**1. 가장 값진 데이터를 파악한다**
가장 먼저 할 일은 가장 값진 데이터를 파악하는 것이다. 이런 데이터를 찾기가 쉽지 않다. 수년 전, 한 CIO는 "데이터가 위치한 모든 장소를 알고 있다고 말하는 것은 '헛소리'"라고 말했다.

데이터베이스, 애플리케이션 데이터 저장소, 클라우드, 백업 미디어와 제거 가능한 미디어에 귀중한 데이터가 저장되어 있다. 실제 데이터를 전달하고 안전하게 만드는 핵심 하위시스템도 포함된다. 모두 개별 보안 방어 체계가 구현되어 있는 액티브 디렉터리, 도메인 컨트롤러, 크리덴셜(Credential) 데이터베이스, DNS, DHCP 네트워크 라우터, 기타 서비스를 예로 들 수 있다.

비즈니스 가치와 민감도를 기준으로 데이터를 범주화해야 한다. 그래야만 핵심 데이터를 가능한 작게 유지할 수 있다. 최소한의 필수 데이터만 저장해야 한다. 애초에 저장하지 않는 것이 가장 안전하기 때문이다.

또 모든 데이터에 상태와 처리 방법, 유효성에 관한 문제를 해결할 오너(책임자)가 있어야 한다. 사용 기간을 표시하고, 사용 기간이 끝나면 폐기 처분해야 한다.

**2. 크리덴셜을 정리한다**
크리덴셜의 '위생'에 만전을 기해야 한다. 영구 계정(permanent membership)을 '0'에 가깝기 만들기 위해 특수(관리자) 계정을 정리해야 한다는 의미다. 관리자는 최소한의 권한(적절하게 충분한 권한)으로 업무를 처리해야 한다. 실제 필요한 때만 권한을 승인해야 한다. 참고로, 이를 'JIT(Just in Time)' 권한이라고 부른다.

각 기업과 기관은 각 권한 그룹의 영구 계정을 검토하고, 영구적인 풀타임 액세스가 필요없는 계정을 없애야 한다. 적절하면서도 엄격한 분석을 통해 영구 계정을 줄일 수 있다. 0~1개의 영구 계정만 유지하는 것이 최상이다.

대부분의 관리자에게 제한된 범위에서 특수 권한이나 관리자 권한을 할당해야 한다. 사전에 만료 시한을 정한 후, 관리자의 관리자 크리덴셜을 없애는 방법으로 이를 처리할 수 있다.

크리덴셜 '위생'은 데이터베이스를 튼튼하게 보호하는데 아주 중요하다. 매번 그런 것은 아니지만 공격자는 자주 비밀 데이터에 액세스하려는 목적에서 특수(관리자) 계정을 공략하기 때문이다. 영구적인 특수(관리자) 계정을 최소화하면, 이들 계정이 해킹되어 악용되는 위험을 낮출 수 있다.

**3. 엄격하게 내부 보안 경계선을 규정한다**
네트워크 방화벽을 이용한 보안이면 충분하다고 생각했던 시절은 지나갔다. 대부분 기업 네트워크의 내외부를 구분하고, 보안 경계선을 따로 만들어야 한다.

사전에 액세스할 수 있는 계정을 규정해야 한다. 호스트 기반 방화벽, 내부 라우터, VLAN, 논리 네트워크, VPN, IPSec, 기타 여러 액세스 관리 도구를 이용해 내부 보안 경계선을 엄격하게 규정할 수 있다. 예를 들어, 절대 다수의 사용자가 다중 계층 애플리케이션의 웹 프론트 엔드에 액세스 할 수 있다. 그러나 백엔드 데이터베이스에 직접 액세스 할 수 있는 사용자는 극소수로 제한해야 한다.

지정된 데이터베이스 관리자, 소수의 지원 서버 및 사용자만 데이터베이스 서버와 프론트 엔드 웹 데이터 베이스, 다중 계층 서비스에 액세스 할 수 있어야 한다. 이렇게 하면 공격자 필요한 크리덴셜 없이 데이터베이스에 직접 액세스하는 것을 막거나, 최소한 감사 필요성을 경고할 수 있다.

**4. 데이터를 항상 암호화 상태로 유지한다**
기존에는 2가지 형태의 암호화를 이용했다. 전송 시 데이터 암호화, 휴지 상태의 데이터 암호화다. 여기에는 전제 조건이 필요하다. 범죄자가 데이터에 액세스 할 수 있는 적법한 크리덴셜을 훔치지 않았어야 한다. 그런데 훔친 경우가 많다.

데이터를 튼튼하게 보호하고 싶다면, 데이터를 위치에 상관 없이 항상 암호화된 상태로 유지해야 한다. 특히 승인되지 않은 장소로 이동된 데이터 암호화가 중요하다. 데이터 유출은 심각한 문제다.

각 데이터를 암호화, 이동 장소에 상관 없이 암호화 상태를 유지하도록 도와주는 솔루션이 많다. 일부는 애플리케이션 서비스로 MADRMS(Microsoft’s Active Directory Rights Management Service)를 예로 들 수 있다. 또한 마이크로소프트 SQL Transparent Data Encryption 등 데이터베이스 내부의 데이터 권한을 암호화하는 솔루션도 있다.

가장 효과적인 데이터 암호화 방법은 뭘까? 누군가 데이터를 훔쳐도, 암호화를 유지해 무용지물로 만드는 방법이다.

**5. 클라이언트를 보호한다**
해커들이 서버에 직접 침투하는 경우는 드물다. 그렇지만 이런 일이 일어난다. SQL 주입 공격과 원격 버퍼 오버플로우 공격을 예로 들 수 있다. 한편, 클라이언트 공격은 훨씬 많다.

데이터를 보호하고 싶다면, 데이터에 액세스하는 사람들을 보호해야 한다. 따라서 1~2주 이내에 중요 패치를 모두 배포해 적용하고, 사용자에게 소셜 엔지니어링을 교육하고, 워크스테이션을 안전하게 구성해야 한다.