DOWNLOAD PDF SECURITY IN DISTRIBUTED COMPUTING

Chapter 1 : Security in Distributed, Grid, Mobile, and Pervasive Computing - CRC Press Book

Distributed computing is achieved in a myriad or ways and there really is no 'one true answer' to the question as phrased. Furthermore data acuity 'How do they ensure that the result they get is the result of the calculus they asked?' is more a function of the implementing system design than of security.

The overall computer security problems in computer systems, computer networks and database systems are explained throughout the book. Furthermore, the computer security problems are covered in balance based on the views and approaches of the business and technology sides. The book is organized with four parts. The introductory part 1 begins with a good overview of the security problems in distributed computing environments. In part 2, the foundations of computer security are discussed. Chapter 3 covers computing security basics by using the trust model which consists of identification, authentication, authorization, confidentiality, integrity, nonrepudiation, and so on. Chapter 4 explains the security architecture model which is based on foundation control and trust models. The following chapter 5 illustrates the foundation model that is based on security policy, security principles, security criteria and standards, and education. Chapter 6 discusses security policies based on corporate principles and business code of conduct. In part 3, all aspects of computer security technologies are covered. The security problems and possible corresponding solutions are explained well in chapters 10 through 12, which are followed by chapter 13 specifying Windows NT security. Chapter 14 explains the security problems and requirements connecting the network to the Internet by using a firewall machine as an example. The following chapter 17 enhances the discussion of the fundamental aspects of DCE of chapter Chapter 18 extends to the illustration of the security features in distributed database systems. In part 4, chapter 20 begins with how to design client-server applications satisfying security issues discussed in the previous chapters. Chapter 21 shows the examples of applying computer security in electronic mail and groupware. The following chapters 22 and 23 discuss how to manage security in the distributed system environments and how to develop a suitable security strategy for organizations. Chapter 24 explains well about all aspects related to auditing in computer systems. The final chapter of this book concludes with describing the security issues in the future. This book is an important reference book for anyone interested in computer security in distributed computing and especially for those beginners in the computer security field. The book is one of the better computer security books that is easy to follow.

Chapter 2 : CFP: Security in Distibuted Computing, special track of PODC

Practical techniques for securing distributed computing systems. This end-to-end guide to safeguarding information assets identifies key issues in computer security, and the technologies that can help organizations respond.

The Minnesota Intru-sion Detection System can detect sophisticated cyberattacks on large-scale networks that are hard to de-tect using signature-based systems. The phenomenal growth in computing power over much of the past five decades has been motivated by scientific applications requiring massive amounts of computation. Data mining is one of these data-centric applications that increasingly drives development of parallel and distributed computing technology. Explosive growth in the availability of various kinds of data in both commercial and scientific domains has resulted in an unprecedented opportunity to develop automated data-driven knowledge discovery tech-niques. Data mining, an important step in this knowledge-discovery process, consists of methods that dis-cover interesting, nontrivial, useful patterns hidden in the data. The data tends to be distributed, and issues such as scalability, privacy, and security prohibit bringing the data together. Such cases require distributed data mining. Into this mix enters the Internet, along with its tremendous benefits and vulnerabilities. The need for cyber-security and the inadequacy of traditional approaches have piqued interest in applying data mining to intru-sion detection. This article focuses on the promise and application of parallel and distributed data mining to cybersecurity. Need for cybersecurity Individuals and organizations attack and misuse computer systems, creating new Internet threats daily. The number of computer attacks has increased exponentially in the past few years,3 and their severity and so-phistication are also growing. The conventional approach to securing computer systems is to design mechanisms such as firewalls, au-thentication tools, and virtual private networks that create a protective shield. However, these mechanisms almost always have vulnerabilities. This has created the need for intrusion detection,5,6 security technology that complements conventional security approaches by monitoring systems and identifying computer attacks. They have several limita-tions. And once someone discovers a new attack and devel-ops its signature, deploying that signature is often delayed. These limitations have led to an increasing in-terest in intrusion detection techniques based on data mining. Additionally, it often discovers rogue commu-nication channels and the exfiltration of data that other widely used tools such as Snort http: The MINDS suite con-tains various modules for collecting and analyzing massive amounts of network traffic. Typical analyses include behavioral anomaly detection, summarization, and profiling. Additionally, the system has modules for feature extraction and for filtering out attacks for which good predictive models exist for example, for scan detection. Independently, each of these modules provides key insights into the network. When com-bined, which MINDS does automatically, these modules have a multiplicative affect on analysis. Notably, this technique enables meaning-ful calculation of the similarity between records containing a mixture of categorical and numerical attributes such as network traffic records. To the best of our knowledge, no other existing anomaly detection technique can find complex behavior anomalies in a real-world environment while main-taining a very low false-alarm rate. A multithreaded parallel formulation of this module allows analysis of network traffic from many sensors in near-real time at the ARL-CIMP. Summarization The ability to summarize large amounts of network traffic can be highly valuable for network security ana-lysts who must often deal with large amounts of data. For example, when analysts use the MINDS anomaly detection algorithm to score several million network flows in a typical window of data, several hundred high-ly ranked flows might require attention. But due to the limited time available, analysts often can look only at the first few pages of results covering the top few dozen most anomalous flows. Because MINDS can summarize many of these flows into a small representation, the analyst can analyze a much larger set of anomalies than is otherwise possible. Our research group has formulated a methodology for summarizing information in a database of transactions with categorical attributes as an optimization problem. These algorithms have helped us better understand the nature of cyberattacks as well as create new signature rules for intrusion detection systems. The system sorts the connections according to the score that the anomaly detection algorithm assigns them. Then, using the pat-terns that the association analysis module generates, MINDS summarizes anomalous connections with the

highest scores. Each line contains the average anomaly score, the number of connections represented by the line, eight basic connection features, and the relative contribution of each basic and derived anomaly detection feature. For example, the second line in figure 2 represents anomalous connections. From this summary, analysts can easily infer that this is a backscatter from a denial-of-service attack on a computer that is outside the network being examined. Such inference is hard to make from individual connections even if the anomaly detection module ranks them highly. Each line contains an anomaly score, the number of connections that line represents, and several other pieces of information that help the analyst get a quick picture. Profiling We can use clustering, a data mining technique for grouping similar items, to find related network connec-tions and thus discover dominant modes of behavior. MINDS uses the Shared Nearest Neighbor clustering algorithm,11 which works particularly well when data is high-dimensional and noisy for example, network data. SNN is highly computationally intensiveof the order O n2 , where n is the number of network connec-tions. So, we need to use parallel computing to scale this algorithm to large data sets. Our group has de-veloped a parallel formulation of the SNN clustering algorithm for behavior modeling, making it feasible to analyze massive amounts of network data. The data consisted of , connections collected over one hour. On a CPU cluster, the SNN algorithm took 10 hours to run and required Mbytes of memory at each node to calculate distances between points. The final clustering step required Mbytes of memory at one node. The algorithm produced 3, clusters ranging in size from 10 to records. Most large clusters corresponded to normal behavior modes, such as virtual private network traffic. However, several smaller clusters corresponded to minor deviant behavior modes relating to misconfigured computers, insider abuse, and policy violations undetectable by other methods. Such clusters give analysts information they can act on immediately and can help them understand their network traffic behavior. Figure 3 shows two clusters obtained from this experiment. This is a policy violation in the organization for which this data was being analyzed. Detecting distributed attacks Interestingly, attacks often arise from multiple locations. In fact, individual attackers often control numerous machines, and they can use different machines to launch different steps of an attack. An intrusion detection system IDS running at one site might not have enough information by itself to detect the attack. Rapidly detecting such distributed cyberat-tacks requires an interconnected system of IDSs that can ingest network traffic data in near real-time, detect anomalous connections, communicate their results to other IDSs, and incorporate the information from other systems to enhance the anomaly scores of such threats. Such a system consists of several autonomous IDSs that share their knowledge bases with each other to swiftly detect malicious, large-scale cyberattacks. Figure 4 illustrates the distributed aspect of this problem. It shows the two-dimensional global Internet Pro-tocol space such that every IP address allocated in the world is represented in some block. The black re-gion represents unallocated IP space. Map of the global IP space. Each red dot in the right-hand box represents a suspicious connection made by a machine to an internal machine on port The righthand box indicates that most of these potential attackers are clustered in specific Internet address blocks. A close examination shows that most of the dense areas belong to the network blocks of cable and AOL users located in the US or to blocks allocated to Asia and Latin America. There are unique sources on the outside trying to contact 1, destinations inside the University of Minnesota IP network space. The total number of involved flows is 1,, which means that most external sources made just one suspicious connection to the inside. If multi-ple sites running the same analysis across the IP space report the same external source as suspicious, it would make the classification much more accurate. Suspicious traffic on port The ideal scenario for the future would be that we bring the data collected at these different sites to one place and then analyze it. Implementing such a system would require handling distributed data, addressing privacy issues, and using data mining tools, and would be much easier if a middleware provided these functions. The distributed network intrusion detection system being developed collaboratively by three uni-versity teams. I thank Devdatta Kulkarni for his volunteer work on integrating the audio and PowerPoint files. His research interests include high-performance computing and data mining. He received his PhD in computer science from the University of Maryland. Contact him at the Dept. This is just a sample from a fellow student. Let our professionals create one just for you.

Chapter 3 : Security in distributed Computing - calendrierdelascience.com

4 Security in Distributed, Grid, Mobile and Pervasive Computing In this chapter, we ï¬•rst review the security concepts related to CDNs and then present several systems, focusing on how they enforce content security.

Energy sector[edit] In distributed generation systems, the risk of cyber attacks is real, according to Daily Energy Insider. An attack could cause a loss of power in a large area for a long period of time, and such an attack could have just as severe consequences as a natural disaster. The District of Columbia is considering creating a Distributed Energy Resources DER Authority within the city, with the goal being for customers to have more insight into their own energy use and giving the local electric utility, Pepco , the chance to better estimate energy demand. The reliability of these estimates is often challenged; the underlying methodology is basically anecdotal. According to the classic Gordon-Loeb Model analyzing the optimal investment level in information security, one can conclude that the amount a firm spends to protect information should generally be only a small fraction of the expected loss i. Some are thrill-seekers or vandals , some are activists, others are criminals looking for financial gain. A standard part of threat modelling for any particular system is to identify what might motivate an attack on that system, and who might be motivated to breach it. The level and detail of precautions will vary depending on the system to be secured. A home personal computer , bank , and classified military network face very different threats, even when the underlying technologies in use are similar. Computer protection countermeasures [edit] In computer security a countermeasure is an action, device, procedure, or technique that reduces a threat , a vulnerability , or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken. Security by design[edit] Security by design , or alternately secure by design, means that the software has been designed from the ground up to be secure. In this case, security is considered as a main feature. Some of the techniques in this approach include: The principle of least privilege , where each part of the system has only the privileges that are needed for its function. That way even if an attacker gains access to that part, they have only limited access to the whole system. Automated theorem proving to prove the correctness of crucial software subsystems. Code reviews and unit testing , approaches to make modules more secure where formal correctness proofs are not possible. Defense in depth , where the design is such that more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds. Default secure settings, and design to "fail secure" rather than "fail insecure" see fail-safe for the equivalent in safety engineering. Ideally, a secure system should require a deliberate, conscious, knowledgeable and free decision on the part of legitimate authorities in order to make it insecure. Audit trails tracking system activity, so that when a security breach occurs, the mechanism and extent of the breach can be determined. Storing audit trails remotely, where they can only be appended to, can keep intruders from covering their tracks. Full disclosure of all vulnerabilities, to ensure that the " window of vulnerability " is kept as short as possible when bugs are discovered. Security architecture[edit] The Open Security Architecture organization defines IT security architecture as "the design artifacts that describe how the security controls security countermeasures are positioned, and how they relate to the overall information technology architecture. It also specifies when and where to apply security controls. The design process is generally reproducible. Security measures[edit] A state of computer "security" is the conceptual ideal, attained by the use of the three processes: These processes are based on various policies and system components, which include the following: User account access controls and cryptography can protect systems files and data, respectively. Firewalls are by far the most common prevention systems from a network security perspective as they can if properly configured shield access to internal network services, and block certain kinds of attacks through packet filtering. Firewalls can be both hardware- or software-based. Intrusion Detection System IDS products are designed to detect network attacks in-progress and assist in post-attack forensics , while audit trails and logs serve a similar function for individual systems. In some special cases, a complete destruction of the compromised system is favored, as it may happen that not all the compromised resources are detected. Today, computer security comprises mainly "preventive" measures, like firewalls or an

exit procedure. A firewall can be defined as a way of filtering network data between a host or a network and another network, such as the Internet , and can be implemented as software running on the machine, hooking into the network stack or, in the case of most UNIX -based operating systems such as Linux , built into the operating system kernel to provide real-time filtering and blocking. Another implementation is a so-called "physical firewall", which consists of a separate machine filtering network traffic. Firewalls are common amongst machines that are permanently connected to the Internet. Some organizations are turning to big data platforms, such as Apache Hadoop , to extend data accessibility and machine learning to detect advanced persistent threats. As a result, as Reuters points out: Yet it is basic evidence gathering by using packet capture appliances that puts criminals behind bars. Vulnerability management Vulnerability management is the cycle of identifying, and remediating or mitigating vulnerabilities , [] especially in software and firmware. Vulnerability management is integral to computer security and network security. Vulnerabilities can be discovered with a vulnerability scanner , which analyzes a computer system in search of known vulnerabilities, [] such as open ports , insecure software configuration, and susceptibility to malware. Beyond vulnerability scanning, many organisations contract outside security auditors to run regular penetration tests against their systems to identify vulnerabilities. In some sectors this is a contractual requirement. Two factor authentication is a method for mitigating unauthorized access to a system or sensitive information. It requires "something you know"; a password or PIN, and "something you have"; a card, dongle, cellphone, or other piece of hardware. This increases security as an unauthorized person needs both of these to gain access. Social engineering and direct computer access physical attacks can only be prevented by non-computer means, which can be difficult to enforce, relative to the sensitivity of the information. Training is often involved to help mitigate this risk, but even in a highly disciplined environments e. Enoculation, derived from inoculation theory , seeks to prevent social engineering and other fraudulent tricks or traps by instilling a resistance to persuasion attempts through exposure to similar or related attempts. This statement is ambiguous. Hardware protection mechanisms[edit] See also: Computer security compromised by hardware failure While hardware may be a source of insecurity, such as with microchip vulnerabilities maliciously introduced during the manufacturing process, [] [] hardware-based or assisted computer security also offers an alternative to software-only computer security. Using devices and methods such as dongles , trusted platform modules , intrusion-aware cases, drive locks, disabling USB ports, and mobile-enabled access may be considered more secure due to the physical access or sophisticated backdoor access required in order to be compromised. Each of these is covered in more detail below. The dongle, or key, essentially creates a secure encrypted tunnel between the software application and the key. The principle is that an encryption scheme on the dongle, such as Advanced Encryption Standard AES provides a stronger measure of security, since it is harder to hack and replicate the dongle than to simply copy the native software to another machine and use it. Another security application for dongles is to use them for accessing web-based content such as cloud software or Virtual Private Networks VPNs. TPMs used in conjunction with server-side software offer a way to detect and authenticate hardware devices, preventing unauthorized network and data access. The firmware or BIOS is programmed to show an alert to the operator when the computer is booted up the next time. Drive locks are essentially software tools to encrypt hard drives, making them inaccessible to thieves. Infected USB dongles connected to a network from a computer inside the firewall are considered by the magazine Network World as the most common hardware threat facing computer networks. Mobile-enabled access devices are growing in popularity due to the ubiquitous nature of cell phones. Built-in capabilities such as Bluetooth , the newer Bluetooth low energy LE , Near field communication NFC on non-iOS devices and biometric validation such as thumb print readers, as well as QR code reader software designed for mobile devices, offer new, secure ways for mobile phones to connect to access control systems. These control systems provide computer security and can also be used for controlling access to secure buildings. Security-evaluated operating system One use of the term "computer security" refers to technology that is used to implement secure operating systems. Many common operating systems meet the EAL4 standard of being "Methodically Designed, Tested and Reviewed", but the formal verification required for the highest levels means that they are uncommon. Secure coding In software engineering, secure coding aims to guard against the accidental introduction of

security vulnerabilities. It is also possible to create software designed from the ground up to be secure. Such systems are " secure by design ". Beyond this, formal verification aims to prove the correctness of the algorithms underlying a system; [] important for cryptographic protocols for example. Capabilities and access control lists[edit] Main articles: Access control list and Capability computers Within computer systems, two of many security models capable of enforcing privilege separation are access control lists ACLs and capability-based security. Using ACLs to confine programs has been proven to be insecure in many situations, such as if the host computer can be tricked into indirectly allowing restricted file access, an issue known as the confused deputy problem. It has also been shown that the promise of ACLs of giving access to an object to only one person can never be guaranteed in practice. Both of these problems are resolved by capabilities. This does not mean practical flaws exist in all ACL-based systems, but only that the designers of certain utilities must take responsibility to ensure that they do not introduce flaws. Capabilities can, however, also be implemented at the language level, leading to a style of programming that is essentially a refinement of standard object-oriented design. An open source project in the area is the E language. As the human component of cyber risk is particularly relevant in determining the global cyber risk [] an organization is facing, security awareness training, at all levels, does not only provides formal compliance with regulatory and industry mandates but is considered essential [] in reducing cyber risk and protecting individuals and companies from the great majority of cyber threats. The focus on the end-user represents a profound cultural change for many security practitioners, who have traditionally approached cybersecurity exclusively from a technical perspective, and moves along the lines suggested by major security centers [] to develop a culture of cyber awareness within the organization, recognizing that a security aware user provides an important line of defense against cyber attacks. Response to breaches[edit] Responding forcefully to attempted security breaches in the manner that one would for attempted physical security breaches is often very difficult for a variety of reasons: Identifying attackers is difficult, as they are often in a different jurisdiction to the systems they attempt to breach, and operate through proxies, temporary anonymous dial-up accounts, wireless connections, and other anonymising procedures which make backtracing difficult and are often located in yet another jurisdiction. If they successfully breach security, they are often able to delete logs to cover their tracks. The sheer number of attempted attacks is so large that organisations cannot spend time pursuing each attacker a typical home user with a permanent e. Note however, that most of the sheer bulk of these attacks are made by automated vulnerability scanners and computer worms.

Chapter 4 : Security In Distributed Computing: Did You Lock th | eBay

Distributed computing is an awesome approach to distribute workload of huge tasks and easily outsource them, if needed. It makes computing tasks scalable and cheap, as cloud computing is involved.

A distributed system is a network that consists of autonomous computers that are connected using a distribution middleware. They help in sharing different resources and capabilities to provide users with a single and integrated coherent network. Techopedia explains Distributed System The key features of a distributed system are: Components in the system are concurrent. A distributed system allows resource sharing, including software by systems connected to the network at the same time. There can be multiple components, but they will generally be autonomous in nature. A global clock is not required in a distributed system. The systems can be spread across different geographies. Compared to other network models, there is greater fault tolerance in a distributed model. The key goals of a distributed system include: Achieving the image of a single system image without concealing the details of the location, access, migration, concurrency, failure, relocation, persistence and resources to the users Openness: Making the network easier to configure and modify Reliability: Compared to a single system, a distributed system should be highly capable of being secure, consistent and have a high capability of masking errors. Compared to other models, distributed models are expected to give a much-wanted boost to performance. Distributed systems should be scalable with respect to geography, administration or size. Challenges for distributed systems include: Security is a big challenge in a distributed environment, especially when using public networks. Fault tolerance could be tough when the distributed model is built based on unreliable components. Coordination and resource sharing can be difficult if proper protocols or policies are not in place. Process knowledge should be put in place for the administrators and users of the distributed model.

Chapter 5 : Distributed Computing | Computer Science and Engineering

A distributed computing system is the system architecture that makes a collection of heterogeneous computers or workstations act and behave as being one single computing system.

For each given chunk of work, give it to two distinct contributors, and compare the results. This will trap most fraudsters and also computing errors due to bad RAM; unfortunately, this also halves the available computing power. This strategy was what used by some friends of mine who briefly got the world record for the N queens problem i. This is a variant of the previous solution, in which you duplicate only some chunks, e. This will trap fraudsters who game results at least occasionally, while incurring much less overhead. You have to keep track of who did what chunk, so that suspicious values can be removed en masse. If the work is of the find-needle-in-haystack persuasion like a distributed key cracking effort , then you can announce a grand prize for who finds it. This will induce a lot of potential cheater to actually do the computations, since not doing them means that they will not get the prize. That one is a generic solution, but the best known algorithm for FHE has the inconvenient drawback of incurring some overhead which is counted in billions -- i. The all-time favourite of anti-cheating strategies: In a similar scenario of distributed computing, namely online games, it does not work well. Simply pray that there will not be too many fraudsters, and that most chunks will be processed as they should. Depending on the type of computation, this may be a valid strategy. For instance, it may work for a key cracking effort, since a few skipped chunks will not matter unless the key is in one of them, which would be bad luck but improbable if most contributors are honest. It would not work for a complete counting where you add all the results each wrong value will alter the total. Some people would fake results, i. They were doing that just to be at a better rank in the list of contributors, ordered by CPU power retrospectively, it was quite obvious that computer owner would act just like car owners do. Their response was an obscure client without source, which was sufficient in that case too few people did the reverse engineering to cheat again. There is also the dual problem, i. This is not an easy question, and it is made harder by closed-source client code Linux includes a nifty system call named seccomp which could be useful for that. And, at least theoretically, this might not be sufficient for complete isolation of a potentially rogue process, because that process will share caches L1 cache, jump prediction

Chapter 6 : Distributed Computing: An Introduction - ExtremeTech

This book is an important reference book for anyone interested in computer security in distributed computing and especially for those beginners in the computer security field. The book is one of the better computer security books that is easy to follow.

Models[edit] Many tasks that we would like to automate by using a computer are of questionâ€”answer type: In theoretical computer science , such tasks are called computational problems. Formally, a computational problem consists of instances together with a solution for each instance. Instances are questions that we can ask, and solutions are desired answers to these questions. Theoretical computer science seeks to understand which computational problems can be solved by using a computer computability theory and how efficiently computational complexity theory. Traditionally, it is said that a problem can be solved by using a computer if we can design an algorithm that produces a correct solution for any given instance. Such an algorithm can be implemented as a computer program that runs on a general-purpose computer: Formalisms such as random access machines or universal Turing machines can be used as abstract models of a sequential general-purpose computer executing such an algorithm. However, it is not at all obvious what is meant by "solving a problem" in the case of a concurrent or distributed system: Three viewpoints are commonly used: Parallel algorithms in shared-memory model All processors have access to a shared memory. The algorithm designer chooses the program executed by each processor. One theoretical model is the parallel random access machines PRAM that are used. Shared-memory programs can be extended to distributed systems if the underlying operating system encapsulates the communication between nodes and virtually unifies the memory across all individual systems. A model that is closer to the behavior of real-world multiprocessor machines and takes into account the use of machine instructions, such as Compare-and-swap CAS , is that of asynchronous shared memory. There is a wide body of work on this model, a summary of which can be found in the literature. Models such as Boolean circuits and sorting networks are used. Similarly, a sorting network can be seen as a computer network: Distributed algorithms in message-passing model The algorithm designer only chooses the computer program. All computers run the same program. The system must work correctly regardless of the structure of the network. A commonly used model is a graph with one finite-state machine per node. In the case of distributed algorithms, computational problems are typically related to graphs. Often the graph that describes the structure of the computer network is the problem instance. This is illustrated in the following example. Different fields might take the following approaches: Centralized algorithms[citation needed] The graph G is encoded as a string, and the string is given as input to a computer. The computer program finds a coloring of the graph, encodes the coloring as a string, and outputs the result. Parallel algorithms Again, the graph G is encoded as a string. However, multiple computers can access the same string in parallel. Each computer might focus on one part of the graph and produce a coloring for that part. The main focus is on high-performance computation that exploits the processing power of multiple computers in parallel. Distributed algorithms The graph G is the structure of the computer network. There is one computer for each node of G and one communication link for each edge of G. Initially, each computer only knows about its immediate neighbors in the graph G; the computers must exchange messages with each other to discover more about the structure of G. Each computer must produce its own color as output. The main focus is on coordinating the operation of an arbitrary distributed system. For example, the Coleâ€”Vishkin algorithm for graph coloring [39] was originally presented as a parallel algorithm, but the same technique can also be used directly as a distributed algorithm. Moreover, a parallel algorithm can be implemented either in a parallel system using shared memory or in a distributed system using message passing. Complexity measures[edit] In parallel algorithms, yet another resource in addition to time and space is the number of computers. Indeed, often there is a trade-off between the running time and the number of computers: If a decision problem can be solved in polylogarithmic time by using a polynomial number of processors, then the problem is said to be in the class NC. Perhaps the simplest model of distributed computing is a synchronous system where all nodes operate in a lockstep fashion. In such systems, a central complexity measure is the number of synchronous communication

rounds required to complete the task. Let D be the diameter of the network. On the one hand, any computable problem can be solved trivially in a synchronous distributed system in approximately 2D communication rounds: On the other hand, if the running time of the algorithm is much smaller than D communication rounds, then the nodes in the network must produce their output without having the possibility to obtain information about distant parts of the network. In other words, the nodes must make globally consistent decisions based on information that is available in their local D-neighbourhood. Many distributed algorithms are known with the running time much smaller than D rounds, and understanding which problems can be solved by such algorithms is one of the central research questions of the field [44]. Typically an algorithm which solves a problem in polylogarithmic time in the network size is considered efficient in this model. Another commonly used measure is the total number of bits transmitted in the network cf. Other problems[edit] Traditional computational problems take the perspective that we ask a question, a computer or a distributed system processes the question for a while, and then produces an answer and stops. However, there are also problems where we do not want the system to ever stop. Examples of such problems include the dining philosophers problem and other similar mutual exclusion problems. In these problems, the distributed system is supposed to continuously coordinate the use of shared resources so that no conflicts or deadlocks occur. There are also fundamental challenges that are unique to distributed computing. The first example is challenges that are related to fault-tolerance. Examples of related problems include consensus problems , [46] Byzantine fault tolerance , [47] and self-stabilisation. Synchronizers can be used to run synchronous algorithms in asynchronous systems. Before the task is begun, all network nodes are either unaware which node will serve as the "coordinator" or leader of the task, or unable to communicate with the current coordinator. After a coordinator election algorithm has been run, however, each node throughout the network recognizes a particular, unique node as the task coordinator. For that, they need some method in order to break the symmetry among them. For example, if each node has unique and comparable identities, then the nodes can compare their identities, and decide that the node with the highest identity is the coordinator. The algorithm suggested by Gallager, Humblet, and Spira [54] for general undirected graphs has had a strong impact on the design of distributed algorithms in general, and won the Dijkstra Prize for an influential paper in distributed computing. Many other algorithms were suggested for different kind of network graphs , such as undirected rings, unidirectional rings, complete graphs, grids, directed Euler graphs, and others. A general method that decouples the issue of the graph family from the design of the coordinator election algorithm was suggested by Korach, Kutten, and Moran. The coordinator election problem is to choose a process from among a group of processes on different processors in a distributed system to act as the central coordinator. Several central coordinator election algorithms exist. A complementary research problem is studying the properties of a given distributed system. The halting problem is undecidable in the general case, and naturally understanding the behaviour of a computer network is at least as hard as understanding the behaviour of one computer. In particular, it is possible to reason about the behaviour of a network of finite-state machines. One example is telling whether a given network of interacting asynchronous and non-deterministic finite-state machines can reach a deadlock.

Chapter 7 : How is distributed computing security ensured? - Information Security Stack Exchange

Automated teller machines, airline reservation systems, online credit card validation, electronic mail, remote file transfer, and global commerce are all based on distributed computing, which basica more.

Author Niklas Abel Categories foxmole , keyidentity Tags distributed , distributed computing , FoxMole , KeyIdentity , python framework , security , twisted Distributed computing is an awesome approach to distribute workload of huge tasks and easily outsource them, if needed. It makes computing tasks scalable and cheap, as cloud computing is involved. Computing time can be rent, which is mostly cheaper compared to buying the necessary hardware. However, outsourcing into foreign networks comes with the advantages and drawbacks of public infrastructure. Public networks cannot be trusted, therefore, traffic should be encrypted and connections should be authorized, which sounds easier than it is when using Python frameworks such as Dispy, Celery or Twisted. Although implemented in the core of the frameworks I used, security is optional, sometimes flawed e. The main priority of official tutorials is to make it work, period â€" so developers test the shown code and use it, without bothering about further security steps. Tutorials showing working code with all needed encryption and authorization steps are rare. Often framework developers are showing the needed parts separated, but not the complete setup. So developers have to spend a lot of time puzzling all needed steps together. Security should be implemented by default, which unfortunately is not often the case in official framework tutorials. It was therefore not easy to find documented literature to implement the frameworks in a secure way. Sometimes, the frameworks did not even offer complete secure solutions. There is still room for improvement, but I hope this blog-post will help developers hardening and securing their software a little bit more. The example code can be found here. Twisted based RPC Twisted is a Python framework for event-driven networking to implement asynchronous network programs. We will use JSON instead to reduce the possible attack surface. For encryption we are using TLS on the transport layer and client certificates for authentication. The way we use it can be adopted to most other Twisted based applications as well. The most common authentication mechanism found seems to be using Twisted with password based authentication protocols like basic authentication or digest authentication defined in RFC There are a few issues with basic authentication as its security highly depends on password complexity and an encrypted connection. The password is sent in clear text through the connection. However, it is not hardened against brute-force attacks. Digest authentication does not send the password in clear text. As it is also password based, its security highly depends on password complexity, as well. For client â€" server connections there is often no need to use password based authentication at all. Instead authentication through certificates can be used to have highly secured authentications. The following tutorial uses certificate based authentication combined with TLS encrypted connections. Therefore, we have to generate certificate authorities as well as key pairs. The keys will then be signed with the generated certificate authorities as shown below: First we have to create our own certificate authority for our clients with an aes encrypted key: However, keep in mind that the keyfile will not be encrypted then: We can add additional clients the same way, if needed. To ensure that no one steals the keys, you should change the rights of the key-files to be read-only for its owners on the final system with: In case you want to harden the solution, I would recommend to use a dedicated system user to execute the scripts from client as well as from server with: Source code for the server and client can be found here. How does it work? If further functions are needed, they can be defined in there. Clients only have to know their names and what arguments have to be passed. At line 32 to 41 it gathers all needed information for connection and authentication. To encrypt our connection we need an sslContext, defined in line It gets the previously defined private key and its certificate as well as the defined sslMethod. The part to add the certificate based authentication is in line 49 to Here we define, that clients should be authenticated and the connection will fail if the client cannot offer a valid certificate. We also define our own verifyCallback function defined in line 13 to 28 as callback for certificate checks. We use it to get further information if a certificate check fails. We add our generated CA certificate in line 55 to only trust and accept certificates which are signed from our clientCA. It uses its own SSL context factory, defined in line 54 to We also define the sslMethod to use SSL.

The function asks the user to type in the password. It saves the password at run time in keypw defined in line As an alternative, a keyfile without a pass-phrase can be used. Therefore a callRemote function is defined in line to If it does, the function will be closed. Finally, in line we start the reactor loop for our client. Conclusion When compared to solutions with basic or digest authentication, the implementation of certificate based authentication is less complex and it offers a higher degree of security. So I would recommend using certificate based authentication together with TLS encryption. Developers ask often for a secure solution, but they have to invest a lot of time in researching to get an accurate solution. I hope I was able to help you getting started with Twisted and using it in a secure way.

DOWNLOAD PDF SECURITY IN DISTRIBUTED COMPUTING

Chapter 8 : Security in Distributed Computing | Scalable Computing: Practice and Experience

Grid is a type of distributed computing system where a large number of small loosely coupled computers are brought together to form a large virtual supercomputer.

Secure distributed data outsourcing e. August 20 , Submissions: Authors are invited to submit original technical papers via http: Electronic submission to papers cyberc. All submitted manuscripts should be prepared as technical papers and may not exceed 8 letter size 8. For more information please see ftp: Submissions not conforming to these guidelines may be returned without review. The submitted manuscripts can be prepared in Word, or Latex using the IEEE templates, but authors should finally submit the manuscript in PDF format and make sure that the file will print on a printer that uses letter size 8. The official language of the meeting is English. Manuscript submission procedure is available over the Web at http: If you have troubles in using the Web submission, you can also submit your paper to papers cyberc. Electronic submissions must be in the form of a readable PDF file. Submitted manuscripts have to present the original unpublished research that is not currently under review for any other conference or journal. Papers not following these guidelines will be rejected without review and further action may be taken, including but not limited to notifications sent to the heads of the institutions of the authors and sponsors of the conference. Submissions received after the due date, exceeding length limit, or not appropriately structured may also not be considered. Manuscripts should present the current research in the areas identified in the call for papers. All submitted manuscripts will be reviewed by experts in the fields and will be judged from the aspects of problem significance, contributions, originality, correctness, technical strength, quality of presentation, and relevance to the conference attendees. Papers will be accepted with Regular Papers and Short Papers with maximal 8 pages and 4 pages in the final version respectively. You can find our all publications recorded in the past 9 years. Contact Us For more information about the conference, please visit www.

Chapter 9 : Security In Distributed Computing: Did You Lock the Door?

Security that is designed for a specific computing environment does not account for the dynamic nature of today's computing environments where virtual servers can be launched on demand anywhere.