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.

Distributed Computing: An Unstoppable Brute Force

Distributed computing allows groups to accomplish work that was not feasible before with supercomputers, due to cost or time constraints. Although the primary functions of distributed computing systems is to produce needed processing power to complete complex computations, distributed computing also reaches outside of the processing arena to other areas such as network usage. When used properly, both areas compliment each other and can produce needed results. When used maliciously, either processing or networking distr...

Copyright SANS Institute
Author Retains Full Rights

AD

Michael Hill

GSEC Certification

Pratical Assignment 1.4b

November 13, 2003

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Table of Contents

Abstract 2

Part I: Introduction 2

1.1 A Brief History of Distributed Computing 2

Part II: The Scale of Distributed Computing 3

2.1 Distributed Protein Folding 3

2.2 The Search for ET 4

Part III: The Negative Sides and Security Implications of Distributed Computing 5

3.1 Distributed.net: Cracking the World's Encryption One Key at a Time 5

3.2 Distributed Denial of Service Attacks 5

3.3 Case and Point: "*Crime gangs extort money with hacking threat*" 7

IV. The Future of Distributed Computing and Possible Solutions 8

4.1 Possible Solutions to Combat Distributed Attacks 9

V Conclusion 9

5.1 The Future of Distributed Computing 9

© SANS Institute 2004, Author retains full rights.

Distributed Computing:
An Unstoppable Brute Force

Abstract:

Distributed computing allows groups to accomplish work that was not feasible before with supercomputers, due to cost or time constraints. Although the primary functions of distributed computing systems is to produce needed processing power to complete complex computations, distributed computing also reaches outside of the processing arena to other areas such as network usage. When used properly, both areas compliment each other and can produce needed results. When used maliciously, either processing or networking distributed attacks can produce a brute force that even the best firewalls or encryption are powerless to prevent. Using distributed computing brute force attacks on encryption algorithms, distributed denial of service attacks, distributed reflective denial of service attacks, and other future forms of malicious attacks, there is much to guard against with these types of computer usage. Distributed computing should be tamed and closely guarded against such uses through efforts to filter out invalid network packets for distributed attacks, and carefully monitoring computer software to ensure that a distributed computing processing, brute force attacks cannot occur.

I. Introduction

1.1 A Brief History of Distributed Computing

During the earliest years of computing, any tasks that required large computations and massive processing were generally left up the government or a handful of large companies. These entities could afford to buy massive supercomputers and the infrastructure needed to support them. With the price of personal computing declining rapidly in price, and supercomputers still very expensive, an alternative was needed. In 1993, Donald Becker and Thomas Sterling introduced Beowulf clustering. Although not the first example of clustering, this was the first time that an effort was made to enable anyone to take off the shelf computers and build a cluster of computers that could rival top supercomputers. The concept behind clustering, in its simplest form, is that many smaller computers can be combined in a way to make a computing structure that could provide all of the processing power needed, for much less money. All of the nodes of a cluster are connected to an isolated internal network and same switch as the serving computer (SC). The serving computer houses the results and distributes new work units to all of the attached nodes. Each node is a single-use computer, allowed to only process the problem that it is given and return the results when finished. Many of the problems that hindered the first clustering efforts still cause problems today. One of the more expensive elements is the dedicated internal network, or interconnects, which link all of the nodes together to the server. Since these nodes are simple banks of processors, security is almost entirely nonexistent and therefore requires a great care in isolating the interconnected network from any outside networking. An additional element contributing to problems is that of suitable software for the clusters to run. Even though all of the nodes work together to process chunks of a complete dataset, the software must still be written to take advantage of the multiples of processors and the individual resources, such as memory, that the nodes contain. To obtain a stable and suitable software layer, it may

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

take months or years to perfect the software to properly process the needed results and use all of the available power. In the end, taking a major step forward for computing power, clustering certainly has its problems and insecurities as a relatively new technology. With distributed computing, it manages to encompass a much wider scope than clustering by allowing nodes to exist anywhere in the world and also be multi-purpose/multi-function machines.

Distributed computing has a similar concept as clustering: take a large problem, break it into smaller units, and allow many nodes to work on the problem in parallel. Where distributed computing strays from this concept is by also allowing the nodes to be multifunction and multipurpose computers that can exist anywhere in the world while being attached to the Internet. Distributed computing can actually take on many different orientations of the nodes, all depending on how the client computers are connected to the Internet. In addition to this element of flexibility, there is also a level of redundancy that does not exist in supercomputing or clustering. With clustering and supercomputing, the data is generally processed only once, due to the large amounts of time that the entire project may take. In distributed computing, it is often the case that work units may be distributed multiple times to multiple nodes. This method serves two functions: to drastically decrease the possibilities of processing errors, and to account for processing that is done on slower CPUs or takes too long to return results. What makes this entire system possible is the application of a small piece of software called a client. This client handles the data retrieval and submission stages as well as the code necessary to instruct the CPU how to process the work unit. Clients vary in side, but most are less than 1-2 megabytes in size. The actual data work units also vary in size, but most are between 250 and 400 kilobytes, so that the hosting/node CPU can handle the process, and users on slower internet connections can easily send and receive the data. Between the small sizes of work units and clients, it seems unreasonable to see a disadvantage to using distributed computing other than the collection and analysis of data. To understand further the actual scale of distributed computing, several real-life examples will be used to detail the actual power that these large networks can create.

II. The Scale of Distributed Computing

2.1 Distributed Protein Folding

Distributed computing can be a tremendous tool for any research or other noteworthy purpose when it is used for positive goals. To give an illustration of the better uses of distributed computing, a group from Stanford University is currently running a distributed system to work on the problem of protein folding. The goal of the project is to try and determine the function of proteins by how they form, or 'fold'. The simple version is that proteins assemble themselves into certain 'folds' which dictate what their function will be. When these proteins fail to fold properly, they result in such diseases as "…Alzheimer's disease, cystic fibrosis, BSE (Mad Cow disease), an inherited form of emphysema, and even many cancers are believed to result from protein misfolding." (Stanford) Continuing from the blueprint of DNA, which specifies the sequence of amino acids, these scientists are taking research to the next level by trying to figure out how proteins, strings of amino acids, form and function. To accomplish this goal, the research group has enlisted the help of their own distributed network people, the Pande Group, to

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

create Folding@Home. Distributed computing enters this challenge by simulating the various folds that protein can form. The concept doesn't appear that it would require massive amounts of computing power to complete it, but the fact is that regular computer simulations just aren't fast enough. A protein fold may occur as fast as a millionth of a second, which is a very long time for a computer to simulate. The team estimates that "…there is a 1000 fold gap between the simulation timescales (nanoseconds) and the times at which the fastest proteins fold (microseconds)." (Stanford) Currently, the group has completed the initial portion of their project in under a year with approximately two-hundred and seventy thousand registered members, of which one-hundred and twenty thousand processors are active. Phase one was to determine the feasibility of using distributed computing with using some of the less complex folding possibilities. The second phase will use data from the first phase to calculate more complex folding possibilities on a much larger scale. This next phase also has managed to secure Intel as a sponsor for the project. By using the resources of distributed computing, this group has been able to successfully accomplish many steps on the path to their ultimate goals. In this instance, distributed computing has performed its role well and may advance the medical field into new areas. By comparison, however, this project should be considered only a very small example of what distributed computing can do. Currently the largest distributed network in operation, SETI@Home has produced some astronomical computational results.

2.2 The Search for ET

Considered to be one of the first distributed computing groups, SETI@Home is the search for extra terrestrials, or perhaps Marvin the Martin if you prefer. The SETI team leases the Arecibo Radio Telescope for a few weeks out of the year. During those weeks of the year, that the SETI team scans as much of the sky as possible and stores the resulting data on 35 gigabyte DLT tapes per day. The tapes are then mailed to Berkley, CA from the Puerto Rico site. Then the tapes are stored in massive tape libraries to be broken down and analyzed for data computations. Having no other practical or affordable way of data mining through all of the data for possible key events, the SETI team formed the SETI@Home project. The project takes the data tapes, breaks small portions of data into 250 kilobyte units, and distributes those units to anyone willing to run the nifty little SETI@Home screensaver. To promote a form of competition, the organization presents real-time statistics on their website, along with awarding certificates of accomplishment to members or teams completing a certain numbers of work units at different milestones. The user-base for the project boasts a registered user-base of four and three-quarters of a million people, with a combined processing power in the neighborhood of 14.19 Teraflops/sec(Seti@Home). To give an idea of the scale and financial implications, the current top-of-the-line offering from the supercomputer manufacturer Cray "features powerful vector processors combined with an interconnect that scales to peak performances of multiple **tens of teraflops"**(Cray, Inc), and the pricing of a single Cray X1 is approximately $2.5 million dollars. So the project is producing tons of power and saving millions of dollars by using a distributed computing setup to mine their data to find communications from little green men. Truly withstanding the test of time, SETI@Home has been in existence since 1998 and continues to compete with a very active user-base.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

The problem of both of these examples and distributed computing, as a whole, is the issue of security and infrastructure. In essence, it is the Internet itself providing a medium of attack. The distributed clients, having control over thousands of computers, is the provided means of attack.

III. The Negative Sides and Security Implications of Distributed Computing

3.1 Distributed.net: Cracking the World's Encryption One Key at a Time

Another group of people pioneering the use of distributed computing is the Distributed Computing Technologies Inc. business running http://www.distributed.net. The initial purpose of this group was to prove that the RC5 encryption algorithm, at the time which was set to replace DES-III encryption, was quite vulnerable to brute force attacks should not be considered as the DES-III replacement. Being another pioneer of distributed technology, the group started their first computation challenge in October 1997 by using brute force to try every possible key in cracking an RC5-64bit challenge. The key took 250 days to locate. After that challenge, the group added another distributed client for nodes that would process the keys for DES-II 1st challenge in February 1998. That key was determined in only 39 days after testing 90% of the possible keys. After reassembling the group and changing their distributed DES client some more, the team continues onward to complete the DES-III challenge in January 1999 by beating the previous record of 56 hours and completing the task in only 22 hours and 15 minutes from the time that the challenge was issued. Having help from the Electronic Frontier Foundation's supercomputer "Deep Crack" and approximately 100,000 volunteer distributed nodes, the two teams were able to brute force test "…245 billion keys per second."(McNatt) Time and time again, the distributed.net team has proven that with enough computing power, even some of the higher encryption algorithms can be broken with persistence. Consider this power when thinking of encryption standards and companies and offices still using older encryption techniques to secure data. If these teams can beat similar methods in less than a day, think of what a mischievous person could do with several hundred thousand clients processing keys unbeknownst to the owners of the computers. If viruses can infect computers and coordinate their efforts to knock out network targets, could they not also be used to compute complex problems and submit results to decrypt even the strongest of algorithms? This should be a question on all security experts' minds, especially when planning for the future.

The power that distributed computing can attain has already been shown through examples of existing projects and their respective numbers. Used in healthy and noteworthy environments, distributed computing works well to provide useful information to those seeking it. It is when it's used for improper reasons that it can create a nearly unstoppable brute force against security features such as encryption and or the bandwidth of the Internet itself.

3.2 Distributed Denial of Service Attacks

One of the areas that distributed computing has been used to stage successful attacks, has been against bandwidth and the structure of the Internet. Distributed denial of service attacks (DDoS) have been among the most successful to target certain sites and IP ranges. The attacks involve the complete saturation of the network with traffic from

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

hundreds of compromised computers working in concert to bring the network or site down. These feats were accomplished with the use of a distributed 'Zombie' client that was installed on PCs that had a security hole in the operating system, notably Microsoft-based. Once the client was installed and running on the computer, it would literally 'phone home' to the controller on a locked, private IRC channel. From there, the controller could tell these compromised computers to attack a specific site and send ICMP packets, or pings, at the fastest rate with the largest amount of data possible, without waiting for any return ACKnowledge signal or the return packet. In the biggest attacks, ping packets were sent out as fast as the sending computers could possibly process them, with a maximum size of 64 kilobytes. In these cases, the attacking machines were computers running Microsoft® Windows®, and the maximum size of a ping packet is 64 kilobytes. A single dialup user sending these out is mere noise, but with a significant group of computers broadcasting these packets of data in unison could easily take down a network, especially if each node of this distributed system had a utilization of 100% of its allotted bandwidth to the internet. This is actually not much of a stretch for most people to realize that the Internet's backbones can only handle so much bandwidth. Unless your firm can pony up large amounts of money for huge connections to the Internet, smaller ones such as T1, T3's and OC3's usually have to suffice. To bring this all to reality, let us consider the following example:

[Example]

Consider an example of what kind of bandwidth a small distributed group could crush. A T1 connection to the Internet has a theoretical bandwidth of 1.5 megabits which equals about 175 kilo**bytes** per second possible throughput up and downstream. Due to the increase in broadband connections to home users, also assume that there are a multitude of users sitting on cable modem connections with vulnerable Microsoft® Windows XP® Professional PCs. The majority of cable modem connections have a limitation on the amount of upstream bandwidth they're allowed, this is usually 256 kilobits per second. Since there are 8 bits in a byte, this amounts to a rough limit of 32 kilo**bytes** per second possible transfer speeds. Already you can see that a single cable modem, if used to 100% of possible capacity, is roughly a third of the bandwidth that a corporate T1 line can handle. Take that single computer and multiply it by ten to twenty and now you see the problem. So why not simply ask the gateway routers from the main Internet trunks to your T1 to filter out this traffic? There are several problems to this solution. To block certain packets with today's equipment, you need to know certain pieces of information in the packets that all of the malicious packets have in common; a common source, type, size, or even the content. With distributed attacks, almost none of this information is the same from one node to another. Even if the attack is using ICMP packets to ping a site to death, blocking those types of packets might stop the current attack, but it would also partially cripple the site by not allowing certain services such as return pings. Current solutions lend themselves to creating more problems.

Compounding these problems is the formation of the packet itself. A normal TCP/IP packet contains, in the header, the source IP of the computer sending the data packet out. This is the receiving computer's way of finding its way back to the sending

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

computer so that it can communicate. This system establishes a form of 'dialogue' with the two computers. In most distributed denial of service attacks, this portion of the packet is 'spoofed' to make it look like the packet came from an entirely different source. According to Steve Gibson of Gibson Research Corp., prior to the release of Windows 2000 and Windows XP, this problem was almost trivial because the Windows 9x and ME series of operating systems did not support full Unix-style sockets. By not fully supporting these types of sockets, the OS didn't allow applications to fully control the formation of data packets leaving these types of sockets. In these cases, the software would try to modify the packet but fail to completely replace the source IP addresses. There was still enough information in the header that the packets could be traced back to the source. However, with the release of Windows 2000 and Windows XP, this single element soon ended due to the facts that both of these operating systems include support for Unix-style sockets. With these types of sockets, software essentially has free reign on the way that packets can be formed. The software is free to create the header how it pleases and thus creates a new problem and form of attack, distributed reflection denial of service attacks, or DRDoS.

Distributed reflection denial of service attacks are similar to DDoS attacks in the manner that they also use a similar client hosted on compromised computers to do all of the work. The key difference to these types of attacks is that instead of using the nodes to attack a certain target, they trick large corporate or other massive sites into doing the attack for them; 'them' being the controller(s). The planning stage would require the attacker to find sites that have very large pipelines connected to the Internet. An example of such sites might be the powerhouses of CNet or Google, where they can accommodate massive amounts of traffic. The attacker chooses as many of these sites as he or she can, and then sets to work. Using the compromised node machines to do the dirty work, the attacker gives the nodes modified packets to send to these large sites. The packets are modified so that the source IP is not the node's IP or a random IP, but the IP of the target site or computer. The process of the communication goes as follows:

1.) Compromised node accepts attack command from creator
2.) Command includes altered headers in packets to send out. The header now has the IP address of the target site set to be the 'Source IP'
3.) Node computer sends packet out to predetermined Internet 'powerhouses' with large pipelines to the Net.
4.) 'Powerhouses' send ACK packets as a return to the initial packet, but they send it to the actual target instead of a random IP or the IP of the compromised Node.
5.) Cumulative power in all of the attacking sites causes denial of service to the target and they are overwhelmed and forced off of the Net.

With very little effort, a creative programmer with malicious intent can easily take down some of the largest sites on the Internet, and with serious financial ramifications. Interestingly enough, during the writing of this paper, a DDoS attack was being used to blackmail a company in Europe into paying out a large amount of money.

3.3 Case and Point: "*Crime gangs extort money with hacking threat*"

In an article from Financial Times.com (FT.com), they reported on Tuesday November 11, 2003 that "…a new type of international extortion racket emerged on Tuesday with revelations that blackmailers have been exploiting computer hacking

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

techniques to threaten the ability of companies to conduct business online." (Nuttall) The article continues onward with saying that these gangs focused on eastern European businesses and attacked businesses repeatedly using distributed denial of service attacks, causing some businesses to lose millions of dollars per day due to downtime. A very real and true example of how the power distributed computing has within its domain of control to those willing to use them for malicious intent. The future of distributed computing will be to gain an understanding and determine the possible solutions needed to prevent and combat these types of attacks.

IV. The Future of Distributed Computing and Possible Solutions

4.1 Possible Solutions to Combat Distributed Attacks

First and foremost, solutions require knowledge of the problems and how these attacks are achieved. One of the best documented examples of distributed denial of service attacks is an account from Steve Gibson at Gibson Research Corp. (http://www.grc.com). His business was attacked by a series DDoS attacks in May 2001 by a thirteen-year old boy commanding over one hundred compromised computers. This person and another cohort managed to take the GRC site down several times in a short period of time and keep it down for hours at a time. The only things that saved Gibson was his knowledge of networking and abilities to contact the vendors operating the edge devices between his two T1's and part of the Internet backbone. Initially Gibson blocked ICMP packets to get his site back online, but knew that was only a temporary stop-gap. He captured many of the packets as they came across his line to build a database of what was coming in and who sent it. Following that line, he asked for users on his newsgroups to submit a copy of the 'Zombie' client which was infecting vulnerable machines. He studied that client for a while and decidedly altered it so that it would not produce any attacks but would still show him what all it was made to do. Eventually the 'Zombie' client led him back to a locked IRC channel where he observed silently in the background watching people use these clients to send out attacks. In the end he simply followed all of the paths back the chain to the source and even though he never pinpointed the exact person attacking, he found out much more than he anticipated and was able to create countermeasures to combat such things. He was able to work with the vendor and design a series of filters for the routers to use so that they could block and discard invalid packets to stop the attacks at that edge of the connection and keep his site online. Gibson also freely shares his experiences to let the Internet community know what all is floating around and can easily strike. The full account of his ordeal is documented on his site at http://grc.com/dos/grcdos.htm.

The second aspect to combating distributed attacks of any form is keeping your operating system as impenetrable as possible. It is always possible that things will sneak up when least expected, but it seems that part of being able to own this level of software is to keep that software patched for security and running only the essential components. It is paramount that all unnecessary services be stopped to prevent intrusion attempts. Even under Linux, UNIX, Mac OS X or other systems, it is crucial that only necessary services, such as mail or file sharing, be run when needed and turned off when they are not. An open door is an easy target for someone looking to break into a computer, so by reducing the chances of finding such doors, these attempts can be seriously decreased.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Finally, it is necessary for an IT professional to know all aspects of his or her computer operating system and environment. Knowledge should especially include what services and programs should be running and which should not. It is always the case during these distributed attacks that the surrogate nodes' operators generally have no clue that their computer is involved in attacks against another system. In cases with multiple servers or workstations, take software inventories if possible or even a simple file index to run a *diff*erentiate program against. It all comes down to how much work is willing to be put forth to protect the investment of an individual or a business.

V. Conclusion

5.1 The Future of Distributed Computing

The future of distributed computing is still quite uncertain since it is one of many new types of computing. The technology has truly shown its worth as a useful research tool as well as its potential for being a threatening tool to cause serious damage to systems and infrastructure, financially and otherwise. An IT security professional should always be on guard for such attacks and know the attacker and technology used to perform such intrusions. Knowledge is the key to ensuring that such attacks can be prevented or at least stopped once they occur, even with it being a brute force.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Works Cited / Resources

1. About the Pande Group. Stanford University. 11 Dec. 2002. <http://www.stanford.edu/group/pandegroup/>.

2. About SETI@Home. Seti@Home. Copyright 2001. <http://setiathome.ssl.berkeley.edu/about_seti/about_seti_at_home_1.html>.

3. [ANNOUNCE] [ADMIN] The secret message is... David McNett. 24 Feb. 1998. <http://lists.distributed.net/hypermail/announce/0039.html>.

4. Beowulf History. Phil Merkey. Copyright 2000-2003 Scyld Computing Corporation <http://www.beowulf.org/beowulf/history.html>.

5. Cray X1 System Specifications. Cray, Inc. November 10, 2003. <http://www.cray.com/products/systems/x1/>.

6. Crime gangs extort money with hacking threat. Chris Nuttall. London. 11 Nov. 2003. <http://news.ft.com/servlet/ContentServer?pagename=FT.com/StoryFT/FullStory&c=StoryFT&cid=1066565805264&p=1012571727088>.

7. Cryptographic Challenges. RSA Laboratories.Copyright 2003. <http://www.rsasecurity.com/rsalabs/challenges/>.

8. Distributed.net completes rc5-64 project (list announcement). David McNett. 25 Sept. 25, 2002. <http://www.distributed.net/pressroom/news-20020926.html>.

9. Distributed Computing: Distributed Communities. Howard Feldman. 22 May 2003. <http://www.onlamp.com/pub/a/onlamp/2003/05/22/distributed.html>.

10. Distributed.net Current Projects, Distributed.net RC5 Challenges, Distributed.net DES Challenges. Distributed Computing Technologies, Inc. Copyright 1997-2003. <http://www.distributed.net/projects.php , http://www.distributed.net/rc5/ , http://www.distributed.net/des/>.

11. RSA Code-Breaking Contest Again Won by Distributed.Net and Electronic Frontier Foundation (EFF). RSA Laboratories. 19 Jan. 1999. <http://www.rsasecurity.com/company/news/releases/pr.asp?doc_id=462>.

12. RSA's DES Challenge III is solved in record time. RSA Laboratories. 18 Jan. 1999. <http://www.rsasecurity.com/rsalabs/challenges/des3/index.html>.

13. The RC5(R) Encryption Algorithm General Information. RSA Laboratories. 7 Apr. 1995. <ftp://ftp.rsasecurity.com/pub/rsalabs/rc5/readme>.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Last Updated: May 22nd, 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**