once, and even though they are "friends," not all of those people will necessarily have our best interests at heart.

- *Unknown private data.* Users are sometimes unaware of what private data is exposed in social media web sites. For example, one tracking firm, Crimson Hexagon, advertises that it "analyzes the entire social internet (billions of blog posts, forum messages, Facebook posts, Tweets, etc.) by identifying statistical patterns in the words used to express opinions on different topics." That translates into their being able to identify you, not by name, perhaps, but to link comments you make on one site with your anonymous postings at another. Your virtual location (site you are visiting) and physical location (such as IP address or GPS coordinates) reveal where you are. Collecting, sharing, and analyzing these data helps commercial advertisers develop effective campaigns, but it also helps evildoers do the same.
- *"Erase" button absent.* Many people have embarrassing moments in their pasts. Fortunately, those moments are known to only a few people, and pieces of evidence (pictures, written descriptions) are often spotty at best. After the night passes or tempers cool, you can rethink and take steps to deal with the situation, perhaps by apologizing or asking a friend to destroy a photograph. With the Internet, words or pictures may have been forwarded beyond your control or even your friends' ability to recall. More significantly, Internet content can be picked up and cached indefinitely by search engines.
- *Application privacy violation.* Some Internet applications unintentionally or intentionally violate users' privacy. They collect and resell data without permission and sometimes even against stated privacy policies. Penalty for such behavior can be lax.
- *Aggregation and inference.* Collecting bits of information, for example, Facebook applications run, ads seen, and friends contacted, each of which may seem innocuous, can lead to the derivation of more information that a user might not want to release. One user might want to protect her home city, but if she has many friends from a single city, that strongly implies her home city.

We detail these threats throughout the chapter.

THREAT: DATA LEAKAGE

Of particular interest to companies and governments is the concept of data leakage. **Data leakage** is another term for inappropriate or unauthorized disclosure, occurring without the knowledge or consent of the person holding the data.

One way data items leak is by public observation: You forget that using a device in public means others can also intercept. One person will hear you on a cell phone, look at something on your computer screen while seated next to you, or see something on your smartphone. Many times the observer is uninterested (and, in fact, often wishes you would just shut up). Occasionally, however, the observer knows your subject and picks up valuable information, which then becomes a blog posting to the world or valuable insight for a competitor.

Are Tweets Protected Speech? Sidebar 18-2

In February 2011, reporter Dana Hedgpeth wrote in the *Washington Post* [HED11] that the U.S. government was attempting to obtain personal information from the Twitter accounts of three people linked to the WikiLeaks investigation. The government's lawyers requested screen names, mailing addresses, telephone numbers, bank account and credit card information, and IP addresses. However, the defendants' lawyers insisted that this information was protected by the First Amendment of the U.S. Constitution.

Although the case before the court addressed the WikiLeaks documents, this government request raises an important question: What data can the government seize from social networks? We have seen in this chapter that users have choices within an application about protecting the privacy of their data in a social network. But can the government override those settings? One of the defendants' lawyers noted that "the users' data would give the government a map of people tied to WikiLeaks and essentially halt free speech online" [HED11]. Government lawyers pointed out that this is a standard request, and that they didn't know if Twitter even collects all the data items requested.

One of the issues raised in this request is whether the current laws apply to Internet technology. "Experts say they were meant to deal with telephone records, not such evolving technology as e-mails and tweets." A lobbyist for the American Civil Liberties Union notes, "We're using tools for accessing information on e-mail, social networking sites that were never contemplated." In a January 28, 2011 blog, a Twitter representative explained the company position: "freedom of expression carries with it a mandate to protect our users' right to speak freely and preserve their ability to contest having their private information revealed."

Public and private personas have merged as the 24x7 work world intersects private lives. Taking a business phone call during a social event, sending an email message during timeout of a sports match, or drafting presentation slides while waiting in a doctor's waiting room are all ways by which work has moved outside the controlled work environment. Outside the office it is easy to forget that eyes and ears are everywhere.

A second way data can be leaked is by direct postings. Just as the line between work and private life has become blurred, so too has the line between corporate and private opinion, judgments, and knowledge. Personal opinions posted to the web can reveal corporate leanings and sensitive corporate data. Or they can be interpreted as reflecting a corporate opinion and not just that of the writer. As Sidebar 18-2 describes, even private messages can sometimes become public.

THREAT: INTRODUCTION OF MALICIOUS CODE

Facebook and other social media sites are excellent places for distribution of malicious code. These sites represent what appears to be a trusted environment with little basis for that trust. Friends might never do anything intentional to harm their friends, but that does not address possible unconscious, unknowing, or ill-advised acts. Precisely because of the interconnectedness of users at networking sites, there will be plenty of

sharing of photos, documents, videos, programs, apps, and other objects capable of containing and transmitting malicious code.

In previous chapters we detailed the technical nature and impact of malicious code. Our focus in this chapter is the nature of social media and the Internet in general that facilitates introducing malicious code.

ATTACK DETAILS: UNINTENDED DISCLOSURE

The disclosure threat in social networking involves several different concepts that come together. First, there is the issue of knowing what is sensitive and what you do not want to release. Second, there is the invisible wall of trust: Face-to-face you can get a sense of someone's reliability and decide at the moment how much to trust that person, but with online interaction you establish a trust relationship—which is really an access control decision, to use the computer security concept—and that relationship remains in effect until you decide or remember to change it.

Think of social network as establishing a big, vaguely structured, loosely connected database. This database contains thoughts, preferences, opinions, activities (or their descriptions), fantasies, friends, and connections. From this database people can draw inferences that may be accurate or false: Jamie is your friend. Jamie likes frogs. Ergo, you like frogs. Obviously, this is faulty logic, although it might also be true. In the next section we explore how people and computers analyze such databases for data connections that lead to unacceptable data disclosure.

Sensitive Data

Some databases contain what is called sensitive data. As a working definition, let us say that sensitive data are data that should not be made public. Determining which data items and fields are sensitive depends both on the individual database and the underlying meaning of the data. Obviously, some databases, such as a public library catalog, contain no sensitive data; other databases, such as defense-related ones, are wholly sensitive. These two cases—nothing sensitive and everything sensitive—are the easiest to handle, because they can be covered by access controls to the database as a whole. Someone either is or is not an authorized user. These controls can be provided by the operating system.

The more difficult problem, which is also the more interesting one, is the case in which *some but not all* of the elements in the database are sensitive. There may be varying degrees of sensitivity. For example, a university database might contain student data consisting of name, financial aid, dorm, drug use, sex, parking fines, and race. An example of this database is shown in Table 18-1. Name and dorm are probably the least sensitive; financial aid, parking fines, and drug use the most; sex and race somewhere in between. That is, many people may have legitimate access to name, some to sex and race, and relatively few to financial aid, parking fines, or drug use. Indeed, knowledge of the existence of some fields, such as drug use, may itself be sensitive. Thus, security concerns not only the data elements but their context and meaning.

TABLE 18-1 Example Database

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey

Furthermore, we must take into account different degrees of sensitivity. For instance, although they are all highly sensitive, the financial aid, parking fines, and drug-use fields may not have the same kinds of access restrictions. Our security requirements may demand that a few people be authorized to see each field, but no one be authorized to see all three. The challenge of the access control problem is to limit users' access so that they can obtain only the data to which they have legitimate access. Alternatively, the access control problem forces us to ensure that sensitive data are not released to unauthorized people.

Several factors can make data sensitive.

- *Inherently sensitive.* The value itself may be so revealing that it is sensitive. Examples are the locations of defensive missiles or the median income of barbers in a town with only one barber.
- *From a sensitive source.* The source of the data may indicate a need for confidentiality. An example is information from an informer whose identity would be compromised if the information were disclosed.
- *Declared sensitive.* The database administrator or the owner of the data may have declared the data to be sensitive. Examples are classified military data or the name of the anonymous donor of a piece of art.
- Part of a sensitive *attribute* or a sensitive *record.* In a database, an entire attribute or record may be classified as sensitive. Examples are the salary attribute of a personnel database or a record describing a secret space mission.
- Sensitive *in relation to previously disclosed information.* Some data become sensitive in the presence of other data. For example, the longitude coordinate of a secret gold mine reveals little, but the longitude coordinate in conjunction with the latitude coordinate pinpoints the mine.

All of these factors must be considered when the sensitivity of the data is being determined.

Types of Disclosures

We all know that some data are sensitive. However, sometimes even characteristics of the data are sensitive. In this section, we see that even descriptive information about data (such as their existence or whether they have an element that is nonzero) is a form of disclosure.

Exact Data

The most serious disclosure is the **exact value** of a sensitive data item itself. The user may know that sensitive data are being requested, or the user may request general data

without knowing that some of it is sensitive. A faulty database manager may even deliver sensitive data by accident, without the user having requested it. In all these cases, the result is the same: The security of the sensitive data has been breached.

Bounds

Another exposure is disclosing **bounds** on a sensitive value, that is, indicating that a sensitive value, y, is between two values, L and H. Sometimes, by using a narrowing technique not unlike the binary search, the user may first determine that $L \leq y \leq H$ and then see whether $L \leq y \leq H/2$, and so forth, thereby permitting the user to determine y to any desired precision. In another case, merely revealing that a value such as the athletic scholarship budget or the number of CIA agents exceeds a certain amount may be a serious breach of security.

Sometimes, however, bounds are a useful way to present sensitive data. It is common to release upper and lower bounds for data without identifying the specific records. For example, a company may announce that its salaries for programmers range from $50,000 to $82,000. If you are a programmer earning $79,700, you would suppose you are fairly well off, so you have the information you want; however, the announcement does not disclose who are the highest- and lowest-paid programmers.

Negative Result

Sometimes we can word a query to determine a **negative result**. That is, we can learn that z is *not* the value of y. For example, knowing that 0 is not the total number of felony convictions for a person reveals that the person was convicted of a felony. The distinction between 1 and 2 or 46 and 47 felonies is not as sensitive as the distinction between 0 and 1. Therefore, disclosing that a value is not 0 can be a significant disclosure. Similarly, if a student does not appear on the honors list, you can infer that the person's grade point average is below 3.50. This information is not too revealing, however, because the range of grade point averages from 0.0 to 3.49 is rather wide.

Existence

In some cases, the **existence** of data is itself a sensitive piece of data, regardless of the actual value. For example, an employer may not want employees to know that their telephone use is being monitored. In this case, discovering a NUMBER OF PERSONAL TELEPHONE CALLS field in a personnel file would reveal sensitive data.

Probable Value

Finally, it may be possible to determine the probability that a certain element has a certain value. To see how, suppose you want to find out whether the president of the United States is registered in the Tory party. Knowing that the president is in the database, you submit two queries to the database:

```
Count(Residence="1600 Pennsylvania Avenue") = 4
Count(Residence="1600 Pennsylvania Avenue" AND Tory=TRUE) = 1
```

From these queries you conclude there is a 25 percent likelihood that the president is a registered Tory.

Direct Inference

Inference is a way to infer or derive sensitive data from nonsensitive data. The inference problem is a subtle vulnerability in database security.

The database in Table 18-2 illustrates the inference problem; this database has the same form as the one introduced in Table 18-1, but we have added more data to make some points related to multiple data items. Recall that AID is the amount of financial aid a student is receiving. FINES is the amount of parking fines still owed. DRUGS is the result of a drug-use survey: 0 means never used and 3 means frequent user. Obviously this information should be kept confidential. We assume that AID, FINES, and DRUGS are sensitive fields, although only when the values are related to a specific individual. In this section, we look at ways to determine sensitive data values from the database.

Direct Attack

In a **direct attack**, a user tries to determine values of sensitive fields by seeking them directly with queries that yield few records. The most successful technique is to form a query so specific that it matches exactly one data item.

In Table 18-2, a sensitive query might be

```
List NAME where
    SEX=M ∧ DRUGS=1
```

TABLE 18-2 Database to Illustrate Inferences

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

This query discloses that for record ADAMS, DRUGS=1. However, it is an obvious attack because it selects people for whom DRUGS=1, and the DBMS might reject the query because it selects records for a specific value of the sensitive attribute DRUGS.

A less obvious query is

```
List NAME where
     (SEX=M ∧ DRUGS=1) ∨
     (SEX≠M ∧ SEX≠F) ∨
     (DORM=AYRES)
```

On the surface, this query looks as if it should conceal drug usage by selecting other non-drug-related records as well. However, this query still retrieves only one record, revealing a name that corresponds to the sensitive DRUG value. The DBMS needs to know that SEX has only two possible values, so that the second clause will select no records. Even if that were possible, the DBMS would also need to know that no records exist with DORM=AYRES, even though AYRES might in fact be an acceptable value for DORM.

Inference by Arithmetic

Another procedure, used by the U.S. Census Bureau and other organizations that gather sensitive data, is to release only statistics. The organizations suppress individual names, addresses, or other characteristics by which a single individual can be recognized. Only neutral statistics, such as count, sum, and mean, are released.

The indirect attack seeks to infer a final result based on one or more intermediate statistical results. But this approach requires work outside the database itself. In particular, a statistical attack seeks to use some apparently anonymous statistical measure to infer individual data. In the following sections, we present several examples of indirect attacks on databases that report statistics.

Sum

An attack by **sum** tries to infer a value from a reported sum. For example, with the sample database in Table 18-2, it might seem safe to report student aid total by sex and dorm. Such a report is shown in Table 18-3. This seemingly innocent report reveals that no female living in Grey is receiving financial aid. Thus, we can infer that any female living in Grey (such as Liu) is certainly not receiving financial aid. This approach often allows us to determine a negative result.

TABLE 18-3 Table Showing Negative Result

Sex	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

Count

The **count** can be combined with the sum to produce some even more revealing results. Often these two statistics are released for a database to allow users to determine average values. (Conversely, if count and mean are released, sum can be deduced.)

Table 18-4 shows the count of records for students by dorm and sex. This table is innocuous by itself. Combined with the sum table, however, this table demonstrates that the two males in Holmes and West are receiving financial aid in the amount of $5,000 and $4,000, respectively. We can obtain the names by selecting the subschema of NAME, DORM, which is not sensitive because it delivers only low-security data on the entire database.

Mean

The arithmetic **mean** (average) allows exact disclosure if the attacker can manipulate the subject population. As a trivial example, consider salary. Given the number of employees, the mean salary for a company and the mean salary of all employees except the president, it is easy to compute the president's salary.

Median

By a slightly more complicated process, we can determine an individual value from the **median**, the midpoint of an ordered list of values. The attack requires finding selections having one point of intersection that happens to be exactly in the middle, as shown in Figure 18-1.

For example, in our sample database, there are five males and three persons whose drug use value is 2. Arranged in order of aid, these lists are shown in Table 18-5. Notice that Majors is the only name common to both lists, and conveniently that name is in the middle of each list. Someone working at the Health Clinic might be able to find out that Majors is a white male whose drug-use score is 2. That information identifies Majors as the intersection of these two lists and pinpoints Majors' financial aid as $2,000. In this example, the queries

```
q = median(AID where SEX = M)
p = median(AID where DRUGS = 2)
```

reveal the exact financial aid amount for Majors.

TABLE 18-4 Count Results

Sex	Holmes	Grey	West	Total
M	1	3	1	5
F	2	1	3	6
Total	3	4	4	11

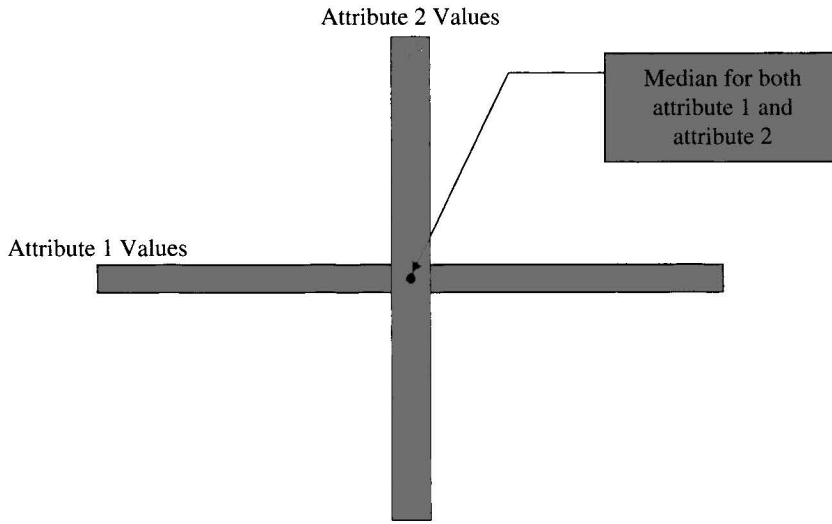

FIGURE 18-1 Intersecting Medians

TABLE 18-5 Tables Showing Drug Use and Aid

Name	Sex	Drugs	Aid
Bailey	M	0	0
Dewitt	M	3	1000
Majors	M	2	2000
Groff	M	3	4000
Adams	M	1	5000
Liu	F	2	0
Majors	M	2	2000
Hill	F	2	5000

Tracker Attacks

As already explained, database management systems may conceal data when a small number of entries make up a large proportion of the data revealed. A **tracker attack** can fool the database manager into locating the desired data by using additional queries that produce small results. The tracker adds additional records to be retrieved for two different queries; the two sets of records cancel each other out, leaving only the statistic or data desired. The approach is to use intelligent padding of two queries. In other words, instead of trying to identify a unique value, we request n–1 other values (where there are n values in the database). Given n and n–1, we can easily compute the desired single element.

For instance, suppose we want to know how many female Caucasians live in Holmes Hall. A query posed might be

count ((SEX=F) ∧ (RACE=C) ∧ (DORM=Holmes))

The database management system might consult the database, find that the answer is 1, and refuse to answer that query because one record dominates the result of the query. However, further analysis of the query allows us to track sensitive data through nonsensitive queries.

The query

q=count((SEX=F) ∧ (RACE=C) ∧ (DORM=Holmes))

is of the form

q = count(a ∧ b ∧ c)

By using the rules of logic and algebra, we can transform this query to

q = count(a ∧ b ∧ c) = count(a) − count(a ∧¬ (b ∧ c))

Thus, the original query is equivalent to

count (SEX=F)

minus

count ((SEX=F) ∧ ((RACE≠C) ∨ (DORM≠Holmes)))

Because count(a) = 6 and count($a \wedge \neg (b \wedge c)$) = 5, we can determine the suppressed value easily: 6 − 5 = 1. Furthermore, neither 6 nor 5 is a sensitive count.

Linear System Vulnerability

A tracker is a specific case of a more general vulnerability. With a little logic, algebra, and luck in the distribution of the database contents, it may be possible to construct an algebraic **linear system of equations** that returns results relating to several different sets. For example, the following system of five queries does not overtly reveal any single c value from the database. However, the queries' equations can be solved for each of the unknown c values, revealing them all.

$$\begin{aligned}
q_1 &= c_1 + c_2 + c_3 + c_4 + c_5 \\
q_2 &= c_1 + c_2 + c_4 \\
q_3 &= c_3 + c_4 \\
q_4 &= c_4 + c_5 \\
q_5 &= c_2 + c_5
\end{aligned}$$

To see how, use basic algebra to note that $q_1 - q_2 = c_3 + c_5$, and $q_3 - q_4 = c_3 - c_5$. Then, subtracting these two equations, we obtain $c_5 = ((q_1 - q_2) - (q_3 - q_4))/2$. Once we know c_5, we can derive the others.

In fact, this attack can also be used to obtain results *other than* numerical ones. Recall that we can apply logical rules to *and* (∧) and *or* (∨), typical operators for database queries, to derive values from a series of logical expressions. For example,

Many sites have privacy policies. Among the common characteristics of the policies are these:

- *Length.* A user might actually read and do something about a two-line policy, but if the policy is many pages long on a different web page, the consumer is tempted to ignore it.
- *Language.* Policies are often written by lawyers to protect the rights of the site owner. Lawyers are noted for precise and comprehensive, not terse and intelligible prose.
- *Mutability.* Many policies end with the caveat that they reserve the right to change the conditions at any time without notice.
- *Nontransferability.* Even for sites with strong restrictions on what they can or will do with collected data, those restrictions are lifted if the site owner is sold to another company with other objectives.
- *Noncomparability.* Even if you were deciding between two applications for which the deciding criterion was your privacy rights, the two policies would be written in free-form prose that would be impossible for you to compare easily side-by-side.

Thus, useful privacy policies remain elusive. Security and privacy researcher Annie Antón and colleagues have been investigating privacy policies for some time. With surveys in 2002 and 2008, they observed changes in users' perceptions [ANT09]. In their 2008 survey, they found that individuals were more uncomfortable with companies trading, sharing, and selling personal data than was the case in 2002. Also, respondents in 2008 were more interested in being informed about safeguards used to protect their sensitive data. Thus, meaningful privacy policies are desirable to users.

Internet social media applications are particularly well supplied with personal data. Messages to friends can be mined for information such as brand names, locations, and preferences. Postings, notices on the wall, photos, likes, hobbies, games, and even friends help refine the image known about a person, thus enabling advertisers to target advertising more and more precisely. The challenge is to write and enforce privacy policies that protect the user's interests when the advertiser pays for and benefits from the user's activity.

CONCLUSION

In this chapter we looked at privacy, collection, and analysis. Our specific example was web applications such as Facebook, but the principles apply to any situation characterized by a mass of personal data: public data from government sources, data collected from customer loyalty or rewards cards, entries to contests and sweepstakes, and internal monitoring activities. As the volume of data about us continues to grow, the opportunity for correlation also expands.

Because the sources of data are diverse and numerous, no single control can protect our privacy. General privacy principles and regulations would help, but in many cases

it would be technically difficult, if not politically challenging, to retrofit controls onto existing collection and analysis activities. Still, it would help to have a general consensus, which is currently lacking, on what kinds of collection and use are or are not acceptable. Without such a general understanding, users have little leverage against the large and well-financed organizations that collect and use such data.

The major points made in this chapter are these:

- Much sensitive data about individuals is collected regularly, often without the user's express consent and sometimes without the user's knowledge.
- Computer analysis can extract individual data from a larger body, so that what may on the surface seem like anonymous data can actually identify a single individual. The problem is especially important as multiple databases are combined; anonymous entities in a large body of data may become more readily identifiable in intersecting extracts from multiple sources.
- Data suppression and modification are ways to reduce the degree to which an individual can be identified. However, in many cases a single anonymizing identifier can allow identification as the pieces of data attached to that one number increase.
- Privacy policies are the user's strongest tool for preserving confidentiality online. However, policies tend to favor the online organization and its data collection goals, not the user's privacy interests.

The security and privacy threats, vulnerabilities, and countermeasures for Internet applications are summarized in Table 18-11.

TABLE 18-11 Threat–Vulnerability–Countermeasure Chart for Internet Applications

Threat	**Consequence**	**Severity**	**Ease of Exploitation**
Disclosure	Loss of confidentiality and privacy	Serious	Easy
Malicious code	System compromise	Serious	Easy

Vulnerability	**Exploitability**	**Prevalence**
Analysis of data	High	High

Countermeasure	**Issues Addressed**	**Mitigation Type**	**Mitigation Effect**	**Effort**
Encryption	Privacy	Technical	Prevention	Moderate, but severely constrains usefulness
User education, awareness	Confidentiality, malicious code	Administrative	Prevention	Low, but low effectiveness
System design	Privacy	Technical	Prevention	Moderate