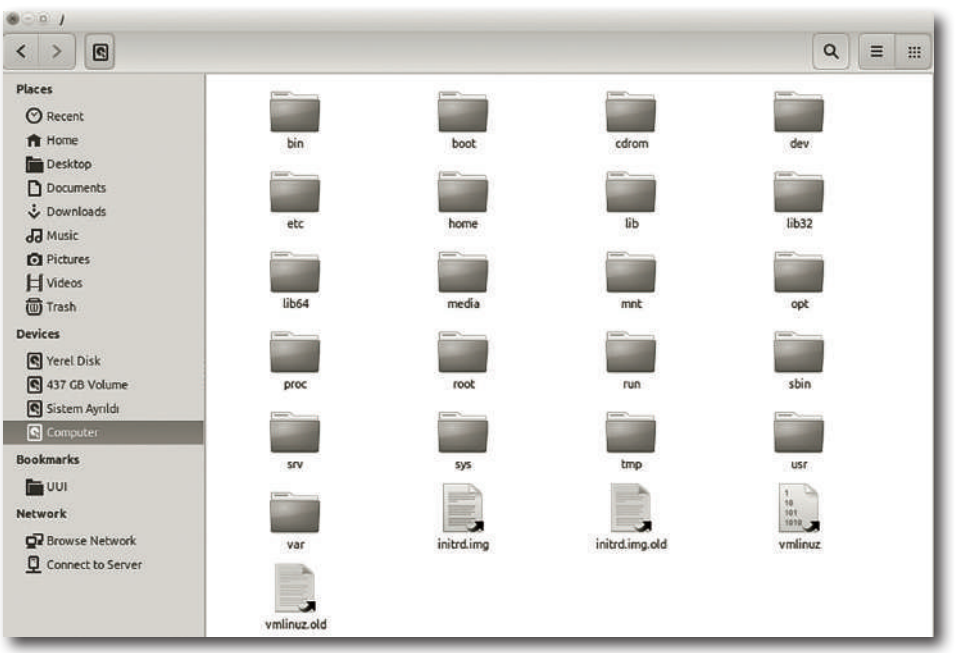
» **/srv:** Sistemin sunduğu hizmetlerle alakalıdır.

» **/tmp:** Geçici dosyaları tutmak içindir.

» **/usr:** Tüm kullanıcılarca paylaşılan verileri içeren dizindir.

» **/var:** Log dosyaları, e-posta ve yazıcı kuyrukları gibi değişken verileri barındırır.

Linux işletim sisteminde bir dizin ya da dosyanın yolu (adresi) ilk olarak kök dizinden başlamaktadır. Kök dizinden başlanarak her bir dizin ismi / ile ayrılır. **Örneğin:** /temp dizini altındaki **odev** isimli dizin içerisinde yer alan **odev1.txt** dosyasına erişmek için yazılması gereken adres; **/temp/odev/odev1.txt** olmalıdır. Şekil 1.2' de Linux tabanlı bir dağıtım olan Ubuntu' nun kök dizin yapısı gösterilmektedir.



Şekil 1.2 Ubuntunun /(Kök) Dizini

TEMEL LINUX KOMUTLARI

Linux gücünü ve esnekliğini **Shell** adı verilen (terminal olarak da bilinir) komut satırından almaktadır. Linux komutlarına hakim bir kullanıcı, pencereler arasında kaybolmak yerine, tek bir terminal penceresinden tüm işlemlerini kolaylıkla halledebilmektedir. Bu kısımda Linux tabanlı işletim sistemlerinde kullanılan temel komutlara değinilecektir. Linuxda dizinleri listeleme ve dizinler arası gezinmede kullanılan en temel komutlar `ls` ve cd'dir. Aşağıda söz konusu komutların açıklamaları ve kullanımları gösterilmiştir.

» **`ls`** Dosyaları listeler.

» **`ls -la`** Gizli dosyalar dahil tüm dosyaları listeler.

» **`cd`** Seçtiğiniz dizinin içine girmenizi sağlar.

```
ahmet-gurel@GUREL:~$ ls
AndroidStudioProjects  Downloads         Music             Templates
bin                    examples.desktop  netbeans-8.0.1    Ubuntu One
Desktop                genymotion        NetBeansProjects  Videos
dev-c++                glassfish-4.1     Pictures          VirtualBox VMs
disk                   JavaFX            Public            workspace
Documents              jdk1.8.0_20       soru19.txt~
ahmet-gurel@GUREL:~$ cd Desktop/
ahmet-gurel@GUREL:~/Desktop$
```

Şekil 1.3 Linux'ta ls ve cd Komutları

Linux tabanlı işletim sistemlerinde çeşitli işlemleri gerçekleştirmek için birçok komut kullanılmaktadır. Komut sayısının fazlalığı ve her bir komutun birçok parametreye sahip olduğu düşünüldüğünde, yardımcı komutlar büyük önem kazanmıştır. Linux işletim sistemlerinde kullanılan komutların sahip olduğu parametreler ve anlamlarını görüntülemek için `–help` ve `man` komutları kullanılmaktadır. Şekil 1.4' de `–help` komutunun `ls` komutu ile birlikte kullanılması gösterilmektedir.

```
ahmet-gurel@GUREL:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      scale sizes by SIZE before printing them.  E.g.,
                               '--block-size=M' prints sizes in units of
                               1,048,576 bytes.  See SIZE format below.
  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of last
                               modification of file status information)
                               with -l: show ctime and sort by name
                               otherwise: sort by ctime, newest first
  -C                         list entries by columns
      --color[=WHEN]         colorize the output.  WHEN defaults to 'always'
                               or can be 'never' or 'auto'.  More info below
  -d, --directory            list directory entries instead of contents,
                               and do not dereference symbolic links
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
      --format=WORD          across -x, commas -m, horizontal -x, long -l,
                               single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
```

Şekil 1.4 Linux'ta –help komut kullanımı

2

Ağ Temelleri

Bu Bölümde

Bu bölümde, Ağ temellerine değinilmiştir.

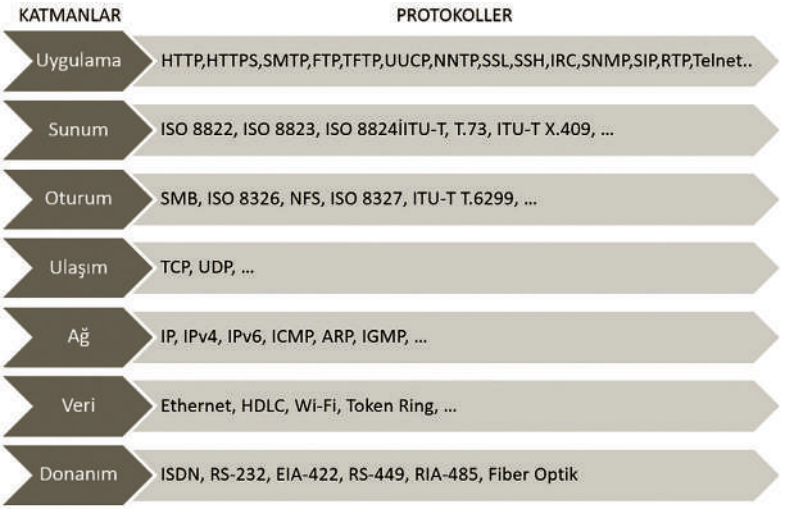
OSI, TCP/IP ve Ağ Protokolleri, önemli portlar ve servisler gibi konular işlendi. Ağ güvenliği ve sızma testleri için iyi seviyede bir ağ bilgisi gerekmektedir.

Kitabın bu bölümünde temel olarak değinilen bu konular hakkında ileri seviye bilgi edinmek için bölümde bulunan her bir başlığı detaylı olarak ileri okumasının yapılması önerilmektedir.

NETWORK NEDİR?

Bilgisayarların iletişim hatları aracılığıyla veri aktarımının sağlandığı sistem, bilgisayar ağıdır.

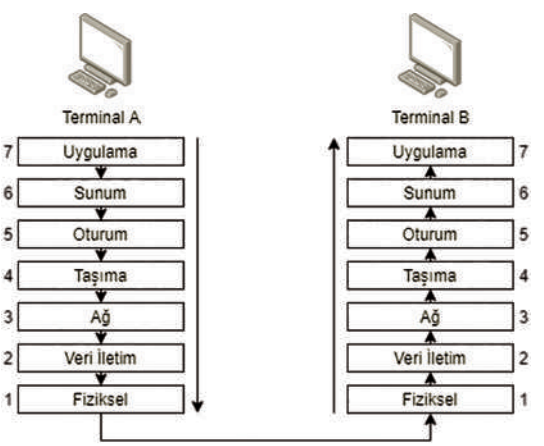
OSI MODELİ



Şekil 2.1 OSI Modeli

Şekil 2.1'de görülen **Open Systems Interconnection** (OSI) modeli **ISO** (International Organization for Standardization) tarafından geliştirimiştir. Bu modelle, ağ farkındalığına sahip cihazlarda çalışan uygulamaların birbirleriyle nasıl iletişim kuracakları tanımlanır.

7 Katmandan oluşan OSI Modelinde her katmanında belli donanımlar ve network protokolleri bulunur. Network haberleşmelerinde OSI Referans modeli kullanılır. Katmanlarda çalışan donanımlara ve protokollere ileriki sayfalarda değineceğiz.

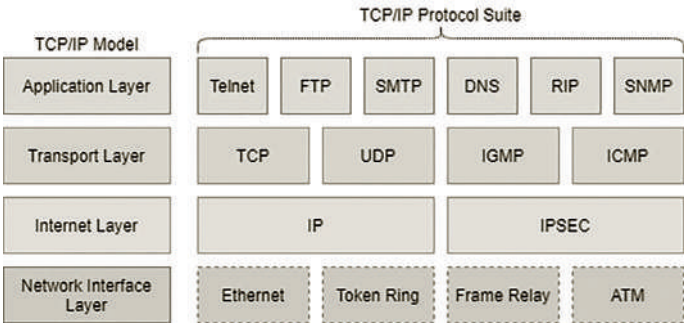


Şekil 2.2 OSI Haberleşme Modeli

TCP/IP

TCP/IP protokolü ilk olarak 80'li yıllarda Amerikan Savunma Bakanlığı (DoD) tarafından OSI tabanlı sistemlere alternatif olarak geliştirilmiştir. DoD'un Amerikan piyasasındaki ana belirleyici olması, bu protokolün Amerikan yazılımlarında standart kabul edilmesine neden olmuştur. İnternet'in babası sayılabilecek ARPANet bu nedenle TCP/IP ile doğmuştur. İnternet kullanımının büyük bir hızla artması ile birlikte TCP/IP OSI üzerinde bir üstünlük kurmuştur.

TCP/IP protokolü yapı olarak iki katmanlı bir haberleşme protokolüdür. Üst Katman TCP (Transmission Control Protocol) verinin iletimden önce paketlere ayrılmasını ve karşı tarafta bu paketlerin yeniden düzgün bir şekilde birleştirilmesini sağlar. Alt Katman IP (Internet Protocol) ise, iletilen paketlerin istenilen ağ adresine yönlendirilmesini kontrol eder. Şekil 2.3' de TCP/IP modeli katmanları ile birlikte gösterilmektedir.



Şekil 2.3 TCP/IP Modeli

» **Uygulama Katmanı (Application Layer):** Uygulama katmanında veriyi göndermek için, verinin türüne göre farklı protokollerle (HTTP, FTP vb.) aktarılmasını sağlamaktadır.

» **Taşıma Katmanı (Host to host or Transport Layer):** Bu katmanda verinin hangi protoller ile aktarılacağı belirlenir. TCP ve UDP protokolleri bu katmanda yer alır.

» **İnternet Katmanı:** Bu katmanda ağ cihazları ile birbirine bağlanmış ağlarla verinin iletimini sağlar. Paketlere IP adresleri eklenerek datagram oluşturulur.

» **Ağ Erişim Katmanı:** Bu katmanda iletişim için gerekli olan fiziksel katmandır. Bu katmanda fiziksel ağ cihazları ve ethernet, Wi-Fi vb. Protokoller çalışır.

TCP/IP protokolünde iki bilgisayar arasında bağlantı 3'lü el sıkışma adı verilen yöntem ile gerçekleştirilmektedir. Bu yöntem A ve B isminde iki bilgisayar üzerinden aşamalı olarak anlatılacaktır.

3

Ağ Sızma Testi

Bu Bölümde

Bu bölümde, sızma testi çeşitleri, metodolojisi, bilgi toplama, ağ haritalama, zafiyet taraması, exploit ve exploit sonrası adımlar hakkında bilgiler verilmektedir.

Bunlarla birlikte Nmap, Nessus ve Metasploit gibi sızma testlerinde çok kullanılan araçşarın detaylı kullanımına değinilmektedir.

SIZMA TESTİ HAKKINDA

Şekil 3.1 Sızma Testi Metodolojisi

Kitabın bu bölümünden itibaren artık sızma testleri üzerine yoğunlaşacağız. GNU/Linux ve Ağ temellerine kısaca değindik bu aşamada başarılı bir sızma testi için izlenmesi gereken adımları, araçları ve yöntemleri göreceğiz.

Şekil 3.1 ve Şekil 3.2'de görüldüğü üzere hedef belirleyerek bu hedefler üzerinde bilgi toplama işlemleri gerçekleştireceğiz. Pasif ve aktif olarak bilgi toplama tamamlandıktan sonra zafiyet tarama ve ardından sızma testi işlemlerini gerçekleştiriyor olacağız. Sızma testi aşamasında her zaman yetkili bir kullanıcı ile sisteme sızılmayabilir bu durumlurda sızma testi sonrası araştırma ile yetki yükseltme işlemleri gerçekleştirilmektedir.

Tüm bu adımlar tamamlandıktan sonra sızma testi izleri sistemden silinerek çıkılır ve bulunan tüm bulgular her adım detaylı ekran görüntüleri ve zafiyetin/zafiyetlerin giderilmesi için gerekli çözüm önerileri sızma testi raporuna yazılır.

Şekil 3.2 Sızma Testi Metodolojisi

» Bilgi Toplama
» Ağ Haritalama
» Zayıflık Tarama süreci
» Penetrasyon (Sızma) Süreci
» Erişim elde etme
» Hak Yükseltme
» Detaylı Araştırma
» Erişimlerin Korunması
» İzlerin Temizlenmesi ve Sistemden Çıkış
» Raporlama

Kısaca toparlamak gerekirse bir sızma testi aşamasında yukarıdaki adımları sıra ile gerçekleştirmek gerekmektedir.

Sızma Testleri (Penetrasyon Testleri) Çeşitleri

Beyaz Kutu (White Box) Sızma Testleri: Testi yapacak kişi, firma tarafından sistem hakkında bilgilendirilir. Bu tip testlerde daha önceden firmada çalışmış/çalışmakta olan ve ağa misafir olarak bağlanan kişilerin sisteme verebileceği hasar test edilir.

Siyah Kutu (Black Box) Sızma Testleri: Bu yöntemde testi yapacak kişiyle herhangi bir bilgi paylaşımı olmaz sadece saldırılacak hedef belirtilir. Bu tip testlerde amaç dışardan bir saldırganın sisteme nasıl erişebileceği ile ilgili bilgi elde edilir.

Gri Kutu (Gray Box) Sızma Testleri: Hem içerden hem dışarıdan yapılan test anlamındadır.

Ağ Sızma Testi İçin Lab Ortamı Kurulumu

Bu kısımda öğreneceğiniz ağ sızma testi uygulamalaır için ortam kurulumuna ihtiyacınız olacaktır. Atak yapmak için kullanacağınız bir adet sızma testi Linux dağıtımı kurunuz. İstediğinizi kurabilirsiniz benim göstereceğim örnekler Kali Linux üzerinde olacaktır.

offensive-security.com/kali-linux-vm-vmware-virtualbox-hyperv-image-download/ adresinden Kali Linux'u indirip sanal makine (Vmware, Virtualbox vb.) üzerinde çalıştırabilirsiniz.

Kali Linux ile saldırıp sızılacak sistemler **Kevgir VM** (***https://canyoupwn.me/kevgir-vulnerable-vm***), **Pluck VM** (***https://www.vulnhub.com/entry/pluck-1,178***), **Metasploitable2 VM** (***https://sourceforge.net/projects/metasploitable/files/Metasploitable2***) ve **MS17-010** zafiyeti içeren ve varsayılan kullanıcı adı ve parola bilgisi ile kurulmuş MSSQL veritabanı içeren bir Windows sanal makineler ile lab kurmanız kitabın bu bölümünde anlatılan konuları uygulamanız açısından faydalı olacaktır.

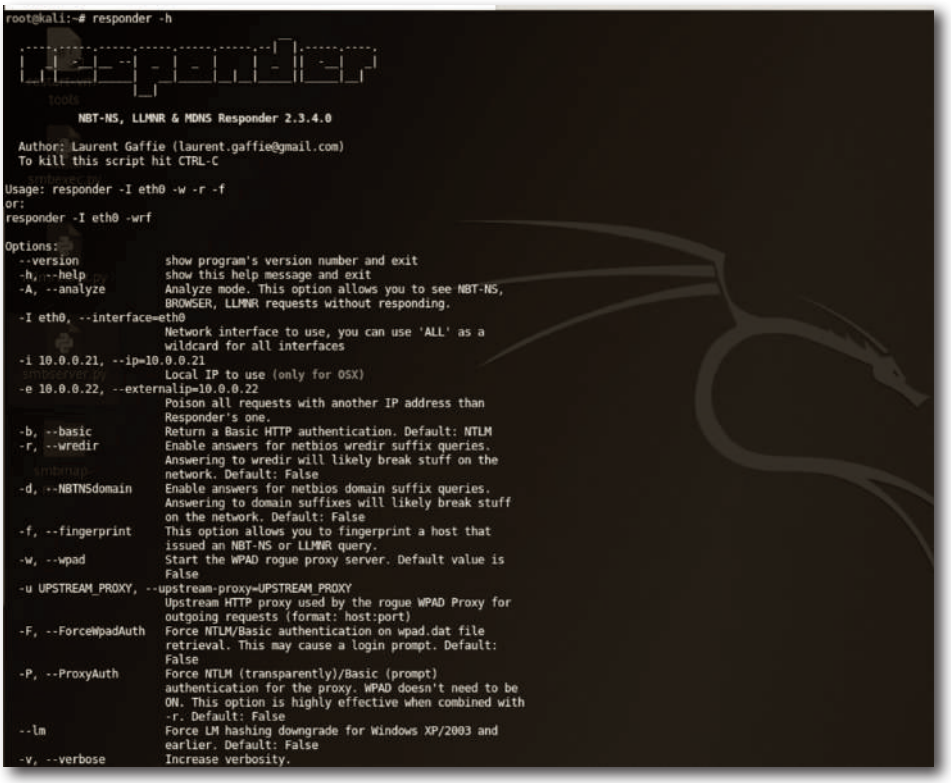
Ağ Tabanlı Saldırılar

LLMNR ve NBT-NS Zehirlenmesi

LLMNR (Link-Local Multicast Name Resolution) ve **NBT-NS** (NetBIOS Name Server) zehirlenmesi zafiyetinde saldırgan kullanıcı adı ve şifrelerin yerel ağda basit bir şekilde kendisine vermesini bekler. LLMNR ve NBT-NS görünüşte zararsız bir bileşen olarak görünür ancak aynı sub-netteki makinelerin DNS zarar gördüğünde hostu tanımak için birbirlerine yardım etmesini sağlar.

Bir makine özel bir hostu çözümlemeye çalışırken DNS çözümü başarısız olduğunda, makine yerel ağdaki diğer makinelere doğru adresi sormak için LLMNR veya NTB-NS

protokolleri üzerinden istekte bulunur. Bu zararsız bir teori gibi görünür ancak büyük bir güvenlik açığı ile saldırganlar bir sistemde tam erişim sağlamak için kullanabilirler. LLMNR ve NBT-NS Poisoning ile NTLMv2Hash bilgisi ele geçirilebilmektedir. Bu hash değeri kırılmaya çalışarak, ele geçirilen parola olduğunda kullanılabilir.



Şekil 3.3 Responder Aracı Yardım Menüsü

LLMNR ve NBT-NS zehirleme atağı için Responder aracı (***https://github.com/lgandx/Responder***) ve Metasploit'in içerisinde bulunan **llmnr_response** modülü kullanılabilir. Şekil 3.3'de görüldüğü üzere responder aracının `help` menüsü görülmektedir.

Gobuster ile Dizin Keşfi

Diğer dirb, dirbuster ve wfuzz ile aynı işlemi yapan hız ve performans olarak kaliteli bir dizin keşif aracıdır. Kullanımı Şekil 3.90'da görülmektedir.

```
Ahmets-MacBook-Pro:Desktop ahmet$ gobuster  dir -u http://php.testsparker.com/ -w wordlist2.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://php.testsparker.com/
[+] Threads:        10
[+] Wordlist:       wordlist2.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/04/20 01:10:12 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/.svn (Status: 301)
/.svn/entries (Status: 200)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/.svn (Status: 301)
/Images (Status: 301)
/Index (Status: 200)
/Products (Status: 200)
/TEST (Status: 301)
/Test (Status: 301)
/artist (Status: 200)
/auth (Status: 301)
/aux (Status: 403)
Progress: 3509 / 20517 (17.10%)
```

Şekil 3.90 gobuster Kullanımı

AĞ HARİTALAMA

NMAP İLE AKTİF BİLGİ TOPLAMA VE AĞ HARİTALAMA

Nmap (Network Map) (***https://nmap.org/***) açık kaynak kodlu gelişmiş bir güvenlik yazılımıdır. Taranan networkun ağ haritasını çıkarabilir,çalışan servisleri tespit edebilir kullanılan işletim sistemi bulunabilir. Hatta **NSE** (Nmap Scripting Engine)'ler kullanarak bazı açıklıklar tespit edilebilir, **brute force** saldırıları gerçekleştirilebilir. Bir network hakkında en detaylı bilgi toplama araçlarından birisidir. Temel Nmap kullanımı ve tarama parametlerini inceleyeceğiz. Nmap konsoldan çalışmaktadır.

Grafiksel arayüz olarak kullanmak içinde Zenmap adlı grafiksel arayüzü bulunmaktadır. Nmap Kali Linux'ta kurulu olarak gelmektedir.

Nmap bir istemciyi veya sunucuyu bir çok farklı şekilde tarayabilir ve buna göre sonuçlar getirir. Bunlar genelde çalışan port,üzeriende çalışan servisler ve işletim sistemi bilgisidir. Portların durumlar şu şekilde gelebilir:

Open (Açık): Portun erişilebilir olduğu üzerinde bir uygulamanın TCP ya da UDP baglantısı kabul ettiği durumdur.

Closed (Kapalı): Port erişilebilir fakat üzerinde uygulama yok TCP ya da UDP bağlantısı kabul etmiyor.

Filtered (Filtreli): Bir paket filtreleme var portun açık kapalı durumuna karar veremiyor.

Unfiltered (Filtresiz): ACK Scan taramasında port erişilebilir fakat açık yada kapalı durumuna karar veremiyor.

Open | Filtered: UDP, IP Protocol, FIN, Null, Xmas Scan için Nmap portların açık veya filtrelenmiş olduğuna karar veremiyor.

Closed |Filtered: Idle Scan için Nmap portların kapalı veya filtrelenmiş olduğuna karar veremiyor.

Nmap Komut Kullanımı

```
nmap [tarama türü] [parametresi] [hedef]
```

Nmap tarama komutu yukarıdaki formatta olacaktır. Nmap'in farklı tarama türleri ve bunları belirten komutlar var, ayrıca hedef kısmı bir ip adresi, domain ya da ip adresi bulunan bir txt dosyası olabilmektedir.

Bu adımdan sonra Nmap tarama türleri ve parametrelerini uygulamalı olarak işleyeceğiz. Sizde Kali Linux sanal makinanızı açarak yine sanallaştırma üzerine kurduğunuz Metsaploitable2, Kevgir VM, Pluck VM ve Windows 7 makinalarının IP adreslerini girerek bu tarama örneklerini kendi bilgisayarınıza kurduğunuz lab ortamında deneyebilirsiniz.

TCP SYN (half open) Scan

Hedefe TCP SYN paketi gönderilir. Portların kapalı olduğu durumlarda hedef makina cevap olarak RST + ACK döner. Portların açık olduğu durumlarda ise hedef makina SYN + ACK bayraklı segment döner. Son olarak RST bayraklı segment göndererek bağlantıyı koparır ve böylelikle TCP üçlü el sıkışma (TCP three-way handshaking) tamamlanmaz.

```
nmap -sS 172.16.0.138
```

TCP Connect Scan

Kaynak makinanın gerçekleştireceği TCP Connect Scan, kapalı portlara yapıldığı zaman RST + ACK paketi döner. Açık portlara yapıldığında SYN + ACK gönderir, kaynak makina ACK bayraklı segment göndererek cevaplar ve üçlü el sıkışmayı tamamlar.

```
nmap -sT 172.16.0.138
```

5

Mobil Uygulama Güvenliği

Bu Bölümde

Bu bölümde, mobil uygulama güvenliği ve sızma testi için gerekli olan Android ve iOS mobil işletim sistemleri hakkında temel bilgiler ele alınmıştır.

Ayrıca OWASP Mobil Top 10 güvenlik açıklıkları, mobil sızma testi için gerekli olan test ortamın kurulması ve mobil sızma testinde kullanılan araçlar hakkında uygulamalı bilgilere yer verilmektedir.

ANDROİD TEMELLERİ

Android, Open Handset Alliance liderliğinde Google firması tarafından akıllı telefon ve tablet bilgisayarlar gibi mobil cihazlar için geliştirilmiş Linux tabanlı işletim sistemidir. Android cihazlarda uygulamaların çalışabilmesi için **apk** uzantılı dosyalar ile uygulamalar yüklenir ve cihazlara dağıtabilir. Günümüzde mobil cihazlarda **Native** ve **Hybrid** uygulamalar kullanılmaktadır.

Native uygulamalar, C++ veya JAVA dilini temel alan Android ile yazılan uygulamalardır. HTML, CSS, JavaScript tabanlı geliştirilen ve bütün platformlara yönelik geliştirilen uygulamalar Hybrid uygulamalardır.

Günümüzde Android cihazların kullanımı oldukça yaygınlaşmıştır. Aşağıda Android cihazların yaygınlaşma sebepleri listelenmiştir;

» Açık kaynak kodlu,
» Linux tabanlı,
» Kullanım Yaygınlığı - Telefonlar, Tabletler, Arabalar...
» Gelişmiş ve ücretsiz yazılım geliştirme ortamı sunması,
» Açık uygulama market,

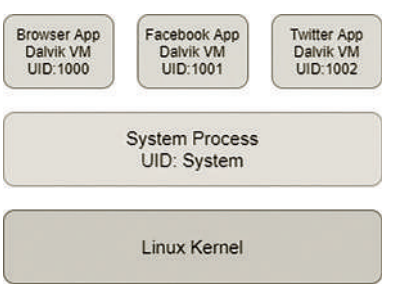
Android işletim sistemi sadece mobil cihazlarda değil çok geniş bir kullanım alanına sahiptir. Bunların başlıcaları;

» Cep telefonları, tabletler, akıllı saatler vs….
» Arabalar, akıllı ev sistemleri
» Mobil bankacılık
» Internet of Things - Nesnelerin İnterneti (IoT)

ANDROİD GÜVENLİK MODELİ

Android güvenlik modelinin belirlenmesinde, Linux güvenlik modeli baz alınmıştır (UID/GUID). Android güvenlik mobil uygulama bazlı izinler kullanılmaktadır. Uygulama izinleri, **androidManifest.xml** dosyasında tanımlanmaktadır. Uygulama kurulumu için uygulamanın sertifika ile imzalanmış olması gerekmektedir. Her bir uygulama farklı bir DVM (Dalvik Virtual Machine) içerisinde çalışmaktadır. Sistem güvenliği açısından kullanıcı kilit rol oynamaktadır. Rootlanmamış bir cihaz için root erişimi mümkün değildir. **su** uygulaması sistemde bulunmaz.

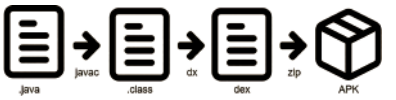
Şekil 5.1' de android güvenlik modeli gösterilmektedir.



Şekil 5.1 Android Güvenlik Modeli

Android Uygulamaları

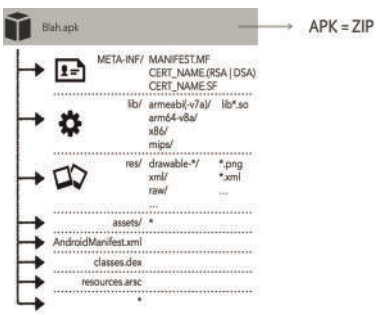
Android uygulamaları yaygın olarak **Java** + **Android SDK** ile geliştirilir. Geliştirilen uygulamalar Android Dalvik VM ve ART ile çalıştırılır. Şekil 5.2'de Android uygulama derlenme döngüsü gösterilmektektedir. Söz konusu döngüde java dosyasından ilk olarak *.class* uzantılı dosya, daha sonra da *.dex* dosyası elde edilmektedir.



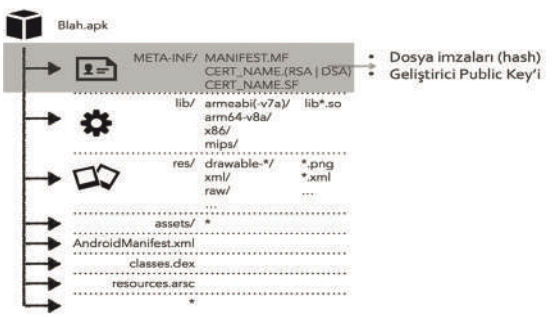
Şekil 5.2 Android Uygulaması Derlenme Döngüsü

Android Application Package File (APK)

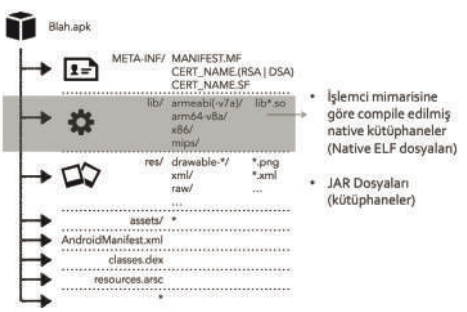
Mobil uygulama güvenliği konusuna geçmeden önce değinilmesi gereken bir diğer kavram APK' dır. Android **Application Package File** (APK) dosyası zip dosya formatına sahip mobil uygulama dosya uzantısıdır. APK dosyasının uzantısı **.zip** olarak değiştirildikten sonra WinZip, WinRAR gibi arşiv programları ile dosya içeriği görüntülenebilir. APK dosyalarının paket içeriğinin özümsenmesi, uygulamalar üzerinde zararlı yazılım analizi gibi güvenlik testlerinin yapılmasında büyük önem taşımaktadır. Bir uygulama APK dosyası içerisinde Şekil 5.5'de görüldüğü üzere **META-INF/, lib/, res/, assets/** gibi bileşenlere sahiptir. APK dosyası içerisinde yer alan bileşenler ve anlamları bu aşamadan sonra şekiller üzerinde gösterilerek incelenecektir.



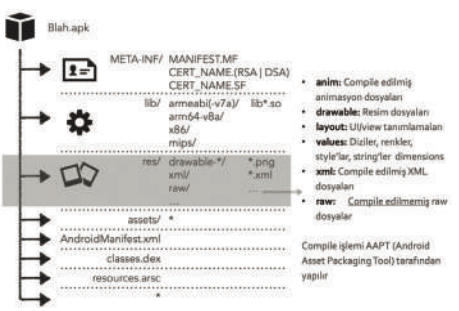
Şekil 5.5 APK Dosya İçeriği



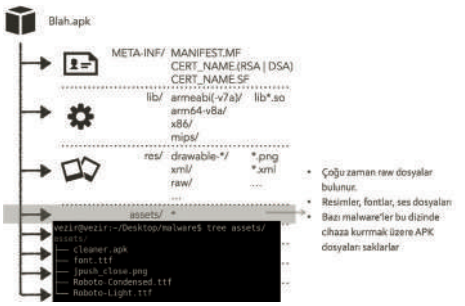
Şekil 5.6 APK META-INF/ Dosya İçeriği



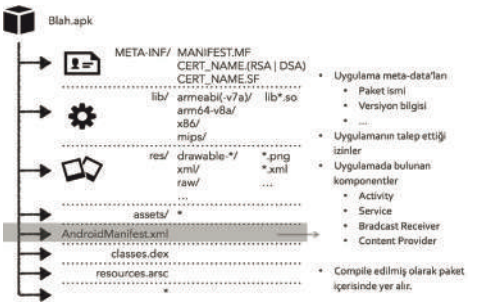
Şekil 5.7 APK lib/ Dosya İçeriği



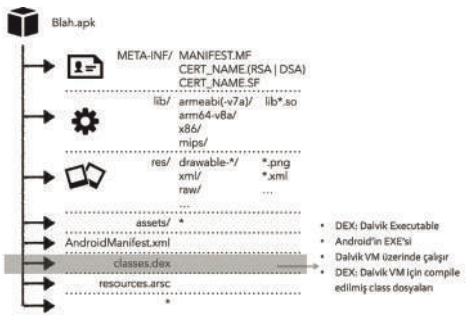
Şekil 5.8 APK res/ Dosya İçeriği



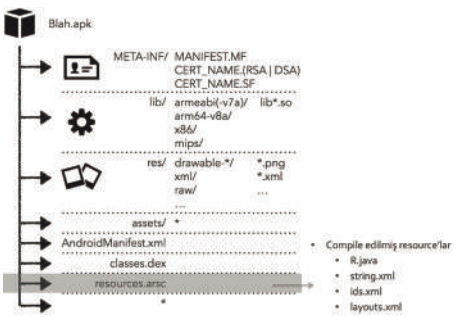
Şekil 5.9 APK assests/ Dosya İçeriği



Şekil 5.10 APK AndroidManifest.xlm Dosya İçeriği



Şekil 5.11 APK classes.dex Dosya İçeriği



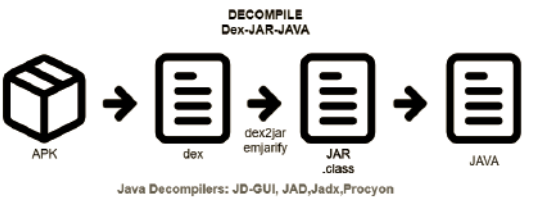
Şekil 5.12 APK resources.arsc Dosya İçeriği

ANDROID UYGULAMALARINDA TERSİNE MÜHENDİSLİK

Android uygulamalar üzerinde tersine mühendislik yöntemi ile gerçekleştirilecek işlemler için **Dex2jar** aracı kullanılabilmektedir. Dex2jar aracı ile class dosyasına dönüştürülmüş olan Android uygulaması, JD-GUI aracı ile kaynak koduna (decompile) geri çevirilebilir. Decompile işleminin yetersiz olduğu durumlarda incelenecek uygulamayı Disassembling işleminden geçirmek gerekebilir.

Şekil 5.13 ve 6.14' de sırası ile dex dosyasından kaynak kod elde etme işlemi ve APK dosyasından kaynak kod elde etme işlemi gösterilmektedir.

Şekil 5.13 Dex dosyasından Java Kaynak Koduna Dönüşüm



Şekil 5.14 APK dosyasından Java Kaynak Koduna Dönüşüm

6

Sosyal Mühendislik Testleri

Bu Bölümde

Bu bölümde, sosyal mühendislik kavramı ve sosyal mühendislik alanında gerçekleştirilebilecek çeşitli test teknikleri ve kullanılabilecek araçlar gösterilecektir.

SOSYAL MÜHENDİSLİK HAKKINDA

Günümüzde bilgi güvenliği zincirinin en zayıf halkası olarak kullanıcılar görülmektedir. Kurumlarda bilgi güvenliğini sağlamaya yönelik pek çok sistem kullanılsa da, söz konusu sistemler tam bir güvenlik sağlayamamakta, bazı saldırıların kullanıcılara ulaşmasını engelleyememektedir. Bu nedenle son kullanıcı seviyesine inmeden kurumsal bir bilgi güvenliğinden bahsetmek mümkün değildir. Kullanıcılara yönelik gerçekleştirilen saldırıların başında oltalama saldırıları gelmektedir.

Sosyal mühendislik, bir hedefin spesifik bilgileri açığa çıkarmaya veya gayri meşru sebeplerle belirli bir eylemde bulunmaya yönelik tüm teknikleri ifade etmektedir. Oltalama saldırıları sosyal mühendisliğin bir türü olup potansiyel mağdurları kimlik bilgileri, banka ve kredi kartı bilgileri gibi hassas bilgileri açığa çıkarmaya ikna etmek için gerçekleştirilmektedir (ENISA).

Saldırı genellikle bir banka veya sosyal bir ağ gibi meşru bir kaynaktan gelen sahte e-postalar ile başlar ve sonrasında kişinin bir zararlı yazılım indireceği ya da istenen bilgileri girebileceği sahte bir web sitesine yönlendirilmesi biçiminde devam eder. Bu sahte iletiler genellikle kişiselleştirilmiş değildir. Sosyal medya gibi ortamlardan elde edilen bilgiler ile yapılan hedef odaklı oltalama saldırıları (Spear Phishing) ise kişiselleştirilmiş saldırılardır.

Sosyal Mühendislik saldırılarında insanların karar verme süreçlerini değiştirmeye yönelik teknikler olduğu gibi, insan davranışları birer açıklık olarak kabul edilir ve bu açıklıklar kullanılarak saldırılar da gerçekleştirilebilir. Sosyal Mühendislik saldırısı yöntemlerinden **Sahte Senaryolar Uydurmak (Pretexting)** yöntemi, genellikle telefonla veya bir web sayfası tarzında gelişen saldırı yöntemidir.

Bu yöntemde hedefin belirlenmesi, hedefe dair bilgi toplanması, hedefle iletişim/ilişki kurulması aşamalarından sonra saldırıya geçilmektedir. Amaç, sahte bir senaryo/hikaye uydurarak bu senaryonun içine serpiştirilmiş tuzaklarla hedeften istenilen bilgiyi almaktır. Kişisel bilgiler, şifreler, özel bir sır vs. yapılan saldırılarda hemen her şekilde elde edilmeye çalışılır. Senaryo kapsamında ayrıca, hedef kişinin zararlı dosyaları kendi bilgisayarına yüklemesi sağlanıp hedefin bilgisayarına sızmak amaçlanmaktadır. Pretexting yöntemi sosyal mühendislik saldırılarında etkili bir yöntemdir. Bu süreçte oltalama tatbikatı sonunda güvenlik zincirinin en zayıf halkası olan **insan** faktörü üzerinden gelebilecek riskler ve saldırıların görülmesi hedeflenmektedir.

Phishing (Oltalama) Saldırıları

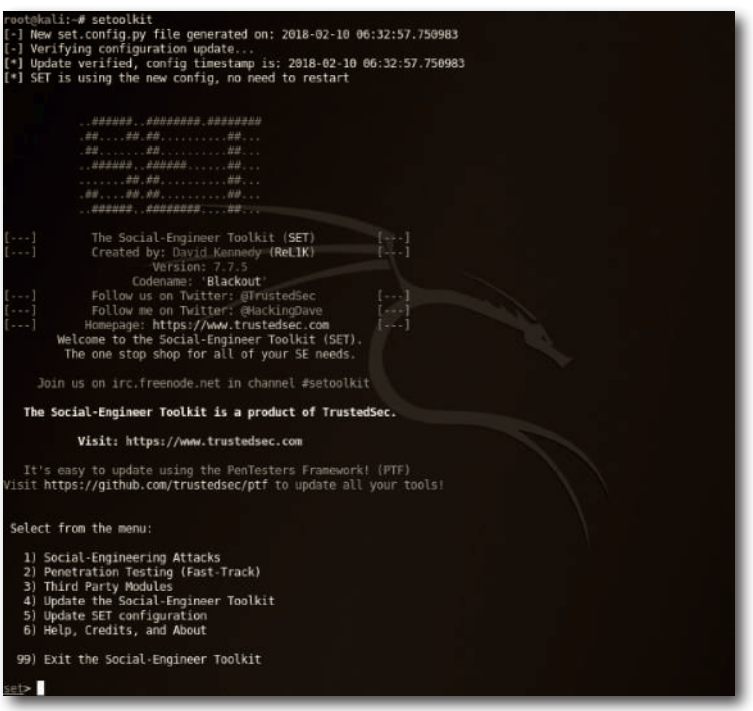
Phishing "Password" (Parola) ve "Fishing" (Balıkavlamak) sözcüklerinin birleştirilmesiyle oluşturulan, Türkçe'ye Yemleme (Oltalama) olarak çevrilmiş bir saldırı çeşididir.

Phishing testleri genellikle test edilen kişinin kurumsal mail adresi gibi hassas bilgilerinin elde edilmesi amacıyla kullanılır. Phishing testlerinde kullanılan bir diğer yaygın senaryo, banka veya resmi bir kurumdan geliyormuş gibi hazırlanan e-posta yardımıyla bilgisayar kullanıcılarının sahte sitelere yönlendirilmesidir. Phishing testlerinde için; Bankalar, Sosyal Paylaşım Siteleri, Mail Servisleri, Online Oyunlar vb. içerikli sahte web sayfaları hazırlanmaktır. Belirtilen türde sahte web sayfalarının oluşturulmasında kullanılan en yaygın araçlardan biri SET'tir.

SET (Social Engineering Toolkit) Kullanımı

SET aracı Kali Linux'de kurulu gelmektedir. SET, oltalama testlerinde kullanılmak üzere istenilen bir siteyi kopyalayabilen ve metasploit ile haberleşebilen bir araçtır. SET ile kopyalanmış ve içerisine zararlı yazılım gömülmüş sahte site, sahte bir domain üzerine yerleştirilerek phishing testleri gerçekleştirilebilmektedir. Terminale setoolkit yazarak SET aracı açılabilmektedir.

Şekil 6.1'de SET aracının konsol üzerinden çalıştırılması gösterilmektedir. SET aracı kullanılarak Phishing testi gerçekleştirmek için Şekil 6.1' de gösterilen menüden 1) Sociel-Engineering Attacks seçilerek devam edilir.



Şekil 6.1 SET (Social Engineering Toolkit) Aracının Giriş Menüsü

Şekil 6.2'de gösterilen menüden 2) `Website Attack Vectors` adımı seçilerek sahte website kopyalama adımı için devam edilir.

Şekil 6.2 SET (Social Engineering Toolkit) Aracının Saldırı Adımları

Şekil 6.3 SET (Social Engineering Toolkit) Aracının Saldırı Adımları

Şekil 6.4'de görüldüğü gibi 2) `Site Cloner` seçilerek internet adresi verilen sitenin klonu oluşturulmaktadır.

Şekil 6.4 SET (Social Engineering Toolkit) Aracı ile Kopyalanan Site

Klonlanmış sahte web sayfası, SET aracının çalıştığı sistem üzerinde yayınlanmaktadır. Bu adımdan sonra adres çubuğuna SET aracının bulunduğu bilgisayarın IP adresini giren bir kişi, kopyalanmış sayfaya ulaşacaktır. Senaryo gereği kopyalanan sayfa ziyaret edilmiş, kullanıcı adı ve parola giriş kısımlarına very girişi yapılmıştır. Kopyalanmış site üzerinden girilen kullanıcı adı ve parola bilgisi aşağıda Şekil 6.5'de görülmektedir.

```
192.168.1.12 - - [10/Feb/2018 06:58:15] "GET / HTTP/1.1" 200 -
directory traversal attempt detected from: 192.168.1.12
192.168.1.12 - - [10/Feb/2018 06:58:15] "GET /index.html HTTP/1.1" 404 -
directory traversal attempt detected from: 192.168.1.12
 directory traversal attempt detected from: 192.168.1.12
192.168.1.12 - - [10/Feb/2018 06:58:15] "GET /index.html HTTP/1.1" 404 -
192.168.1.12 - - [10/Feb/2018 06:58:15] "GET /index.html HTTP/1.1" 404 -
directory traversal attempt detected from: 192.168.1.12
192.168.1.12 - - [10/Feb/2018 06:58:30] "GET /users/email_available?email=ahmet HTTP/1.1" 404 -
directory traversal attempt detected from: 192.168.1.12
192.168.1.12 - - [10/Feb/2018 06:58:32] "GET /users/email_available?email=ahmetgurel HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
PARAM: authenticity_token=2ec9384197ad3b15c06b0fab4bff2d2c6a08a3e6
POSSIBLE PASSWORD FIELD FOUND: password=testPa
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


[*] WE GOT A HIT! Printing the output:
PARAM: authenticity_token=2ec9384197ad3b15c06b0fab4bff2d2c6a08a3e6
POSSIBLE PASSWORD FIELD FOUND: password=testparolas
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


[*] WE GOT A HIT! Printing the output:
PARAM: authenticity_token=2ec9384197ad3b15c06b0fab4bff2d2c6a08a3e6
POSSIBLE PASSWORD FIELD FOUND: password=testparolasi
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


directory traversal attempt detected from: 192.168.1.12
192.168.1.12 - - [10/Feb/2018 06:58:44] "GET /users/email_available?email=ahmetgurel HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
PARAM: authenticity_token=2ec9384197ad3b15c06b0fab4bff2d2c6a08a3e6
POSSIBLE PASSWORD FIELD FOUND: password=testparolasi
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Şekil 6.5 SET (Social Engineering Toolkit) Aracı ile Kullanıcı Adı ve Parola Ele Geçirilmesi

SET aracı yanında oltalama saldırılarında kullanılabilecek bir diğer araç HTTrack'tir. HTTrack ile hedef siteleri kopyalayıp, kullanıcı adı ve parola bilgilerinin kaydedilmesini sağlayan kodlar ekleyerek sahte web siteleri oluşturulabilmektedir. Bu işlemleri gerçekleştirebilmek için web programlama dillerine hakim olmak gerekmektedir.

7

Kablosuz Ağ Testleri (WEP/WPA/WPA2 Parola Elde Etme)

Bu Bölümde

Bu bölümde, kablosuz ağların çalışma mekanizması, WEP, WPA ve WPA2 şifreleme yöntemleri ve bunlara yönelik yapılan sızma testleri hakkında bilgi verilmektedir.

KABLOSUZ AĞLARIN ÇALIŞMA MEKANİZMASI

Açılımı **Wireless Fidelity** olan Wi-Fi yani **Kablosuz Ağ Bağlantısı**, küçük radyo dalgaları üreten bağlantı noktalarını kullanan sistemlerdir. Wi-Fi ya da Kablosuz bağlantı teknolojisine sahip olan cihazlar ağa bağlanabilir ve fiziksel (kablolu) bağlantı gerekmeden veri transferi yapabilir. Wi-Fi standartları çeşit ve özelliklerine IEEE 802.11, 802.11a, 802.11b, 802.11g ve 802.11n, ve IEEE 802.11n standartlarına görebelirlenir Bunlardan en çok karşımıza çıkan 802.11b'dir ve 2.4Ghz'lik yayılma aralığına sahiptir.

Kablosuz ağlarda çift yönlü radyo haberleşmesi kullanılır. Radyo dalgaları ile haberleşme üç çeşit olmaktadır. Bunlar alıcı (receiver), verici (transmitter) ve alıcı-verici (trans-receiver) olarak adlandırılır.

Alıcılar: Adından da anlaşılabileceği üzere sadece radyo sinyallerini alabilen fakat gönderme özelliği barındırmayan aygıtlardır. Buna örnek olarak televizyonları gösterebiliriz.

Vericiler: Sadece radyo sinyalleri gönderebilen ama alma yetileri olmayan elektronik devrelerdir. Bunlara örnek olarak radyo verici istasyonları, televizyon verici İstasyonları vb. Sistemler gösterilebilir.

Alıcı-Vericiler: Hem alma hem verme özellikleri olan aygıtlardır. Bunlara örnek olarak cep telefonu baz istasyonları, cep telefonları gösterilebilir.

Wi-Fi ağınızın güvenliğini sağlamak için ortaya çıkmış şifreleme yöntemleri vardır, bu şifreleme iyöntemleri WEP, WPA, WPA2 ve WPA3 dür. Bir sonrakinin çıkış amacı bir öncekinde bulunan güvenlik açıklıkları nedeniyle, yeterli güvenliğin sağlanamamasından kaynaklanmaktadır.

WIRED EQUIVALENT PRIVACY (WEP) ATAKLARI

Wi-Fi ağınızın güvenliğini sağlamak için ortaya çıkmış şifreleme yöntemleri vardır.

Bu şifreleme yöntemlerinden biri WEP (Wired Equivalent Privacy)'dir. Wired Equivalent Privacy (WEP), dünyada en çok kullanılan Wi-Fi güvenlik algoritmasıdır. Bunun sebebi ise, geriye uyumluluğu ve birçok router'ın kontrol panelinde ilk sırada yer alması. WEP 64-bit olarak çıktı fakat sonra 128-bit'e çıkarıldı. Günümüzde 256-bit WEP şifrelemesi mevcut olsa da, 128-bit şifreleme halen en yaygın olarak kullanılanıdır.

Algoritmadaki bir çok düzeltmeye ve arttırılan anahtar boyutuna rağmen, WEP standardında zaman içinde birçok güvenlik açığı keşfedildi. Ücretsiz araçlar (Aircrack vb.) ile kolaylıkla kıralabilmeltedir. WEP 2004 yılında resmi olarak bitirildi. WEP ağına bağlanılırken aşağıdaki gibi kimlik doğrulama gerçekleşir.

Şekil 7.1 WEP Kimlik Doğrulaması

WEP Algoritmasının 3 Önemli Amacı:

» Kimlik Doğrulama (Authentication)

» Gizlilik (Privacy)

» Bilgi Değiştirme Kontrolu (Message Modification Control)

WEP'in teknik altyapısı RC4 (Rivest Cipher) akış şifreleme algoritmasına dayanır. RC4'ün amacı, verilen bir gizli anahtar ile geniş uzunlukta rastgele sayılar üretmek ve daha sonra bu akışla göndericide düz metin mesajı şifrelemektir. Mesajın şifrelenmesinin çözümü ve şifrelemesi temel olarak XOR fonksiyonu ile yapılmaktadır. WEP onay sisteminde RC4 ile iki defa XOR'lanan sonuç aynı değeri vermektedir. IV (Initialization Vector) paketleri WEP'de düşük boyutta olduğu için tekrara neden olabiliyor. IV (Initialization Vector) gerçek anahtara eklenilip RC4 işleme giriyor dolayısıyla her IV değiştirildiğinde RC4 tekrar şifrelemeye başlıyor. Yeterince tekrar etmeyen IV (Initialization Vector) paket topladığında WEP parolası ele geçirilebilmektedir.

WEP Parolasını Ele Geçirme Adımları:

Başlamadan önce;

» Kendi cihazımızın MAC adresi = iwconfig ya da ifconfig komutu ile görebilirsiniz.

» BSSID = Hedef Mac Adresi

» ESSID = Hedef Kablosuz Ağ adı

» Channel = Yayın yapılan kanal numarası

» Access Point (AP) yani erişim noktası, kablosuz ağ yayının yapıldığı modem ya da verici cihazlara verilen isimdir.

Bu terimlerin açılımını öğrendikten sonra Kali Linux üzerindeki Aircrack araç ailesini kullarak kıracağız. Donanım olarak TP-LINK TL-WN727N adaptörünü kullanacağım.

8

SCADA Sistemleri ve Güvenliği

Bu Bölümde

Kitabın bu bölümünde, Scada sistemleri hakkında temel bilgiler anlatılarak, scada sistemlerinin güvenliği ve sızma testi adımları gösterilecektir.

SCADA Sistemleri

SCADA açılımı Supervisory Control and Data Acquisition yani **Merkezi Denetleme Kontrol ve Veri Toplama** sistemidir. Kritik sistemlerin ve tesislerin merkezi olarak izlenmesine olanak veren sistemdir.

SCADA Sistemlerinin Başlica Özellikleri

» Grafik Arayüz,

» İzleme Sistemi,

» Alarm Sistemi,

» Veri Toplama, Analiz ve Raporlama Sistemleridir.

SCADA üç temel bölümden oluşur;

» Uzak Uç Birim (Remote Terminal Unit (RTU))

» İletişim Sistemi

» Kontrol Merkezi Sistemi (Ana Kontrol Merkezi AKM – Master Terminal Unit MTU)

İletişim Protokolleri

RS232 / RS422 / RS485 / Modbus protokolleri, bir Modbus cihazını bir veya daha fazla genel seri cihaz (örneğin, yazıcılar, barkod tarayıcılar ve benzeri) ile birçok cihaz bilgisayarlara veya birbirlerine bağlanabilmeye izin verir.

SCADA'nın Kullanim Alanlari

» Nükleer Tesisler

» Elektrik Tesisleri

» Su Toplama-Arıtma-Dağıtım Tesisleri

» Trafik Kontrol Sistemleri

» Otomotiv Endüstrisi

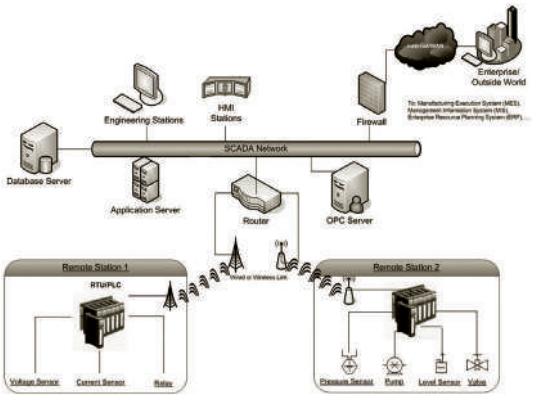
» Doğalgaz Tesisleri

» Kısaca aklımıza gelebilecek endüstriyel otomasyon sistemlerinde kullanılmaktadır.

SCADA Ağ Yapisi ve Güvenliği

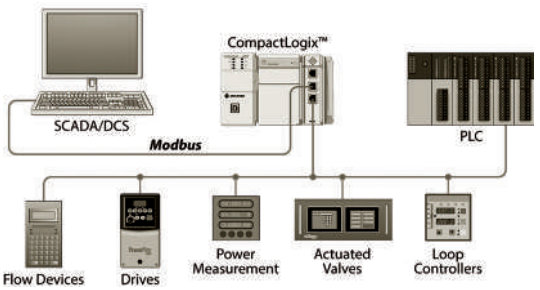
Şekil 8.1'de görülen örnek bir SCADA Network topolojisidir. Uzakdaki PLC istasyonlarından sensörlerden toplanan verileri toplayarak merkezi olarak denetlenmesini sağlamaktadır.

Bir sonraki görseli inceleyebilirsiniz.



Şekil 8.1 Örnek SCADA Ağ Topolojisi

Şekil 8.2'de görülen Modbus protokolunun SCADA içinde kullanım örneği görülmektedir. MODBUS protokolü seri port ve internet protokollerini kullanarak haberleşme yapar. (MODBUS RTU, MODBUS ASCII, MODBUS PLUS, MODBUS TCP/IP)



Şekil 8.2 MODBUS Protokolü

SCADA Sistemlerinde Güvenlik Riskleri

» APT (Advanced Persistent Threat) Saldırıları
» SCADA Yazılımlarında bulunan zafiyetler
» SCADA Network Protokol ve Cihazlarında bulunan zafiyetler
» Dışa açık SCADA Kontrol ve Yönetim Panellerinde Ön tanımlı parola kullanımı
» Kullanılan işletim sistemlerinde bulunan zafiyetler

yukarıda ki adımlar SCADA sistemlerinde güvenlik açısından büyük risk oluşturmaktadır.

SCADA Sistemlerinde Güvenlik için Kontrol Edilebilecek Noktalar

» Cihazlar fabrika varsayılan kimlik bilgileri kullanıyor mu?
» SCADA Network Protokol ve Cihazları güncel mi?
» PLC'lere erişim yalnızca yetkili makineler için beyaz listeye alınmış mı? Her yerden erişilebilir olmamalıdır.

» SCADA ağı şebekenin geri kalanından ayrı mı? Eğer değilse, kurumsal iş istasyonlarından PLC'lere ulaşmayı deneyin.

» SCADA kontrol merkezine fiziksel erişim sınırlı mı?

» Denetleyici makineden internete erişebilir misin?

» SCADA ağında çalışan açık metin hizmetleri var mı?

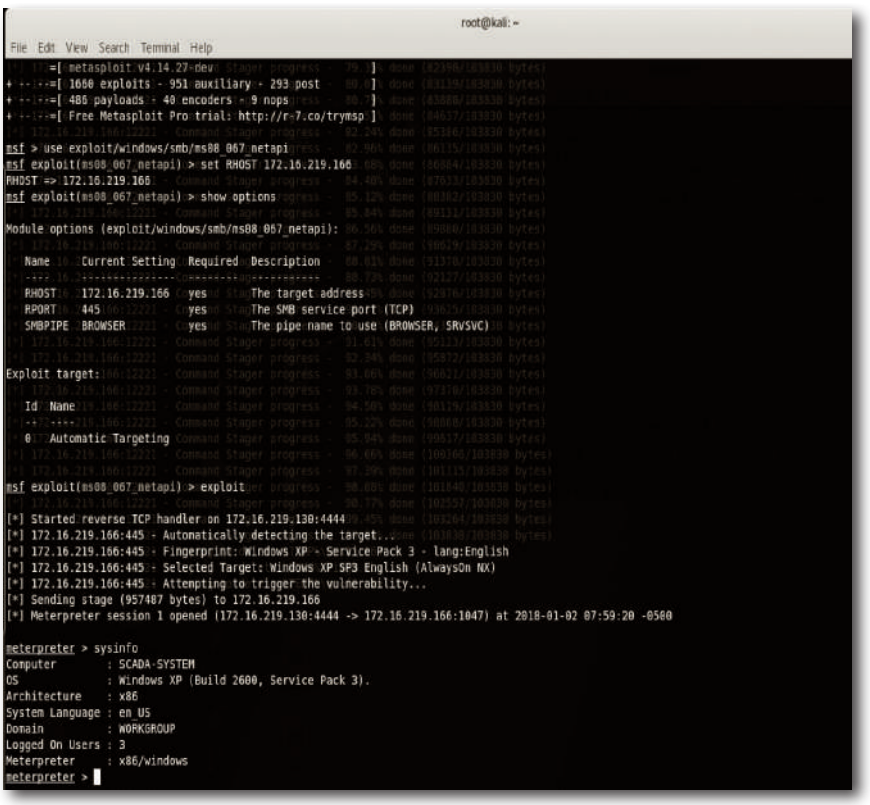
» Kurum sıkı bir parola politikası izliyor mu?

» Denetleyici makineleri, iş istasyonları ve sunucular güncel mi? (Eski sürüm zafiyerleri MS08-067, MS17-010 vb.)

» Anti-virüs yazılımı çalıştırıyorlar mı? ve uygulama beyaz listesine zorla uygulanıyor mu?

SCADA Sızma Testi Uygulamaları

Bu kısımda SCADA sistemler üzerinde gerçekleştirilebilecek bazı test senaryoları verilmiştir. Standart bir SCADA Network'unde bulunabilecek bir Windows XP işletim sistemi üzerine ABB MicroSCADA Pro SYS600 9.3 yazılımı ve Modbus slave simulator yazılımı olan **Modbus Pal 1.6b** yazılımları kurularak test sistemi oluşturulmuştur. Oluşturulan sistem üzerinden örnek birkaç senaryo incelenecektir.



Şekil 8.3 MS08-067 ile Sızılan WindowsXP

9

Bellek Taşması Zafiyetleri ve Exploit Geliştirme

Bu Bölümde

Bu bölüm altında, PCMan FTP Server 2.0.7 uygulaması üzerinden Stack Tabanlı Buffer Overflow Zafiyeti için exploit kodunu geliştireceğiz.

PCMan FTP Server 2.0.7 uygulaması üzerinden Stack Tabanlı Buffer Overflow Zafiyeti için `exploit` kodunu geliştireceğiz. PCMan FTP Server 2.0.7 uygulamasını:

https://www.exploit-db.com/apps/9fceb6fefd0f3ca1a8c36e97b6cc925d-PCMan.7z adresi üzerinden indirebilirsiniz.

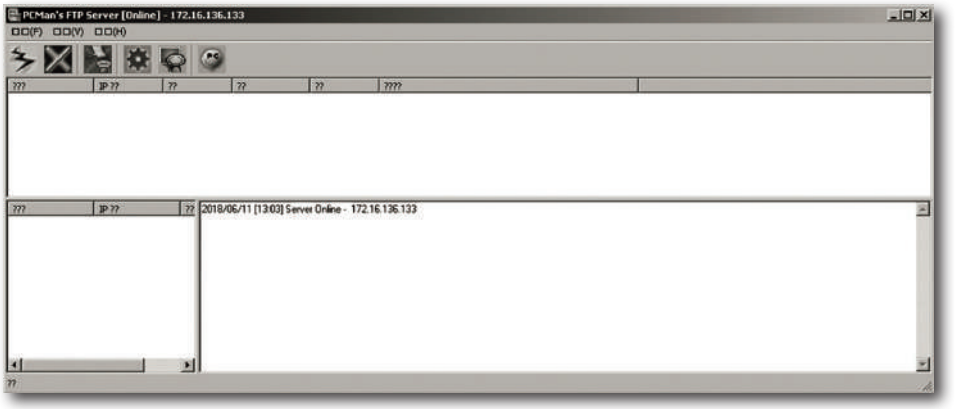
Uygulamalı işlemlere başlamadan önce bellek taşması yani buffer overflow zafiyetini incelemek gerekmektedir. Adından da anlaşılacağı üzere bellek taşması, sınırlı boyuttaki bellek alanına, planlanan miktarın üzerinde verinin kopyalanması ile yaşanan taşma durumudur. Örneğin uygulama üzerinde bir değişkene 10 karakterlik boyut tanımlandığını düşünelim. Eğer kullanıcı tarafından girilen veride 20 karakterlik bir değer varsa, fazladan girilen 10 karakter çalışma alanının dışına çıkarak uygulamanın crash olmasına ve diğer Register alanlarına yazılmasına sebep olacaktır. Bu konuyu daha derinlemesine anlayabilmek için uygulamanın çalışma anında kullandığı bellek ve adreslerin temel seviyede ne işe yaradığını bilmek gerekmektedir.

EIP: Instruction pointer olarak geçmektedir. CPU'nun an itibariyle code segment'i içerisindeki hangi instruction'i çalıştıracağını göstermektedir.

ESP: Stack pointer anlamına gelmektedir. Stack veri yapısında LIFO (Last In Firs Out) yani stack'e son giren ve ilk çıkacak elemanı göstermektedir.

FUZZING

Fuzzing; hedef uygulamayı hatayı hata vermeye zorlamak vermek amacıyla üretilen ve hedefe gönderilen veriler olarak adlandırılabilmektedir. Bu başlık altında Immunity Debugger uygulaması ile PCMan FTP Server uygulamasına yönelik olarak yapılan fuzzing işlemi anlatılacaktır. PCMan FTP Server uygulması windows üzerinde indirip çalıştırıldığında Şekil 9.1'de gösterilen ekran açılmaktadır.



Şekil 9.1 PCMan FTP Server 2.0.7 Uygulaması

PCMan FTP Server uygulaması TCP 21 numaralı port üzerinden çalışmaktadır. Söz konusu uygulamanın kurulu olduğu bilgisayar, test bilgisayarı tarafından nmap ile tarandığında, uygulamanın versiyon bilgisi ve kurulu olduğu işletim sistemi bilgilerine ulaşılmıştır. Söz konusu işleme ait tarama görüntüsü Şekil 9.2'de gösterilmektedir.

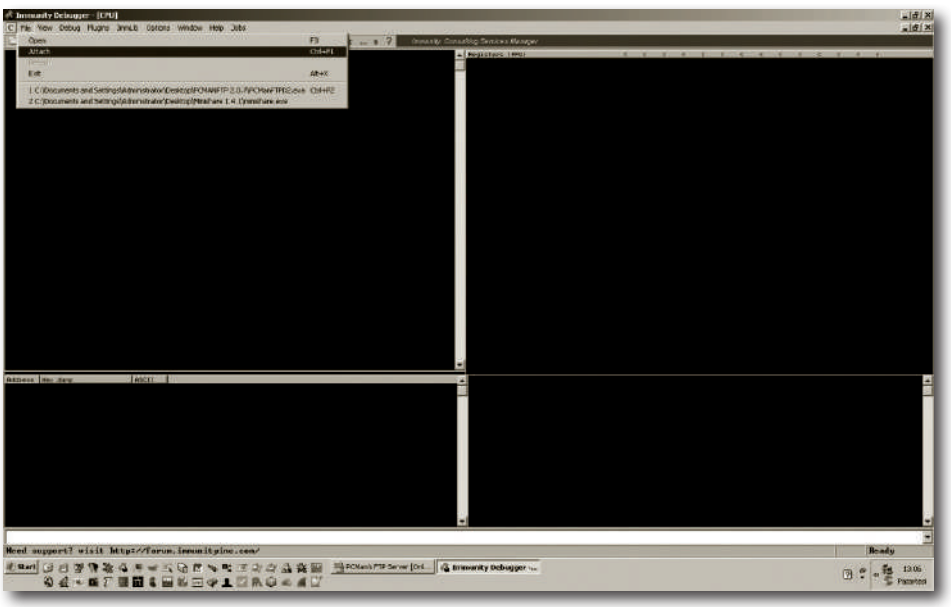
```
root@kali:~# nmap -sS -sV 172.16.136.133
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-13 13:01 EDT
Nmap scan report for 172.16.136.133
Host is up (0.00035s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE      VERSION
21/tcp  open  ftp          PCMan's FTP Server 2.0
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:10:31:F1 (VMware)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_x
p

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.28 seconds
root@kali:~#
```

Şekil 9.2 PCMan FTP Server 2.0.7 Sisteminin Nmap Tarama Sonucu

PCMan FTP Server 2.0.7 uygulamasında Stack Buffer Overflow zafiyetini tespit edebilmek için bir debugger'a ihtiyacımız olacak bu yazıda söz konusu işlem için `Immunity Debugger` kullanılmıştır. Immunity Debugger'ı kurduktan sonra, çalışan PCMan FTP Server 2.0.7 uygulamasını Attach edip çalıştırmamız gerekmektedir.

Yukarıda belirtilen işlem için, Şekil 9.3'de görüldüğü üzere **File**>**Attach** dedikten sonra gelen process listesinden uygulamamıza ait olan PCMan FTP Server'ı seçiyoruz.



Şekil 9.3 Immunity Debugger Attach İşlemi -1

Ahmet Gürel

gurelahmet.com 

twitter.com/ahmettgurell 

instagram.com/ahmettgurel 

facebook.com/gurelahmet 

github.com/ahmetgurel 

lnkedin.com/in/ahmetgurell 

bit.ly/3ysSMt2 

udemy.com/user/ahmetgrel 

bugcrowd.com/ahmet 

ahmetgurel.yazilim@gmail.com 

Dr. Mehmet Ali Yalçınkaya

bit.ly/3dOBcYo 

twitter.com/myalcinkaya 

instagram.com/myalcinkaya 

facebook.com/ maliyalcinkaya32 

mehmetyalcinkaya@ahievran.edu.tr