Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

The Risks Involved With Open and Closed Public Key Infrastructure

Over the past couple of decades, on-line communication, especially electronic mail and on-line shopping, has changed the way that people transfer sensitive information to and from each other. As long as these methods of communication will be used, there needs to be a way to keep this information secure. One solution to help us solve this problem is Public Key Infrastructure (PKI). There are two types of PKI models: open and closed. Each one has its advantages, but there is a certain level of risk and liability involved...

Copyright SANS Institute
Author Retains Full Rights

AD

Automate more Web application scanning.
watchfire

The Risks Involved With Open and Closed Public Key Infrastructure

Philip Hlavaty
GIAC GSEC Practical Assignment
Version 1.4b
February 24, 2003

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Abstract

Over the past couple of decades, on-line communication, especially electronic mail and on-line shopping, has changed the way that people transfer sensitive information to and from each other. As long as these methods of communication will be used, there needs to be a way to keep this information secure. One solution to help us solve this problem is Public Key Infrastructure (PKI). There are two types of PKI models: open and closed. Each one has its advantages, but there is a certain level of risk and liability involved with each model. This paper will provide a basic overview of PKI and its components. It will then discuss the advantages and disadvantages of both the open and closed PKI models. Finally, this paper will present some of the risks and liability issues involved with PKI. In particular, it will discuss the enormous risks behind the open PKI model and why it never flourished in the marketplace.

Goals of PKI

Before we can discuss the risks involved with Public Key Infrastructure, we must first understand how it works and what it is intended to accomplish. PKI in and of itself is not a technology. Rather, it is a method of deployment by which other security systems are founded. It is a basis by which several different components, such as digital signatures and certificate authorities, can work together to provide a high level of Internet security. PKI does not authenticate and audit data; however, it does support these important necessities of electronic security. [10]

Every network security infrastructure should provide a solution to four major security problems: confidentiality, data integrity, data authentication, and non-repudiation. In order for an infrastructure to ensure *confidentiality*, it must ensure that the information being transmitted is not disclosed to any unauthorized users. *Data integrity* is achieved when the transmitted information is not tampered with or altered in any way. If users can verify the identity of the sender, then they have achieved *data authentication*. Finally, a secure infrastructure will ensure that a receiver can undeniably prove that information came from a particular sender. This is known as *non-repudiation*. [10] The next section will describe the components that PKI uses to address these security problems.

Components of PKI

Public Key Infrastructure is composed of a variety of components that work in unison to ensure secure communication. In order to understand how PKI as a whole works, we must first understand how each component works and how they are related to each other.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Digital Certificates and Key Pairs

PKI is based on data encryption that is the result of a pair of encryption keys. One key is considered to be "public" because it is widely available for others to use. The other key is considered "private" because it is accessible only to a certain individual. It is the responsibility of this individual to make sure that his private key stays hidden from the public. This is the fundamental building block of PKI. [8]

If you encrypt data using the public key, only the private key can decrypt it, and vice versa. A user can freely send out his public key to other users because he knows that data encrypted with his public key can only be decrypted with his private key. [2] This type of data retrieval is considered to be much more secure than just a simple password because it requires the knowledge of both the public and the private key. A password, on the other hand, could be more easily guessed than a pair of encryption keys. [9] The public/private key pair ensures *confidentiality* because unauthorized users cannot determine the contents of the information unless they obtain knowledge of both keys.

The main problem with this scenario is how to ensure that the sender of a public key really is who he says he is. This problem is solved with digital certificates. A digital certificate is an electronic document that declares a public key holder is who he claims to be. These certificates act as online identification. [3] Digital certificates handle the *data authentication* problem because they are used to determine who sent a particular message.

Certificate Authorities

A Certificate Authority (CA) issues digital certificates. It is also responsible for the generation, distribution, and management of public keys. In order to obtain a digital certificate, a user will establish a trust relationship with a CA. The CA will then authenticate the user according to guidelines in the CA's Certificate Practices Statement (CPS). The user is then issued a digital certificate. [10] Even though individuals may not directly trust each other, they can establish an indirect trust relationship through a CA.

Most CAs are composed of a number of components, but the two most important components of a CA are the Registration Authority (RA) and the certificate repository. The RA performs most of the administrative tasks of a CA. It registers users for a digital certificate, and it verifies all the information that goes into a digital certificate. The RA may even perform diligence tests on a user to keep the information up to date. [10] The certificate repository is a public database that keeps a record of current certificates and revocation lists. Keeping updated revocation lists is critical because certificates need to be deactivated if users no longer need them or if a user's private key has been compromised. The certificate repository ensures that only legitimate digital certificates and key pairs are being used. [2]

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Digital Signatures and Hashing Algorithms

Digital signatures and hashing algorithms accomplish two of the four PKI goals. They ensure the *integrity* of the information being sent, and they solve the *non-repudiation* problem by not allowing the sender to dispute that he was the originator of the sent message. In order to understand how digital signatures and hashing algorithms accomplish these goals, let's look at how they work.

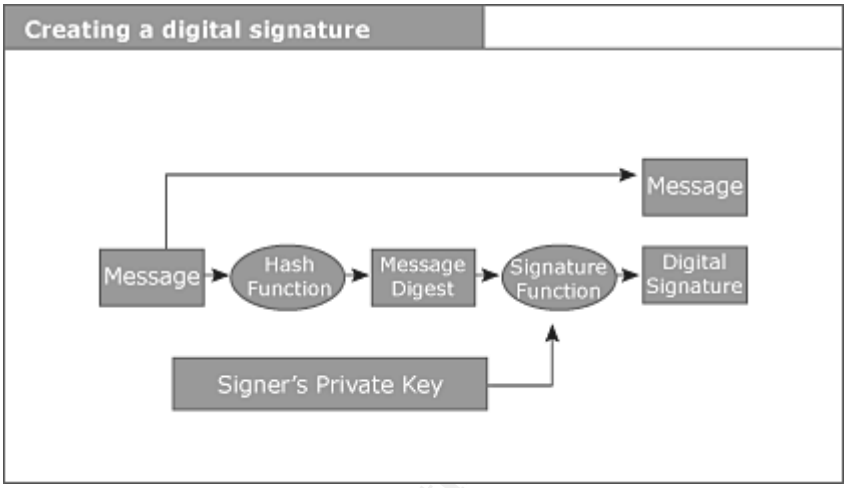

Figure 1 – Creating a Digital Signature. Taken from http://www.digsigtrust.com/images/pki_2.gif

Figure 1 is a visual representation of how a digital signature is created. First, a message is encrypted with a hash algorithm, which is an extremely complicated mathematical equation. A hash is known as a one-way algorithm because once a message has been encrypted, there is no way to decrypt the message. Some common examples of hash algorithms are MD5 and SHA-1. If the SHA-1 hashing algorithm is used, the result is a 160-bit string of digits known as a message digest. In the case of MD5, the message digest is 128 bits long, but it is just as secure as the SHA-1 message digest. Each message digest is unique to the original message.

Next, the message digest is encrypted with the sender's signature function. The sender's private key is incorporated into the signature function. The result is a digital signature. A digital signature will be different for every document that is signed, even if it is signed with the same private key. This is because the message digest is different every time. In order to ensure the confidentiality of the original message, it is encrypted with the sender's private key. The recipient can then decrypt it with the sender's public key. When the digital signature is complete, it is included with the (encrypted) original message and the sender's public key and sent off to the recipient. [8]

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

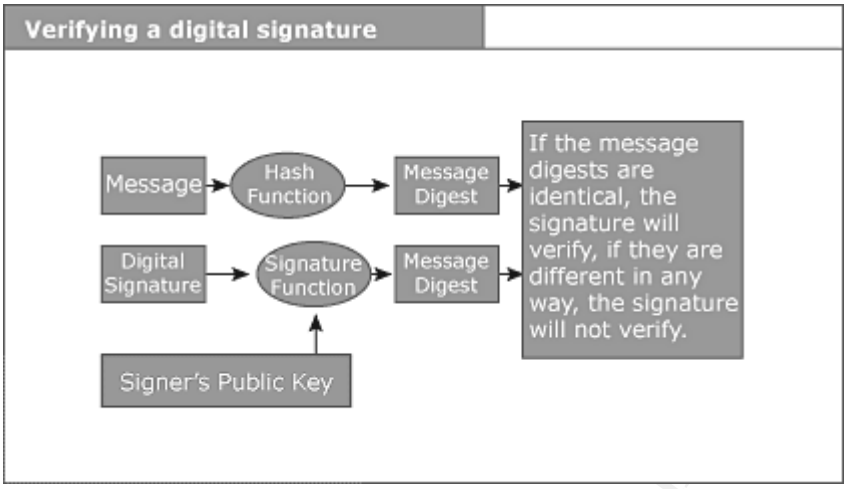

Figure 2 - Verifying a Digital Signature. Taken from http://www.digsigtrust.com/images/pki_3.gif

Figure 2 shows how the recipient of a message verifies the digital signature. The recipient will first decrypt the original message using the sender's public key to recreate the original plaintext. The user will then encrypt the original message with the same hashing algorithm that the sender used. This will recreate the message digest. Next, the digital signature is decrypted with a signature function. However, in this instance, the signature function consists of the sender's public key, not his private key. This will create another message digest. If the two digests are identical, then the signature is verified. As a result, the recipient knows that the message has not been altered in any way during transit (i.e., *integrity* has been assured). It also means that *non-repudiation* has been achieved because the sender cannot deny that he sent the message. If either of the message digests is off by just a single bit, then the recipient can safely determine that the message has either been altered or intercepted during transmission. [8]

Closed PKI vs. Open PKI

Once an organization decides to use a particular Public Key Infrastructure, it must decide exactly how to implement it. The organization must decide whether it wants to put the majority of the responsibility on an outside CA or if it wants to handle the situation itself. The next two sections will define closed PKI and open PKI, and they will discuss the advantages and disadvantages of each method.

Closed PKI

A closed PKI model can be described as "a contract or series of contracts that identifies and defines the rights and responsibilities of all parties to a particular transaction." [1] Every desktop that participates in a closed PKI architecture must have the proprietary PKI software installed on it. Without this software, machines cannot ensure that data transfer is secure.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

In a closed PKI architecture, a proprietor issues certificates within a specific, bounded context. For example, the manager of an online mall can act as his own CA. Thus, he can enter into contractual relationships with customers and merchants, and he can issue digital certificates to them. In this situation, the manager knows exactly how his certificates will be used. [1]

One major advantage that a closed PKI architecture has over an open one is risk management. In the above scenario, the manager creates his relationships under his own terms. Therefore, he can accurately predict the potential risks and losses he might face. Another major advantage of a closed PKI is the secrecy of its code. The code driving the PKI is not readily available to the public. As a result, it is safe to assume that the PKI is secure. However, this could be a potential drawback. Since the code is not put under public scrutiny, it is difficult to know whether or not the code and the components behind a PKI are truly safe.

Opponents of a closed PKI architecture claim that it puts a heavy burden on each desktop's owner because he is responsible for the proprietary software installation, problem troubleshooting, and the cost of maintaining and updating the software. They also argue that it is not easy to expand a closed PKI enterprise. For example, a system administrator could easily ensure that all internal desktops are equipped with the correct proprietary software to ensure secure communications. However, if the internal desktops are communicating with applications on external desktops, then the proprietary software must also be installed on the external machines. In a situation like this, maintenance becomes a nightmare. If the software on the internal machines needs to be upgraded, the external machines probably need to receive the same upgrades or else there could be compatibility issues. [5]

Open PKI

In an open PKI, the architecture is readily available for inspection by the public so that people can determine whether or not their systems are compatible with the PKI. Individuals make a partnership with a third-party CA to obtain a digital certificate, and they agree on the interface that is used to ensure secure communications. Since the PKI vendor handles the infrastructure, no proprietary software needs to be installed on the customer's desktops. [5]

In order for an open PKI solution to be successful, it must prove to be interoperable with the products and services of all of its clients. An open PKI solution should integrate seamlessly into a client's infrastructure so as not to disrupt any programs or services already in effect. [6] This kind of solution can be difficult to achieve because of the wide variety of business architectures that exist. Customers will more than likely have multiple applications already deployed, and the PKI must integrate smoothly into the existing architecture so that no services are disrupted.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The one major advantage that an open PKI architecture has over a closed one is that it can be easily expanded due to the fact that a third-party CA handles all of the installation and maintenance of the PKI software. However, the third-party CA's software may not integrate seamlessly into the customer's existing architecture. When a customer chooses to use an open PKI, he is taking a huge chance on whether the PKI will actually provide a viable security solution or if the PKI will cause more harm than good.

While an open PKI solution may take some of the responsibility away from the customer, it may also make it more difficult for him to assess the risk of potential monetary losses due to theft or fraud. In a closed PKI, a proprietor can issue digital certificates to whomever he wants. As a result, he can predict (and, in some cases, control) the losses his company may face. There is no way to make this assessment under an open PKI architecture. [1] The next section discusses many of the risks involved with PKI, especially the increased liability risks that are involved in an open PKI and why it may be in the best interest for a customer to choose a closed PKI instead.

PKI Risks

Total security of an electronic transaction is difficult, if not impossible, to achieve. Although PKI is designed to make electronic transactions secure, there are still instances when the architecture can break down and a security breach can occur. Most of these instances occur because of human error or carelessness. This section will take a look at some of the risks involved with PKI.

Private Key Protection

The cornerstone of PKI is the public/private key pair. In particular, the users' private keys are the most important components of PKI. If a customer's private key is stolen, the thief could use that private key to digitally sign documents and make it appear as though the customer signed them. If a CA's private key is stolen, the thief could use that key to generate numerous fraudulent digital certificates and issue them to unwitting customers.

So, then, how are these critically important pieces of information supposed to be kept safe? The most basic way to protect your private key is store it on your desktop and protect it with a password. However, as this paper has mentioned already, passwords can be easily guessed, especially if simple passwords like "abcdefgh" or "12345678" are chosen. [2] Another way to protect your private key is with the use of "smart cards." These cards are meant to be used once and then discarded. Each card that contains the private key has its own unique password. A user inserts the card into a desktop, logs on using the unique password, and does not have to log on again for the rest of the day. While this system is much more secure than a password-protected desktop, it does have its drawbacks. If a user were to step away from his desktop for a minute, the private key is left unprotected unless the user locks his computer. A more significant

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

drawback is the cost of using a smart-card system. Since each card is only used once, the cost of multiple cards can add up over time. Additionally, desktops must be equipped with card-reading hardware and software. [4] When faced with this kind of choice, most individuals would rather save money than take the extra step to increase security.

Trust

Although PKI is intended to provide a level of trust between individuals, carelessness and unforeseen circumstances can cause users to question the trust relationships they have with others. The previous section mentioned that an attacker could gain access to someone's private key, and he could then generate messages that appear as if the private key's owner signed them. In this type of situation, it would be almost impossible for the private key's owner to prove that he did not send the message because of how PKI is designed to solve the non-repudiation problem.

A user can take every necessary precaution to make sure that his end of the PKI architecture is secure, but he cannot ensure that other users (or even the CA) are doing their part. Did the CA do a thorough job of verifying the identity of all of its clients? Is the CA making sure that all outdated and compromised certificates are not being reused or recycled? How secure are the CA's computers? How secure are the other users' computers? These questions can certainly raise some doubt about the overall security of the PKI architecture. [4]

Open PKI Liability Risks

When it comes to e-commerce, fraud and theft are inevitable. Even though the PKI architecture is designed to prevent these things from happening, Internet crime will continue to occur. If all parties that participate in an open PKI act reasonably, then fraud and theft can be kept to a minimum. However, mistakes will happen, and major problems can occur when one or more parties act unreasonably (i.e., the proper precautionary procedures are not followed). Of course, when theft does occur, everyone wants to know who is at fault. This liability problem is one that is difficult to solve in an open PKI.

The main difference between closed and open PKI is risk management. In a closed PKI, liability allocation is very manageable because the potential liability exposure is limited in scope. Since the proprietor (who also assumes the role of CA) of a closed PKI controls who he makes contractual agreements with, he can accurately predict and prepare for potential losses. The proprietor can either absorb the losses himself, or he can pass it off to his customers. [1]

In an open PKI model, third-party CAs and their customers also form agreements through contracts. As such, each party should exercise some degree of "reasonable care" to ensure that the potential for loss is kept at a minimum. This last statement can be the basis of major problems for all parties involved in an open PKI because of the phrase "reasonable care." What constitutes reasonable

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

care, and how can any party of an open PKI be sure that all other parties are doing their parts to ensure security?

Certificate Authorities bear most of the responsibility in exercising reasonable care in an open PKI model because they are the centers of the architecture. As such, they should put forth the most effort of all parties. First of all, they should verify the identity of their customers. Although there should be some level of trust between customers and the CAs, the CAs need to take whatever steps necessary to deter fraudulent customers. Next, they should take extremely good care of their own private key. If an attacker managed to obtain the CA's private key, he could create an unlimited number of forged certificates. The most obvious way that CAs can protect their private keys is to make sure their internal systems are secure. Therefore, a CA needs to make sure that the vendors from whom they purchase their hardware and software are reliable and trustworthy. In order to make sure that they are holding up to the public's standards, CAs should publish a CPS that clearly states the practices they use to issue certificates. [7] Public scrutiny is a great way to make sure you are doing your job correctly.

Not all of the responsibility falls on the shoulders of the CAs. The customers of a CA should also take the necessary precautions to protect their own data. For starters, they can provide accurate information about their identity to the CAs. Anyone who fails to do so is acting unreasonably and could be held accountable if any crimes are committed. Customers should review their contracts and certificates to ensure there are no erroneous statements. They should frequently check the most up-to-date information provided by the CAs, especially the certificate revocation lists. Customers bear a huge amount of risk if they use certificates that are no longer supported by their CA. Most importantly, customers should ensure that their private keys are kept safe, and they should notify their CA immediately if there is reason to believe that a private key has been compromised. [7]

If a crime is committed as a result of one of the open PKI parties acting unreasonably, then that party should obviously be held accountable and should bear the liability. However, instances could occur when all parties involved in an open PKI act reasonably and losses are still suffered. Who bears the liability in a situation like this? In 1995, a group known as the Information Security Committee of the American Bar Association's Section of Science and Technology (the "ABA Committee") collaborated with the state of Utah in order to create a set of guidelines to allocate the liability involved with digital certificates. They enacted an influential model known as the Utah Digital Signature Act (the Utah Act).

Under the Utah Act, a government agency acts as a "top level" CA by which all other CAs must abide. This government-level CA creates policies and provides regulatory oversight for all CAs in the private sector. The Utah Act also requires that all parties abide by a set of guidelines that define "reasonable care." For

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

example, the CA must ensure that the information in a digital certificate is accurate while the customer cannot provide false information to the CA. [1] However, customers must be aware that even if they follow all the guidelines, they can still be held accountable for data and financial loss. If a licensed CA complies with all the requirements specified in the Utah Act with regards to false or forged digital certificates, it is not held liable for any losses a customer may face. Therefore, the customer is left with the burden of having to bear those losses. If a customer exercises reasonable care and is the victim of a crime, it is still his responsibility to provide "clear and convincing" evidence that his digital certificate was used under false pretenses. [7]

Although the Utah Act was meant to benefit and protect an open PKI architecture, it inadvertently had the opposite effect. Third-party CAs like Verisign never flourished because the Utah Act hindered many customers from wanting to participate in an open PKI. The risk of having to bear potentially huge losses that could not be reimbursed outweighed the rewards of having a digital certificate. Also, there was no safety net for the customers similar to the safety net provided to credit card holders by the Electronic Funds Transfer Act (EFTA). The EFTA says that if a credit card holder exercises reasonable care and his card is stolen, then he is not liable for more than $50 in losses prior to his reporting the stolen card. The Utah Act created no such safety net. Even though the Utah Act provided many safeguards for CAs, there was also the potential for CAs to suffer huge losses if they did not exercise reasonable care. Even the tiniest mistake could lead to crimes that would cause customers to suffer huge losses, and the public needed to be reassured that they would be reimbursed for those losses. [7] Customers decided to either forgo digital certificates or become their own CA in a closed PKI. It may take more time and effort to implement a closed PKI over an open PKI, but there is significantly less risk involved.

Summary

Although there is no perfect solution for securing the transfer of electronic data, Public Key Infrastructure appears to be an excellent solution on the surface. It combines the functionalities of public/private key pairs, digital certificates, and hashing algorithms in an attempt to provide confidentiality, data integrity, data authentication, and non-repudiation. Both the open and closed PKI models have their advantages, but the open PKI model introduces an increased liability risk that is not appealing to most users. In fact, these liability risks are the main reason that many third-party Certificate Authorities no longer exist. PKI is a well-designed architecture that has not been fully utilized, and, unfortunately, it probably never will be. However, it will likely provide a foundation for future security architectures.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Bibliography

[1] Biddle, C. Bradford. "Legislating Market Winners: Digital Signature Laws and the Electronic Commerce Marketplace." World Wide Web Journal, Volume II, Issue 3, Summer 1997. URL: http://www.w3j.com/7/s3.biddle.wrap.html. (28 January 2003)

[2] Conry-Murray, Andrew. "Strategies & Issues: Public Key Infrastructure Nuts and Bolts." Network Magazine, 5 November 2001. URL: http://www.networkmagazine.com/article/NMG20011102S0008/1. (26 January 2003)

[3] "Digital Certificates & Encryption." URL: http://www.spitzner.net/digcerts.html. (3 February 2003)

[4] Ellison, Carl, and Bruce Schneier. "Ten Risks of PKI: What You're Not Being Told about Public Key Infrastructure." Computer Security Journal, Volume XVI, 1 November 2000. URL: http://www.counterpane.com/pki-risks.pdf. (3 February 2003)

[5] "Enterprise Key Management." 2003. URL: http://www.esign.com.au/whitepapers/enterprise/pki/diff3.shtml. (27 January 2003)

[6] Just, Mike, Steve Lloyd, and Chris Voice. "Entrust's Open PKI Solution: Interoperability and Support Standards." Version 1.0. August 2000. URL: http://www.entrust.com/resources/pdf/open_pki.pdf. (28 January 2003)

[7] Hunter, Bruce, et al. "The Role of Certification Authorities in Consumer Transactions: A Report of the ILPF Working Group on Certificate Authority Practices." Internet Law and Policy Forum. 14 April 1997. URL: http://www.ilpf.org/groups/ca/draft.htm. (9 February 2003)

[8] "PKI Basics Digital Signatures and Public Key Infrastructure (PKI) 101." URL: http://www.digsigtrust.com/support/pki_basics.html. (27 January 2003)

[9] "Public Key Infrastructure (PKI) with Comtrust™: A Superior Way of Meeting HIPAA e-Security Requirements." 2002. URL: http://www.marcinkoadvisors.com/Services/IT5.pdf. (3 February 2003)

[10] Weise, Joel. "Public Key Infrastructure Overview." Sun BluePrints™ Online, August 2001. URL: http://www.sun.com/solutions/blueprints/0801/publickey.pdf. (5 February 2003)

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 23rd, 2013

Upcoming SANS Training

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

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced