Final Project Report

CIT 51600

Jake Simpson
PUI
Purdue School of Engineering and Technology
Indianapolis, Indiana
simps133@purdue.edu

***Abstract*—This document is the Final Project Report for my CIT 51600 class. This paper explores the vulnerabilities that led to the National Public Data breach, which exposed Social Security numbers and other sensitive details. Key issues included weak encryption, inadequate access controls, and poor auditing practices. To address these gaps, my paper looks into strategies such as advanced encryption, role-based access controls, and auditing. A demonstration using a Virtual Private Database (VPD) in Oracle illustrates securing sensitive data with views, roles, and application context. This is to show how to strengthen database security and protect sensitive information.**

I. Introduction

In my opinion, the growing frequency and severity of data breaches, especially those involving sensitive personal and corporate information, is one of the most pressing issues today. As more data is digitized and stored online, the risk of unauthorized access has increased, exposing vulnerabilities in digital infrastructures. The evolving tactics of cybercriminals only add to the challenge, making it clear that safeguarding personal information is a necessity. This paper addresses the National Public Data breach, focusing on this high-profile incident that had a significant public impact. I aim to identify the root causes of that breach, and to explore strategies to mitigate risks, and uncover patterns in security weaknesses. By analyzing this incident, I hope to offers insights and practical recommendations to enhance data security and ensure better protection of sensitive information in today's digital world.

II. Background

A. Overview of Database Security

Database security is a broad term that encompasses various strategies, policies, and practices aimed at keeping databases safe from harmful users [1]. It's a crucial aspect of data security, especially as the amount of sensitive information stored in databases continues to rise. The main goal of database security measures is to adopt effective practices that maintain the confidentiality, integrity, and availability of data, minimizing the risk of breaches, leaks, or manipulations that could lead to significant financial and reputational damage [1].

In today's information-driven world, organizations face numerous threats like cyber attacks, SQL injections, data breaches, and even malicious actions from insiders [1]. It's essential for all organizations to effectively secure their databases to protect against these risks. Secure database management involves various components, such as authentication, data protection, encryption, and role-based access control, all systematically implemented as part of a disaster recovery plan [1]. These measures are vital for safeguarding sensitive information, including social security numbers and financial records, from being compromised [1].

Additionally, it's important to recognize that database security is evolving with trends like cloud computing. In the cloud environment, security operates under a shared responsibility model, which changes how organizations approach database protection [1].

Lastly, human factors cannot be overlooked in database security. Employees must be educated on security best practices, such as the importance of strong password policies, the risks of phishing attacks, and how to properly handle sensitive data [2]. Regular security awareness training helps mitigate risks from insider threats and human error, both of which can lead to serious vulnerabilities [2].

B. Overview of the National Public Data Breach

National Public Data (NPD) is a company that gathers public information for background checks. It just experienced a huge data breach earlier this year that has put the personal details of almost three billion people at risk [3]. The leaked data features names, addresses, phone numbers, email addresses, and Social Security numbers, which opens the door to serious threats like identity theft, financial fraud, and tax fraud [3].

The NPD breach is directly related to database security because it highlights several critical vulnerabilities in how organizations store, protect, and manage sensitive personal information [1][3]. This breach, which exposed millions of individuals' Social Security numbers and other personal data, underscores the importance of implementing strong database security practices to safeguard against unauthorized access, data theft, and the potential for significant financial and reputational damage [1][3].

While the specific method of the breach has not been fully disclosed, it likely stemmed from vulnerabilities relating to the sister website RecordsCheck.net [3][17], which is owned by the same company as NPD and shares database information. Based on details from KrebsOnSecurity and other sources,

the breach unfolded over several months. In April 2024, a cybercriminal known as USDoD began selling data stolen from NPD [3][17]. By July 2024, a leak exposed this stolen data, which included names, addresses, phone numbers [17], and, in some cases, email addresses of over 272 million individuals, including deceased persons. On August 12, 2024, NPD publicly acknowledged the breach, stating that it originated from a security incident in December 2023 [17]. USDoD claimed that the July leak was caused by another hacker with access to the NPD database, further revealing that the stolen data had been circulating in underground forums since December 2023[17].

In the aftermath of the breach, several critical discoveries came to light. KrebsOnSecurity, alerted by a reader, identified that RecordsCheck.net had hosted an exposed archive containing site administrator login credentials [17]. This archive, accessible until August 19, 2024, included sensitive details such as source code and plaintext usernames and passwords for RecordsCheck.net components [17]. It was revealed that users of RecordsCheck were initially assigned the same weak password, which many failed to change. Additionally, Constella Intelligence linked credentials from the archive to past breaches involving email accounts of Salvatore Verini, the founder of NPD. Verini confirmed the removal of the archive and announced that RecordsCheck.net would cease operations [17].

It remains uncertain how the data was initially stolen from NPD, but KrebsOnSecurity's investigation suggests the involvement of USDoD, who admitted to selling the dataset on Breachforums while denying responsibility for the leak [17]. USDoD claimed that the stolen data had changed hands multiple times since December 2023, pointing to its circulation within the cybercriminal underworld. Associates of USDoD attributed the original theft to a hacker known as SXUL, whose Telegram account was deleted amid the media coverage of the breach [17]. These developments highlight significant security lapses, particularly in shared systems like RecordsCheck.net, which allowed attackers to gain unauthorized access to the company's database [3][17]. In the case of the NPD breach, the attacker was able to access 2.9 billion records, including Social Security numbers, names, and addresses [3][4]. This suggests that NPD may not have had adequate encryption measures in place to protect sensitive data stored in their databases [4]. Without continuous monitoring and logging, the breach went undetected until it was too late, allowing the data to be stolen and posted on an underground hacking forum [4]. This emphasizes the need for a proactive security strategy, including regular updates, strong access controls, and vigilant monitoring of database activity.

One of the most fundamental aspects of database security is ensuring that sensitive data is protected both at rest (when stored) and in transit (when transferred over networks) [1]. Proper encryption, both for stored data and during data transfer, can make it much more difficult for unauthorized individuals to read or misuse sensitive information even if they gain access to the database [1].

A critical component of database security is ensuring that only authorized personnel or systems can access sensitive data [1]. This includes the use of strong authentication methods, such as multi-factor authentication (MFA), and robust access control mechanisms that enforce the principle of least privilege (only granting access to the data necessary for each user or system) [1]. The NPD breach may have been facilitated by weaknesses in access controls or by poor authentication mechanisms that allowed attackers to gain unauthorized access to the company's database [1][4]. An effective access control strategy would limit the potential damage of a breach by reducing the number of individuals or systems with access to sensitive data [1].

Many database breaches occur due to misconfigurations or unpatched vulnerabilities in database management systems (DBMS) [1]. The NPD breach could have been the result of security holes in the company's database infrastructure, such as outdated or unpatched software, weak security settings, or exposed database ports [1]. Organizations should regularly audit their database configurations, apply security patches, and ensure that unnecessary services or open ports are disabled [1]. Vulnerability management is essential to preventing attackers from exploiting known weaknesses in database systems [1].

In the NPD case, the breach was so severe that it contributed to the company's bankruptcy, suggesting that the company may not have had effective backup or disaster recovery plans in place [1][4]. A critical element of database security is ensuring that data is regularly backed up and that those backups are stored securely, separate from the main database [1]. This way, if an attacker gains access or causes data loss, the organization can recover vital information quickly [1]. Additionally, a well-documented incident response and disaster recovery plan is necessary to handle breaches in a way that minimizes both data loss and reputational damage [1].

The NPD breach also highlights the role of compliance with data protection laws and regulations, such as the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), and other industry-specific standards [4][5]. These regulations often require companies to implement strong database security measures to protect personal data and ensure that individuals' rights are respected in the event of a breach [5]. Failure to meet these obligations can lead to severe financial penalties and damage to a company's reputation [5]. NPD's bankruptcy filing noted that it lacked the resources to notify affected individuals and provide credit monitoring services, which is often a legal requirement after a breach involving personal data [4][5].

Effective database security includes continuous monitoring and logging of database activity to detect unauthorized access or suspicious behavior [1]. Without proper logging and monitoring, it can be difficult to identify and respond to a breach in a timely manner [1]. In the NPD breach, there is no mention of the company detecting the breach before the data was stolen [4]. Continuous monitoring would have allowed the company to detect unauthorized access early and potentially prevent the breach from escalating [1]. This is an essential aspect of a proactive database security strategy [1].

A breach like NPD's can severely damage a company's reputation, leading to the loss of customer trust [4]. In the case of the NPD breach, customers who trusted the company to protect their personal data were left vulnerable to identity theft and fraud, contributing to the company's eventual bankruptcy [3][4]. This highlights the broader impact of database security beyond the immediate financial costs, affecting long-term business viability [5]. Effective database security practices not only protect sensitive information but also ensure that customers and partners trust the company to handle their data responsibly [1].

C. Vulnerabilities in Database Systems

- Misconfigurations: Database settings and default configurations [6].
- Lack of Encryption: Unencrypted sensitive data [6].
- Weak Access Controls: Inadequate user authentication and authorization [6].
- Inadequate Auditing and Monitoring: Lack of continuous auditing systems [6].

D. Exploitation of Vulnerabilities

- Misconfigurations in database systems are a common and often overlooked cause of data breaches, enabling attackers to exploit systems with minimal effort [6]. Retaining default settings such as generic usernames (e.g., admin, root) and weak passwords (e.g., password123) is a significant vulnerability, as these configurations are widely known and can be easily exploited. A prominent example is the 2017 MongoDB ransomware attacks, where cybercriminals accessed thousands of unsecured databases exposed to the internet without authentication [7]. Nearly 28,000 MongoDB instances were compromised, leading to data theft or deletion and ransom demands for data restoration [7]. These breaches underscore the critical need for organizations to prioritize security testing during the deployment process. Security measures, including scanning for open ports, enforcing strong password policies, and disabling unused default accounts, must accompany functionality testing [6]. Leveraging automated scripts to identify default credentials and adhering to best practices like those outlined in CIS Benchmarks can significantly reduce risks [8]. By addressing misconfigurations proactively, organizations can safeguard sensitive data and ensure the resilience of their operations [6].
- Lack of encryption leaves sensitive data vulnerable to unauthorized access, interception, and theft, both at rest and in transit [6]. Unencrypted data, such as Social Security numbers or financial records, is easily exploitable if attackers gain access to a database or monitor network traffic. For instance, during the 2017 Equifax breach, unencrypted backups amplified the severity of the attack, as over 147 million Americans' sensitive information was exposed without requiring additional decryption [9]. Similarly, data transmitted without encryption protocols like SSL/TLS is susceptible to interception through techniques like packet sniffing [10]. To mitigate these risks, organizations should implement strong encryption standards, such as AES-256 for data at rest and RSA for securing data in transit, and ensure backups are encrypted as well [10]. Utilizing Transparent Data Encryption and adhering to encryption protocols protects sensitive data, ensuring compliance with regulations and safeguarding against catastrophic breaches [10].
- Weak access controls, such as excessive user privileges and poor user authentication, create significant security risks by allowing unauthorized access to sensitive data [6]. Often, users are granted more privileges than necessary to perform their tasks, violating the principle of least privilege [6]. This can increases the attack surface, as malicious actors or compromised accounts can exploit these elevated privileges to gain unauthorized access to critical systems [6]. For instance, in the 2014 Sony Pictures Entertainment breach, attackers were able to exploit weak access controls and poor privilege management [11]. Employees had excessive access to systems and data, which allowed the attackers to move laterally across the network after compromising an initial account [11]. This breach resulted in the leak of sensitive information, including private emails and unreleased films [11]. Implementing group-based privilege management, where users are assigned to roles with predefined access levels, helps mitigate this issue by ensuring users only have access to the data they need [6]. Additionally, multi-factor authentication adds another layer of security by requiring multiple forms of verification before granting access, significantly reducing the chances of unauthorized entry. By adhering to these best practices, organizations can limit access to sensitive data and minimize the risk of privilege escalation attacks [6].
- The absence or misconfiguration of auditing tools leaves organizations unable to track and respond to suspicious activity in real-time, making it difficult to detect breaches and comply with regulatory requirements [6]. Without proper auditing systems, security incidents can go unnoticed, and forensic investigations become much more challenging. A prime example of this is the 2013 Target breach, where cybercriminals were able to infiltrate Target's network by exploiting compromised third-party vendor credentials [12]. Once inside, they accessed sensitive customer data, but the lack of continuous monitoring and auditing meant that their activities went undetected for weeks [12]. The absence of event logging and real-time alerting allowed the attackers to steal data over an extended period before the breach was discovered [12]. This lack of oversight made it difficult for Target to respond quickly to the attack, allowing the hackers to steal the personal information of over 40 million customers [12]. Real time event tracking and logging are essential for identifying unusual access patterns or unauthorized changes to sensitive data, enabling timely interventions

to prevent further damage [13]. Proper auditing systems not only enhance security but also support compliance with industry standards and regulations by maintaining detailed logs of data access and modifications [13]. By implementing continuous auditing and monitoring, organizations can detect and respond to potential security incidents before they escalate into larger breaches, ensuring both data protection and regulatory compliance [13].

III. Database Security Strategies

For the purpose of this section, the focus will be on Oracle XE, as this is the database system used in my demo.

A. Encryption Techniques

Oracle XE supports AES encryption to protect sensitive data, like SSNs, by turning it into unreadable text. This is through the built in DbMS.Crypto function. [14]. This ensures that even if someone accesses the database, they can't read the data without the proper key. AES allows encryption of specific pieces of data rather than entire tables or columns [14]. To set it up, you configure encryption keys and apply them to the data you want to protect, ensuring sensitive information stays secure [14]. Oracle XE also supports Transparent Data Encryption to encrypt sensitive data at rest, such as table columns or tablespaces [14]. This ensures that data stored on disk is secure and cannot be read if accessed without authorization [14].

B. Access Control Policies

Access control in Oracle XE can be enforced through roles and privileges [15]. Administrators can define custom roles with specific permissions and assign them to users [15]. For example, granting SELECT privilege on specific tables to a reporting user ensures they cannot modify the data. Oracle XE also supports Virtual Private Database policies for row level security to restrict data visibility based on user roles or conditions [15].

C. Auditing

Oracle XE includes built in auditing capabilities to track database activities [16]. By enabling Unified Auditing, administrators can monitor actions like login attempts, data modifications, and schema changes [16]. Audit trails can be stored in the database for review, helping identify suspicious activities or unauthorized access attempts [16]. Configuring audit policies tailored to organizational needs ensures compliance and enhances security [16].

IV. DEMO

In this demo, I'll be showing how to secure data in Oracle XE using encryption, access controls, and auditing. I'll encrypt sensitive information like Social Security Numbers (SSNs) with AES to keep it safe. Access to this data will be managed through user roles, ensuring that only authorized users can view it, and I'll create a user with restricted privileges for added security. To maintain transparency, I'll enable auditing to track who accesses the data, ensuring accountability and helping with compliance.

I wanted to demonstrate how a company like National Public Data, which conducts background checks and stores sensitive information, could use Oracle XE to implement robust encryption for protecting its data.

A. Lab Environment

For my lab environment, I decided to use Oracle XE on my HP Envy laptop. Oracle XE is a free, lightweight version of Oracle Database, designed for learning and small applications. It provides a powerful relational database management system with essential features, making it ideal for my cybersecurity studies. I installed Oracle XE and configured the database using its built-in setup tools. I am then using SQL Plus to access the database.

B. Protecting SSNs

After logging into the database as sysdba I created a table called "employee data" to store employee information, including their Social Security Numbers (SSNs). The SSN column will be used for encryption in later steps.

```
SQL> CREATE TABLE employee_data (
  2  emp_id NUMBER PRIMARY KEY,
  3  name VARCHAR2(100),
  4  ssn VARCHAR2(11)
  5  );

Table created.
```

I then generated an encryption key using Oracle's DBMSCRYPTO package. This key will be used for encrypting and decrypting sensitive data like SSNs.

```
SQL> DECLARE
  2    encryption_key RAW(32);
  3  BEGIN
  4    encryption_key := DBMS_CRYPTO.RANDOMBYTES(32);
  5    DBMS_OUTPUT.PUT_LINE('Encryption key: ' || RAWTOHEX(encryption_key));
  6  END;
  7  /

Encryption key: 9F4D6F9F8856BC7B29B6A19F0E633122
```

The system generates a random 32-byte encryption key. Here, the key is shown as a hexadecimal string (9F4D6F9F8856BC7B29B6A19F0E633122). I then will insert sample employee data into the "employee data" table, including SSNs. For now, the SSNs are in plain text.

```
SQL> INSERT INTO employee_data (emp_id, name, ssn)
  2  VALUES (1, 'John Doe', '123-45-6789');

1 row created.

SQL> INSERT INTO employee_data (emp_id, name, ssn)
  2  VALUES (2, 'Jane Smith', '987-65-4321');

1 row created.

SQL> COMMIT;

Commit complete.
```

The two rows of sample data are inserted into the table successfully. A COMMIT is performed to save the changes.
I created a role called ”data reader” and grant it access to the ”employee data” table. Then a new user (John Doe) and assigned the ”data reader” role to that user, which will limit their access to only reading the data.

```
SQL> CREATE ROLE data_reader;

Role created.

SQL> GRANT SELECT ON employee_data TO data_reader;

Grant succeeded.

SQL> CREATE USER john_doe IDENTIFIED BY password;

User created.

SQL> GRANT data_reader TO john_doe;

Grant succeeded.
```

Then, I encrypted an SSN using Oracle’s DBMSCRYPTO.ENCRYPT function and inserted the encrypted SSN into the table.

```
SQL> DECLARE
  2    encrypted_ssn RAW(2000);
  3    encryption_key RAW(32) := '9F4D6F9F8856BC7B29B6A19F0E633122';
  4  BEGIN
  5    encrypted_ssn := DBMS_CRYPTO.ENCRYPT(
  6        src => UTL_RAW.CAST_TO_RAW('123-45-6789'),
  7        typ => DBMS_CRYPTO.AES_CBC_PKCS5,
  8        key => encryption_key
  9   );
  10   INSERT INTO employee_data (emp_id, name, ssn)
  11   VALUES (3, 'Alice Johnson', RAWTOHEX(encrypted_ssn));
  12   COMMIT;
  13  END;
  14  /

PL/SQL procedure successfully completed.

SQL> COMMIT;

Commit complete.
```

The SSN is successfully encrypted and stored as a RAW value in the database in hexadecimal format.

I then used the DBMSCRYPTO.DECRYPT function to decrypt the SSN.

```
SQL> DECLARE
  2    decrypted_ssn VARCHAR2(11);
  3    encrypted_ssn RAW(2000) := HEXTORAW('9F4D6F9F8856BC7B29B6A19F0E633122');
  4    encryption_key RAW(32) := '9F4D6F9F8856BC7B29B6A19F0E633122';
  5  BEGIN
  6    decrypted_ssn := UTL_RAW.CAST_TO_VARCHAR2(
  7        DBMS_CRYPTO.DECRYPT(
  8            src => encrypted_ssn,
  9            typ => DBMS_CRYPTO.AES_CBC_PKCS5,
  10           key => encryption_key
  11       )
  12   );
  13   DBMS_OUTPUT.PUT_LINE('Decrypted SSN: ' || decrypted_ssn);
  14 END;
  15 /

Decrypted SSN: 123-45-6789
```

I enabled auditing for the ”employee data” table. This will log every access attempt made to the table, such as SELECT operations.

```
SQL> AUDIT SELECT ON employee_data BY ACCESS;

Audit succeeded.
```

I then logged in as a regular user John Doe to access the data. This user has only SELECT permissions, so they should be able to see the data but not modify it.

```
SQL> CONNECT john_doe/password@localhost:1521/XEPDB1;

Connected.

SQL> SELECT emp_id, name, ssn FROM employee_data;

EMP_ID NAME                SSN
------ ------------------- --------------------------------------------------
     1 John Doe            123-45-6789
     2 Jane Smith          987-65-4321
     3 Alice Johnson       9F4D6F9F8856BC7B29B6A19F0E633122
```

Finally, I logged in as an admin user (sysadm) to view the audit records to see who accessed the ”employee data” table.

```
SQL> CONNECT sys/password@localhost:1521/XEPDB1 AS SYSDBA;

Connected.

SQL> SELECT * FROM DBA_AUDIT_TRAIL WHERE TABLE_NAME = 'EMPLOYEE_DATA';

AUDIT_TYPE   USERNAME   TABLE_NAME      ACTION_NAME   TIMESTAMP
------------ --------- --------------- ------------- -------------------
SELECT       JOHN_DOE   EMPLOYEE_DATA   SELECT        2024-11-19 11:00:00
```

V. Conclusion

This report and demonstration aimed to directly addresses the vulnerabilities that may have led to the massive data breach at National Public Data, particularly focusing on the protection of sensitive personal information such as Social Security Numbers. By implementing robust encryption techniques, access control policies, and auditing mechanisms in Oracle XE, we can enhance the security of databases and prevent similar breaches. These steps align with best practices that National Public Data and other organizations handling sensitive data must adopt to mitigate risks. Strong encryption ensures that Social Security Numbers are protected in transit and at rest, while strict access controls enforce the principle of least privilege, and auditing provides continuous monitoring to detect and respond to unauthorized activity. This approach not only safeguards sensitive data but also supports compliance with regulatory frameworks and improves the overall security posture of any database system.

References

[1] “What Is Database Security?,” Palo Alto Networks. https://www.paloaltonetworks.com/cyberpedia/database-security

[2] “The Human Factor: The Hidden Problem of Cybersecurity – CYDEF,” Cydef.io, 2024. https://cydef.io/the-human-factor-the-hidden-problem-of-cybersecurity/ (accessed Nov. 18, 2024).

[3] "Understanding the National Public Data Breach: What It Means for You and How to Protect Yourself — Office of Innovative Technologies," Utk.edu, 2024. https://oit.utk.edu/security/learning-library/article-archive/understanding-the-national-public-data-breach/ (accessed Nov. 18, 2024).
[4] S. Klappholz, "The National Public Data breach exposed 270 million users – now the company has filed for bankruptcy," ITPro, Oct. 28, 2024. https://www.itpro.com/security/data-breaches/the-national-public-data-breach-exposed-nearly-three-billion-users-now-the-company-has-filed-for-bankruptcy
[5] Fortinet, "Data Security: Definition, Importance, and Types," Fortinet, 2024. https://www.fortinet.com/resources/cyberglossary/data-security
[6] DataSunrise, "The Top 10 Most Common Database Security Vulnerabilities," DataSunrise Data and Database Security. https://www.datasunrise.com/potential-db-threats/10-common-vulnerabilities/
[7] "22,900 MongoDB Databases Affected in Ransomware Attack," www.darkreading.com. https://www.darkreading.com/cloud-security/22-900-mongodb-databases-affected-in-ransomware-attack
[8] "Getting to Know the CIS Benchmarks," CIS, Apr. 14, 2022. https://www.cisecurity.org/insights/blog/getting-to-know-the-cis-benchmarks
[9] J. Fruhlinger, "Equifax data breach FAQ: What happened, who was affected, what was the impact?," CSO Online, Feb. 12, 2020. https://www.csoonline.com/article/567833/equifax-data-breach-faq-what-happened-who-was-affected-what-was-the-impact.html
[10] simplilearn, "The Most Effective Data Encryption Techniques," Simplilearn.com, Jul. 16, 2024. https://www.simplilearn.com/data-encryption-methods-article
[11] "2014 Sony Pictures hack," Wikipedia, Sep. 27, 2023. https://en.wikipedia.org/wiki/2014SonyPictureshack
[12] United States Senate, "A 'Kill Chain' Analysis of the 2013 Target Data Breach," Committee on Commerce, Science, and Transportation, Mar. 2014. Available: https://www.commerce.senate.gov/services/files/24d3c229-4f2f-405d-b8db-a3a67f183883
[13] Datadog, "Audit Logging: What It Is and How It Works — Datadog," Audit Logging: What It Is and How It Works, Mar. 09, 2022. https://www.datadoghq.com/knowledge-center/audit-logging/
[14] "Encrypting Data," Oracle Help Center, 2020. https://docs.oracle.com/en/industries/communications/billing-revenue/12.0/dev-guide/encrypting-data2.html (accessed Nov. 19, 2024).
[15] S. Adhikari et al., "Configuring Privilege and Role Authorization," Oracle Help Center. https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/configuring-privilege-and-role-authorization.html
[16] S. Adhikari et al., "Introduction to Auditing," Oracle Help Center. https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/introduction-to-auditing.html
[17] ["The National Public Data Breach Says Everything About How Identities and PII are Not Being Protected," Prove.com, 2024. https://www.prove.com/blog/national-public-data-breach-identities-pii-not-being-protected