**Giriş: Active Directory'nin Gözden Kaçan Güç Merkezi - AD CS ve Certipy**

Active Directory (AD), bir kurumun dijital kimliklerinin yönetildiği merkezi bir sistemdir. Kullanıcılar, bilgisayarlar ve parolalar genellikle ilk akla gelen bileşenler olsa da, arka planda çalışan ve çoğu zaman gözden kaçırılan çok daha güçlü bir yapı vardır: **Active Directory Certificate Services (AD CS)**. AD CS, ağ üzerindeki kullanıcılar, cihazlar ve hizmetler için dijital sertifikalar üreten ve yöneten bir "dijital kimlik fabrikası" gibidir. Doğru yapılandırıldığında güvenliği artırır, ancak yanlış yapılandırıldığında, standart bir kullanıcıdan Domain Admin'e uzanan şok edici yollar açabilir. İşte bu noktada, bu yanlış yapılandırmaları bulup sömürmek için tasarlanmış olan **Certipy** aracı devreye girer.

Bu yazı, Certipy'nin ne olduğunu, neden bu kadar önemli olduğunu, nasıl kullanıldığını ve AD CS güvenliğinizi test etmek için hangi pratik senaryoları uygulayabileceğinizi adım adım açıklayacaktır.

**1. Certipy Aracı Nedir?**

**Certipy**, Active Directory Certificate Services (AD CS) ortamındaki güvenlik zafiyetlerini ve yanlış yapılandırmaları tespit etmek, numaralandırmak (enumerate) ve sömürmek (exploit) için geliştirilmiş Python tabanlı bir sızma testi aracıdır. Güvenlik araştırmacıları Will Schroeder (@harmj0y) ve Lee Christensen (@tifkin\_) tarafından yayımlanan "Certified Pre-Owned" adlı whitepaper'da detaylandırılan saldırı vektörlerini otomatize eder.

Özünde Certipy, AD CS altyapısını tarayarak aşağıdaki gibi kritik sorulara yanıt arar:

* Hangi sertifika şablonları (certificate templates) mevcut?
* Hangi kullanıcılar veya gruplar bu şablonlardan sertifika talep edebilir?
* Bu şablonlarda, ayrıcalık yükseltmeye olanak tanıyan tehlikeli ayarlar var mı? (Örn: Talep edenin istediği bir kimliği sertifikaya eklemesine izin verme)
* Sertifika yetkilileri (Certificate Authorities - CA) üzerinde kimlerin yönetici hakları var?

Bu bilgileri toplayarak, saldırganın veya sızma testi uzmanının bir etki alanında nasıl yetki yükseltebileceğine dair net bir yol haritası çizer.

**2. Neden Kullanılır?**

Certipy, hem saldırgan (Red Team) hem de savunma (Blue Team) perspektifinden kritik bir araçtır.

* **Sızma Testi Uzmanları ve Saldırganlar İçin:** AD CS, genellikle göz ardı edilen bir saldırı yüzeyidir. Certipy, bu "kör noktayı" aydınlatır. Standart bir kullanıcı hesabıyla, bir yöneticinin kimliğine bürünmeyi sağlayan bir sertifika talep etme veya bir Domain Controller'ın kimliğini taklit etme gibi son derece etkili saldırıları gerçekleştirmek için kullanılır. Bu, geleneksel parola kırma veya zafiyet tarama yöntemlerine göre çok daha sessiz ve hızlı bir yetki yükseltme yöntemidir.
* **Güvenlik Yöneticileri ve Savunma Ekipleri İçin:** En iyi savunma, saldırgan gibi düşünmektir. Certipy, sistem yöneticilerinin ve güvenlik ekiplerinin kendi AD CS altyapılarını proaktif olarak denetlemelerine olanak tanır. Aracı kullanarak tehlikeli yapılandırmaları bir saldırgan fark etmeden önce tespit edip düzeltebilirler. Bu, kurumun güvenlik duruşunu güçlendirmek için paha biçilmez bir denetim aracıdır.

**3. Nasıl Kurulum Yapılır?**

Certipy'nin kurulumu oldukça basittir ve genellikle siber güvenlik odaklı bir Linux dağıtımı (örn: Kali Linux) üzerinde yapılır.

**Gereksinimler:**

* Python 3 ve pip
* Git

**Kurulum Adımları:**

1. **Aracı GitHub'dan Klonlayın:**  
   Terminali açın ve aşağıdaki komutu çalıştırın.

git clone https://github.com/ly4k/Certipy.git

1. **Dizine Geçin:**

cd Certipy

1. **Gerekli Python Kütüphanelerini Yükleyin:**  
   setup.py dosyası aracılığıyla kurulum yapmak en kolay yoldur. Bu komut, gerekli tüm bağımlılıkları otomatik olarak yükleyecektir.

python3 setup.py install

Alternatif olarak, pip ile de kurulum yapabilirsiniz:

pip3 install .

Bu adımlardan sonra certipy komutu sistem genelinde kullanılabilir hale gelecektir.

**4. Kullanım Parametreleri**

Certipy, farklı işlevler için alt komutlara sahiptir. En temel komutlar find, req ve auth'tur.

**Genel Parametreler:**

* -u <user>: Kullanıcı adı.
* -p <password>: Parola.
* -hashes <LM:NT>: Parola yerine NTLM hash'i.
* -dc-ip <IP\_ADDRESS>: Domain Controller IP adresi (DNS sorunları için).
* -target <TARGET\_HOST>: Hedef etki alanının FQDN'si (örn: dc01.domain.local).
* -ca <CA\_NAME>: Hedef Sertifika Yetkilisi (CA) adı.
* -debug: Hata ayıklama için ayrıntılı çıktı verir.

**Alt Komutlar:**

* **find:** AD CS ile ilgili yapılandırmaları numaralandırmak ve zafiyet aramak için kullanılır.
  + -vulnerable: Sadece sömürülebilir ("vulnerable") olarak işaretlenmiş şablonları ve CA'ları gösterir.
  + -enabled: Sadece aktif olan şablonları gösterir.
* **req:** Belirli bir şablondan sertifika talep etmek için kullanılır.
  + -template <TEMPLATE\_NAME>: Kullanılacak şablonun adı.
  + -upn <user@domain>: Sertifikanın içine yazılacak olan UPN (User Principal Name). Kimlik taklit etmek için kullanılır.
* **auth:** Talep edilen bir sertifikayı (.pfx dosyası) kullanarak kimlik doğrulaması yapmak için kullanılır.
  + -pfx <PFX\_FILE>: Kimlik doğrulaması için kullanılacak sertifika dosyası.
* **shadow**: "Shadow Credentials" saldırısını otomatize eder. Bir hesaba yeni bir sertifika ekleyerek o hesap için kalıcı bir arka kapı oluşturur.

**5. Etkili Kullanımın 5 Örneği ve Açıklamaları**

Aşağıdaki örnekler, domain.local adında bir etki alanında, pentester adında standart bir kullanıcı ve DC01 adında bir Domain Controller olduğunu varsaymaktadır.

**Örnek 1: Genel Durum Tespiti (Enumeration)**  
İlk adım her zaman ortamı tanımaktır. Bu komut, tüm CA'ları ve sertifika şablonlarını listeler ve potansiyel zafiyetleri işaretler.

**Komut:**

certipy find -u 'pentester@domain.local' -p 'Password123' -target domain.local -dc-ip 192.168.1.10

**Açıklama:**  
Bu komut, pentester kullanıcısının kimlik bilgileriyle AD'ye bağlanır, tüm sertifika yetkililerini ve şablonlarını listeler. Çıktıda, her şablonun izinlerini, ayarlarını ve en önemlisi "Vulnerable: True" veya "Vulnerable: False" bilgisini göreceksiniz. Bu, saldırı yüzeyini anlamak için temel adımdır.

**Örnek 2: Sadece Zafiyetli Şablonları Bulma**  
Büyük ortamlarda yüzlerce şablon olabilir. Sadece işinize yarayacak olanları görmek için çıktıyı filtrelemek en iyisidir.

**Komut:**

certipy find -u 'pentester@domain.local' -p 'Password123' -target domain.local -vulnerable

**Açıklama:**  
-vulnerable parametresi, Certipy'ye sadece bilinen ESC1, ESC2, ESC3 gibi saldırı vektörlerine karşı savunmasız olan şablonları göstermesini söyler. Bu komut, doğrudan hedefe yönelik bilgi sağlayarak zaman kazandırır. Çıktıda genellikle şablonun neden zafiyetli olduğu da açıklanır (örn: "Enrollee Supplies Subject").

**Örnek 3: Zafiyetli Bir Şablondan Sertifika Talep Etme (ESC1 Saldırısı)**  
Diyelim ki VulnerableTemplate adında bir şablon buldunuz ve bu şablon, talep edenin sertifikaya alternatif bir isim (Subject Alternative Name - SAN) eklemesine izin veriyor. Bu, kimlik taklit etmenin anahtarıdır.

**Komut:**

certipy req -u 'pentester@domain.local' -p 'Password123' -ca 'ca-server.domain.local' -target DC01.domain.local -template VulnerableTemplate -upn administrator@domain.local

**Açıklama:**  
Bu komut, VulnerableTemplate şablonunu kullanarak bir sertifika talep eder. -upn administrator@domain.local parametresi ile, oluşturulan sertifikanın Administrator kullanıcısına ait olduğunu iddia ederiz. Başarılı olursa, Certipy administrator.pfx adında bir dosya oluşturur. Bu dosya, artık Administrator kullanıcısının dijital kimliğidir.

**Örnek 4: Elde Edilen Sertifika ile Kimlik Doğrulama ve TGT Alma**  
Önceki adımda administrator.pfx dosyasını elde ettiniz. Şimdi bu kimliği kullanarak Domain Controller'dan bir Kerberos TGT (Ticket Granting Ticket) talep edeceksiniz. Bu bilet, domain admin yetkilerine sahip olduğunuzun kanıtıdır.

**Komut:**

certipy auth -pfx administrator.pfx -dc-ip 192.168.1.10

**Açıklama:**  
auth komutu, .pfx dosyasını kullanarak DC'ye kimlik doğrulaması yapar. Başarılı olursa, Administrator kullanıcısı için bir TGT elde eder ve bunu administrator.ccache dosyasına kaydeder. Artık export KRB5CCNAME=administrator.ccache komutuyla bu bileti kullanarak secretsdump.py gibi araçlarla tüm domain hash'lerini çekebilir ve etki alanının tam kontrolünü ele geçirebilirsiniz.

**Örnek 5: Kalıcı Erişim için "Shadow Credentials" Saldırısı**  
Bu gelişmiş teknik, bir kullanıcı veya bilgisayar hesabına gizlice bir sertifika ekleyerek kalıcı bir arka kapı oluşturur. Bu sayede, hesabın parolası değişse bile sertifika ile erişime devam edebilirsiniz.

**Komut:**

certipy shadow auto -u 'pentester@domain.local' -p 'Password123' -target-account pentester

**Açıklama:**  
shadow auto komutu, pentester hesabının kendisi için yeni bir sertifika oluşturup hesaba kaydeder. Bu işlem, hesabın msDS-KeyCredentialLink özelliğini değiştirerek yapılır. Saldırı başarılı olduktan sonra pentester.pfx adında bir dosya oluşur. Artık bu PFX dosyasını kullanarak certipy auth komutuyla, parolasını bilmeseniz bile pentester olarak kimlik doğrulayabilirsiniz.

**Sonuç ve Savunma Yöntemleri**

Certipy, AD CS'nin bir kurum için ne kadar güçlü bir güvenlik kalkanı ya da ne kadar tehlikeli bir Aşil tendonu olabileceğini gözler önüne seren mükemmel bir araçtır. Bu aracın varlığı, sistem yöneticilerine ve güvenlik uzmanlarına net bir mesaj verir: **Active Directory Certificate Services yapılandırmalarınızı düzenli olarak denetleyin ve sıkılaştırın!**

Bu tür saldırılardan korunmak için:

* Kullanılmayan veya güvensiz sertifika şablonlarını devre dışı bırakın.
* Şablon izinlerini "en az ayrıcalık" ilkesine göre yapılandırın. Özellikle Enroll ve Autoenroll haklarını kimin sahip olduğuna dikkat edin.
* "Enrollee supplies subject" (Talep eden konuyu belirtir) gibi tehlikeli ayarları devre dışı bırakın.
* Sertifika yetkilisi (CA) sunucularını, Domain Controller'lar kadar kritik varlıklar olarak koruyun.
* Certipy gibi araçları düzenli olarak kendi sistemlerinizde çalıştırarak proaktif bir şekilde zafiyetleri tespit edip kapatın.

Unutmayın ki dijital dünyada en büyük zafiyet, genellikle en az anlaşılan ve en çok göz ardı edilen sistemlerden kaynaklanır. AD CS de bu sistemlerin başında gelir.