Making Database Security an IT Security Priority

SANS ANALYST PROGRAM

Sponsored by Oracle

Making Database Security an IT Security Priority

A SANS Whitepaper – November 2009

Written by Tanya Baccam

Introduction

Organizations are becoming more concerned about data security, especially as the intrinsic value of our data continues to increase. However, database security often gets overlooked. Managing organizational assets such as data, as well as overall information security concerns, are two of the key technology areas having a large affect on companies today. Although it is often difficult to put an exact price tag on the data we store, we do know data is an extremely valuable asset, and the compromise and/or exposure of such information can cause significant damage to business and company reputation. As a result, a security strategy needs to be developed to address information security risks, including data security. Throughout this paper we discuss security strategy and the key controls that should be considered in an organization's security strategy as they relate specifically to database security and protecting an organization's information assets.

Security Strategy Overview

A security strategy is an overall plan to mitigate risk. While many organizations today have a security strategy, what is sometimes missed or not adequately addressed is database security. A security strategy must mitigate the overall possibility of harm or loss to a company's data. Furthermore, a security strategy must address the business data concerns from a legal, statutory and contractual perspective. Multiple regulatory requirements and standards, such as PCI, HITECH/HIPAA, GLBA, SOX, privacy laws and others have required organizations to address information security risks. According to the SANS Database Audit and Compliance Survey, 74 percent of the respondents felt their organizations utilized data that is or might be considered regulated.[1] Additionally, contracts are increasingly requiring organizations to consider security at all layers. Addressing database security in a proactive manner can save organizations a significant amount of money and reduce the overall risk exposure.

When building a security strategy, the following key steps must be taken:

- Define organizational control objectives.
- Identify an approach to meet the objectives.
- Identify controls to meet the objectives.
- Identify metrics and/or benchmarks to manage the controls.
- Implement the controls.
- Test and maintain the controls going forward.

Preventive, detective and corrective controls should be implemented to cover people, process and technology for all information security assets.

[1] www.sans.org/reading_room/analysts_program/DatabaseAudit_Feb08.pdf

Why a Database Security Strategy?

If regulatory or contractual requirements are not enough reason to address database security as part of an overall security strategy, let's look at a few other key facts that encourage us to consider database security as a part of our security strategy.

Databases are increasingly being targeted by attackers. In Verizon's 2009 Data Breach Investigations Report it was noted that 30 percent of breaches were against databases, as Figure 1 illustrates. The only other asset that had a larger percentage of breaches was POS systems, and many of them still contain or interact with a database.

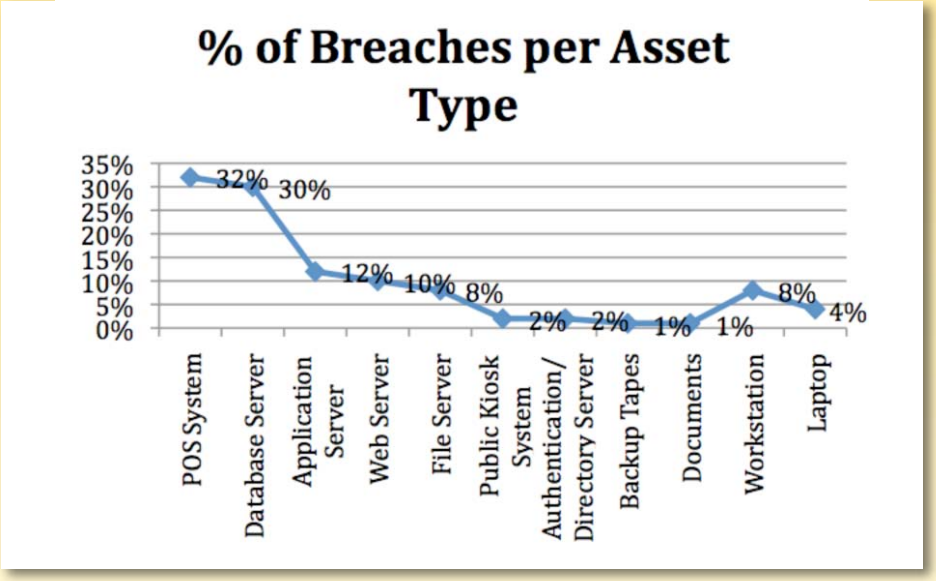

Figure 1: Percentage of Breaches per Asset Type[2]

[2] www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf

What is even more concerning is the percentage of records compromised when a database is breached. When a POS system is breached, only six percent of the records were compromised. However, when a database is breached, 75 percent of the records were compromised! Of course, the database is a much larger overall risk area to have a larger impact on an organization. No other asset had nearly as high a percentage of total records compromised, as Figure 2 illustrates.

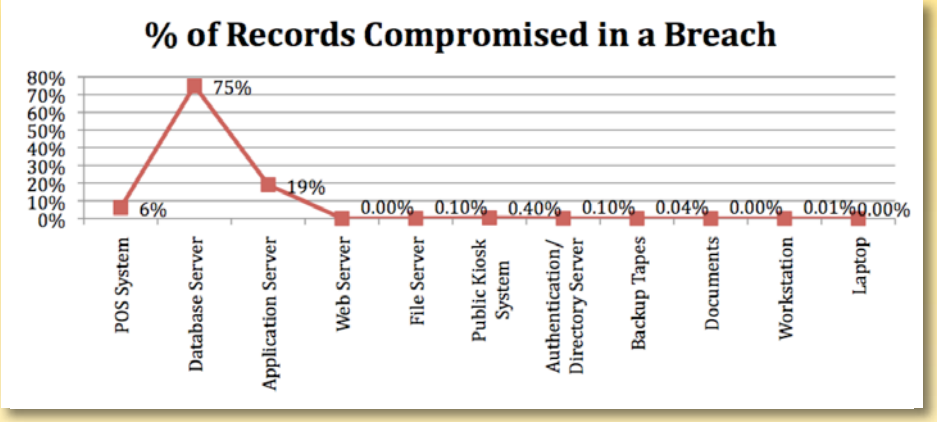

Figure 2: Percentage of Records Compromised per Asset Type[3]

Data plays an extremely important role in a typical organization's environment. Security has historically addressed keeping the external attackers out of networks and operating systems. More recently, the focus has been on security of applications. Organizations spend large amounts of resources adding firewalls, IDS, IPS, policies, operating systems controls, access controls and other security controls to end points and on the network. By doing so, organizations believe they are protected. However, without controls directly around the data, they've left open an opportunity for an internal attacker who is authorized to access and transfer data from the database.

For an attacker, a compromise of a database can have an immediate and large financial payoff. For organizations, a database compromise can have a large, negative financial impact on their business. By way of example, TJX has set aside $197 million in reserves to deal with the theft of 94 million records.[4] And, as of May 2009, Heartland had spent $12.6 million on costs related to its January 2009 breach, according to an article in SCMagazine.[5]

[3] www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf
[4] www.usatoday.com/money/perfi/credit/2009-01-21-visa-mastercard-credit-security-breach_N.htm
[5] scmagazineus.com/126-million-spent-so-far-to-respond-to-Heartland-breach/article/136491/

Making Databases a Priority

Databases, by their nature, are complex. Many security professionals simply do not have the background to understand the risk and security issues related to various brands and versions of databases. This leaves security in the hands of DBAs, who spend less than five percent of their time on database security, according to a Forrester Research report. The report stated that many enterprise DBAs are unaware of which databases, tables, and columns contain sensitive data, either because these are legacy applications and/or because no documentation of the data models and their properties exists.[6]

Even with full knowledge of database assets, databases are more difficult to protect uniformly because there are unique security implementation procedures for the databases themselves, as well as with the applications interacting with them. Forrester estimates that more than 90 percent of enterprises support more than one type of database in their environment. Enterprises today have to support hundreds and thousands of production databases with various business applications running on them. Business applications interacting with the databases can pose significant risks as additional application layer vulnerabilities may be introduced.

In order to effectively protect the organization's data, database security must become a management priority. Without management's support and strategic direction, database security will continue to be overlooked.

While implementation of firewalls, IDS, IPS, policies, operating system controls, access controls, and other security controls is certainly important, and "defense in depth" should always be emphasized, these layers of security may be automatically bypassed if the attacker is an insider. Furthermore, the database may be left wide open and accessible from any internal system, or at least many more systems than necessary. There may be no real structure to the security implemented in the database, and simple vulnerabilities such as default accounts may give an attacker access.

Companies should obviously continue to protect their networks, servers and other systems with the tools they're currently using. However, they can no longer rely exclusively on perimeter controls to protect data inside their organizations. There are many, like the Jericho Forum, proclaiming the death of network perimeters and calling for more data centric protections. Whether you agree or disagree that the perimeter is dead, controls do need to be moved closer to the data being protected. Then, if an attacker does bypass network security or other internal controls, the database layer security controls can still be effective. In other words: Keep the security controls close to the action!

[6] Reference: Jonathan Penn, Forrester Market Overview: IT Security in 2009, Forrester Research

As indicated in Figure 3, adding controls as close to the data as possible places the final layer of security around the data stored in databases, protecting it from both external and internal attacks.

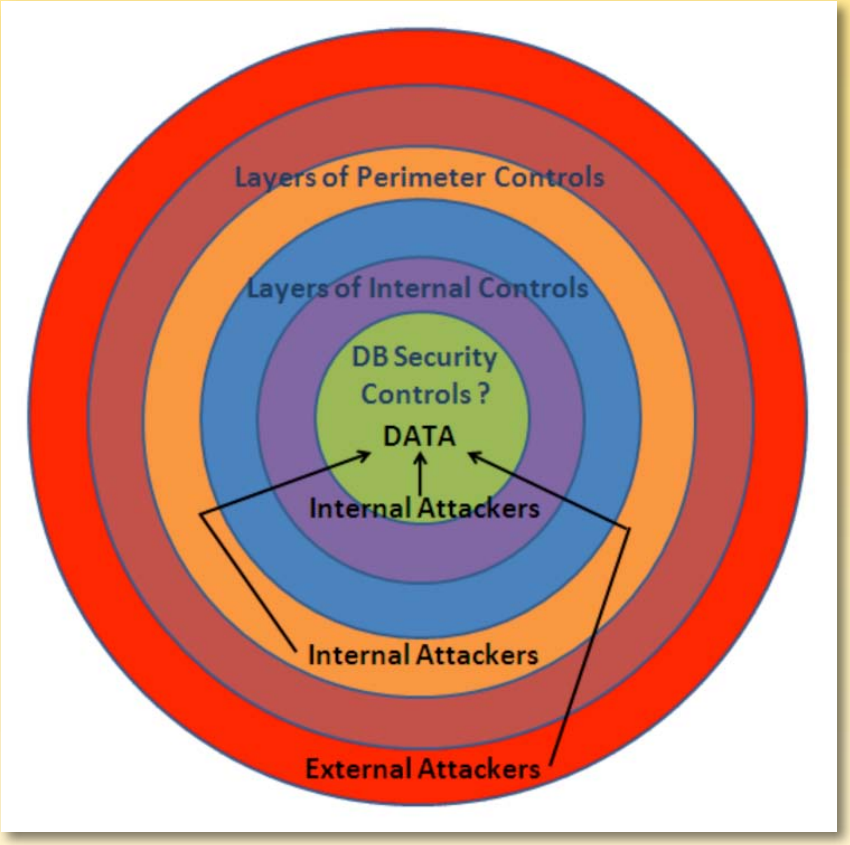

Figure 3: Layers of Controls and Protection from Attack Vectors

In summary, database security must be a part of the strategic plan:

- Databases recently saw the largest percentage of compromised records per incident and are one of the top compromised assets.
- The number of incidents continues to increase.
- Management support is critical to ensure successful implementation of security controls.
- Many security departments and professionals do not understand and/or look at database security.
- Controls should be applied as close to the data (in the database) as possible to protect from both external and internal attackers.

Database Security Considerations

We know we need to address database security as part of our overall security strategy. So, the question becomes, what key areas should be addressed? The following areas are critical areas we discuss throughout the remainder of this paper:

- Access controls
- Encryption
- Auditing
- Separation of environments
- Secure configuration

There are other controls that should also be considered, such as physical security, policies, user awareness training, and backups, among others, that organizations have already addressed for other critical systems. We will not directly address those here, except to refer to them as important security controls. The controls we look at in this document are controls specific to the database environment that are less likely to be addressed at the database layer of the enterprise.

Access Controls

Access controls not only protect against external and internal attackers, they can also protect from the mistakes that users introduce—errors that can potentially have as big an impact on operations as an external or internal attacker. For example, a user drops or deletes a business critical object, such as a required database table that they assume is not in use. Access controls can also minimize the impact of other risks that may affect the database, such as application risks that have a direct impact on the security of the database on the backend. There are many resources that illuminate and provide statistics on the number of application risks that exist.[7] For example, Web application SQL injection attacks[8] pose one of the most significant risks to databases, according to the SANS Institute Top 20[9] and the Verizon report.

In fact, when SQL injection was used, the attacker was able to compromise an average of 79 percent of the records, according to Verizon's report. SQL injection can be used to query data the attacker should not have access to, run built-in packages in the database, and potentially even get a command prompt on the underlying operating system.

[7] An example of one such site is www.webappsec.org/projects/statistics/.
[8] http://projects.webappsec.org/SQL-Injection
[9] www.sans.org/top20/

Access controls should be deployed based on the principle of least privilege. Access controls should be applied to two primary categories of users: administrators and standard end users.

For administrators, each DBA should be limited to only the functionality they need to do their job. Many times default functionality, including default roles, provide many more privileges than are necessary because they are designed for ease of use. In such cases, default roles such as 'DBA' should not be used, and instead specific roles for administrative activities should be designed to grant only necessary privileges. This may require multiple roles to be set up for different levels or types of administrators, as Figure 4 illustrates.

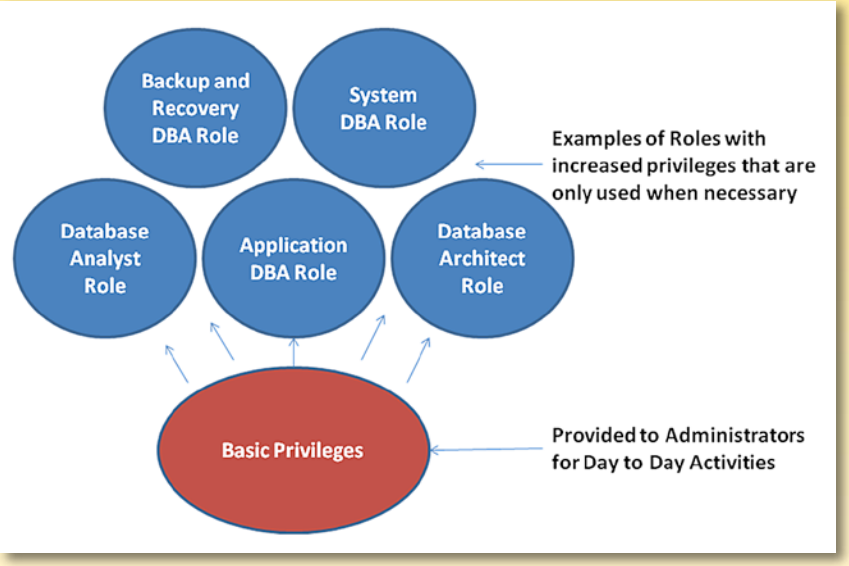

Figure 4: Example of Access Controls for Administrators

Even organizations that do not have multiple administrators should limit administrative access to only what is necessary for that administrator at a given point in time. Again, this can minimize the impact of errors and security incidents. With an Oracle database, for example, you may also use one of the options called Database Vault,[10] which can control what administrators can do and what data they can see within a database. For access control, considerations can also be given to multifactor authentication and authorization, which can also be implemented in many different ways, including with Database Vault and other built-in functions.

[10] www.oracle.com/database/security/database-vault

Applications need to give further consideration to security in order to protect the backend database. If a single account is used to connect to the backend database, the database may lose the capability to effectively track the individual user's activity. Applications should use solutions such as Virtual Private Database (VPD)[11], Label Security[12] and Secure Application Roles,[13] which allow dynamic control over what information is available to end users, based on their environment or context. Login triggers that monitor who a user is, and information, such as where the user is coming from, can be used to limit what a user can do when connecting to a database. These are powerful features that, when implemented correctly, can significantly decrease the overall risk the database faces from an access control perspective.

For regulatory purposes, access controls are often a foundational component that organizations must address because the access controls directly impact what information users can access. Centralized management of these controls can also reduce the risk of improperly applying access controls to any user—administrative or nonadministrative. While organizations have been working to centrally manage the provisioning and deprovisioning of users for years, they often overlook database user provisioning in their plans. Centrally enforcing password management policies and authorization can be beneficial to the overall database environment. Enterprise User Security[14] is a way to meet the need to manage users externally in an enterprise wide identity management system.

Encryption

Encryption use is growing in response to regulatory compliance mandates and auditor requirements for such controls. Encryption is a strong security control when implemented correctly. However, some organizations view encryption as a magical solution that will solve all their data security problems. While no database security strategy would be complete without data encryption, the reality is that we need to understand the entire environment, and then protect data in two states: data in transit and data at rest. Databases provide different solutions for encrypting data in transit and data at rest. Often, two solutions are needed to address each of the risks posed.

[11] www.oracle.com/technology/deploy/security/database-security/virtual-private-database/index.html
[12] www.oracle.com/database/security/label-security
[13] www.oracle.com/technology/deploy/security/database-security/secure-application-roles/index.html
[14] www.oracle.com/technology/deploy/security/database-security/enterprise-user-security/index.html

Data in transit refers to the data that is traveling across a network. We want to ensure that data is being encrypted as it traverses the network, when possible, or at least make sure the network is properly segmented so an insider cannot easily sniff the network traffic. As a recent illustration of why it is important to encrypt sensitive traffic internally, TJX had a very large breach in 2006, where payment card and customer data was captured using a sniffer to monitor the internal network traffic. End to end encryption is a strong protection against data leakage from internal attacks. Encryption solutions for traffic include SSL, SSH, IPsec, and others, often provided natively by databases.

Data at rest refers to data that is being stored within the database. Data can be encrypted at different tiers as part of an application, but that practice can pose potential challenges for key management and system updates. By encrypting the data in the database itself, you centralize the encryption controls for the data at its source. Encryption can be applied per column or for an entire tablespace. It is also important to note that encryption must also be applicable to data being stored for backup purposes. This is also data at rest that must be properly protected.

Auditing

Ultimately, we can and must implement the access controls talked about previously; however, in order for users to get their jobs done, we must grant some access. Therein lies the risk. You have to allow the DBA and users to do their job, so you have to give some privileges. But how do you know if privileges are being abused? And how does the DBA or user know that they are protected and are not going to be accused of abusing privileges they have been granted when they have not?

You can mitigate these risks by implementing and properly managing the audit trails within the database. Auditing allows you to track user activity, including administrator activity. Many options exist for auditing, including standard database auditing features. At a minimum, activities such as the following should be audited for regulatory requirements:

- Administrative activity
- Logon and logoff activity
- Failures
- Use of system privileges
- Use of "all" privileges
- Alterations to the database structure
- Critical object access

Customized auditing for the environment may also be necessary. For example, when users read or change specific records, you may want to record this activity. Solutions such as Fine Grained Auditing (FGA) can be used to monitor such activity and customize auditing based on predefined conditions.[15]

Storing audit records on a secondary system and centralizing the audit records is also critical. By storing records on a secondary system there is reduced likelihood that unauthorized activity recorded in the audit trail can be changed. Furthermore, by centralizing the audit records on a single system, it is easier to review and monitor activity across databases instead of looking at each database individually. In an Oracle environment, this functionality can be provided through Audit Vault.[16]

Separation of Environments

Separation of production, test, QA and development or similar environments has been an information security best practice for many years and has become embedded in many regulatory requirements and standards and audit programs. Environments and the data stored in them need to be separated because developers and development environments pose a significant risk to the organization because of the access they have. Ideally, developers should only have access to the development environment, unless there is a business need, such as troubleshooting a problem that cannot be duplicated in the development environment. If access does need to be given, developers should be given read-only roles with minimal privileges and their actions must be audited. Roles should be created proactively so, when a problem does arise, unnecessary access is not granted as a quick solution.

[15] www.oracle.com/technology/deploy/security/database-security/fine-grained-auditing/index.html
[16] www.oracle.com/database/security/audit-vault

Production data should not be propagated to other environments, such as development or test. However, the data can be manipulated to enable transfer of the data to other environments without leaving it open to compromise. For example, Oracle's Data Masking[17] can automate this process of de-identifying data. Data Masking can take sensitive information such as social security or credit card numbers and replace them with realistic but simulated values while still maintaining referential integrity so applications continue to work. These fully functional masked databases are safe to move to nonproduction environments. If attackers do get access to these other environments, which are typically less secure then production, the data is useless to them. This separation is illustrated in Figure 5.

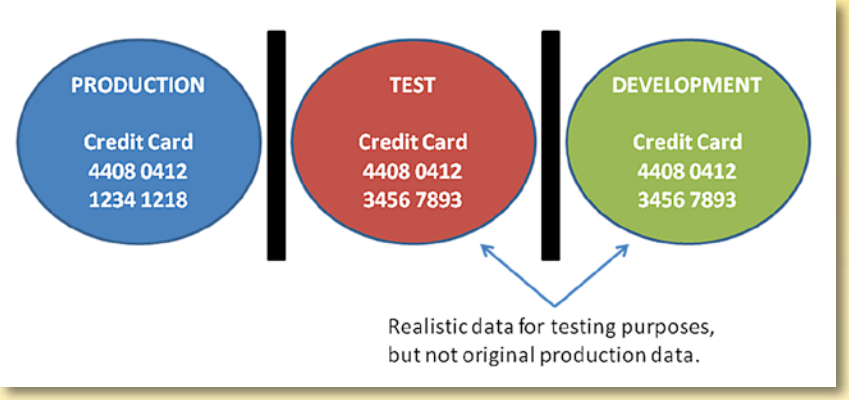

Figure 5: Proper Separation of Production, Test and Development Environments, Including Separation of Data

Secure Configuration

Ultimately, we are responsible for securing systems within the environment, and that includes the database. Unfortunately, as mentioned previously, many security professionals simply do not understand or have not looked at database security. The challenge with a database is that the complexity involved can often be overwhelming! Complexity is the enemy of security and can lead to mistakes. However, with the right tools, the complexity can be managed successfully.

[17] www.oracle.com/database/security/data-masking

To properly address database security, tools are required to automate the entire secure configuration life cycle. This includes database discovery, security scanning, configuration lock down, automated remediation, and so on. Security consideration needs to be given to the database itself, as well as the surrounding environment, including the underlying operating system and applications. Here are key areas that should be considered when securing a database:

- Default accounts
- Users and roles
- Exposed passwords
- Patching
- Privileges and permissions
- Parameter settings
- Password management
- Profiles
- Auditing
- Listener security

Database vendors have begun taking the hardening process more seriously. Solutions such as Oracle Configuration Management Pack[18] can help monitor and identify configuration vulnerabilities. Once a database is properly secured, configuration vulnerabilities need to be continuously monitored for potential changes, because well-intentioned users may modify the configuration of a system, thereby leaving it in a vulnerable state.

Not only is it important to monitor for configuration drift for security purposes, but there are also many regulatory requirements today that require a system to be configured in a particular state. By continuously monitoring for changes, management can be notified automatically if a database no longer meets an organization's security requirements and/or regulatory requirements.

[18] www.oracle.com/database/security/secure-configuration

Summary

Sometimes it is incorrectly assumed that doing nothing will cost an organization nothing. In reality, doing nothing related to the database security strategies discussed in this paper can cost an organization millions. For example, Heartland has publicly admitted to paying nearly $13 million on breach-related costs, and that number will most likely increase. Many regulatory requirements, standards and audits that are conducted today require the controls discussed in this paper.

Organizations need to consider database security as a part of their overall security strategy. Database security should be sold to the organization as a whole. After all, the most critical asset an organization has is its data. Security is everyone's responsibility. This means the DBA must play a role, as well as the security department and end users with access. Executives should clearly define the roles of DBAs, security departments, auditors and others within an organization to ensure that database security risks have been addressed and that proper separation of duties is maintained when new controls are implemented. By providing an overall strategy for the environment, executives can encourage different departments to work together to meet the organization's overall security requirements.

The good news, today, is that organizations are becoming more aware of their database security issues. Now they need to wrap robust database security strategies into their overall security and compliance strategies. Fortunately, database vendors have added some excellent security enhancements over the years that can greatly reduce the overall risk the database faces within a given environment. Such advancements will help organizations put security close to their critical data and better meet regulatory requirements.

[22] http://en.wikiquote.org/wiki/Mark_Twain

About the Author

Tanya Baccam is a SANS senior instructor as well as a SANS courseware author. She is the current author for the SANS Security 509: Securing Oracle Databases course. She also provides many security consulting services for clients, such as system audits, vulnerability and risk assessments, database audits, and Web application audits. Today much of her time is spent on the security of databases and applications within organizations. Tanya has also played an integral role in developing multiple business applications and currently holds the CPA, GCFW, GCIH, CISSP, CISM, CISA, CCNA, and OCP DBA certifications.

SANS would like to thank this paper's sponsor:

Last Updated: May 27th, 2018

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Atlanta 2018	Atlanta, GAUS	May 29, 2018 - Jun 03, 2018	Live Event
SANS Rocky Mountain 2018	Denver, COUS	Jun 04, 2018 - Jun 09, 2018	Live Event
SANS London June 2018	London, GB	Jun 04, 2018 - Jun 12, 2018	Live Event
SEC487: Open-Source Intel Beta Two	Denver, COUS	Jun 04, 2018 - Jun 09, 2018	Live Event
DFIR Summit & Training 2018	Austin, TXUS	Jun 07, 2018 - Jun 14, 2018	Live Event
Cloud INsecurity Summit - Washington DC	Crystal City, VAUS	Jun 08, 2018 - Jun 08, 2018	Live Event
Cloud INsecurity Summit - Austin	Austin, TXUS	Jun 11, 2018 - Jun 11, 2018	Live Event
SANS Milan June 2018	Milan, IT	Jun 11, 2018 - Jun 16, 2018	Live Event
SANS Cyber Defence Japan 2018	Tokyo, JP	Jun 18, 2018 - Jun 30, 2018	Live Event
SANS Oslo June 2018	Oslo, NO	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS ICS Europe Summit and Training 2018	Munich, DE	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Philippines 2018	Manila, PH	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Crystal City 2018	Arlington, VAUS	Jun 18, 2018 - Jun 23, 2018	Live Event
SANS Minneapolis 2018	Minneapolis, MNUS	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Cyber Defence Canberra 2018	Canberra, AU	Jun 25, 2018 - Jul 07, 2018	Live Event
SANS Paris June 2018	Paris, FR	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS Vancouver 2018	Vancouver, BCCA	Jun 25, 2018 - Jun 30, 2018	Live Event
SANS London July 2018	London, GB	Jul 02, 2018 - Jul 07, 2018	Live Event
SANS Cyber Defence Singapore 2018	Singapore, SG	Jul 09, 2018 - Jul 14, 2018	Live Event
SANS Charlotte 2018	Charlotte, NCUS	Jul 09, 2018 - Jul 14, 2018	Live Event
SANSFIRE 2018	Washington, DCUS	Jul 14, 2018 - Jul 21, 2018	Live Event
SANS Malaysia 2018	Kuala Lumpur, MY	Jul 16, 2018 - Jul 21, 2018	Live Event
SANS Pen Test Berlin 2018	Berlin, DE	Jul 23, 2018 - Jul 28, 2018	Live Event
SANS Cyber Defence Bangalore 2018	Bangalore, IN	Jul 23, 2018 - Jul 28, 2018	Live Event
SANS Riyadh July 2018	Riyadh, SA	Jul 28, 2018 - Aug 02, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LAUS	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS Pittsburgh 2018	Pittsburgh, PAUS	Jul 30, 2018 - Aug 04, 2018	Live Event
SANS August Sydney 2018	Sydney, AU	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS San Antonio 2018	San Antonio, TXUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS Boston Summer 2018	Boston, MAUS	Aug 06, 2018 - Aug 11, 2018	Live Event
Security Awareness Summit & Training 2018	Charleston, SCUS	Aug 06, 2018 - Aug 15, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, IN	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS Amsterdam May 2018	OnlineNL	May 28, 2018 - Jun 02, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced