Red Team / AV-EDR Bypass Araçları

Genel Bakış

Bu belge, siber güvenlik dünyasında özellikle Red Team (kırmızı takım) operasyonlarında kullanılan, Antivirüs (AV) ve Endpoint Detection and Response (EDR) sistemlerini atlatmaya yönelik araçları içermektedir. Bu araçlar, hedef sistemlere sızma sonrası çeşitli görevleri gizli ve etkili bir şekilde yerine getirmek için geliştirilmiştir.

PDF'de incelenecek araçlar:

1. Manjusaka – Yeni nesil C2 (Command & Control) framework

2. SharpSploit – C# diliyle yazılmış post-exploitation kütüphanesi

3. Ebowla – Payload şifreleyerek AV tespitinden kaçmayı sağlayan araç

4. ScareCrow – AMSI ve ETW gibi güvenlik önlemlerini atlatan payload çalıştırıcısı

-Bu araçlar ayrı görevleri olsa da Red Team operasyonlarının zincirleme adımlarında birbirini tamamlayan rollere sahiptir.

1. Manjusaka – Çin Menşeli Yeni Nesil C2 Framework

Nedir?

Manjusaka, Çin kaynaklı, yeni nesil bir Command and Control (C2) framework'tür. Cobalt Strike'a benzer şekilde çalışır, fakat açık kaynak ve modern teknolojilerle geliştirilmiştir. Red Team çalışmalarında hedef sistemler üzerinde uzaktan kontrol ve yönetim sağlar.

Özellikleri

✓Çok platformlu destek (Windows ve Linux)

✓Rust diliyle yazılmış agent yapısı

✓Web tabanlı GUI üzerinden kontrol

✓İçerisinde çok sayıda yerleşik modül barındırır (keylogger, ekran görüntüsü alma, dosya yönetimi vb.)

✓Kullanımı kolay ve görsel olarak güçlü bir arayüz sağlar

Kullanım Alanları

✓Red Team operasyonlarında hedefle bağlantı kurmak

✓Eğitim ve test laboratuvarlarında gerçek saldırı senaryolarını simüle etmek

✓APT (Advanced Persistent Threat) davranışlarını modellemek

Kurulum Aşamalar

- Ön Gereksinimler:

•Python 3.x

•Go dili

•Rust kurulumu

•Redis (arka plan veritabanı olarak)

Kurulum Adımları:

```
git clone
https://github.com/ylbsEC
/Manjusaka.git
cd Manjusaka
pip install -r requirements.txt
go build
```

- Web Arayüzü Başlatma: Kurulum sonrası web arayüzü üzerinden agent oluşturma, yönetim ve bağlantı izleme işlemleri yapılır.

Dikkat Edilmesi Gerekenler

-Sadece izinli ortamlarda kullanılmalıdır. Gerçek sistemlere karşı kullanımı yasa dışıdır.

-AV ve EDR sistemleri tarafından tespit edilebilir, bu nedenle gelişmiş kamuflaj stratejileriyle kullanılması önerilir.

-Redis ve Rust gibi sistem araçlarının doğru kurulumu kritiktir.

2. SharpSploit – C# ile Yazılmış Sızma Sonrası Araç Kütüphanesi

Nedir?

SharpSploit, Red Team ve penetration testing faaliyetlerinde kullanılmak üzere geliştirilmiş, C# diliyle yazılmış açık kaynaklı bir post-exploitation (sızma sonrası) kütüphanedir. PowerSploit'in .NET sürümü olarak düşünülebilir ve modern .NET uygulamalarıyla uyumludur.

Özellikleri:

✓Modüler yapıdadır – İstenilen işlevler kod içerisine eklenebilir.

✓.NET Framework ile yazılmış güvenlik araçları geliştirmek için temel sağlar.

✓Reflective DLL Injection, shellcode çalıştırma, credential dumping, keystroke logging gibi özellikleri vardır.

✓AV/EDR atlatımı için özel yazılmış teknikleri destekler.

Kullanım Alanları:

•Red Team operasyonlarında özel payload ve sızma sonrası modüller geliştirmek

•AV/EDR atlatımı için .NET temelli özel araçlar oluşturmak

•PowerSploit kullanan yapıları modern sistemlerde kullanıma uygun hale getirmek

•Güvenlik araştırmaları ve eğitimlerde pratik çözümler geliştirmek

Kurulum Aşamaları

- Ön Gereksinimler:

-Windows işletim sistemi

-Visual Studio

-.NET Framework (4.0 ve üzeri)

Adımlar:

1 .SharpSploit GitHub sayfasına gidin:

https://github.com/cobbr/SharpSploit

2. Kodu klonlayın:

```
git clone
https://github.com/cobbr
/SharpSploit.git
```

3. Visual Studio ile projeyi açın.

4. Gerekli bağımlılıkları ekleyin ve build (derleme) işlemini yapın

5. Kütüphaneyi kendi .NET projenize dahil ederek dilediğiniz gibi kullanabilirsiniz.

Dikkat Edilmesi Gerekenler

✓AV/EDR sistemleri tarafından bazı modüller imza bazlı olarak tespit edilebilir, bu yüzden kaynak kod üzerinde özelleştirme yapılmalıdır.

✓Yalnızca izinli ve kontrollü test ortamlarında kullanılmalıdır.

✓Kodun düzgün çalışabilmesi için .NET sürüm uyumluluğu ve çalışma zamanı (runtime) ortamları kontrol edilmelidir.

✓Gerçek sistemlerde çalıştırmadan önce offline test edilmesi önerilir.

3. Ebowla – Payload Şifreleme ile AV Atlatma Aracı

Nedir?

Ebowla, oluşturulan payload'ları şifreleyerek antivirüs (AV) yazılımlarının tespitinden kaçırmak için kullanılan bir obfuscation ve encryption framework'üdür. Özellikle Meterpreter ve benzeri payload'ları şifreleyip farklı programlama dillerine entegre ederek tahmin edilebilir davranışları gizlemeyi amaçlar.

Özellikleri

✓Payload'ı AES, XOR, vb. algoritmalarla şifreleyebilir

✓Sonuçta elde edilen şifreli payload'ı C, Python, Powershell, Go gibi dillere gömebilir

✓Statik analiz araçlarını yanıltmak için davranışların yapısı değiştirilir

✓Özelleştirilebilir encoder ve decoder mekanizması içerir

✓Hem 32-bit hem 64-bit destek sunar

Kullanım Alanları

✓AV/EDR sistemlerini bypass ederek zararlı payload'ın hedef sisteme sızmasını sağlamak

✓Eğitim ve test ortamlarında gerçek saldırı senaryolarını simüle etmek

✓Mevcut shellcode/payload'ları özelleştirerek gizlemek

✓Penetrasyon testlerinde özel binary oluşturmak

Kurulum Aşamaları

- Gereksinimler:

•Python 2.7 (Ebowla Python 3 ile uyumlu değildir)

•Wine (Windows payload oluşturmak için Linux'ta çalışıyorsan)

•Mono (bazı C# işlemleri için)

Kurulum Adımları:

1. GitHub üzerinden klonlayın:

```
git clone
https://github.com/Genetic
-Malvare/Ebowla.git
cd Ebowla
```

2. Konfigürasyon dosyasını düzenleyin:

-Payload tipi, şifreleme metodu, dil seçimi (örneğin: config.json)

3. Payload şifreleme işlemi:

```
python generate_payload.py
config.json
```

4. Üretilen şifreli payload çıktısını istenen dile gömerek, compile edin (örneğin: C, Python, PowerShell)

Dikkat Edilmesi Gerekenler

✓Python 2.7 kullanmak zorundasın, yeni sürümlerle çalışmaz.

✓Üretilen dosya sadece decoder içeren sisteme uygun compile edilmelidir.

✓AV sistemleri tarafından bazı davranışlar hâlâ tespit edilebilir, şifreleme metodunu çeşitlendirmek önemlidir.

✓Kodun gerçek bir sisteme uygulanması yasal izin gerektirir, aksi durumda hukuki sonuç doğurabilir.

4. ScareCrow – AMSI ve ETW Bypass ile Payload Çalıştırıcı

Nedir?

ScareCrow, modern Windows sistemlerinde çalışan, AMSI (Antimalware Scan Interface) ve ETW (Event Tracing for Windows) gibi güvenlik önlemlerini atlatmak amacıyla tasarlanmış bir payload loader (yükleyici) aracıdır. Bu araç, özellikle shellcode'ların sessiz ve fark edilmeden çalıştırılması için geliştirilmiştir.

Özellikleri:

✓AMSI Bypass: Antivirüs yazılımlarının script analizine müdahalesini engeller.

✓ETW Patchleme: Davranış izleme sistemlerinden kaçmayı sağlar.

✓SysWhispers2 desteği ile Native API çağrılarını gizli şekilde kullanır.

✓Go diliyle yazılmıştır – taşınabilir ve hızlıdır.

✓Payload'ı EXE veya DLL formatında üretme imkanı sunar.

✓Komut satırı üzerinden özelleştirilebilir üretim.

Kullanım Alanları:

✓AV/EDR sistemlerinden kaçınarak zararlı kodu çalıştırmak

✓Red Team operasyonlarında "stealth" (gizlilik) odaklı saldırılar

✓PowerShell gibi araçlara karşı alınan modern güvenlik önlemlerini devre dışı bırakmak

✓Eğitim ortamlarında AMSI/ETW mimarisini test etmek

Kurulum Aşamaları:

- Gereksinimler:

•Go dili (GoLang)

•Windows için Mingw-w64 (derleyici)

•Git

- Adımlar:

1. Kod deposunu klonlayın:

```
git clone
https://githup.com/optiv
/ScareCrow.git
cd ScareCrow
```

2. Go bağımlılıklarını yükleyin:

```
go mod tidy
```

3. Payload oluşturun:

```
./ScareCow -payload
<shellcode.bin> -domain
<target>  -format exe
```

4. İstenilen formatta EXE ya da DLL çıktısı alabilirsiniz.

Dikkat Edilmesi Gerekenler:

✓Gerçek sistemlere uygulanması yasal değildir; yalnızca izinli test ortamlarında kullanılmalıdır.

✓ Payload şifrelenmemişse bazı AV sistemleri yine de tespit edebilir.

✓ ETW patch’leri bazı sistemlerde olay kaydı eksikliğine yol açabilir.

✓ Go dili ile derlenen dosyalar bazı sistemlerde büyük boyutlu olabilir.