1章
ペネトレーションテストの基本

ペネトレーションテストとは、攻撃者がセキュリティコントロールを回避し、組織のシステムへアクセスするために使う可能性のある手法をシミュレートする方法である。これは単にスキャナや自動化ツールを使い、報告書を作成するだけにとどまらない。また一晩ではペネトレーションテスターの専門家になれない。達人になるには、何年もの修練と実務経験が必要である。

現在セキュリティ業界内で、ペネトレーションテストの評価と定義の方法に変化が現れている。Penetration Testing Execution Standard（PTES）は、ペネトレーションテストを再定義しており、これは初心者および熟練したペネトレーションテスターの両方に影響を与え、またセキュリティコミュニティの主要メンバーにも受け入れられている。その憲章は、ペネトレーションテストの実行に求められる基本原則の確立により、ペネトレーションテストの真の意味を定義し、それに対する認識を高めるためのものだ。ペネトレーションテストを初めて知った、あるいはPTESに馴染みがない場合は、http://www.pentest-standard.org/でさらに詳細を学んでほしい。

1.1 PTESのフェーズ

PTESのフェーズは、ペネトレーションテストを定義するために設計されたものであり、またクライアントである組織に対し、この種の評価を誰が行った場合でも、費やされる努力の基準を保証するものである。基準は7つのカテゴリーに分類され、各カテゴリーには異なる努力レベルがあり、これは攻撃を受ける組織がどこまで求めるかによる。

1.1.1 事前契約のやり取り

事前契約のやり取りは通常、クライアントとペネトレーションテストの範囲および条件について話し合う際に行われる。事前契約では契約の目標を伝えるのが重要だ。またこの段階は、全範囲に及ぶ完全なペネトレーションテストに何が期待できるかを顧客に説明する機会でもある。つまり「何のテストができ、何がテストされるか」の制限がないテストについて、説明するということだ。

1.1.2 インテリジェンスギャザリング

インテリジェンスギャザリング段階では、攻撃する組織について、ソーシャルメディアネットワーク、Googleハッキング、ターゲットのフットプリンティングなどにより、ありとあらゆる情報を収集する。ペネトレーションテスターにとって最も重要なスキルの1つが、ターゲットの行動、運用方法、最終的にはどう攻撃できるかなど、ターゲットについて知る能力である。ターゲットについて集めた情報は、設置されているセキュリティコントロールの種類について、貴重な見解を提供してくれる。

インテリジェンスギャザリングでは、ターゲットのシステムをゆっくりと探るうちに、そこに置かれた保護メカニズムが何であるか識別を試みることになる。たとえばある組織が、外向きのデバイスのポートのあるサブセットにおけるトラフィックのみを許可している場合、ホワイトリストに登録されたポート以外を検索するとブロックされる。まずブロックまたは検出されてもいい使い捨てのIPアドレスから探ることで、このようなブロックに関する挙動をテストするのは、いいアイディアである。一定のしきい値を超えると、Webアプリケーションファイアウォールが、それ以上の要求をブロックするような環境に置かれているWebアプリケーションをテストする場合でも同様のことが言える。

この種のテストにおいて攻撃を検出されないようにするには、テスターまたはテスターのチームへとリンクしていないIPアドレスからスキャンを実行する。通常、インターネット上に存在するシステムは日々攻撃を経験しているため、最初の調査ではバックグラウンドノイズとみなされ、検出されないだろう。

主要攻撃に用いるIPアドレスとはまったく範囲の異なるIPアドレスから、かなり派手なスキャンを実行するほうが、意味がある場合もある。この場合、使用するツールに対して、組織がどれだけ正しく対応するかを見る手助けになる。

1.1.3 脅威モデリング

脅威モデリングは、インテリジェンスギャザリングで収集した情報を用いて、ターゲットとするシステム上の既存の脆弱性を識別する。脅威モデリングを実行する際、最も効果的な攻撃方法、探している情報の種類、組織をどう攻撃するかを判断する。脅威モデリングでは組織を敵として見て、攻撃者と同じように弱点を見つけ出す。

1.1.4 脆弱性分析

最も実行可能性の高い攻撃方法が見つかったら、ターゲットへのアクセス方法を考えなければならない。**脆弱性分析**では、先の段階で集めた情報を組み合わせ、どの攻撃の実行可能性が高いかを理解する。特に脆弱性分析は、ポートスキャンと脆弱性スキャン、バナー収集によって得られた結果や、インテリジェンスギャザリングで収集した情報を考慮する。

1.1.5 エクスプロイト

エクスプロイトは、おそらくペネトレーションテスト最大の見せ場の1つだが、精密さよりもブルートフォースで行われる場合が多い。エクスプロイトは成功がほぼ疑う余地のない場合に限り、実行すべきである。もちろん、そのエクスプロイトの作動を防ぐための、予定外の保護措置をターゲットが取っている場合もあるが、脆弱性を発現させる前にシステムが脆弱であるかどうかを把握すべきである。むやみにエクスプロイトの猛攻を仕掛け、シェルに期待するのは生産的ではない。わずらわしいだけで、ペネトレーションテスターにもクライアントにも何の利益ももたらさない。まずは下調べをし、それから成功の確率が高いよくリサーチしたエクスプロイトを実行しよう。

1.1.6 ポストエクスプロイト

ポストエクスプロイトフェーズは、1つまたは複数のシステムの侵害後に始まるが、まだその段階にはほど遠い。

ポストエクスプロイトは、どんなペネトレーションテストにおいても重要である。この段階こそが、平均的なありふれたハッカーと読者とを差別化し、ペネトレーションテストから得た貴重な情報とインテリジェンスを実際に提供する場である。ポストエクスプロイトは、特定のシステムをターゲットとし、重要なインフラを特定して、企業が守ろうとしている最も重要な情報またはデータを狙う。次々にシステムをエクスプロイトする場合、攻撃がビジネスに最大の影響を与えるのを実演することになる。

ポストエクスプロイトでシステムを攻撃するとき、さまざまなシステムが何をしているのか、またそれぞれのユーザーの役割が何かを判断する時間を取るべきである。たとえばドメインインフラストラクチャシステムを侵害して、エンタープライズ管理者としてシステムを動かす、あるいはドメインの管理者レベルの権限を取得したとしよう。ドメインは掌握したが、Active Directoryとやり取りするシステムはどうだろう。従業員に給与を支払うのに用いられる、主要な財務アプリケーションはどうか。システムを侵害し、次の給料日の際、すべてのお金を会社から引き出して、海外口座へ移すことができるだろうか。知的財産はどうだろうか。

仮にクライアントが大手ソフトウェア開発企業で、製造業務で使用するカスタムのアプリケーションを顧客に出荷しているとしよう。ソースコードにバックドアを組み込み、顧客全員にダメージを与えられるだろうか。どうすればブランドの信頼を失墜させられるだろうか。

ポストエクスプロイトはこうした微妙なシナリオの1つであり、どんな情報が利用できるのか、またその情報を使って何のメリットがあるのかを時間をかけて調べなければならない。攻撃者は通常、侵害したシステム上で同様に膨大な時間を費やしている。悪意ある攻撃者のように思考しよう。クリエイティブになり、迅速に行動し、自動化ツールではなく自分の感覚に頼ろう。

1.1.7 報告書の作成

報告書の作成は、ペネトレーションテストにおける最も重要な要素である。レポーティングによって

自分が行ったことやその手段を伝えるのは当然だが、何より大切なのは、ペネトレーションテスト中に見つかった脆弱性を企業がどう修正すべきかを伝えることである。

ペネトレーションテストを実行するときは、組織がまず考えない、攻撃者としての視点から作業する。テストで取得した情報は、企業の情報セキュリティプログラムの成功にとっても、また将来の攻撃を防ぐうえでも重要である。発見事項を集めて報告する際に、単に技術的な脆弱性を部分的に修正するのではなく、見つかった内容を企業がどのように利用して認識を高め、発見した問題を修正し、全体のセキュリティを向上させられるかについて考えてほしい。

最低限、報告書を要旨、経営層向けプレゼンテーション、技術的知見に分けよう。技術的知見は、クライアントがセキュリティホールを修正するのに利用するが、これはまたペネトレーションテストの重要なポイントでもある。たとえば、クライアントのWebベースのアプリケーションでSQLインジェクションの脆弱性を見つけた場合、クライアントに対し、すべてのユーザー入力をサニタイズし、パラメータ化されたSQLクエリを利用し、権限に制限のあるユーザーアカウントでSQLを実行し、カスタムのエラーメッセージを生成することを推奨するだろう。

クライアントが推奨を受け入れて、ある特定のSQLインジェクションの脆弱性を修正したら、SQLインジェクションから本当に守られていることになるだろうか。そうではない。サードパーティアプリケーションの安全性確保に失敗しているといった根本的な問題が、SQLインジェクション脆弱性を引き起こしている可能性があるからだ。こうした問題も当然修正される必要がある。

1.2 ペネトレーションテストのタイプ

PTESの7つのカテゴリーを基本的に理解したところで、ペネトレーションテストの主な2つのタイプである「overt（公開）」と「covert（秘密）」を見てみよう。公開ペネトレーションテスト、つまり「ホワイトハット」テストは、企業側が完全に理解している状況で行われる。一方秘密のテストは、未知の、未発表の攻撃者の行動をシミュレートする設計となっている。どちらのテストにも長所と短所がある。

1.2.1 公開ペネトレーションテスト

公開ペネトレーションテストでは、作業する企業とともに潜在的なセキュリティ脅威を見出し、企業のITまたはセキュリティチームに、企業システムを案内してもらうことになる。公開テストの主な長所は、社内情報にアクセスできるのと、ブロックされる心配なしに攻撃を仕掛けられることである。短所となるのは、クライアントの事故対応プログラムを効果的にテストできない、あるいはセキュリティプログラムが特定の攻撃をどれだけ検出できるかが判断できない点だろう。時間が限られていて、インテリジェンスギャザリングなどの一部のPTESステップを想定しない場合、公開テストが最良の選択肢となる。

1.2.2 秘密ペネトレーションテスト

公開テストと異なり、承認された秘密ペネトレーションテストは、攻撃者の行動を模倣して設計さ

れ、企業が知らないうちに実行される。秘密テストは、社内のセキュリティチームの検出能力と、攻撃への対応能力を試すために行われる。

秘密テストにはお金も時間もかかり、公開テストよりも高度なスキルが求められる。セキュリティ業界のペネトレーションテスターから見れば、本物の攻撃をよりそっくりにシミュレートしているために、秘密テストのほうが好まれる場合が多い。秘密攻撃は、予備調査での情報収集能力にかかっている。したがって秘密テスターとしては、一般にターゲットの脆弱性を大量に見つけようとするのではなく、検出されずにシステムにアクセスする最も簡単な方法を見つけることになる。

1.3 脆弱性スキャナ

脆弱性スキャナは、任意のシステムまたはアプリケーションに影響を与えるセキュリティ上の欠陥を見出すために使われる、自動化ツールである。脆弱性スキャナは通常、ターゲットで実行されているサービスと、オペレーティングシステムの**フィンガープリンティング**（つまりバージョンとタイプの識別）を行う。OSのフィンガープリンティングを実行したら、脆弱性スキャナを使って、脆弱性が存在するかどうかを判断するチェックを行う。こうしたチェック能力はその開発者の能力の範囲と等しいため、どんな自動化ソリューションであっても、システムの脆弱性を見逃したり、誤って伝えたりすることがある。

最新の脆弱性スキャナは、誤検出を最小限に抑えたすばらしい性能を発揮しており、多くの組織は時代遅れのシステムを発見したり、攻撃者にエクスプロイトされる可能性のある新たな問題を検出したりするのに役立てている。

脆弱性スキャナはペネトレーションテストで非常に重要な役割を果たしており、特に公開テストでは、検出回避を気にせずに複数の攻撃を仕掛けることができる。脆弱性スキャナで収集した豊富な知識は非常に貴重なものだが、それに依存しすぎないよう注意してほしい。優れたペネトレーションテストは自動化できず、システムへの攻撃を成功させるには、知識とスキルが必要となる。熟練したペネトレーションテスターになると、脆弱性スキャナはほとんど使わず、システムを侵害するための自分の知識と経験に頼ることが多くなる。

1.4 まとめ

ペネトレーションテストが初めて、あるいは正式な方法論を用いたことがない場合、PTESを勉強しよう。どんな実験でもそうだが、ペネトレーションテストを実行する場合、正確かつ柔軟で、繰り返し利用できるプロセスを用いるようにしてほしい。ペネトレーションテスターとして、インテリジェンスギャザリングと脆弱性分析を可能な限り熟練させ、シナリオを見せられたときにそれにすぐ順応し、優位に立てるようにしよう。