We Can Share
Sensitive, Private, or Classified Information
Responsibly

Table of Contents

We Can Share Sensitive, Private, or Classified Information Responsibly

Purpose

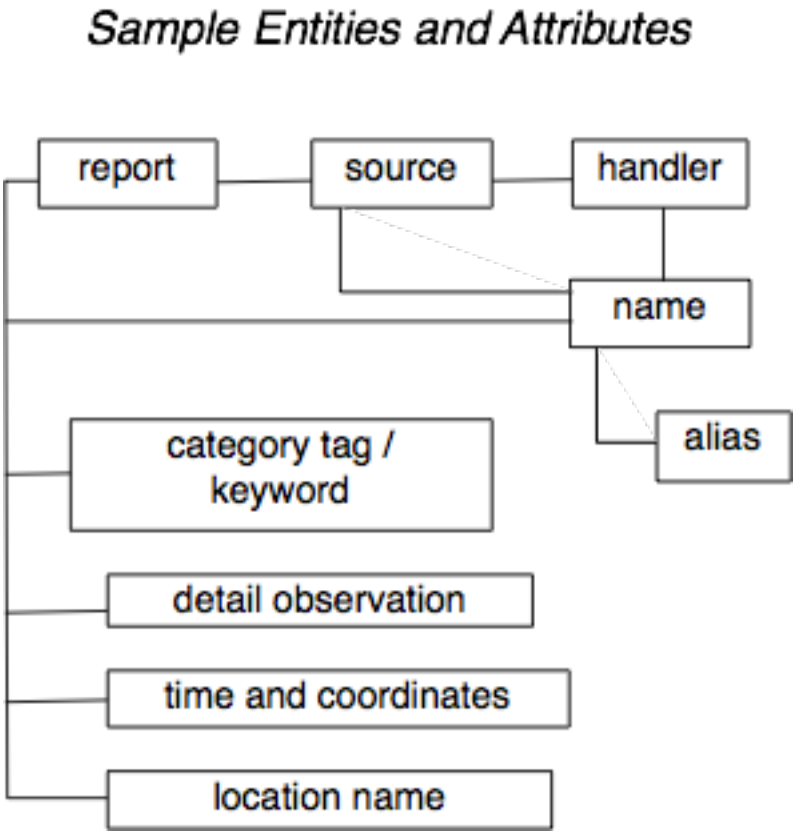
In this report, we explain how to share sensitive, private or classified information responsibly. Agencies dealing with classified information should recommend and deploy the method promptly because of failures demonstrated by published incidents. The deployment will only seem more urgent when the larger body of classified incidents is examined.

The brief scenario discussed here cannot encompass all aspects of responsible information sharing. It will be necessary to engage end users in discussions and to develop a comprehensive set of scenarios prior to constructing a production. Nevertheless, the scenario presented here illustrates important features of any safe and effective information-sharing environment. The method we advocate, *blind encrypted data matching* for *need-to-know* based access control, is fully explained elsewhere and public domain software provides a handy reference implementation to facilitate development and deployment of production systems.

Knowledge Structure

To explain the method, we adopt a set of entities and attributes that are likely to be found in the knowledge base for classified intelligence reports. Although entities and attributes are realistic sounding, they reflect no actual system and far fewer in number than one will find in a real system. We use them only for purposes of illustration

We start with a discrete incident report that has a variety of attributes including keywords, location name, and, where applicable, the geographic coordinates and time as well as other details of the observed activity. The report is attributed to a source, a named source; therefore, the source entity is connected to a name attribute. Sources may have other attributes such as reliability and productivity. In addition, sources have handlers who manage and possibly fund the source. The handlers have names. Of course, there may be aliases or misspellings for any named person or organization. These entities are illustrated in the following figure:



Queries and MultiLevel Security

Suppose we put all our intelligence reports in one basket, implement the knowledge structure shown above and run some queries. Here are a few queries that were interesting back in 2002.

Query 1: How many reports with the tag "WMD" were received from sources in location "Iraq" in the last year. Hypothetical answer: 180.

The answer is interesting because it suggests that WMD are a factor to guide decision making.

Query 2: How many unique source names are found in the intermediate tables of results returned for query 1. Hypothetical answer: 20.

This answer raises a cautionary flag. The sources are unusually productive and there aren't as many as one might have expected.

Query 3: How many distinct groups of sources are found in the intermediate table produced by Query 2 where a "distinct group" is defined as a set of source names whose set-members do not share a handler name with any members of another

distinct group. In other words, a distinct group is independently managed and funded. Historical answer: 1.

This answer is shocking. All the sources were funded and managed from one handler - historically Ahmed Chalabi who was known to be politically motivated as well as highly compensated for his sources' productivity. Very likely, however, the only query that was ever executed in 2002 was query 1 for the highly defensible reason that source names are highly sensitive and should not be available for query processing.

Consider the following quotation:

> *"Naming her in this way would have compromised every operation, every relationship, every network with which she had been associated in her entire career" Joseph C. Wilson referring to the public exposure of CIA handler Valerie Plame Wilson.*

One should add to this lament the knowledge that many high officials in the former Soviet Union who cooperated with the US during the long tenure of CIA turn-coat Aldrich Ames suddenly disappeared and were no longer available as information sources. Considering that Ames was unlikely to be the only turn-coat and most such situations are surely resolved without public disclosure, a reasonable person will conclude that the names of sources must be protected from insider threats. Yet this is difficult, both technically and operationally.

Old fashioned security relies on a multilevel security system that protects handler names more carefully than source names and source names more carefully than report contents. A typical modern example of multilevel security is provided by the Accumulo project (accumulo.apache.org) launched originally by the NSA. However, such multilevel security software leaves major operational problems unresolved: Who applies the access tags? Who develops the rules? Who enforces uniform rules across all agencies and locations? Who matches users, roles and access privilege? Nothing makes these operational problems easy but implementing a multilevel security system is fraught with risk.

Consider the WikiLeaks case. The security tags for Hilary Clinton's memos were applied in the State Department. The access rules for field offices of the Army Intelligence Service were written by the Army and enforced in Iraq. This operational disconnect across agencies doomed the security of the information sharing system. The system allowed Bradley Manning to download Secretary Clinton's memos from his location in Iraq across SIPRNET and then redistribute the classified memos via WikiLeaks. Obviously, wise men can fix such problems after the horse leaves the barn. However, future problems cannot be fully anticipated. Consider the problems posed by Isaac Asimov in "I Robot". Relying from just three obviously infallible

rules, the intelligent machines in Asimov's tales came up with the most outrageous actions when presented with unpredictable future situations. Just as Isaac Asimov could construct a story that placed his rule-based robots in situations that compromised their mission, a clever hostile party can manipulate people, systems, and events to defeat a rule-based multi-level security systems.

Press reports incline to focus on the information lost through the security breaches; however, history will surely find that the most destructive damage wrought by WikiLeaks will fall upon information sharing. Given our current security arrangements, no realistic person can expect that active duty intelligence officers will allow all of their sensitive information to fall into the big basket of a centralized information sharing system. The active duty officers have too much concern for their sources - not to mention their own careers! Federation or centralization represents irresponsible information sharing. Consequently, responsible local authorities will keep separate, local records for sensitive fields like "source" and "handler". Data that could cause damage locally will not be federated globally. The resulting database becomes, in database terms, horizontally-partitioned into geographically separate, local, secure partitions. This outcome is the responsible course of action for those involved but it prevents information sharing. A query can no longer use the full range of a database attribute. There can be no responsible information sharing if fear of inadvertent data release causes responsible parties to block the sharing.

In fact, it is doubtful if the query 3, discussed above, could be run even now given the security compartments that are quite reasonably in place. So failures to share information will lead to damaging incidents as we can illustrate with another historical example.

How to Reverse History and Share Information - Advice for the Future

We next turn to a highly public failure to share information: the December 2009 attempt to blow-up an airplane perpetrated by Umar Farouk Abdulmutallab. The would-be suicide attacker was a known entity in 4 databases, but the attacker nevertheless boarded a flight. Fortunately, he failed in his attack and was arrested. Only afterwards did anyone share the data and that might have stopped the terrorist before the flight departed.

It is not hard to see why information sharing failed. Recognizing this threat required an analysis of 4 databases held in three countries. The source of the information in Yemen was an especially sensitive fact. Even the existence of an intelligence source within a radically-oriented madrasa was highly classified because that source was risking his life to collect information.

Is there a scenario in which this historical event could have been turned into a routine police action to round up a suspected individual? Here is one.

- Suppose a report is filed every time a multiple-entry visa is issued for a foreign national. The suspect in this case would have such a report issued in the US.
- Suppose a report is issued every time a multiple-entry visa for a foreign national is revoked for cause. The suspect in this case would have such a report issued in the UK.
- Suppose a report is filed in the US when authorities classify an individual as a radical imam. The imam in this reported event, Anwar al-Awlaki , would have such a report in the US.
- Suppose a report is filed in Yemen whenever a foreign national studies with a radical imam residing in Yemen. Again, this suspect was implicated in such a report.
- Suppose a report is filed whenever a responsible party reports that an individual may be aiding terrorism. The suspect in this case was reported to CIA officers in Nigeria by the suspect's father, a respected banker, and the report was duly forwarded to a CIA database in the US.

Now suppose we can share the information that was gathered and run some routine queries:

Query 4: Is there a person with a multiple entry visa to the US who has been denied a visa elsewhere? Answer: yes, the suspect had a US visa but the similar visa in the UK was revoked.

Query 5: Is there a person with a multiple entry visa to the US who has studied with a radical imam? Answer: yes, the suspect studied with a radical imam in Yemen.

Query 6: Is there a person with a multiple entry visa to the US who is suspected of terrorist sympathies? Answer: Yes, the CIA had such a report on the suspect from reliable sources in the suspect's home country, Nigeria.

Was that so hard? Yes! It was too hard in 2009 and remains very hard today because different nations don't fully cooperate on information sharing. Information is collected. Analysts work on local data looking for intelligence gold. However, information is not routinely shared; therefore, major discoveries are routinely overlooked by the analysts who are locked into compartmentalized data. But, to a degree, such low-quality intelligence analysis may be preferred as the lesser of two evils. Imperfect sharing protects against the insider threat - which includes not just the rare turn-coat threat but the more common threat of inept or inadvertent insider actions.

Indeed in this scenario, it would be stupid to share large amounts of information with Yemen not knowing the sympathies of all individuals in Yemen with insider access. Yemen must feel likewise about American authorities. Any breach of security

such as the world saw with the Valerie Plame outing, the Aldrich Ames espionage, or the WikiLeaks would spell an abrupt, painful ending for the intelligence sources in Yemen who cooperated with Yemeni authorities. Mistrust is mutual across agencies and nations! This fact must guide the development of a better sharing system. That is where the new approach of blind encrypted data matching becomes important.

With blind encrypted data matching, all report matching and all query processing is performed on data that is strongly encrypted. With strong encryption, the data is safe outside its local security compartment. It is also useless outside the local security perimeter except for the approved purpose of finding matching, related data, or answers to queries. Only a small amount of carefully reviewed and approved information is actually shared thereby minimizing the risk to the mission.

In our visionary scenario, the US, UK and Yemen cooperate by allowing blind encrypted data matching (BEDM). Through this process, the authorities are alerted that there is a reported individual who should trigger followup action. At this point, nobody can identify that individual because the answer is encrypted. However, the blind agent provides the rationale to move forward and approve the sharing of selected classified information.

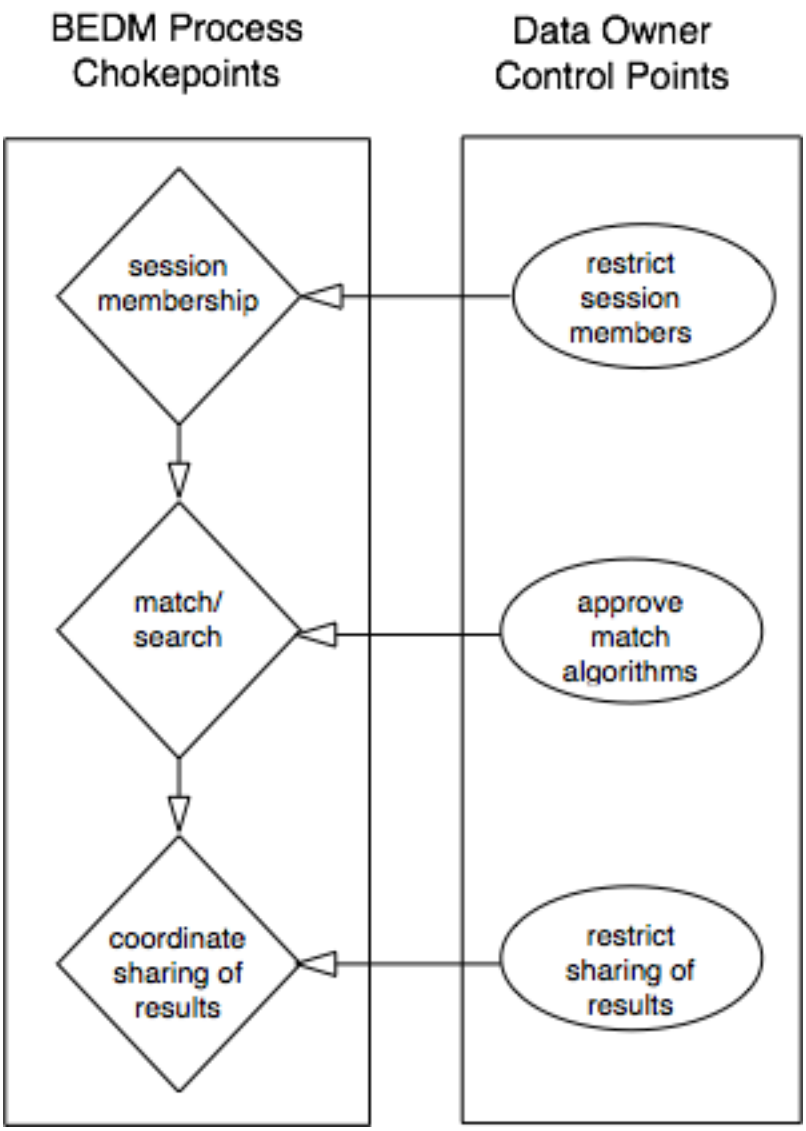
Our scenario outlines a vision for information sharing through blind encrypted data matching. Frequently, this vision prompts the question: who operates the blind agent? We don't emphasize the answer in the scenario because that is a policy matter and we prefer to discuss policy after discussing data matching and responsible information sharing. In general, however, the blind agent should be run by someone who is trusted to be neutral with respect to the data owners. A related question is what special precautions are needed for encrypted - but admittedly still federated - data. The design of special security provisions is discussed in a related document on software architecture[1].

[1] The WWN Architecture for a Secure Information Sharing System

Thwarting Hostile Information Access

An information system may have hundreds of thousands of users. Among them are certain to be a few people with hidden sympathies for ideological causes or with financial problems that might tempt them to sell information. Also people can be negligent or thoughtless leading them to make serious errors. Inevitably, some fraction of the activity on an information system will be hostile to the mission.

A system like BEDM is a major advance in security over prior notions of multilevel access control, but it is not a barrier to all hostile insider activity. Consequently, our scenario must illustrate how the technology incorporated in blind encrypted data matching (BEDM) enables a knowledgeable system manager to set up rules and policies to maximize security. The central innovation of BEDM is a multistep process that replaces the simplistic ask/receive one-step model adopted by current analysis systems. During the BEDM process, the system places limits on what can be delivered to a user through a series of choke points illustrated in the diagram to the right.



All search and matching performed by a blind agent takes place in a cycle called a "session". For each session, there is a fixed set of data owners. A data owner has control over the session through a series of choke points. First, the owner can review and approve all of the other data owners who participate in a session. If there is one party that is considered unreliable, other parties can refuse to participate in the

session with that party. Secondly, the data owners may review and must approve the algorithms and searches prior to advancing to the matching phase of the session. The importance of this will be explained below.

Finally, the BEDM process has rules that restrict sharing. Technically speaking, a blind agent could be set up to automatically share each discovery, but that is not a good idea. Instead, there are policies in place to limit what can be shared and how it is shared. Such policies are suggested and approved in advance by the data owners. Thus there is a flow of encrypted information during a session but the creation and distribution of the encrypted results are restricted at each of three choke-points illustrated above.

Let us illustrate the necessity of the choke-point function in the context of our historical scenario involving Umar Farouk Abdulmutallab. Suppose his mentor, Anwar al-Awlaki, had successfully placed an agent in a US intelligence agency. If that agent wanted to find and reveal the sources used by Western intelligence services in Yemen, then a legally formulated SQL query such as: "`select * from source where location = 'Yemen'`" would suffice to reveal the sources[2]. Using the choke point for "approve match algorithms", a conscientious data owner would reject such a broad query in advance.

Many hostile queries can be eliminated because they fall into the category of "too broad"; in other words, they are fishing expeditions for valuable facts. Also, we note that the previous query refers to very sensitive information, the source names, without offering any sensitive information in return. There is no reciprocity in this request, no evidence of a matter of mutual interest, and therefore no motivation to share information with the author of this request.

Generally speaking, we share only with people who need to know. If a person doesn't know what they are looking for and asks only broad questions, then the query does not deserve a reply. With that in mind, analysts and policy makers will create rules and policies to protect the sharing environment. Here are few rules that might be enforced at a choke-point:

Rule 1: If the match involves an intelligence source, then block the sharing operation pending further investigation.

---

[2] For those unfamiliar with the term SQL, it is a computer language that is widely used for asking questions that require data analysis and retrieval. Our query is fairly generic and says, in English "give me the names of all sources in Yemen". In practice the query would be specialized for the terminology of the intelligence agency, but an insider has that necessary knowledge. Although other query languages are in use, we use the best known of these languages to illustrate the point about the requirement for information sharing rules.

Rule 2: If the match involve sharing with an unreliable data owner, then block sharing pending further investigation and approval.

Rule 3: If the match entails a lopsided data exchange whereby one data owner would give up a great deal of classified information in exchange for very limited information from the counter parties, then block sharing and request a modification of the matching.

The choke points are important tools and represent an improvement over less secure information sharing environments. However, they are still rule-based and suffer from the same deficiency that plagues rule-based access in multilevel security systems: the consequences of rules cannot be anticipated. Let us examine the possible consequences for choke point rules in the context of the earlier scenario.

Consider an intelligence analyst who wishes to pursue a line of inquiry based on "suspicion by association". The basic idea is simple. If, for example, Anwar al Awaki is a mentor to terrorists and teaches at el-Elman University in Yemen, then it is reasonable to ask for lists of dates when Awaki taught there and who else was attending on the same date. If a new name crops up often in such a "suspicion by association" query, that new name is a candidate for investigation.

Now consider that this inquiry depends on someone providing names and dates from el-Elman University. An analyst cooperating with the opposition could tease out the name of the information source with some innocuous but clever queries presented in sequence. Each query would ask about activity at el-Eman on a particular date, say 5 March 2007. Then another query would ask for another date and then so forth through a series of dates. A good agent would even space the queries out to avoid drawing attention. To accomplish the purpose, the opposition has provided its insider counterspy with lists of dates when various suspicious individuals were observed to be in attendance at el-Eman. The names on the opposition's list were placed there by al-Qaeda operatives because the persons named are suspected informers for the Yemeni government or a western power. The key to the attack on the information sharing system is that the database will have no reports on the days when an informer was not present in the madrasa. Reports jump when a informer is present. Informers will stand out in the pattern of hits on the series of queries.

It is unlikely that simple rules will prevent this line of attack. Moreover, consider the following subtle point that has not been mentioned up until now. The choke points can block data sharing, but certain attacks only need to know that the rule has been activated. In the preceding example, the hostile agent only counted positive matching operations on dates - there was no need to see the actual secret data. Likewise, a hostile agent could ask outright: "Is X an informer in el-Eman

University" and a positive encrypted match would reveal the answer even if the encrypted data is never shared.

The game of spy versus counterspy is complicated and never fully played out. The federated data base coupled with access controls may be the current state of the art for information sharing but it is wide open to the insider threat. Counterespionage agents must be very busy! Introducing blind encrypted data matching will preserve the advantage of information sharing while blocking most insider attacks. However, clever insiders still have methods of attack and we block many of those with the rules at the choke points. When we reach this stage, we will have security and responsible information sharing, but still a residual risk. Counterspies cannot retire yet! That conclusion merits two other observations.

First, although access control alone is insufficient for responsible information sharing, access control can be and should be included in a blind encrypted data matching system. To illustrate this, consider the query: "Is X an informer in el-Eman University". If the answer is yes, it not only answers the query but shows that the person posing the query desires to know the name of informers. If that person has access-rights to that knowledge, then perhaps this is a legitimate query. However, if the person posing the query has no access-right to the knowledge, then this is a suspicious even hostile query that should be followed-up by counterintelligence operatives.

Second, it would be ideal if blind encrypted matching systems helped the essential work of counterintelligence but the blindness of the matching agent actually impedes it. Counterintelligence analysts are limited in their ability to inspect any of the work of the blind agent because the parameters of algorithms and search queries used in the matching process are encrypted. For that reason, a full system must include the adjudication extension. This important feature is outside the scope of the scenario but it is discussed in the Software Architecture document[1] cited earlier.

Conclusions

In this document, we have outlined a scenario constructed from pieces of historical fact that is reasonably representative of one opportunity for information sharing. When the opportunity presented itself, the opportunity was wasted. Discussion of this scenario shows how failure is built-in to current information systems and points the way to an improved system.

Our scenario is a simple scenario. The imagined threat and the defense described here both lack the variety and subtlety of real life. A lack of scope is indeed a limitation of any short singular scenario, but it should not be a limitation on the western defense against aggression and terrorism. The fact that life is harder than

our scenario supposes only emphasizes the need to acquire the best tools to prevail in the struggle.

Responsible information sharing using blind encrypted data matching is at least one or two orders of magnitude more secure than legacy technology. It can be gamed and defeated as is true of any system, but that defeat is far less likely. Therefore, BEDM can be seen as an enabling factor to safely expand the scope of responsible information sharing and thereby to support a wider, more competent scope of intelligence analysis activities.