Securing an Application: A Paper on Plastic

Joe Rhode

Securing an Application: A Paper on Plastic

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4b Option 2
Administrivia Version 2.3

Joseph S. Rohde
January 20, 2003

Abstract

This paper discusses the process of integrating a credit card application to the front end of already existing accounting and payments processing applications. It discusses the information risk analysis process needed to drive out a plan to secure the sensitive credit card information and the action plan to implementing the mitigated controls. Securing an application involves much more than the mechanics of creating access groups and granting permissions. It involves establishing business management alliances, building relationships with technical subject matter experts, and creating an environment for open dialog between these entities as well as with the members of the project team. Today, applications tend to be multi-platform, complex, integrated with purchased vendor products, and, many times, linked to external (to the company) customers and businesses. In this environment, the jobs of security professionals are complex, requiring the need to integrate dissimilar security solutions in order to provide the level of risk tolerance suitable to the application and yet complying with governmental laws and industry regulations. Networking with and leveraging off other security professionals' knowledge is both prudent and essential to providing the most appropriate, cost effective solution for employers and/or business customers.

Introduction

In order to compete in any of today's markets, acceptance of credit cards for payment of products and services is nearly essential. Businesses with an Internet presence most certainly lose potential sales if there is no facility to complete a transaction using a credit card or other third party guarantor of payment for the product or service. My employer (hereafter referred to by the fictional name ABCBest1) recently made a decision to accept credit cards for payment of its products. Prior to this decision, payments were accepted only in the form of cash or checks.

Prior to the addition of this new process, the entire financial handling system of ABCBest1 was designed so that money credited to an account had only one entry point, an accounting department data entry process. The security needed to process cash payments tended to be procedural based with multiple money handlers/balancers along with "checks and balances" processes to guard against fraud. Check processors were usually MICR based readers that fed accounting processes which prepared checks for deposit, created check sums and credit accounts. These processes were cross balanced to assure integrity and deter fraud through embezzlement. Security based controls were centered on authorization to the network and access to the applications. The controls were either at the system access control level to data (e.g. read, write, and update) or to specific applications, which restricted function by their design.

By its nature, the design addressed confidentiality by limiting access to the employees who needed to process payments and work with the various accounting processes. Integrity of payments and their proper handling was also addressed through the checks and balances. Fraud was limited to overdrawing an account or falsely using another's check. Availability was addressed at the enterprise level. Overall, it was not a high risk process.

Identifying the problem, vulnerability, and risk

ABCBest1's decision to accept credit cards for payment of its products created the need to modify the security environment to allow the organization to accept and process card not present credit card transactions as a method to purchase a product or service.

Whenever a new application is required, the first step involved in the risk assessment process is that of identifying the new information. (Actually, reviewing existing information in light of a new application may be prudent.) This new information must be studied through the process of asset classification. , One of the ten security objectives contained in ISO17799, based on the British Standard BS7799, is "Asset Classification and Control". This section focuses on the categories of confidentiality, integrity, and availability (CIA). At ABCBest1, the model used to classify assets is based on this standard and involves answering questions with regard to these CIA categories. These questions center on terms of fraud, competitive advantage, employee moral, additional costs and others. Upon answering these types of questions for each category, the security practitioner is able to identify the area in which to focus efforts to mitigate the risks inherent to this information.

In this case, the new information included credit card number, expiration date, and customer name. When assessing this information, it was discovered that the three pieces of information, when used together, provided the holder of that information a tool to commit fraud. Disclosure of this information beyond the scope of its intended use also created a potential legal problem for the custodian of the information. With these types of breaches becoming increasingly common in the news media, it became intuitive that **confidentiality**, disclosure of information only to individuals with a business related need to access the information, of this sensitive credit card information emerged as the primary objective to address.

Another issue, **integrity**, the assurance of the accuracy of the information, rose to the surface, since the new information needed to be accurate in order to apply the payment to the account while being able to collect the funds from the issuing member banks.

Finally, **availability,** the requirement of the application/information to be active and accessible when needed, was not to be overlooked since the entire process, beginning with the internet application and flowing through the network to the issuing bank, needed to be available in order to transact business.

The vulnerability in a global sense was the network, which included not just the internal servers and network connections but also the access to the information given to people, the Internet, network administrator capabilities, and external business relationships. A major task was breaking these vulnerabilities down into smaller, addressable pieces. Following this task, each of the vulnerabilities needed to be assessed and mitigated.

The threat was assessed as external as well as internal. Since business would be transacted over the public network, the Internet, the threat of hackers attempting to intercept the information was clear. Since it has been shown that internal risks of personnel gaining access to the information are even greater than external, the controls to protect the information needed to continue on the internal network as well.

Risk is the uncertainty of and potential loss of information or the accuracy of the information due to the CIA factors. In this case, the information at risk was the credit card number and expiration date. If an unauthorized party gained access to this information, there was a risk of fraud. If this information were altered intentionally or accidentally, a risk of not crediting the appropriate customer account or not approving a potential sale existed. The higher the risk, the more attention was given to mitigating the risk.

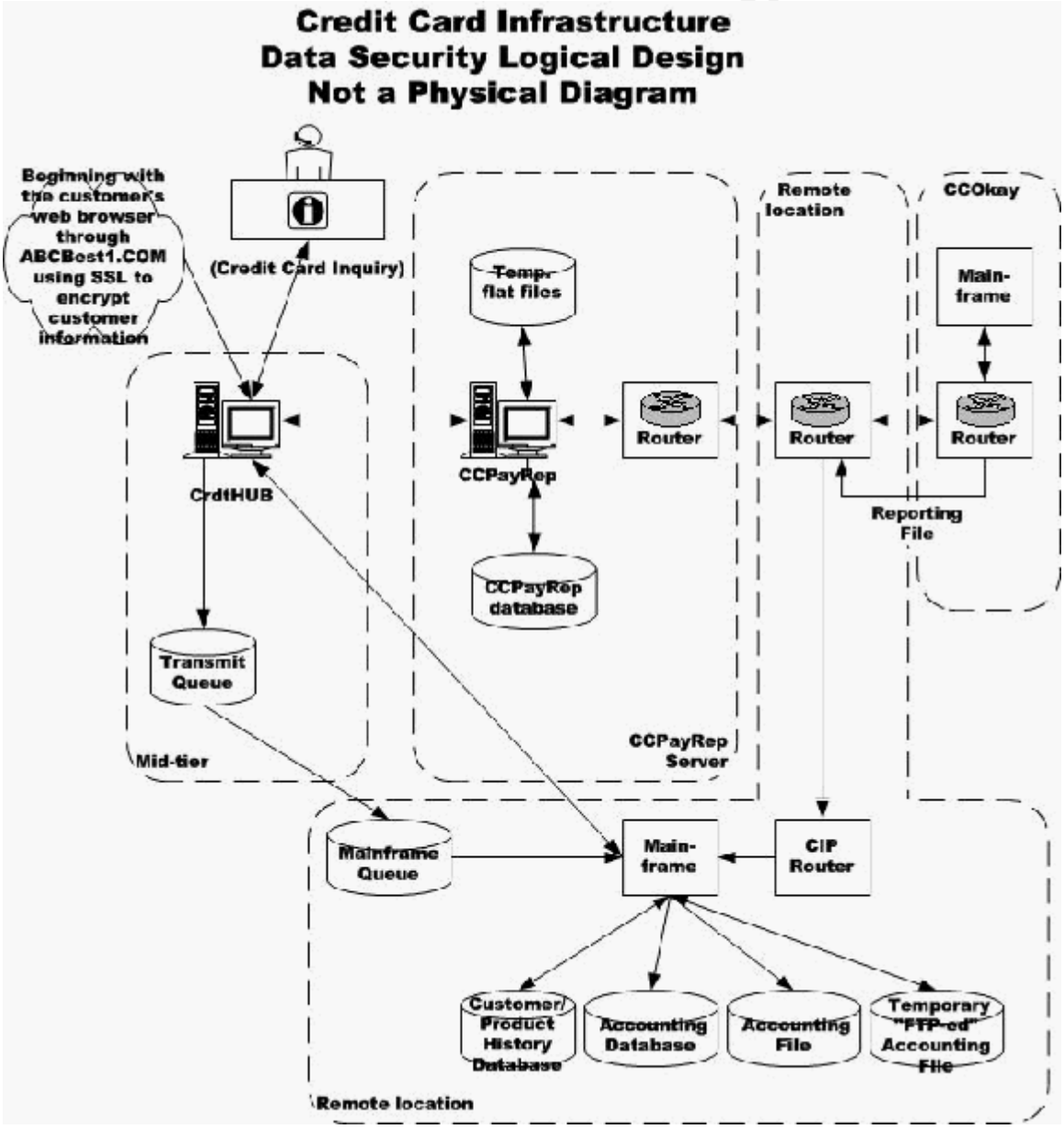
Approaching and solving the problem

The need to identify the exact path that the data traveled was crucial to providing the appropriate security solution. It was mandatory that a relationship of understanding and trust be established with the system designer/network designer and lead developers/programmers. The need to establish this working relationship became even more evident as the development progressed, since some of the security solutions had to be implemented at the program level.

Security awareness is not just a high-level, thou-shall-not-post-passwords-on-the-terminal advertisement at the corporate level by the executive offices. It is conceptual and practical information to be shared with and taught to project members. If the project members understand why security is important and how it can help maintain and increase company profitability, they are much more likely to help look for vulnerabilities, especially where the average security practitioner does not have the knowledge to tread--inside the developed program code. The developers must “buy in” to the need for secure data handling processes in order to alert the security practitioner with enough lead time to make changes to the security design and respond to changes in direction of the programming

solutions. Security practitioners must help create this cooperative "we-are-all-on-the-same-side" environment by providing a willing-to-mitigate, willing-to-work-together spirit.

Following the identification of the schematic flow of data, the technical details as to the hardware and software approved/available/needed was researched and identified. Since the entire credit card processing application involved communicating from external entities on the front end (customers providing credit card information to purchase goods/services) and external entities on the other end (credit card purchase approving/processing companies), a mutually accepted method to securely communicate with these business partners needed to be identified.



Because ABCBest1 already had an established Internet presence, the network was already set up with a secure communications process, 128 bit SSL. The other end required some negotiation with a vendor. The function of the vendor (hereafter referred to as the fictional company CCOkay) was to receive some credit card transaction information from ABCBest1 to determine if the customer had enough credit available to complete the transaction. In addition, at the end of every day, a file of accumulated credit card transactions was to be transmitted via the file transfer protocol process (FTPed) to CCOkay for settlement.

Since the data sent to CCOkay was very similar to the data received from the customer, the handling requirements were also the same, encryption at least the 128 bit strength level. CCOkay was not set up to handle communications with ABCBest1's 128 bit requirement. The systems designers pointed out that other external relationships with which this sensitive information was shared already existed. With some of these relationships, ABCBest1 routers were deployed at external locations with leased line connections back to ABCBest1. CCOkay agreed to this solution since the impact to their company was minimal. Additionally, IPSEC 3DES router to router encryption on this connection was implemented.

Once the credit card transaction information was received from the Internet web server, it was routed to, still protected by SSL, an internally developed program (hereafter referred to as the fictional program named CRDTHUB). One of the purposes of CRDTHUB was to format the payment transaction to be sent to the third-party, NT-based, payment processing software (hereafter referred to as the fictional third-party product CCPayRep). Another purpose was to send the payment transaction to the OS/390 (mainframe) platform to be warehoused in a database for future needs of inquiry or refunds/reversals. The function of CCPayRep was to receive the transaction from CRDTHUB, format the information in order to communicate with CCOkay, and send the information to CCOkay for verification that the funds were available in the cardholders account to make the purchase.

CCPayRep came with an application program interface (API). This API was to be used by CRDTHUB when passing the transaction information to CCPayRep for the verification process. In actuality, the performance of this application was so poor, that the internal development area of ABCBest1 replaced the API with internally developed code that would make a call to CCPayRep using an industry standard version of SSL, 128 bit. A corresponding module was developed for the front end of CCPayRep to decrypt the information in order for CCPayRep to send the information out for available fund verification.

As discussed above, CCOkay expected and required the information unencrypted. For this reason, CCPayRep sent the transaction unencrypted to CCOkay. To mitigate the risk this created, a router to router encryption schema was added to protect the data path from the point the data left CCPayRep to the

point where the data was released to CCOkay's control. Once CCOkay approved the funds, a reply was returned across the same router to router connection to CCPayRep. In actuality, this return path did not require encryption since no credit card information was included as part of the reply.

When CCPayRep received the reply, it performed two functions. First, it created a transaction record, which was to be used later to create an end of day settlement file. To prevent unauthorized access to this information, CCPayRep stored the information on its privately-controlled database using CCPayRep's proprietary version of SSL. Secondly, it forwarded the reply to CRDTHUB across the encrypted path discussed earlier.

Business requirements also included updating a customer/product history database residing on the mainframe platform, a progress tracking database to help resolve problems for transactions in progress, and accounting and reporting databases on the mainframe. The multi-platform customer inquiry application required access to the customer/product history database residing on the mainframe platform, which was used to interrogate the database to resolve customer-initiated inquiries.

When determining how to send information to the customer/product history database, no business requirement of immediate updates existed. A short time lag was acceptable. Therefore, the technical side of the project team decided to use MQSeries. CRDTHUB used MQSeries to send information from the NT platform to the mainframe platform using a series of queues.

Naturally, as a security analyst, this writer saw another place where this sensitive data would be exposed and made a series of attempts to contact IBM MQSeries experts to discuss encryption techniques available to be used in MQSeries. The contact was successful, but the news was not positive. Although there was an encryption capability built in to MQSeries, it forced all the information within the data stream to be encrypted and unpredictable results were obtained when translating data from an ASCII based platform to an EBCDIC based platform and back again. To complicate the issue further, this OS/390 based customer/product history database was going to be used for the customer inquiry application. This meant that the information had to be searchable and readable. Therefore, encrypting the entire data stream was not the solution of choice and ABCBest1 was forced to find a solution to encrypt only the sensitive information.

Because of its recent popularity and marketing hype, Public Key Infrastructure (PKI) was examined first. Since ABCBest1 already had a project team assigned to implementing PKI, internal resources were available to be utilized. After numerous meetings and attempts at prototyping a solution, it was quickly realized that the PKI solution was clunky and caused the creation of vulnerability. When encrypting the same information two different times using PKI, the result produced two different encrypted values, and, therefore, could not be used as the

search criteria by the customer inquiry application. The vulnerability was to allow the information to be searched; a large portion of the credit card number (last eight of the sixteen total digits of the credit card number) needed to remain unencrypted in order to provide enough uniqueness. This solution was a possibility but not preferred.

While working on this solution, ABCBest1 asked its database administrator and developer to write the application solution so that the impact to the application code was minimized when the ultimate solution was discovered and retrofitted into the code. As a part of an interim solution, the developer wrote a Caesar cipher to protect the credit card information stored on the mainframe. Although very elementary to decipher, at first glance it appeared as if it were encrypted. Surprisingly, this complied with the credit card issuer's security requirements. However, ABCBest1 was not comfortable with the solution and continued to pursue a better alternative.

Shortly after the application reached the production environment, a more desirable solution was discovered. This solution was to use Microsoft's crypto API. Crypto API enabled the encryption to be on a single field, namely, credit card number. In addition, each time crypto API encrypted the credit card number, the result had the same encrypted value. Therefore, the need to carry a portion of the credit card number unencrypted was no longer necessary. Now the inquiry application encrypted the credit card number (given to the customer service representative by the caller/customer) and the result was used as a search criterion to select the purchase transaction from the customer/product history database. Since there was no other use for this credit card number on this platform (OS/390), no ability to decrypt was needed, thus, eliminating the need to address the security of the process or the key on the mainframe.

Another area that required attention was the CCPayRep application. As mentioned earlier, the CCPayRep application maintained its own proprietary encryption routine to use in housing each transaction made that day. However, prior to the end of day settlement process, the entire days worth of transactions were accumulated in an unencrypted form within an NT based flat file. This file was then FTPed to CCOkay for settlement. The exposure that existed was retaining this flat file for five days, in unencrypted text. This provided a nice easy place for a person to go to capture five days worth of credit card transactions from which to steal credit card numbers. The risk was mitigated by designing an application that monitored an NT based folder, which is where the flat files were created, looking for a certain file extension. The application would then call the crypto API-based encryption routine to encrypt the entire file. This resulting file was then given a new file extension and replaced the source file.

This trigger which initiated the process was in the way CCPayRep processed the flat file. When CCPayRep created the file, it assigned one extension. Upon completing the FTP transfer, CCPayRep reassigned the extension so that it knew

it had completed the transfer. This second extension is what ABCBest1's application looked for. Whenever the application discovered one of these extensions, the encryption process took place. Therefore, upon completing the file transfer and encryption processes, only one file remained which was the encrypted result file. In the event the file had to be retransmitted, the file would be de-encrypted and the file extension would be changed back to its pre-FTPed value. (This was important because only a couple of non-IT business partners have the authority to execute the de-encryption process and only when directed to by their management.) Then the entire automated process started over, ending with the file being encrypted again.

Following the transfer of the settlement file, a reporting file was created by CCOkay and transmitted back to ABCBest1. The difference from the router to router environment used by the approval and settlement processes was that this file had to be routed to the mainframe platform instead of to the NT platform. The file contains information to be used as input to the accounting process to balance purchases to payments process. This reporting file also possibly contained information about charge backs, disputed amounts in which the customer contacts the member bank instead of working with the company which originated the charge. This situation raised yet another place where a credit card number was exposed.

At this point, ABCBest1 had little experience with encryption to an OS/390 directly from a network router. Upon consulting with ABCBest1's network, router, and mainframe specialists, it was discovered that this could be solved with relatively low cost. The company already owned a mainframe channel attached router (CIP router) capable of receiving encrypted traffic. Other than labor, the only cost was to increase the memory capabilities on the router and add an additional process card to handle any performance issues to the potential additional load on the router due to encryption. Once the information entered the mainframe, a job ran to strip off any charge back credit card numbers and wrote them to a paper report to be followed up manually. Naturally, procedures for safe handling and disposal of these paper documents containing the credit card numbers had to be created. After the credit card numbers were stripped off, the information was absorbed into the accounting process and its established controls. No further work with accounting or following applications was done at this point.

The customer inquiry process was a stand alone process designed for the customer services call center representatives to use in the event a customer wanted to inquire of the status of a purchase, question a charge, or requested a refund. This process had to be designed to accommodate a large number of call center representatives while not negating all the mitigated controls established to process the credit card information throughout this credit card processing system. This was accomplished using standard NT groups and access permissions to the inquiry process.

Options to search for a completed credit card transaction residing on the customer/product history database visited included searching on last four digits of credit card number along with expiration date, searching on last eight digits of credit card number along with expiration date and amount of transaction, and searching on reference number. All of these options had issues of call center representatives being able to discover credit card information with the possibility of allowing them to personally profit from them. (Random number combinations of a subset of the credit card number could be entered in hopes of retrieving a number of matches with the entire sixteen positions of the number in readable form.) In addition, call centers were known for high turnover, which increased the probability of fraudulent activity.

A fourth option, to allow the call center representative to search for a transaction, was discovered and selected as the solution. This option used the same encryption technique as used in the MQSeries process mentioned above. When the customer called in, he/she provided the full 16-position credit card number (without expiration date). This number was then encrypted within the inquiry application. The encrypted value was used as the search value to find the transaction in the customer/product history database. Only exact matches were returned. Expiration date was never displayed. The advantage of using this method was that the call center representative could not do random generic searches for valid credit card numbers while at the same time receiving the transactions about which the credit card holder was inquiring. This solution also set the stage for electronic reversals and refunds initiated through this application.

Reflecting on the resolution

These solutions did not come easily. When the application was first implemented, many areas of compromise existed. In the interest of making the functionality of using credit cards to purchase products/services, most of the encryption solutions just discussed were not in place. In spite of the requirement of "end to end" encryption of credit card information, the implemented controls were the mitigated security controls that were in place by the pre-established production date. This left ABCBest1's security analysts rather uncomfortable.

Consequently, they worked together to create a document outlining the exposures, possible ramifications, and mitigation options and requested a meeting with the middle management over the business area responsible for this application. After explaining the document, starting with the information involved and possible risks if the information was compromised, they requested the management signoff. The risks included potential lawsuits arising from the fraudulent use of customers' credit card information held in ABCBest1's custodianship, inaccurate application of payment, additional costs to investigate problems, and fines assessed by the credit card company for non-compliance to

credit card handling regulations. Once the management understood what was written in the document and the amount of risk that they were accepting, they were open to the pursuit of more acceptable security based controls and the retrofitting of these into the application. That one meeting opened the door to a follow up work order with an approved scope and budget.

Many valuable lessons were learned through this process, one of which was that when business management was properly informed of the information risks, the security analyst gained an advocate to protecting the company information, not only for one project/ application, but for the future as well. Another important lesson was to make sure that the various security practitioners agreed in front of the project team. It is easy to have our favorite solutions to exposures; however, in many cases more than one solution that will mitigate the vulnerability exists. If the views of other security practitioners are listened to, the ability to choose the most appropriate solution for a given situation increases. These differences should be used as possible alternatives, but the security team must present itself as united when approaching the project team. This will result in a better reputation for information security as a whole and much more cooperation during future projects.

Conclusion

During the past ten years, it has been this writer's observation as a security practitioner that the technical environment has become increasingly complex. In the past, the mitigated security solutions were usually implemented on a single computer platform. Occasionally, physical security came into play. Integrity of the information was controlled by platform security or using separate transactions. Today, the technical environment is complex. Understanding one security implementation (e.g. NT security) will no longer suffice as a knowledge base for the security practitioner. The security professional must be well versed in understanding network topologies, vulnerabilities to internal networks as well as external connections, available mitigation solutions, and the soft skills to be able explain the concepts to the business partners.

This project, to implement a secure infrastructure to accept credit cards for payment, used many dimensions of the security professional's repertoire. This project required the knowledge of risk assessment techniques, NT security administration tasks and commands, various encryption solutions along with available APIs, mainframe security administration commands, and router capabilities. One area which should not be minimized was the use of communication skills when working with the business partner. Once the business partner understood risks and the mitigation solutions, the entire task for the security practitioner was reduced to the technical dimension. The willingness to implement alternate solutions and to provide a lower than desired

level of security while waiting for the ultimate solution contributed greatly to creating a cooperative environment with the business partner.

Finally, it is important to build a network of security knowledgeable individuals. Technology is evolving at break neck speed and it is nearly impossible for one person to keep up with all of the changes. Knowing whom to contact to help with technical details for various challenges contributes to timely solutions.

References

Books:

Fisch, Eric A. and Gregory B. White. Secure Computers and Networks: Analysis, Design, and Implementation. Boca Raton, Florida: CRC Press LLC, 2000.

Tipton, Harold F. and Micki Krause, ed. Information Security Management Handbook. 4th edition. Boca Raton, Florida: Auerbach Publications, an imprint of CRC Press LLC, 2000.

URLs:

Crypto API white paper:
http://online.securityfocus.com/library/899

ISO17799:
http://www.iso-17799-security-world.co.uk/what.htm

MQSeries and encryption:
http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/SG245306.html?Open

Standard of good practice:
http://www.isfsecuritystandard.com/index_ie.htm

VISA secure credit card handling requirements:
http://www.usa.visa.com/business/merchants/cisp_requirements.html

Last Updated: January 25th, 2020

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Las Vegas 2020	Las Vegas, NVUS	Jan 27, 2020 - Feb 01, 2020	Live Event
SANS Vienna January 2020	Vienna, AT	Jan 27, 2020 - Feb 01, 2020	Live Event
SANS Security East 2020	New Orleans, LAUS	Feb 01, 2020 - Feb 08, 2020	Live Event
SANS New York City Winter 2020	New York City, NYUS	Feb 10, 2020 - Feb 15, 2020	Live Event
SANS Northern VA - Fairfax 2020	Fairfax, VAUS	Feb 10, 2020 - Feb 15, 2020	Live Event
SANS London February 2020	London, GB	Feb 10, 2020 - Feb 15, 2020	Live Event
SANS Dubai February 2020	Dubai, AE	Feb 15, 2020 - Feb 20, 2020	Live Event
SANS Cairo February 2020	Cairo, EG	Feb 15, 2020 - Feb 20, 2020	Live Event
SANS San Diego 2020	San Diego, CAUS	Feb 17, 2020 - Feb 22, 2020	Live Event
SANS Scottsdale 2020	Scottsdale, AZUS	Feb 17, 2020 - Feb 22, 2020	Live Event
SANS Brussels February 2020	Brussels, BE	Feb 17, 2020 - Feb 22, 2020	Live Event
Open-Source Intelligence Summit & Training 2020	Alexandria, VAUS	Feb 18, 2020 - Feb 24, 2020	Live Event
SANS Training at RSA Conference 2020	San Francisco, CAUS	Feb 23, 2020 - Feb 24, 2020	Live Event
SANS Jacksonville 2020	Jacksonville, FLUS	Feb 24, 2020 - Feb 29, 2020	Live Event
SANS Secure India 2020	Bangalore, IN	Feb 24, 2020 - Feb 29, 2020	Live Event
SANS Manchester February 2020	Manchester, GB	Feb 24, 2020 - Feb 29, 2020	Live Event
SANS Zurich February 2020	Zurich, CH	Feb 24, 2020 - Feb 29, 2020	Live Event
SANS Northern VA - Reston Spring 2020	Reston, VAUS	Mar 02, 2020 - Mar 07, 2020	Live Event
SANS Munich March 2020	Munich, DE	Mar 02, 2020 - Mar 07, 2020	Live Event
SANS Secure Japan 2020	Tokyo, JP	Mar 02, 2020 - Mar 14, 2020	Live Event
ICS Security Summit & Training 2020	Orlando, FLUS	Mar 02, 2020 - Mar 09, 2020	Live Event
Blue Team Summit & Training 2020	Louisville, KYUS	Mar 02, 2020 - Mar 09, 2020	Live Event
SANS Jeddah March 2020	Jeddah, SA	Mar 07, 2020 - Mar 12, 2020	Live Event
SANS St. Louis 2020	St. Louis, MOUS	Mar 08, 2020 - Mar 13, 2020	Live Event
SANS Dallas 2020	Dallas, TXUS	Mar 09, 2020 - Mar 14, 2020	Live Event
SANS Paris March 2020	Paris, FR	Mar 09, 2020 - Mar 14, 2020	Live Event
SANS Prague March 2020	Prague, CZ	Mar 09, 2020 - Mar 14, 2020	Live Event
Wild West Hackin Fest 2020	San Diego, CAUS	Mar 10, 2020 - Mar 11, 2020	Live Event
SANS Doha March 2020	Doha, QA	Mar 14, 2020 - Mar 19, 2020	Live Event
SANS Secure Singapore 2020	Singapore, SG	Mar 16, 2020 - Mar 28, 2020	Live Event
SANS Norfolk 2020	Norfolk, VAUS	Mar 16, 2020 - Mar 21, 2020	Live Event
SANS London March 2020	London, GB	Mar 16, 2020 - Mar 21, 2020	Live Event
SANS San Francisco East Bay 2020	OnlineCAUS	Jan 27, 2020 - Feb 01, 2020	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced