Parola Güvenliği Kontrol Araçları için 2025 Yılı Teknik ve Trend Analizi

1. Giriş: Parola Güvenliğinin Evrimi ve 2025 Vizyonu

Bu rapor, 2025 yılı için parola güvenliği kontrol araçları alanındaki en son ve en etkili ilk 10 tekniği/trendi derinlemesine inceleyerek, Python tabanlı bir parola güvenlik analiz ve öneri aracı geliştiren projeye rehberlik etmeyi amaçlamaktadır. Projenin temel hedefi, kullanıcının girdiği parolanın güvenliğini analiz etmek ve gerekirse güçlü bir parola önerisi sunmaktır. Bu kapsamlı analiz, projenin teknik temelini güçlendirecek ve geleceğe yönelik güvenlik stratejilerine ışık tutacaktır.

Dijital bağımlılığın artmasıyla birlikte, her bir çevrimiçi oturum açma işlemi potansiyel bir hedef haline gelmektedir.[1] Kimlik bilgisi ihlalleri, veri ihlallerinin %94'üne neden olan temel güvenlik açığı olmaya devam etmektedir.[2] Geleneksel parola sistemleri, yapay zeka (YZ) destekli saldırılar, kimlik avı (phishing), kimlik bilgisi doldurma (credential stuffing) ve kaba kuvvet saldırıları gibi gelişmiş tehditler karşısında yetersiz kalmaktadır.[3] Bu durum, parola güvenliği yaklaşımlarında köklü bir değişimi zorunlu kılmakta ve proaktif savunma mekanizmalarının önemini artırmaktadır.

2025 yılında tehdit ortamı, sadece saldırı hacmindeki artışla değil, aynı zamanda saldırı metodolojisindeki temel bir dönüşümle karakterize edilmektedir. YZ destekli parola tahmin etme, derin sahte (deepfake) saldırıları, YZ güdümlü kötü amaçlı yazılımlar ve otomatik kimlik bilgisi doldurma gibi yöntemlerin karmaşıklığı sürekli artmaktadır.[3] Geleneksel, kural tabanlı güvenlik sistemleri, bu yeni tehditlere karşı reaktif kalmakta ve hatta "eskimiş" hale gelebilmektedir.[3] Bu durum, statik savunma mekanizmalarından dinamik, uyarlanabilir ve öngörücü sistemlere geçişin kritik bir gereklilik olduğunu göstermektedir. Siber güvenlik alanındaki bu "silahlanma yarışı", savunmacıların da YZ'yi kullanarak YZ destekli saldırılara karşı koymasını zorunlu kılmaktadır.[6] Dolayısıyla, bir parola güvenlik aracı, yalnızca mevcut parolaları değerlendirmekle kalmayıp, aynı zamanda gelecekteki YZ destekli saldırı vektörlerini öngörebilen ve bunlara karşı koyabilen proaktif mekanizmalar içermelidir. Bu, aracın kullanıcılara daha ileriye dönük

güvenlik stratejileri sunması gerektiğini ortaya koymaktadır.

2. 2025 Yılı İçin En Son ve En Etkili Parola Güvenliği Teknikleri/Trendleri

2.1. NIST 2025 Yönergeleri: Karmaşıklıktan Uzunluğa ve Parola İfadelerine Geçiş

Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından yayınlanan 800-63B Özel Yayını, parola güvenliği için bir kıyaslama noktası belirlemektedir.[2] 2025 yönergeleri, parola karmaşıklığı (büyük/küçük harf, sayı, özel karakter karışımı) yerine parola uzunluğuna odaklanmaktadır. Minimum 8 karakter, ancak tercihen 12-16 karakter veya daha uzun (15+ karakter, hatta 64 karaktere kadar) parolalar veya parola ifadeleri önerilmektedir.[1] Bu değişim, uzun parolaların kaba kuvvet saldırılarına karşı çok daha dirençli olmasından ve kullanıcılar için hatırlanmalarının daha kolay olmasından kaynaklanmaktadır.[2]

Ayrıca, NIST, zorunlu periyodik parola değişikliklerini, bir ihlal kanıtı olmadıkça önermemektedir, çünkü araştırmalar bu durumun genellikle daha zayıf parola seçimlerine yol açtığını göstermektedir.[2] Tüm yazdırılabilir ASCII ve Unicode karakter setlerinin kullanımını teşvik etmekte, parola ipuçlarını ve bilgi tabanlı kimlik doğrulama sorularını yasaklamaktadır.[2]

Bu yönergeler, hem güvenliği artırmayı hem de kullanıcı deneyimini basitleştirmeyi amaçlamaktadır. Kullanıcıların daha kolay hatırlayabileceği ancak kırılması zor parolalar oluşturmasına olanak tanıyarak, parola tekrar kullanımı ve zayıf parola seçimi gibi riskli alışkanlıkları azaltır.[2] Ayrıca, düzenleyici standartlara uyumluluğu artırır ve parola sıfırlama taleplerini azaltarak maliyet tasarrufu sağlar.[8]

Python aracı, kullanıcının girdiği parolanın uzunluğunu kontrol etmeli ve NIST 2025 yönergelerine uygun olarak uzun parola veya parola ifadesi kullanımını teşvik etmelidir. Karmaşıklık kurallarını zorlamak yerine, uzunluğa ve benzersizliğe odaklanan bir puanlama sistemi geliştirebilir. Unicode karakter desteği, aracın küresel kullanıcılar için

daha esnek olmasını sağlar.

NIST'in karmaşıklıktan uzunluğa geçişi ve zorunlu periyodik parola değişikliklerini kaldırması, kullanıcı davranışlarıyla ilgili gözlemlere dayanmaktadır.[2] Araştırmalar, sık parola değişikliklerinin genellikle daha zayıf seçimlere yol açtığını ve karmaşık kuralların tahmin edilebilir kalıplara veya kolayca tahmin edilebilir değişikliklere neden olduğunu göstermektedir.[8] Bu durum, teknik olarak ne kadar sağlam olursa olsun, aşırı yük getiren veya kullanıcılar için sezgisel olmayan güvenlik politikalarının zararlı olabileceğini vurgulamaktadır. Bu tür politikalar genellikle parolaların not alınması veya yeniden kullanılması gibi riskli alışkanlıklara yol açar.[2] Güncellenmiş NIST yönergeleri, sağlam güvenliği geliştirilmiş kullanılabilirlikle birleştirmeyi amaçlayan pragmatik bir evrimi temsil etmekte ve böylece daha iyi kullanıcı uyumu ve genel güvenliği teşvik etmektedir. Bu bağlamda, Python aracının parola önerileri ve analizleri, teknik sağlamlığın yanı sıra kullanıcı psikolojisini de dikkate almalıdır. Kullanıcı dostu ve hatırlanabilir parola ifadeleri önermek, teknik olarak güçlü ancak akılda tutulması zor parolaları zorlamaktan daha etkili olacaktır. Bu, aracın sadece "teknik olarak güçlü" bir parola değil, aynı zamanda "kullanılabilir ve sürdürülebilir güçlü" bir parola önermesi gerektiğini göstermektedir.

Aşağıdaki tablo, NIST 2025 yönergelerindeki temel değişiklikleri özetlemektedir:

Tablo 1: NIST 2025 Yönergeleri Temel Değişiklikleri

Özellik	Mevcut NIST Yönergeleri	2025 NIST Yönergeleri
Parola Süresi	Her 60-90 günde bir	Yalnızca bilinen bir ihlalde
Parola İpuçları	İzin verilir	Yasaklanmıştır
Karakter Kümeleri Desteği	Sınırlı	Tüm ASCII ve Unicode destekli
Parola Yöneticileri Teşviki	Sınırlı	Şiddetle teşvik edilir
Minimum Parola Uzunluğu	Minimum 8 karakter	Minimum 12-16 karakter
Karmaşıklık Gereksinimleri	Gerekli (büyük/küçük harf/özel karakter karışımı)	Gerekli değil; uzunluğa odaklanma

2.2. Zorunlu Güvenliği İhlal Edilmiş Kimlik Bilgisi Taraması ve Kara Listeleme

Bu teknik, yeni oluşturulan veya mevcut parolaların, bilinen veri ihlallerinde ifşa edilmiş veya yaygın olarak kullanılan, kolayca tahmin edilebilir parolaların listelerine (kara listeler) karşı gerçek zamanlı olarak taranmasını gerektirir.[1]

pwnedpasswords gibi Python kütüphaneleri, Have I Been Pwned API'si aracılığıyla bu tür kontrolleri k-anonimlik prensibiyle (düz metin parolayı sunucuya göndermeden SHA-1 hash'inin ilk 5 karakterini kullanarak) güvenli bir şekilde gerçekleştirebilir.[13] Bu, bir parolanın daha önce bir ihlalde ortaya çıkıp çıkmadığını veya yaygın bir zayıf parola olup olmadığını tespit eder.

Bu yöntem, kullanıcıların "123456" veya "password" gibi kolay tahmin edilebilir veya daha önce sızdırılmış parolaları kullanmasını engeller.[1] Bu, saldırganların kimlik bilgisi doldurma saldırılarıyla birden fazla hesaba erişmesini zorlaştırır ve genel hesap güvenliğini önemli ölçüde artırır.[7] Veri ihlallerinin büyük bir kısmının kimlik bilgisi sorunlarından kaynaklandığı düşünüldüğünde, bu proaktif önlem hayati önem taşır.

Python aracı, parola girişini anında bilinen ihlal veri tabanlarına ve yaygın zayıf parolaların kara listelerine karşı kontrol etmelidir. pwnedpasswords kütüphanesi [13] doğrudan entegre edilebilir ve kullanıcılara "Bu parola daha önce sızdırıldı" veya "Bu parola çok yaygın ve tahmin edilebilir" gibi anında ve eyleme geçirilebilir geri bildirim sağlayarak daha güçlü parolalar seçmelerine yardımcı olur.

Geleneksel parola politikaları öncelikli olarak minimum uzunluk, belirli karakter türlerinin dahil edilmesi gibi statik, dahili kurallara odaklanmıştır.[14] Ancak, özellikle ihlal edilmiş kimlik bilgilerine karşı kara listeleme vurgusu [2], dinamik, harici tehdit istihbaratının parola doğrulama sürecine dahil edilmesine yönelik önemli bir değişimi temsil etmektedir. Bu, artık sadece bir parolanın

nasıl oluşturulduğu değil, aynı zamanda *halihazırda* tehlikeye atılıp atılmadığı veya bilinen kötü listelerin bir parçası olup olmadığı ile ilgilidir. Bu durum, kimlik bilgisi doldurma gibi gerçek dünya saldırı vektörlerini [6] ve ihlallerin yüksek bir yüzdesinin kimlik bilgisi sorunlarından kaynaklandığını doğrudan ele almaktadır.[4] Değer, yalnızca sözdizimsel olarak zayıf parolaları önlemekle kalmayıp, aynı zamanda

anlamsal olarak bilinen kötü parolaların kullanımını engellemekten gelmektedir. Bu nedenle, Python aracının parola değerlendirme mantığı, yalnızca kural tabanlı (uzunluk, karakter türleri) değil, aynı zamanda gerçek zamanlı tehdit istihbaratına dayalı (kara listeleme) olmalıdır. Bu yaklaşım, aracın daha proaktif, güncel tehditlere karşı daha dirençli ve saldırganların kullandığı güncel yöntemlere karşı daha etkili olmasını sağlar.

2.3. Çok Faktörlü Kimlik Doğrulama (MFA) Evrimi ve Kimlik Avına Direnç

MFA, bir kullanıcının kimliğini doğrulamak için birden fazla doğrulama faktörü (bilgi, sahip olunan şey, biyometrik özellik) gerektiren kritik bir güvenlik katmanıdır.[7] Parola tek başına yeterli olmadığında, MFA, parola çalınsa bile yetkisiz erişimi engeller.[12] 2025'te MFA'nın benimsenmesinin artması beklenmektedir [12], özellikle Google Cloud ve AWS gibi büyük platformların bunu zorunlu hale getirmesiyle.[16] Kimlik avına dayanıklı MFA yöntemleri (örneğin, SMS veya sesli arama tabanlı MFA'dan uzaklaşma) önem kazanmaktadır, çünkü SIM-jacking gibi saldırılar bu tür zayıf MFA yöntemlerini atlayabilir.[12] FIDO standartları ve geçiş anahtarları gibi yöntemler, kimlik avına karşı dayanıklı MFA olarak öne çıkmaktadır.[17]

MFA, parola güvenliğinin temel bir bileşenidir çünkü parolaların zayıf veya tahmin edilebilir olabileceği veya sızdırılabileceği gerçeğini kabul eder.[16] Parola güvenliği ihlallerinin %81'inin kimlik bilgisi ihlallerinden kaynaklandığı göz önüne alındığında, MFA ek bir güvenlik katmanı sağlayarak hesap ele geçirme riskini önemli ölçüde azaltır.[19] MFA, kimlik avı saldırılarına karşı da koruma sağlayabilir.[16]

Python aracı doğrudan MFA'yı uygulamazken, kullanıcılara MFA'nın önemini vurgulamalı ve mümkün olduğunda etkinleştirmelerini önermelidir. Güçlü parola önerileri sunarken, bu parolanın MFA ile birlikte nasıl daha etkili olacağını açıklayabilir. Araç, pyotp gibi kütüphanelerle [20] TOTP/HOTP tabanlı MFA kodlarını doğrulama yeteneğini göstererek, bir uygulamanın MFA entegrasyonuna yardımcı olabilecek örnekler sunabilir.

Araştırma materyalleri [12], "günümüzde parolaların yeterince güçlü olmadığını" ve MFA'nın "ek koruma sağladığını" açıkça belirtmektedir. Bu durum, parola algısında temel bir değişimi ifade etmektedir: parolalar artık

tek veya *birincil* savunma mekanizması olarak değil, daha geniş, çok katmanlı bir güvenlik mimarisi içinde *temel bir katman* olarak görülmektedir.[21] "Kimlik avına dayanıklı MFA"ya artan vurgu [12], tüm MFA yöntemlerinin eşit koruma sağlamadığını ve saldırganların daha zayıf MFA uygulamalarını (örn. SMS tabanlı MFA için SIM-jacking) atlatmak için geliştiğini kabul etmektedir. Bu durum, daha güçlü, daha dirençli MFA türlerine geçişi zorunlu kılmaktadır. Bu nedenle, Python aracının parola güvenlik analizi, parolanın kendi içsel gücünü değerlendirmenin yanı sıra, kullanıcının genel kimlik doğrulama stratejisindeki yerini ve MFA ile nasıl tamamlandığını da dikkate almalıdır.

Araç, kullanıcılara parolanın tek başına yeterli olmadığını ve MFA'nın neden vazgeçilmez bir tamamlayıcı olduğunu anlatmalıdır. Bu yaklaşım, parolanın "güçlü" olmasının ötesinde, "güvenli bir ekosistem içinde güçlü" olmasının önemini vurgulamaktadır.

2.4. Parolasız Kimlik Doğrulama: Geçiş Anahtarları (Passkeys) ve FIDO Standartları

Parolasız kimlik doğrulama, geleneksel parolalara olan bağımlılığı ortadan kaldıran veya azaltan yöntemleri ifade eder. Geçiş anahtarları (Passkeys), FIDO İttifakı tarafından desteklenen ve kullanıcıların kullanıcı adı ve parola girmesine gerek kalmadan web sitesi veya uygulamadaki bir kullanıcı hesabına bağlı FIDO kriptografik kimlik bilgileridir.[18] Geçiş anahtarları, cihazda depolanan ve kullanıcının cihazını (biyometrikler, PIN, desen vb.) açtığı şekilde kilidini açtığı gizli bir anahtar çifti kullanır. Kimlik avına dayanıklıdır, her zaman güçlüdür ve paylaşılan bir sır yoktur.[18] Küresel parolasız kimlik doğrulama pazarının 2025'te 20 milyar doları aşması beklenmektedir.[17]

Bu yöntem, kimlik avı, kimlik bilgisi doldurma ve diğer uzaktan saldırı türlerini azaltarak güvenliği önemli ölçüde artırır.[18] Kullanıcılar için daha hızlı ve daha güvenli oturum açma deneyimleri sunar, şifre sıfırlama ihtiyacını ve müşteri desteği maliyetlerini azaltır.[18] İnsan hatasından kaynaklanan veri ihlallerinin %68'ini azaltmaya yardımcı olur.[19] Geçiş anahtarları, parolaların aksine çalınamaz ve saldırganların kullanabileceği oturum açma verisi içermez.[18]

Python aracı doğrudan geçiş anahtarları oluşturmasa da, parolasız kimlik doğrulama trendini vurgulamalı ve kullanıcılara bunun gelecekteki birincil kimlik doğrulama yöntemi olacağını belirtmelidir. Araç, kullanıcıları geçiş anahtarlarının sunduğu güvenlik ve kolaylık hakkında eğitebilir ve mevcut parolalarını güçlendirme çabalarının geçiş anahtarları gibi daha ileri çözümlere doğru bir adım olduğunu açıklayabilir. FIDO standartlarına uyumluluk, aracın uzun vadeli alaka düzeyini artırır.

Parolasız kimlik doğrulamanın hızlı yükselişi [18] ve büyük endüstri oyuncuları tarafından benimsenmesi [17], sadece "parola güvenliği" kavramının ötesine geçerek daha geniş bir "kimlik güvenliği" kavramına doğru derin bir değişimi işaret etmektedir. Geçiş anahtarları sadece geliştirilmiş parolalar değildir; parolayı temelden ortadan kaldırarak, bunun yerine cihaza bağlı kriptografik anahtarlara ve biyometrik verilere dayanmaktadır.[18] Bu, odağın

paylaşılan bir sırrı (parola) güvence altına almaktan, kullanıcıya özgü nitelikler (biyometrikler) veya cihaz sahipliği aracılığıyla *kimliği* güvence altına almaya kaydığını göstermektedir. Ayrıca, araştırma materyalleri sürekli olarak "geliştirilmiş kullanıcı deneyimi", "daha hızlı oturum açma süresi" ve "daha yüksek oturum açma başarı oranları"nı [18] temel faydalar olarak vurgulamakta, bu da kullanım kolaylığının artık sadece güvenlik etkinliği değil, aynı zamanda güvenlik benimsenmesi için kritik bir itici güç olduğunu göstermektedir.

Bu nedenle, Python aracının kapsamı, sadece parolanın kendisini analiz etmekle kalmayıp, kullanıcının genel kimlik doğrulama ekosistemini de dikkate almalıdır. Araç, kullanıcılara parolasız çözümlerin geleceği hakkında bilgi vererek, mevcut parola alışkanlıklarını iyileştirmenin yanı sıra, daha ileri kimlik doğrulama teknolojilerine geçiş yapmaları için bir teşvik görevi görebilir. Bu durum, aracın bir "parola güvenlik denetleyicisi" olmaktan çıkıp, bir "kimlik doğrulama sağlığı danışmanı" rolüne evrilme potansiyelini işaret etmektedir.

Aşağıdaki tablo, farklı kimlik doğrulama yöntemlerini güvenlik, kullanıcı deneyimi ve kimlik avına direnç açısından karşılaştırmaktadır:

Tablo 2: Kimlik Doğrulama Yöntemlerinin Karşılaştırması

Yöntem	Güvenlik Düzeyi	Kullanıcı Deneyimi	Kimlik Avına Direnç	Uygulama Karmaşıklığı
Geleneksel Parolalar	Düşük-Orta (uzunluğa/karmaşıklığa bağlı)	Orta (hatırlama zorluğu, sıfırlama)	Düşük (kolayca kimlik avına yakalanabilir)	Düşük
SMS/E-posta OTP ile MFA	Orta (paroladan daha iyi, ancak SIM-jacking riski)	Orta (SMS/e-posta bekleme)	Düşük-Orta (telefoni zafiyetleri)	Orta
Uygulama Tabanlı OTP ile MFA	Yüksek (parola ve OTP uygulaması)	Orta (uygulama açma, kod girme)	Orta (kimlik avı siteleri OTP isteyebilir)	Orta
Donanım Anahtarları ile MFA	Çok Yüksek (kriptografik, fiziksel anahtar)	Orta (anahtarı taşıma zorunluluğu)	Çok Yüksek (kimlik avına dayanıklı)	Orta-Yüksek
Biyometrik MFA	Yüksek (benzersiz, taklit	Yüksek (hızlı, cihaz	Orta (fiziksel biyometrikler	Orta

	edilmesi zor)	entegrasyonu)	taklit edilebilir)	
Geçiş Anahtarları (Passkeys)	Çok Yüksek (kriptografik, kimlik avına dayanıklı)	Çok Yüksek (parolasız, cihaz kilidiyle entegre)	Çok Yüksek (kimlik avına dayanıklı)	Orta-Yüksek

2.5. Parola Gücü Analizi için Yapay Zeka ve Makine Öğrenimi Uygulamaları

Yapay Zeka (YZ) ve Makine Öğrenimi (ML), parola güvenliğini artırmada dönüştürücü bir rol oynamaktadır.[3] Geleneksel kural tabanlı sistemlerin aksine, YZ/ML algoritmaları, büyük veri kümelerini (sızdırılmış parolalar, kullanıcı davranışları, saldırı kalıpları) analiz ederek parolaların gücünü değerlendirebilir, zayıf noktaları belirleyebilir ve hatta saldırı vektörlerini tahmin edebilir.[5] Bu modeller, parolanın uzunluğu, karakter çeşitliliği, tekrar eden desenler, klavye yürüyüşleri ve sözlük kelimeleri gibi özellikleri çıkararak parolanın kırılma olasılığını tahmin eder.[14] Karar Ağaçları ve Yığınlanmış Modeller gibi algoritmalar, güçlü ve zayıf parolaları sınıflandırmada yüksek doğruluk göstermiştir.[23]

YZ/ML, geleneksel yöntemlerle tespit edilemeyen karmaşık ve gizli kalıpları ve zayıf noktaları ortaya çıkarabilir. Saldırganların da YZ kullanarak parola tahmin yeteneklerini artırdığı bir ortamda [4], savunmacıların da YZ'yi kullanması bir "siber güvenlik silahlanma yarışı"nın parçasıdır.[6] Bu, parola güvenlik araçlarının daha akıllı, adaptif ve proaktif olmasını sağlar. YZ, %60'tan fazla ihlalin kimlik bilgisi sorunlarından kaynaklandığı bir ortamda, işletmelerin güçlü parola politikaları uygulamalarını zorunlu kılmaktadır.[4]

Python aracı, parola gücü analizi için Scikit-learn [26] gibi kütüphaneler kullanarak ML modelleri entegre edebilir. Bu, parolanın basit kural setlerinin ötesinde, gerçek dünya saldırı verilerine dayalı olarak daha doğru bir "güvenlik puanı" almasını sağlar. Örneğin, kullanıcının parolasının bilinen bir saldırı kalıbına uyup uymadığını veya bir YZ tahmin aracı tarafından kolayca tahmin edilip edilemeyeceğini değerlendirebilir.

Parola gücü kontrolüne yönelik geleneksel yaklaşım, minimum uzunluk, karakter türleri ve yaygın sözlük kelimelerinden kaçınma gibi kriterlere odaklanan kural tabanlıydı.[7] Ancak, araştırma materyalleri, YZ/ML modellerinin [5] "sızdırılmış parolaların geniş veri kümelerini" ve "kullanıcı davranışlarını" analiz ederek "zayıf parolalarla ilişkili kalıpları ayırt edebildiğini ve yeni oluşturulan parolaların gücünü tahmin edebildiğini" açıkça

vurgulamaktadır. Bu durum,

kuralcı yaklaşımdan (bir parolanın *ne içermesi gerektiği*) *öngörücü* analize (bir parolanın kırılma veya tahmin edilme olasılığı) doğru önemli bir sıçramayı temsil etmektedir. Dahası, araştırma materyalleri, saldırganların artık çevrimiçi davranışlara dayalı olarak son derece doğru parola tahminleri oluşturmak için YZ'den yararlandığını vurgulamaktadır.[4] Bu durum, savunma amaçlı YZ'nin saldırı amaçlı YZ'ye karşı koymak için kritik olduğu bir "siber güvenlik silahlanma yarışı" yaratmaktadır.[6] Bu nedenle, Python aracının parola değerlendirme mantığı, parolanın yalnızca mevcut güvenlik standartlarına göre değil, aynı zamanda gelecekteki YZ destekli saldırı vektörlerine karşı da dirençliliğini değerlendirmelidir. Bu, aracın statik bir denetleyici olmaktan çıkıp, dinamik bir risk değerlendirme ve tahmin motoru haline gelmesini sağlar. Aracın, saldırganların kullandığı yöntemleri taklit ederek (örneğin, bir YZ parola tahmincisi gibi davranarak) parolanın zayıflığını test etmesi, kullanıcılara daha gerçekçi ve eyleme geçirilebilir geri bildirimler sunabilir.

2.6. Oturum İçi Anomali Tespiti için Yapay Zeka ve Makine Öğrenimi

YZ/ML, ağ trafiğini ve kullanıcı davranışını sürekli olarak izleyerek normal aktivite desenlerinden sapmaları tespit edebilir.[3] Bu, bir kullanıcının tipik giriş zamanları, cihaz türleri, coğrafi konumları ve erişim sıklığı gibi verileri analiz ederek bir "normal davranış taban çizgisi" oluşturur.[5] Olağandışı bir giriş denemesi bu kalıptan saptığında, YZ sistemleri bunu şüpheli olarak işaretleyebilir ve güvenlik protokollerini tetikleyebilir.[5] Bu, çalınan parolaların kullanıldığı durumlarda bile gerçek zamanlı kimlik doğrulama ve tehdit tespiti sağlar.

Bu teknik, parola tabanlı saldırıların (kimlik bilgisi doldurma, kaba kuvvet) başarılı olması durumunda bile, oturum içi anomali tespiti ek bir güvenlik katmanı sunar. Saldırganın ele geçirilmiş kimlik bilgileriyle sisteme girmesi durumunda dahi, alışılmadık davranışlar (örn. farklı konumdan giriş, olağandışı erişim kalıpları) tespit edilerek erişim kısıtlanabilir veya ek doğrulama istenebilir.[5] Bu, geleneksel parola kontrollerinin ötesinde sürekli bir koruma sağlar.

Python aracı, kullanıcının parolasını kontrol ederken, bu parolanın oturum içi anomali tespiti mekanizmalarıyla nasıl desteklenebileceğine dair öneriler sunabilir. Doğrudan bir oturum içi izleme aracı olmasa da, bu tür sistemlerin önemini vurgulayarak kullanıcının genel güvenlik duruşunu iyileştirmesine yardımcı olabilir. Özellikle,

parolanın kendisi zayıf olsa bile (ancak diğer tekniklerle güçlendirilmişse), anomali tespiti ek bir koruma sağlayabilir.

Parola gücü denetleyicileri parola oluşturma veya ilk giriş noktasına odaklanırken, "oturum içi anomali tespiti" [3] güvenliği ilk oturum açma olayının ötesine taşımaktadır. Bu durum, güvenliğin tek seferlik bir geçit kontrol süreci değil, kullanıcının oturumu boyunca sürekli, devam eden bir doğrulama olduğunu ifade etmektedir. Temel varsayım, ilk kimlik doğrulama (parola, MFA) başarılı olsa bile, bir oturumun yine de tehlikeye atılabileceğidir (örn. oturum ele geçirme yoluyla veya meşru kullanıcının cihazının ele geçirilmesi durumunda). Bu durum, kullanıcı davranışının sapmalar açısından sürekli izlendiği "sürekli kimlik doğrulama" kavramına doğrudan yol açmaktadır.[21] Bu nedenle, Python aracının parola önerileri, parolanın sadece ilk giriş anındaki gücünü değil, aynı zamanda sürekli kimlik doğrulama bağlamında nasıl bir rol oynadığını da ele almalıdır. Araç, kullanıcıya parolanın bir başlangıç noktası olduğunu ve gerçek zamanlı davranış analizinin neden sürekli bir güvenlik katmanı olarak gerekli olduğunu anlatmalıdır. Bu, parolanın "güçlü" olmasının ötesinde, "oturum boyunca güvenli" olmasının önemini vurgulamaktadır.

2.7. Davranışsal Biyometrikler ile Sürekli Kimlik Doğrulama

Davranışsal biyometrikler, kullanıcıların cihazlarla etkileşimlerindeki benzersiz kalıpları (tuş vuruşu dinamikleri, fare hareketleri, dokunmatik ekran kaydırmaları, yürüyüş şekli, ses tanıma) analiz ederek kimlik doğrulama sağlar.[21] Fiziksel biyometriklerin aksine, davranışsal biyometrikler sürekli ve gerçek zamanlı aktiviteyi analiz eder, bu da onları devam eden kullanım için daha uyarlanabilir kılar.[21] YZ ve ML entegrasyonuyla, sistemler zamanla kullanıcı davranışındaki değişikliklere uyum sağlayabilir ve dinamik güvenlik profilleri oluşturabilir.[21] Bu, oturum boyunca kimliğin sürekli olarak doğrulanmasını sağlar.

Parolaların veya PIN'lerin aksine, davranışsal özellikler benzersizdir ve dolandırıcılar tarafından taklit edilmesi neredeyse imkansızdır.[27] Kimlik bilgilerinin çalınması durumunda bile, dolandırıcıların meşru kullanıcıları taklit etmesini zorlaştırır.[27] Sürekli kimlik doğrulama sağlayarak oturum ele geçirme riskini azaltır ve gerçek zamanlı dolandırıcılık tespitine olanak tanır.[21] Ayrıca, özel donanım gerektirmemesi ve pasif izleme özelliği sayesinde kullanıcı deneyimini kesintiye uğratmaz.[27]

Python aracı, doğrudan davranışsal biyometrikleri uygulayabilir. pynput gibi

kütüphanelerle temel tuş vuruşu dinamiklerini kaydedebilir ve kullanıcıların benzersiz yazma ritimlerini profilleyebilir.[32] Bu veriler, kullanıcının parolasını girerkenki davranışını analiz etmek ve potansiyel anormallikleri (örn. bot tarafından giriş, farklı bir kişinin girişi) tespit etmek için kullanılabilir. Bu, parolanın kendisi doğru olsa bile, girişin meşru bir kullanıcıdan gelip gelmediğini anlamak için ek bir doğrulama katmanı sunar.

Geleneksel kimlik doğrulama yöntemleri (parolalar, hatta fiziksel biyometrikler) genellikle oturum açma noktasında tekil, aktif bir doğrulama adımı içerir. Davranışsal biyometrikler [21] ise "sürekli kimlik doğrulama" ve "pasif izleme" kavramlarını tanıtarak bu durumu temelden değiştirmektedir. Bu, güvenlik kontrollerinin sürekli olarak arka planda, açık bir kullanıcı müdahalesi gerektirmeden veya kullanıcı deneyimini kesintiye uğratmadan gerçekleştiği anlamına gelir. Bu yaklaşım, sık ve aktif kimlik doğrulama istemleriyle ilişkili kullanıcı yorgunluğunu ve sürtünmeyi doğrudan ele alırken, aynı zamanda oturum boyunca anormallikleri tespit ederek güvenliği artırmaktadır. Buradaki temel düşünce, güvenliği "görünmez" ancak her yerde mevcut kılmaktır. Bu nedenle, Python aracının parola güvenlik önerileri, sadece başlangıçtaki parola gücünü değil, aynı zamanda kullanıcının oturum boyunca nasıl sürekli olarak doğrulanabileceğini de içermelidir. Araç, davranışsal biyometriklerin parolanın kendisini tamamlayarak nasıl daha sürtünmesiz ve güçlü bir güvenlik deneyimi sağlayabileceğini açıklayabilir. Bu durum, "güvenliğin görünmez olması" ve "kullanıcı deneyimini önceliklendiren güvenlik" trendini yansıtmaktadır.

2.8. Uyarlanabilir Kimlik Doğrulama Politikaları

Uyarlanabilir kimlik doğrulama (risk tabanlı kimlik doğrulama olarak da bilinir), kullanıcı davranışına ve bağlama göre kimlik doğrulama gereksinimlerini dinamik olarak ayarlayan bir güvenlik yaklaşımıdır.[28] Konumsal faktörler (konum, cihaz, zaman), kullanıcı davranış kalıpları ve geçmiş erişimler gibi bağlamsal sinyalleri sürekli bir risk değerlendirme sürecinin parçası olarak kullanır.[28] Düşük riskli senaryolarda (örn. bilinen bir cihazdan ve konumdan giriş), sistem daha az doğrulama isteyebilir (örn. sadece parola). Ancak yüksek riskli senaryolarda (örn. yeni bir konumdan veya şüpheli bir cihazdan giriş), ek doğrulama (örn. MFA) veya erişim engelleme talep edebilir.[28] Makine öğrenimi, kullanıcı davranış kalıplarını analiz ederek ve anormallikleri tespit ederek bu değerlendirmeleri geliştirir.[28]

Bu yaklaşım, güvenlik ile kullanıcı deneyimi arasında kritik bir denge kurar. Düşük riskli durumlarda gereksiz sürtünmeyi en aza indirirken, yüksek riskli durumlarda güvenliği

artırır.[28] Kimlik tabanlı saldırıları gerçek zamanlı olarak tespit edip durdurarak, güvenliği önemli ölçüde artırır ve MFA yorgunluğunu azaltır.[28] Dağıtık iş gücü ve uzaktan çalışma ortamları için özellikle faydalıdır.[28]

Python aracı, kullanıcının parolasının gücünü değerlendirirken, bu parolanın uyarlanabilir kimlik doğrulama sistemleri bağlamında nasıl kullanılabileceğini açıklayabilir. Örneğin, "Bu parola, uyarlanabilir bir sistemde düşük riskli bir giriş için yeterli olabilir, ancak yüksek riskli bir senaryoda ek doğrulama gerektirecektir" gibi geri bildirimler sağlayabilir. Araç, pyotp gibi kütüphanelerle MFA entegrasyonu için temel sağlayarak [20], uyarlanabilir politikaların bir parçası olarak MFA'nın nasıl tetiklenebileceğini gösterebilir.

Geleneksel kimlik doğrulama sistemleri ikili bir mantıkla çalışır: erişim yalnızca kimlik bilgilerine göre verilir veya reddedilir.[28] Ancak uyarlanabilir kimlik doğrulama, gelişmiş bir

risk değerlendirme katmanı sunmaktadır.[28] Bu durum, güvenliğin artık sabit bir kural kümesi değil, bir kimlik doğrulama girişiminin belirli bağlamına göre uyarlanmış dinamik bir yanıt olduğu bir paradigma değişimini ifade etmektedir. Bu yaklaşım, bir parolanın "gücünün" mutlak olmadığını, ancak konum, cihaz, zaman ve kullanıcı davranışı gibi çevresel koşullara büyük ölçüde bağlı olduğunu açıkça kabul etmektedir.[28] Bu, güvenliğe daha olgun ve incelikli bir yaklaşımı temsil etmekte ve tek tip politikalardan uzaklaşmaktadır. Bu nedenle, Python aracının parola güvenlik analizi, parolanın "mutlak" gücünün yanı sıra, farklı bağlamsal senaryolarda (örn. bilinen cihazdan/konumdan giriş ile yeni/şüpheli cihazdan/konumdan giriş) nasıl bir risk profili oluşturduğunu da değerlendirmelidir. Araç, kullanıcılara parolanın gücünün, kullanıldığı ortamın riskine göre değişebileceğini ve uyarlanabilir politikaların bu dinamizmi nasıl yönettiğini anlatmalıdır. Bu durum, aracın yalnızca bir "güç denetleyicisi" olmaktan çıkıp, bir "risk danışmanı" rolüne bürünme potansiyelini göstermektedir.

2.9. Gelişmiş Parola Yöneticisi Entegrasyonu ve Kullanımı

Parola yöneticileri, karmaşık ve benzersiz parolaları güvenli bir şekilde depolayan ve oluşturan araçlardır.[1] NIST 2025 yönergeleri, parola yöneticilerinin kullanımını şiddetle teşvik etmektedir.[8] Bu araçlar, kullanıcıların her hesap için farklı, uzun ve güçlü parolalar oluşturmasına ve hatırlamasına yardımcı olur, böylece parola tekrar kullanımını ve zayıf parola seçimini önler.[1] Güvenilir parola yöneticileri, bağımsız

inceleme siteleri ve kullanıcı tavsiyeleri aracılığıyla bulunabilir.[7]

İnsanların güçlü parolalar oluşturma ve hatırlama konusunda iyi olmadığı kanıtlanmıştır.[7] Parola yöneticileri, bu insan faktörü zayıflığını ortadan kaldırarak, kullanıcıların güvenlikten ödün vermeden çok sayıda karmaşık parolayı yönetmesini sağlar.[7] Veri ihlallerinin büyük bir kısmının kimlik bilgisi sorunlarından kaynaklandığı göz önüne alındığında, parola yöneticileri bu riski azaltmada kritik bir rol oynar.

Python aracı, kullanıcıya güçlü parolalar önerirken, bu parolaların bir parola yöneticisinde nasıl güvenli bir şekilde saklanabileceğini ve yönetilebileceğini vurgulamalıdır. Araç, kullanıcılara parola yöneticisi kullanmanın genel güvenlik duruşlarını nasıl iyileştireceğini açıklayabilir. Ayrıca, aracın kendisi, parola yöneticileriyle entegre olabilecek veya onlara benzer parola oluşturma ve depolama mekanizmalarını (ancak tam teşekküllü bir parola yöneticisi değil) örnekleyebilecek özellikler sunabilir.

Araştırma materyalleri, parola güvenliğindeki temel bir gerilimi sürekli olarak vurgulamaktadır: güçlü parolaların insanlar tarafından hatırlanması genellikle zordur.[7] Bu durum, parolaların yeniden kullanılması, zayıf parolaların seçilmesi veya not alınması gibi istenmeyen kullanıcı davranışlarına yol açmaktadır.[1] Parola yöneticileri, bu boşluğu etkili bir şekilde kapatan birincil teknolojik çözüm olarak sunulmaktadır.[1] Bu araçlar, kullanıcıların ezberleme yükü olmadan yüksek güvenlik (uzun, karmaşık, benzersiz parolalar) elde etmelerini sağlamaktadır. Bu durum, etkili güvenlik çözümlerinin yaygın benimsenmeyi ve dolayısıyla gerçek dünya güvenlik iyileştirmelerini sağlamak için doğası gereği kullanıcı dostu olması gerektiği yönündeki daha geniş bir eğilimi işaret etmektedir. Siber güvenlikte "insan faktörünü" yönetmekle ilgilidir. Bu nedenle, Python aracının parola öneri mekanizması, sadece parolanın teknik gücünü değil, aynı zamanda parola yöneticileri aracılığıyla yönetilebilirliğini de dikkate almalıdır. Araç, kullanıcıya "Bu parola güçlüdür ve bir parola yöneticisiyle kolayca yönetilebilir" gibi geri bildirimler sunarak, güvenlik ve kullanılabilirlik arasındaki dengeyi vurgulamalıdır. Bu durum, aracın sadece bir teknik denetleyici değil, aynı zamanda kullanıcıların güvenlik alışkanlıklarını iyileştirmeye yardımcı olan bir rehber rolü üstlenmesi gerektiğini göstermektedir.

2.10. Gerçek Zamanlı Parola Paylaşımı ve Maruz Kalma Tespiti

Bu teknik, şirket içi iletişim araçları (Slack, Microsoft Teams gibi) ve diğer işbirliği

platformlarındaki gerçek zamanlı konuşmalarda parola paylaşımını veya hassas kimlik bilgilerinin maruz kalmasını tespit etmeyi içerir.[36] Mimecast gibi çözümler, endüstri lideri Doğal Dil İşleme (NLP) ve Makine Öğrenimi (ML) modellerini kullanarak bu platformlardaki mesajları sürekli olarak analiz eder ve parola paylaşımı gibi politika ihlallerini gerçek zamanlı olarak tanımlar.[36] Bu, şirket içi veri sızıntılarını ve saldırı yüzeyini azaltmaya yardımcı olur.

Çalışanların işbirliği araçlarında parolaları paylaşma eğilimi, kuruluşlar için önemli bir güvenlik açığı oluşturur.[36] Tek bir sızdırılmış parola bile, bir markanın fikri mülkiyetine, itibarına ve pazar payına felaketle sonuçlanan zararlar verebilir.[36] Bu tür proaktif tespit, ihlalleri önler ve düzenleyici uyumluluğu sürdürmeye yardımcı olur. Bu, teknik kontrollerin ötesinde insan davranışının da güvenlik için kritik olduğunu göstermektedir.

Python aracı doğrudan bir kurumsal izleme çözümü olmasa da, parola güvenliği eğitim ve farkındalık modülleri sunabilir. Kullanıcılara, parolaların asla paylaşılmaması gerektiğini ve işbirliği platformlarında hassas bilgilerin nasıl korunacağını hatırlatabilir. Aracın kendisi, kullanıcının girdiği parolanın bilinen bir "paylaşım kalıbına" uyup uymadığını (örn. "şirketadı123") kontrol etmek için ML modellerini kullanabilir.

Birçok trend harici tehditlere ve teknik parola özelliklerine odaklanırken, bu trend [36] doğrudan kritik bir dahili güvenlik açığını ele almaktadır: insan davranışı, özellikle de işbirliği araçları içinde parola paylaşımının riskli uygulaması. Bu durum, en sağlam teknik güvenlik önlemleri olsa bile, insan hatası, ihmal veya farkındalık eksikliğinin önemli güvenlik boşlukları yaratabileceğini vurgulamaktadır. Çözüm, sadece teknik tespiti değil, aynı zamanda bir "güvenlik kültürü"nü teşvik etmeyi de içermektedir.[36] Bu, parola güvenliğinin kapsamını sadece teknik doğrulamadan, iç tehditleri veya kazara ifşaları azaltmayı amaçlayan organizasyonel politika uygulamasına ve kullanıcı eğitimine kadar genişletmektedir. Bu nedenle, Python aracının parola güvenlik önerileri, sadece teknik parametreleri değil, aynı zamanda kullanıcıların parola kullanım alışkanlıklarını ve güvenlik farkındalığını da hedeflemelidir. Araç, kullanıcılara parola paylaşımının risklerini ve bunun neden kaçınılması gerektiğini açıkça belirtmelidir. Bu durum, aracın bir "teknik denetleyici" olmaktan çıkıp, bir "güvenlik danışmanı" rolüne evrilme potansiyelini göstermektedir, zira en güçlü parola bile yanlış kullanıldığında zayıf hale gelmektedir.

3. Sonuç ve Python Tabanlı Araç İçin Öneriler

2025 yılı ve sonrası için parola güvenliği, statik kural setlerinden dinamik, uyarlanabilir ve kullanıcı merkezli yaklaşımlara doğru önemli bir evrim geçirmektedir. NIST yönergeleri, uzunluk ve parola ifadelerine odaklanarak karmaşıklıktan uzaklaşmakta, zorunlu parola değişikliklerini kaldırmaktadır. İhlal edilmiş kimlik bilgisi taraması ve kara listeleme, bilinen zayıflıkları proaktif olarak engellemektedir. Çok faktörlü kimlik doğrulama (özellikle kimlik avına dayanıklı yöntemler) ve parolasız kimlik doğrulama (geçiş anahtarları gibi) parolaya olan bağımlılığı azaltarak güvenlik katmanlarını artırmaktadır. Yapay zeka ve makine öğrenimi, parola gücü analizi, oturum içi anomali tespiti ve davranışsal biyometrikler aracılığıyla tehditleri öngörme ve bunlara karşı koymada merkezi bir rol oynamaktadır. Uyarlanabilir kimlik doğrulama, risk bağlamına göre güvenlik gereksinimlerini ayarlarken, parola yöneticileri kullanıcı deneyimini ve genel güvenliği iyileştirmektedir. Son olarak, gerçek zamanlı parola paylaşımı tespiti, insan faktöründen kaynaklanan riskleri azaltmaktadır. Bu trendler, hem teknik sağlamlığı hem de kullanıcı dostu yaklaşımları bir araya getirerek daha dirençli bir dijital güvenlik ortamı oluşturmayı hedeflemektedir.

Python tabanlı parola güvenlik aracının 2025 ve ötesinde etkili olabilmesi için aşağıdaki geliştirme yönleri önerilmektedir:

1. **NIST Uyumluluğu ve Akıllı Öneriler:** Aracın parola gücü değerlendirme mantığını NIST 2025 yönergeleriyle (uzunluk odaklı, parola ifadesi desteği, Unicode karakter desteği) tamamen uyumlu hale getirilmelidir. Karmaşıklık kurallarını zorlamak yerine, uzun ve hatırlanabilir parola ifadeleri önermeye odaklanılmalıdır. Aracın önerileri, kullanıcının hatırlayabileceği ve benimseyebileceği parolalar sunarak kullanıcı davranışını olumlu yönde etkilemelidir.
2. **İhlal Edilmiş Parola Taraması Entegrasyonu:** Kullanıcının girdiği parolayı Have I Been Pwned gibi bilinen ihlal veri tabanlarına karşı gerçek zamanlı olarak kontrol etmek için pwnedpasswords gibi Python kütüphaneleri entegre edilmelidir. Bu, kullanıcılara parolalarının daha önce sızdırılıp sızdırılmadığına dair anında ve eyleme geçirilebilir geri bildirim sağlayacaktır.
3. **Yapay Zeka Destekli Parola Gücü Analizi:** Geleneksel kural tabanlı kontrollerin ötesine geçerek, parolanın kırılma olasılığını tahmin etmek için makine öğrenimi modelleri (örn. Scikit-learn ile Karar Ağaçları veya Yığınlanmış Modeller) kullanılmalıdır. Bu modeller, parolanın uzunluğu, karakter çeşitliliği, tekrar eden desenler, klavye yürüyüşleri ve sözlük kelimeleri gibi özellikleri analiz ederek daha doğru bir güvenlik puanı sağlayabilir.
4. **Parolasız Kimlik Doğrulama Bilgilendirmesi:** Araç, doğrudan parolasız kimlik doğrulama sağlamasa da, kullanıcılara geçiş anahtarları ve FIDO standartları gibi parolasız çözümlerin geleceği hakkında bilgi vermelidir. Parola önerileri sunarken, bu önerilerin daha ileri kimlik doğrulama yöntemlerine geçiş için bir basamak

olabileceği vurgulanmalıdır.

5. **Davranışsal Biyometrik Analiz Potansiyeli:** Parola giriş anındaki kullanıcı davranışını (örn. tuş vuruşu dinamikleri) analiz etmek için pynput gibi kütüphanelerle temel davranışsal biyometrik özellikler entegre edilebilir. Bu, parolanın kendisi doğru olsa bile, girişin meşru bir kullanıcıdan gelip gelmediğini anlamak için ek bir doğrulama katmanı sunacaktır.
6. **Uyarlanabilir Kimlik Doğrulama Bağlamı:** Parolanın gücünü değerlendirirken, farklı bağlamsal senaryolarda (örn. bilinen cihazdan/konumdan giriş ile yeni/şüpheli cihazdan/konumdan giriş) nasıl bir risk profili oluşturduğunu açıklanmalıdır. Araç, kullanıcılara parolanın gücünün, kullanıldığı ortamın riskine göre değişebileceğini ve uyarlanabilir politikaların bu dinamizmi nasıl yönettiğini anlatmalıdır.
7. **Parola Yöneticisi Kullanımını Teşvik:** Güçlü parolalar önerirken, bu parolaların bir parola yöneticisinde nasıl güvenli bir şekilde saklanabileceği ve yönetilebileceği vurgulanmalıdır. Parola yöneticilerinin, kullanıcıların güvenlikten ödün vermeden karmaşık parolaları yönetmelerine nasıl yardımcı olduğunu açıklanmalıdır.
8. **Kullanıcı Farkındalığı ve Eğitim Modülleri:** Parola paylaşımının riskleri ve işbirliği araçlarında hassas bilgilerin nasıl korunacağı gibi konuları içeren kısa eğitim modülleri veya hatırlatıcılar eklenmelidir. Bu, insan faktöründen kaynaklanan güvenlik açıklarını azaltmaya yardımcı olacaktır.
9. **Gelişmiş Parola Önerileri:** Sadece güçlü parolalar değil, aynı zamanda "parola ifadeleri" (passphrases) gibi hatırlanması kolay, ancak uzun ve rastgele kelimelerden oluşan kombinasyonlar önerilmelidir. Bu, kullanıcıların hem güvenli hem de kullanılabilir parolalar oluşturmasına yardımcı olacaktır.
10. **Geri Bildirim ve İyileştirme Mekanizmaları:** Araç, kullanıcının girdiği parolanın neden zayıf olduğunu veya güçlü bir önerinin neden tercih edilmesi gerektiğini açık ve anlaşılır bir dille açıklamalıdır. Bu, kullanıcıların güvenlik bilincini artıracak ve daha iyi parola alışkanlıkları geliştirmelerine yardımcı olacaktır.

Alıntılanan çalışmalar

1. Strong Password Best Practices For 2025 | Cyble, erişim tarihi Temmuz 9, 2025, https://cyble.com/knowledge-hub/strong-password-best-practices-2025/
2. NIST Password Guidelines: 2025 Updates & Best Practices - StrongDM, erişim tarihi Temmuz 9, 2025, https://www.strongdm.com/blog/nist-password-guidelines
3. The Future of AI Data Security: Trends to Watch in 2025 - CyberProof, erişim tarihi Temmuz 9, 2025, https://www.cyberproof.com/blog/the-future-of-ai-data-security-trends-to-watch-in-2025/
4. Password-guessing AI: how to defend against it - NordPass, erişim tarihi Temmuz 9, 2025, https://nordpass.com/blog/password-guesser-ai/

5. AI in Password Security: Predicting and Preventing Credential- Based Attacks, erişim tarihi Temmuz 9, 2025, https://www.researchgate.net/publication/388525778_AI_in_Password_Security_Predicting_and_Preventing_Credential-_Based_Attacks
6. AI arms race: How AI will be used by cyber-attackers (and defenders) - Specops Software, erişim tarihi Temmuz 9, 2025, https://specopssoft.com/blog/ai-in-cybersecurity-arms-race-attackers-defenders/
7. The State of Password Security 2025 Report - Bitwarden, erişim tarihi Temmuz 9, 2025, https://bitwarden.com/resources/the-state-of-password-security/
8. 2025 NIST Password Guidelines: Enhancing Security Practices - Scytale, erişim tarihi Temmuz 9, 2025, https://scytale.ai/resources/2024-nist-password-guidelines-enhancing-security-practices/
9. The New Rules for Strong Passwords in 2025 - GadellNet, erişim tarihi Temmuz 9, 2025, https://gadellnet.com/the-new-rules-for-strong-passwords-in-2025/
10. What are the NIST password guidelines in 2025? - TrustCommunity, erişim tarihi Temmuz 9, 2025, https://community.trustcloud.ai/article/nist-password-guidelines-2025-15-rules-to-follow/
11. The Complete Guide to NIST Password Guidelines (2025 Update ..., erişim tarihi Temmuz 9, 2025, https://drata.com/blog/nist-password-guidelines
12. 2025 Multi-Factor Authentication (MFA) Statistics & Trends to Know - JumpCloud, erişim tarihi Temmuz 9, 2025, https://jumpcloud.com/blog/multi-factor-authentication-statistics
13. lionheart/pwnedpasswords: A Python Library and CLI for the Pwned Passwords v2 API, erişim tarihi Temmuz 9, 2025, https://github.com/lionheart/pwnedpasswords
14. password-strength - PyPI, erişim tarihi Temmuz 9, 2025, https://pypi.org/project/password-strength/
15. Building a Password Strength Checker in Python | by ryan - Medium, erişim tarihi Temmuz 9, 2025, https://medium.com/@ryan_forrester_/building-a-password-strength-checker-in-python-6f723d20511d
16. What is Multi-Factor Authentication (MFA)? - SentinelOne, erişim tarihi Temmuz 9, 2025, https://www.sentinelone.com/cybersecurity-101/identity-security/what-is-multi-factor-authentication-mfa/
17. 2025 Passwordless Identity Assurance Statistics Released - Cyber Security Tribe, erişim tarihi Temmuz 9, 2025, https://www.cybersecuritytribe.com/cyber-security-announcements/2025-passwordless-identity-assurance-statistics-released
18. Passkeys: Passwordless Authentication - FIDO Alliance, erişim tarihi Temmuz 9, 2025, https://fidoalliance.org/passkeys/
19. Passwordless Authentication Adoption Trends in 2025 - JumpCloud, erişim tarihi

Temmuz 9, 2025, https://jumpcloud.com/blog/passwordless-authentication-adoption-trends
20. pyauth/pyotp: Python One-Time Password Library - GitHub, erişim tarihi Temmuz 9, 2025, https://github.com/pyauth/pyotp
21. Behavioural biometrics seen as key layer, not replacement for passwords - Security Brief UK, erişim tarihi Temmuz 9, 2025, https://securitybrief.co.uk/story/behavioural-biometrics-seen-as-key-layer-not-replacement-for-passwords
22. Expert Insights Podcast: #64 – Passwordless Authentication and the Rise of Passkeys, erişim tarihi Temmuz 9, 2025, https://fidoalliance.org/expert-insights-podcast-64-passwordless-authentication-and-the-rise-of-passkeys/
23. Password Strength Detection via Machine Learning: Analysis, Modeling, and Evaluation, erişim tarihi Temmuz 9, 2025, https://arxiv.org/html/2505.16439v1
24. Ankit152/Password-Strength-Classifier: A Machine Learning model that predicts whether the password is weak, medium or strong. - GitHub, erişim tarihi Temmuz 9, 2025, https://github.com/Ankit152/Password-Strength-Classifier
25. [Literature Review] Password Strength Detection via Machine Learning: Analysis, Modeling, and Evaluation - Moonlight | AI Colleague for Research Papers, erişim tarihi Temmuz 9, 2025, https://www.themoonlight.io/en/review/password-strength-detection-via-machine-learning-analysis-modeling-and-evaluation
26. Software for Data Analysis: Python Libraries - Research Guides, erişim tarihi Temmuz 9, 2025, https://guides.lib.uci.edu/dataanalysis/pythonlibraries
27. Behavioral Biometrics: The Passwordless Future of Access Control - MojoAuth, erişim tarihi Temmuz 9, 2025, https://mojoauth.com/ciam-101/behavioral-biometrics-access-control
28. What is Adaptive Authentication? - CrowdStrike.com, erişim tarihi Temmuz 9, 2025, https://www.crowdstrike.com/en-us/cybersecurity-101/identity-protection/adaptive-authentication/
29. What is Adaptive Authentication and Authorization? - 1Kosmos, erişim tarihi Temmuz 9, 2025, https://www.1kosmos.com/authentication/adaptive-authentication/
30. You Might Never Need to Change Your Password Again - Newsweek, erişim tarihi Temmuz 9, 2025, https://www.newsweek.com/passwords-ai-cybersecurity-biometrics-2094604
31. What is Behavioral Biometrics? - IBM, erişim tarihi Temmuz 9, 2025, https://www.ibm.com/think/topics/behavioral-biometrics
32. Coding Intelligence: Building a Behavioural Biometrics Engine in Python | by Aghede David, erişim tarihi Temmuz 9, 2025, https://medium.com/@davidaghede598/coding-intelligence-building-a-behavioural-biometrics-engine-in-python-a0f12d4f6748
33. A Lightweight Behavioral Biometric Framework using Python and Flask for Continuous Authentication in Online Banking - ResearchGate, erişim tarihi

Temmuz 9, 2025, https://www.researchgate.net/publication/391460812_A_Lightweight_Behavioral_Biometric_Framework_using_Python_and_Flask_for_Continuous_Authentication_in_Online_Banking
34. Adaptive authentication rules settings in a policy - SecureAuth Product Docs, erişim tarihi Temmuz 9, 2025, https://docs.secureauth.com/2104/en/adaptive-authentication-rules-settings-in-a-policy.html
35. June 2025 Security Tips - University of Rochester, erişim tarihi Temmuz 9, 2025, https://tech.rochester.edu/june-2025-security-tips/
36. Real-time Password Detection In Chat Conversations - Mimecast, erişim tarihi Temmuz 9, 2025, https://www.mimecast.com/use-cases/password-detection/