ORACLE VM

An Oracle Technical White Paper
September 2012

Secure Deployment of Oracle VM Server for SPARC

ORACLE

Introduction

Virtualization is widely used in consolidation projects and as a basis for new server deployments. The benefits in energy efficiency, space, cooling as well as cost are immediately apparent, as many applications don't require the full computational power of even the smallest available new server today. However, in addition to all these benefits, virtualization also introduces some additional aspects to system security.

Oracle VM Server for SPARC (previously called Sun Logical Domains or LDoms) technology is one implementation of virtualization based on the UltraSPARC T1, T2, T2 Plus and SPARC T3 and T4 processors. This paper will help you understand the general security concerns in virtualized environments as well as the specific additional threats that arise out of them. It will discuss these threats, their relation to Oracle VM Server for SPARC and how to mitigate the risk with a set of appropriate counter measures. Based on these, some general recommendations for secure deployments – both for Oracle VM Server for SPARC and for virtualized systems in general – will be given, using a generalized model of security classes as an example.

Engineered Systems such as the SPARC SuperCluster also make use of virtualization. As part of their integrated design concept, the principles discussed in this paper have been carefully engineered into these systems to make full use of all the security capabilities available in these solutions. As such, they are a good example to show how the principles discussed in this paper can be put to work in reality. More details can be found in the recently published whitepaper “SPARC SuperCluster Security Principles and Capabilities” [1]

To make good use of this paper, you should have some basic understanding of how Logical Domains work and how they are configured. It is recommended that you have read and understood the very good introduction presented in the Sun Blueprint “Beginners Guide to LDoms” [2] or the more recent whitepaper “Oracle VM Server for SPARC: Enabling A Flexible, Efficient IT Infrastructure” [3].

Acknowledgements

This paper is based on the study “Sicherer Einsatz der LDom-Technologie von Sun” by Steffen Gundel[1] of cirosec GmbH[2]. The description of the threats and counter measures in chapter 2 especially are heavily based hereon. I would like to thank Steffen Gundel for his excellent work, which was written in 2009 by order and for account of Sun Microsystems GmbH.

I would further like to thank my collegues Jan Brosowski, Martin Müller and Glenn Brunette as well as Michael Ramchand and Darren Moffat for their valuable comments to both style and content of this paper. They were truly helpful.

[1] Steffen.Gundel@cirosec.de

[2] http://www.cirosec.de/

Security in a Virtualized Environment

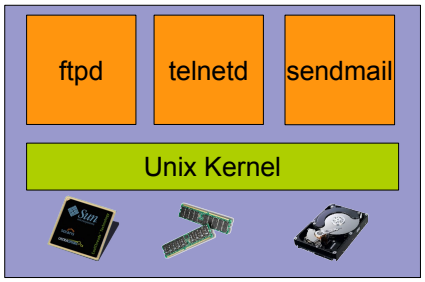

Figure 1: Virtualized Services in Unix

Virtualization is nothing new in the IT industry. Essentially, it was part of the first Unix systems in the early 1970s, which were capable of running several processes simultaneously, thus virtualizing the hardware resources of the server for the benefit of multiple users and services (for example, telnetd, sendmail, ftpd at the time). From a security perspective, each of these services provided a potential entry point for an attacker to try and take over not only the individual service, but to go on from there and seize the entire server. Today's security best practice is to secure each service running on a server as well as the operating system itself. In some cases, you even separate these services by deploying them in different servers to reduce the impact of a successful breach. This needs now to be carried over into the emerging virtualized world.

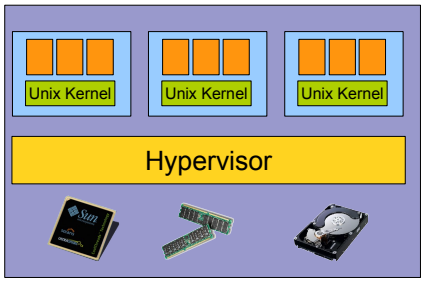

Figure 2: Virtualized Unix Servers

All virtualization technologies come with an additional layer of hardware or software: the hypervisor. It is in charge of configuring and managing both the hardware and the configuration of the guest systems. Just like the Unix kernel virtualized CPU, memory and IO resources for several (guest) processes, the hypervisor virtualizes these same resources for several guests operating systems. The guests themselves appear to be whole and unique server systems. This new component is a potential target for attacks. Of course it needs to be secured, just like all conventional components. And likewise, the security implications of the overall system need to be evaluated and considered when deploying the server, the guests and the services within them.

From a security point of view, you always have to assume that there is a – yet unknown – design or implementation flaw that will allow an attacker to do things he or she is not permitted to do. The most typical example would be a buffer overflow in some network daemon that allows the attacker to execute some arbitrary code with the privileges of that daemon. The attacker would then attempt to elevate his privileges to those of root and take over control of the server. Likewise, in a virtualized environment, an attacker would try to exploit some weakness in the hypervisor or its supporting software components to be able to hijack the whole system, including other guests. Logic dictates that we can never be completely sure that no errors or security holes exist. We might well be very certain, but not 100% sure. Therefore, we always have to deploy systems in a way that minimizes risk of damage in case of a breach.

The Execution Environment

In the section above, the hypervisor was named as the one additional component that is used to add virtualization to a server. However, while the hypervisor is at the heart of this technology, there are

typically other components that are required to operate the system. Depending on the individual virtualization product, these components may be tightly integrated with the hypervisor to form a monolithic-looking product, or they may be only loosely coupled, allowing for a more flexible, modular deployment and configuration. In all cases, we can separate several functions that need to be implemented:

- The hypervisor itself. Whether implemented in hardware, in software or both, this component "owns" all the hardware and grants the guest systems access to it in one way or another.
- A configuration utility to configure the hypervisor, and thus the virtual hardware which is available to guest systems.
- The IO-component of the virtualization solution. This component owns the IO hardware like disk and network devices and provides virtualized devices to the guests. Depending on the implementation, these might need specialized hardware drivers to access the virtual devices.
- A virtual console to communicate with the guest before network access is possible.

In case of Oracle VM Server for SPARC, there are these components, as shown in figure 3 below:

- The **Hypervisor** – It is implemented as firmware for the platform and relies heavily on the hardware support built into the CPU.
- The **Control Domain** – a specialized guest that configures the hypervisor.
- The **IO Domain** – a guest that owns some or all of the available IO devices of the platform.
- The **Service Domain** – a guest that offers some sort of service to other guests. For example, a service domain could provide console access to other guests or provide virtual disks. In the latter case, it would also be an IO domain.

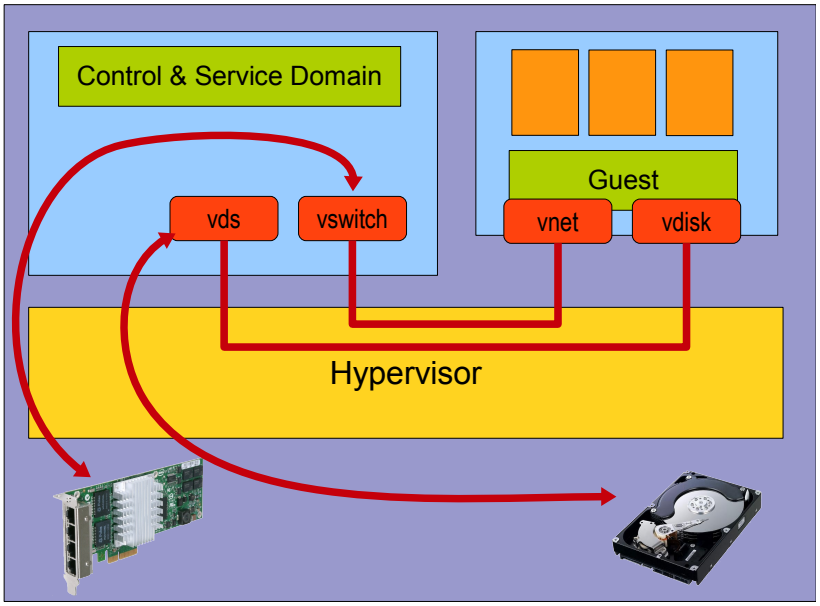

Figure 3: Components of Oracle VM Server for SPARC Virtualization

All these components together form the "execution environment" – the infrastructure upon which the production guests rely. They are described in more detail in the Blueprint "Beginner's Guide to LDoms"[2] and the recent whitepaper "Oracle VM Server for SPARC: Enabling A Flexible, Efficient IT Infrastructure" [3].

Usually, these components are not strictly separated. In fact, the most simple configuration would bundle all of these functions in one domain (as shown in Figure 3), which would then be control domain, IO domain and service domain for all other guests. However, there are good reasons to have at least a second IO domain for redundant IO configurations. While this is done primarily to improve serviceability of the platform, we will later see that there are benefits with regards to system security as well. For more details on configuration options, check the LDom Administration Guide [4] or the Blueprint mentioned above.

Deployment Models

The above description of the execution environment silently assumes the most typical deployment model for LDoms: A **consolidation scenario** where the platform is used to host a number of guest domains, usually between 10 and 50 (or more), depending on the compute requirements of the guests and the size of the hardware platform. With the introduction of the SPARC T4 CPU and especially with the T4-4 server, a different model of operation became more popular: **Hardware partitioning**. In this model, a small number of domains, not more than one per CPU, is each assigned its own portion of physical IO resources, usually one or two complete PCIe root complexes. When properly configured, these domains are fully independent of each other, in other words, do not require any other domain's services to boot and run. In the terminology mentioned above, these domains would be IO domains, but not service domains. Since they usually own a whole PCIe root complex, the term "root domain" has become common for these domains to distinguish them from IO domains in the consolidation use case. A prominent example of this model is the SPARC SuperCluster. When configured in this way, the functional and security characteristic of the execution environment is significantly different to the consolidation scenario mentioned above. From a security perspective, this difference in deployment will need to be taken into account when evaluating the threats and their potential for damage. In the remainder of this paper, the implications of each of these deployment models will be pointed out whenever they become relevant.

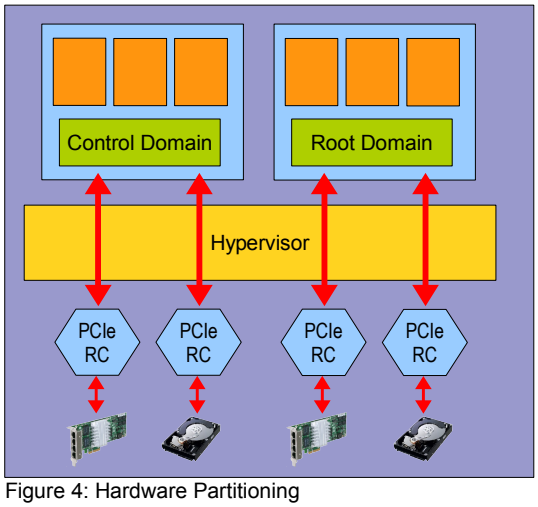

Figure 4: Hardware Partitioning

Securing the Execution Environment

When looking at the security of virtualized environments it is important to understand that each guest is running an operating system which, as such, is a potential target for a normal attack. This does not change, just because the guest is running on a hypervisor vs. directly on the server hardware. These attacks, although usually a prerequisite for any attack on the execution environment, are not the subject of this paper. Instead, once an attacker has identified a platform as being virtualized, he or she may try to attack the execution environment itself.

There are several potentially interesting attack targets in the execution environment of Oracle VM Server for SPARC. Figure 3 above shows the different components of a simple LDom setup, where the control, IO & service domain provides both network and disk services to one guest. These services are implemented through daemons and kernel modules running in the control domain. They communicate with their corresponding device drivers in the guest using Logical Domain Channels (LDC) which are provided by the hypervisor. An attacker would try to find and abuse an error in the design or implementation of any of these components to try and break the isolation of the guest system, either to execute arbitrary code for example in the service domain, or to disrupt normal operations of some or all of the platform.

To do this, the attacker has to be able to communicate with the system in some way or another. Essentially, this means the exchange of network packets with one of the network ports of the platform chassis. The goal would be to either break in through an error in the virtual networking system directly, or to first hijack an available guest through normal means and then continue the attack on the execution environment from within this guest system. It is theoretically possible to shape network packets in such a way that they will penetrate any preceding network equipment (like routers and firewalls) unharmed and only do their damage at the target. However, the much more likely scenario is to assume that the attacker has either hijacked an adjacent system to launch his or her attack, or uses a already penetrated guest domain as an intermediate system.

Virtualization Means Shared Hardware Resources

Another aspect of virtualized environments is hardware sharing. Usually, this is the main motivation for using virtualization techniques in the first place – consolidating several servers into one. However, sharing resources also implies the possibility that one virtualized guest system can starve other guests on the same system off the resources they need, possibly to the point where they come close to a complete halt. Again, these side-effects are not generally new – two processes in a Unix system also compete for resources like network bandwidth or memory and can potentially do harm to one another in this way. However, when deploying numerous systems side by side, this has to be considered.

In Oracle VM Server for SPARC, not as many types of resources are shared as in other implementations. CPU and memory is allocated to each guest exclusively, thus preventing abuse through excessive CPU usage or memory allocation. Disk and network, on the other hand, are typically provided by service domains to many guests in consolidation deployments. In deployments where LDoms are used for mere hardware partitioning, the system can be configured without any sharing of hardware resources between guests.

The rest of this paper will study the different possibilities for an attack, either directly on the execution environment, or indirectly through side effects of shared resources. We will further discuss measures to counter these threats.

Attack & Defense

This paper will discuss numerous potential attacks and how to counter them. The approaches are different for each attack. The targets are different, as well as the type of damage in case of success. However, these attacks all have one thing in common: They try to overcome or eliminate the isolation of the different virtualized servers running on one platform. The ultimate goal of an attack on a virtualization platform is to break system isolation – to attack one guest from another by manipulating the hypervisor or a part of the overall execution environment. Of course, there is much more to system security in general. Each individual guest system needs to be protected like any standalone server, with all the best practices that evolved over the last few decades. The threats discussed in this paper are the ones that are new because of virtualization. The execution environment, with the hypervisor at its heart, is what implements these virtual servers. It is therefore also the ultimate target of all attacks related to virtualization.

In this chapter, we will approach a system running LDoms, starting at the perimeter – the datacenter environment – and slowly closing in discussing each component, until finally reaching the hypervisor itself. On our way, we will identify what threats need to be considered, and how to counter them effectively. These are the individual elements we will discuss:

- The Operational Environment
- The Execution Environment
  - The ILOM
  - The Hypervisor
  - The Control Domain
    - LDoms Manager
  - The IO Domain
  - The Service Domain
- The Guest Domains

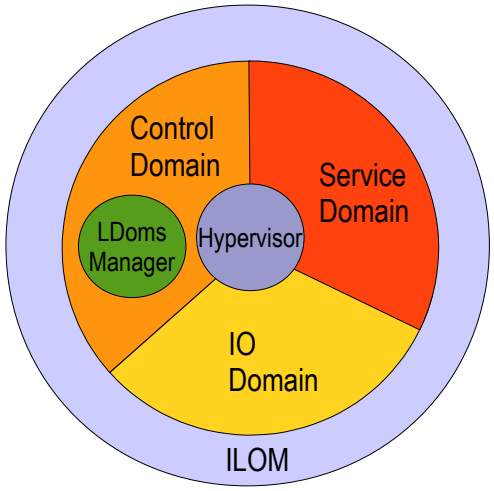

Figure 5: The Components of the Execution Environment

The Operational Environment – a First Line of Defense

All systems are operated by their administrators. They are usually located in a datacenter, they are connected by networking equipment, perhaps they use storage connected through a SAN. Each of these components is again operated by administrators – either the same ones that also care for the systems, or others. Depending on the size of the datacenter, there might be groups with clearly separated areas of responsibility. Or perhaps there is only one administrator. In all cases, there are administrators. And even the best administrator will eventually make a mistake. Administrators are also prime targets for social engineering attacks or plain bribes. Finally, the administrator configures

these systems and components according to an overall architecture, which in turn was designed by some member of the overall IT organization. All of this, the components and all the personnel involved in its operation, make up the operational environment of a server, be it virtualized or not. This is the first line of defense against any security incident.

Real servers are part of our physical world. We can see and touch them, stick labels to their chassis and attach different color cables to them. In a virtualized world, none of this is true anymore. There aren't any cables, so they can't have colors. There aren't any chassis (of each individual server, that is), so we can't stick labels to them. Many servers now live in just one chassis, and what makes things worse, they could even move between chassis without anyone seeing the difference by looking into the datacenter. In short, the virtualized world has a tendency to be more complex, and definitely requires more, and more accurate documentation.

Because virtualization adds an additional layer of software between the actual hardware and the guest systems running the production services, there is also additional administrative work that needs to be done. As mentioned above, it is subject to human error and social engineering. Taking all this into account, there are three distinctive threats that can be countered at the level of the operational environment.

Threat #1: Unintentional Misconfiguration

The main concern in virtualized deployments is to sustain server isolation. This includes separation of network segments, deployment of servers in security zones and possibly segregation of administrative access. All of this requires careful configuration of virtual hardware resources. Since none of these resources is actually visible, like colored cables for instance, errors are easier to make and harder to detect. Possible errors include:

- unnecessary communication channels between production guests and the execution environment
- unnecessary access to network segments, which also includes the risk of an attack over this interface
- unintentional shorts between different security zones
- misplacement of guests in a wrong security zone because of errors in guest migration
- insufficient hardware allocation, leading to unexpected overloading

Evaluation

"To err is human"[3], and there is nothing we can do about that. However, there is much we can do to make errors less likely. How likely they actually are has much to do with the level of training, the motivation and level of load on the administrators, as well as how the virtual environment is planned and how the virtual infrastructure is implemented. As in the non-virtualized world, administrators can accidentally destroy masses of data in a short time. The infamous Unix command rm -rf * exemplifies what damage an administrator can do if he or she isn't concentrating well enough. This stays true in virtualized environments, just that now the administrator has more choice for the kind of error.

[3] Seneca the Younger, http://en.wikipedia.org/wiki/Ex_officio#E

Counter Measures

Preventing human error is a near-impossible task. However, there are many things you can do to minimize the chance for such errors. The following counter measures help to achieve this.

Counter Measure #1: Operational Guidelines

Just like for any other platform, appliance, application or, essentially, datacenter, you need to define operational guidelines for your Oracle VM Server for SPARC environment. These guidelines are the foundation of all operational security measures, defining how and what should be done. Regular checks should be performed to make sure these guidelines are still up to date and adequate, and also that they are actually being followed in everyday operations. They should include:

- Patch Management for all components of the environment
- Change Management to allow for well defined, retracable and secure implementation of changes
- Regular check of logfiles
- Monitoring of the integrity and availability of the environment

There are several other, more technical measures that can be taken to reduce the risk of unintentional actions. These are all related to access control for LDoms Manager, and are discussed in chapter "LDoms Manager" on page 23.

Threat #2: Errors in the Architecture of the Virtual Environment

Virtualized hosts need disk and network access. Both are usually implemented by some sort of virtualization technique for network and disk. When moving from dedicated servers to virtualized environments, the disk configurations don't usually change very much. SANs with the methods of mapping LUNs to hosts or host groups are commonly used in enterprise datacenters today, and virtualization doesn't introduce any new features or requirements.

With networking, this is different. Where individual servers usually need no more than 2-4, sometimes perhaps 8 network ports, this number would need to be multiplied by the number of guests hosted by one virtualization platform if no sharing of interfaces was possible. This is unrealistic in most cases, since the larger platforms are capable of supporting 100 or more guest systems. Therefore, network virtualization needs to be employed in most cases to satisfy all networking needs. Today, VLAN tagging is often used to carry more than one Ethernet segment over a single wire and network port. Other techniques like NIC virtualization are just now evolving, for example in Solaris 11 Express[4]. However, the isolation of these network segments could be circumvented, with network packets ending up in the wrong segment.

When planing the virtual environment, keeping up the isolation of different security zones needs to be carefully considered, thinking not only about the shared hardware of the actual server platform, but also about other shared components like network switches, SAN switches etc. If the overall

[4] http://hub.opensolaris.org/bin/view/Project+crossbow/

architecture doesn't provide for a breach in isolation, it is possible that an attacker can jump from one security zone to another, thus further increasing the potential amount of damage he or she could do.

Evaluation

This threat is a catch-all. Keeping up the isolation of guests and security zones is the primary goal of everyone involved – the developers of the virtualization hardware and software, the administrators and the architects that design the complete virtualized solution. The architecture needs to be designed in a robust way, anticipating possible errors and attacks, and providing for the worst case with second and third lines of defense. This is not trivial, and the design is usually the result of many compromises between highest security, manageable complexity and cost. However, if done right, such an architecture will help confine potential security issues. Errors in the design, on the other hand, carry the potential of amplifying the danger of other threats, as everyone relies on alleged security nets that in truth have holes no one knows about.

Counter Measures

Just like the description of the threat is very general, so are the measures available to address it. Plan with security in mind.

Counter Measure #2: Carefully Assigning Guests to Hardware Platforms

As described above, breaking the isolation is the ultimate goal of any attack, and also the worst possible case for a virtualized environment. Usually, individual guest systems can be grouped into groups with equal security requirements and privileges, or security zones. Assigning only guests within one such zone to a certain hardware platform assures that even a breach of isolation will not allow the crossing into a different security zone.

Counter Measure #3: Planing LDoms Migration

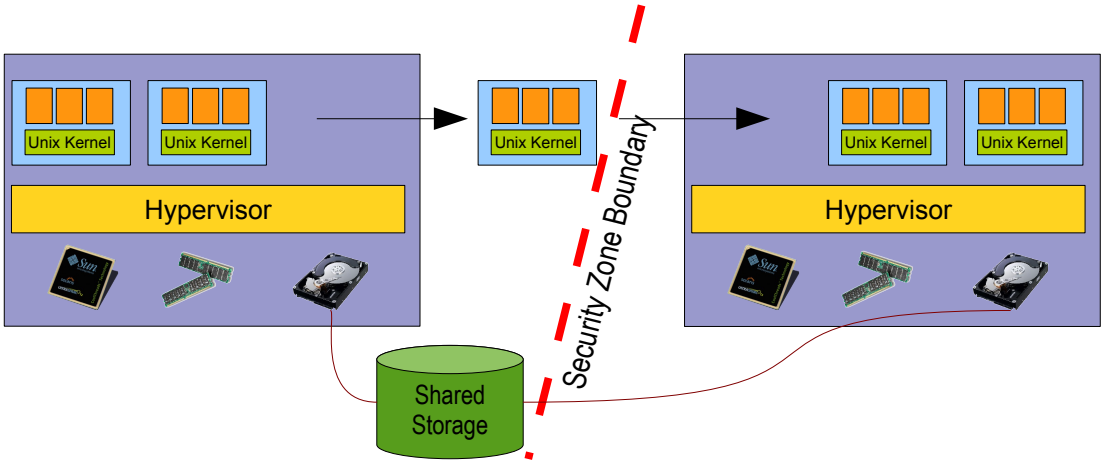

Figure 6: Domain Migration across Security Boundaries

Every security concept contains several security zones with different security requirements. These zones are typically isolated from each other to reflect these requirements. When deploying virtualized systems, these zones should be respected, and all the guests of one system should be located in only one zone. In this case, an attacker who successfully breaches guest isolation will not also reach out of the initially compromised security zone. The feature of Live Domain Migration (which is also available in other virtualization platforms) has the potential to break this scheme, if a guest is inadvertently migrated to a platform living in a different security zone. To avoid this, carefully planing guest migration and taking measures to disallow the migration across security zone boundaries is recommended. As of Oracle VM Server for SPARC 2.2, domains with physical IO resources can not be live migrated, although it is conceivable that this limitation might be removed in future versions, but cold migration is still possible, which makes this counter measure applicable to all deployment models.

Counter Measure #4: Correct Virtual Connections

In the physical world, individual systems are usually grouped, and groups are isolated from each other for security or other reasons. This is unchanged in a virtualized environment. However, unlike in the physical world, network connections are invisible and it is hard to keep track of all connections. Errors in virtual connections might give guests access to network segments they should not be connected to, perhaps circumventing firewalls or security groups.

To reduce the risk of implementation errors, explicit planning and documentation of all virtual and physical connections is vital. Plan your guest's connections with simplicity and manageability in mind. Document your plan, and check the correctness of your implementation against your plan before you go into production – and at regular intervals after that.

Counter Measure #5: VLAN Tagging

VLAN tagging can be used to consolidate several Ethernet segments onto one physical network. This feature is also available in virtual switches. To mitigate the risks involved with software errors in the implementation of virtual switches, configure one virtual switch per VLAN. To further protect against errors in the Ethernet driver, refrain from using tagged VLANs. However, the probability for such errors is low, as this vulnerability of tagged VLANs is well known and previous intrusion tests on Oracle's Sun SPARC T-Series platform with Oracle VM Server for SPARC have not shown it vulnerable.

Counter Measure #6: Virtual Security Appliances

Security appliances like packet filters or firewalls are themselves instruments of isolation. They are used to protect the isolation of security zones. Porting such a service from a legacy SPARC system to Oracle VM Server for SPARC is technically easy, and consolidating several such devices onto one platform to benefit from the possibilities of (free) virtual networking within the platform is tempting. However, they are subject to all the same threats as any other guest domain, including a possible breach of isolation. Therefore, all aspects of risk and security should be carefully considered before deciding to virtualize such a service.

Threat #3: Side Effects Through Shared Resources

As mentioned on page 6, the sharing of resources in virtualized environments opens the potential for attacks by overloading one or more of these resources to a point where other components are negatively affected. In the worst case, this will render the affected resource unusable to one or more guest domains, effectively stopping the guest. This would be a typical Denial-of-Service type attack.

In the case of Oracle VM Server for SPARC, not all possible resources are affected by this threat. CPU and memory resources are assigned exclusively to each guest, preventing abuse in most cases. Cache thrashing of cache areas shared between strands assigned to two guests or overloading of memory bandwidth are possible exceptions. However, they would only slow down, not completely stop affected guests. Disk and network services however are typically shared between several guests in consolidation scenarios, provided by one or more service domains. The assignment and distribution of these resources needs to be considered carefully. On the upside, any configuration that permits maximum performance and resource utilization will at the same time minimize the risk of side effects, while scenarios using hardware partitioning, where little to no sharing takes place, are not affected.

Evaluation

It is fairly easy to saturate a network link or overload a disk assigned to a single guest. If other guests share that same physical resource – the same network link or storage space on the same physical disk, side effects through common use are unavoidable. The worst possible damage would be the unavailability of a service offered by an affected guest for the duration of the attack. The attack target would not be compromised, no data would be lost, and service restoration would be quick. Furthermore, minimizing the effects of this threat is relatively easy. As it is limited to network and disk

resources on Oracle VM Server for SPARC, and also only applies to consolication scenarios and not to hardware partitioning, this threat can be considered a minor threat, although it should be kept in mind, like all others.

Counter Measures

This threat is limited in impact. It can be further mitigated by following these recommendations:

Counter Measure #7: Carefully Assigning Hardware Resources

Guest domains should only be assigned hardware resources they actually need. Any unused resource, like a network port or DVD drive required during installation only, should be unassigned once it is no longer needed. This helps keep the overall installation clean and reduces the possible entry points for an attacker.

Counter Measure #8: Careful Assignment of Shared Hardware

Shared hardware resources like physical network ports present a possible target for Denial-of-Service attacks as described above. This is true for all types of sharing – both virtual networks using Ldom vSwitches and virtual networking using SR-IOV virtualization. Carefully selecting which guests share which hardware resource limits the impact of such attacks to one group of guests. Guests sharing hardware resources could, for example, be grouped by the same availability or security requirements. Beyond grouping, resource controls of different kinds can be applied. Since CPU and memory are assigned exclusively to each guest, there is no need for further control to avoid impacting other guests[5]. However, disk and network resources could be considered here. Separating disk access through dedicated physical access paths or through dedicated virtual disk services (vds) can help to mitigate cross-impacting. Similar measures are possible for network resources.

Summary

Just like in the non virtualized world, a well run datacenter, architectures that are clean and simple, documentation that is up to date and easy to use and administrators that are motivated and well trained are factors that contribute to reliability and security of an IT operation. All the counter measures discussed above have to do with understanding the technical details of your deployment and their security implications. Plan carefully, document well, and keep your architecture as simple as possible. Make sure you understand the implications of virtualized hardware, and you will be well prepared for a secure deployment of Oracle VM Server for SPARC.

The Execution Environment

The components of the Execution Environment as shown in Figure 5 have been described in section "The Execution Environment" on page 3. Particularly in a consolidation scenario, each of these

[5] CPU and memory resource controls should be considered to contain the application running within a guest. This is a recommended procedure for all deployments. For best results, use Solaris Zones.

components provides certain services which together form the overall platform to run the production guest systems. Their configuration is vitally important for the integrity of the system. In this section, we will have a closer look at each of these components and potential threats specific to each. Of course, there are also some threats and counter measures applicable to more than one of these components. They will be discussed the first time they are touched.

Depending on the configuration for a consolidation scenario or for hardware partitioning, the individual components' threat level can be very different. For example, an IO domain that offers disk and network IO services to several dozen guest systems presents a different kind of attack surface and a different level of risk of damage compared to an IO domain that uses all its IO for its own purposes and offers no IO services, as would be typical for a hardware partitioning scenario. These differences need to be considered when designing the overall security architecture of the system.

In a theoretical approach where we don't look at technical implementation details, the Execution Environment as a whole is a potential target for an attacker. Generally, an attacker will not care whether the component under attack is the control domain or an IO domain, as long as the attack is successful. To address this overall attack target, the first threat to discuss is

Threat #4: Manipulation of the Execution Environment

By manipulating the execution environment, any level of control can be reached. This can be achieved in numerous ways, for example by installing manipulated firmware in the ILOM to snoop on all guest IO from within an IO domain. If an attacker has control of the configuration utility of the execution environment, he or she can access and change the system's configuration in any way suitable to his or her overall goals. This is a threat that applies to all virtualization solutions. Depending on the implementation, protecting against it will require different counter measures. In Oracle VM Server for SPARC, this means that once the attacker has control of the control domain, any possible reconfiguration of the system is at his or her disposal. If an IO domain is compromised, changes to attached storage, for example boot disks of guest systems, is possible.

Evaluation

With regards to damage, this is the most severe threat of all. Once an attacker has gained the ability to manipulate the execution environment, he or she practically owns the system. Damage is maximal, the attacker has access to all attached storage, all network ports and can potentially read and manipulate all data, covert or openly. Actually getting to this point requires a successful break-in into either the ILOM or any of the domains of the execution environment, either over the network or through some error in the virtualization stack. However, both ILOM and these domains can not usually be attacked directly. Therefore, a successful attack is very hard to perform.

Counter Measures

Many of the measures listed below are standard security practice and should be implemented on any system. Together, they present an additional layer of protection around the execution environment that further reduces the risk of intrusion and manipulation. Although this threat also applies to other

virtualization solutions, the counter measures are obviously not always independent of the underlying technology. The counter measures discussed here apply to all domains of the execution environment.

Counter Measure #9: Secure Interactive Access Paths

As is best practice on any production system, only those system accounts that are required should actually exists and be active. All other accounts should either be eliminated or disabled to prevent their misuse. While this is true for all systems, guest and execution environment alike, it should be a hard requirement for the latter. Accounts that are required for administration should be secured with strong passwords and should not be shared between different domains. Depending on the sensitivity of the system in question, implementing two-factor authentication or a two-person rule for certain actions should be considered.

To allow for complete traceability and accountability, no anonymous logins like "root" should be used. Rather, implement RBAC to grant individual administrators access to those functions they are entitled to. Of course, network access for administrative purposes should always use encrypting tools like ssh, and the administrator's workstations should be treated as high security systems.

Counter Measure #10: Minimizing Solaris

Software that isn't installed can not be hacked. Furthermore, it can not be active unintentionally, perhaps opening service ports that should not exists at all. This general security practice of course also applies here, especially to the domains of the execution environment. For details you should consult the Oracle VM Server for SPARC Administration Guide [4].

Counter Measure #11: Hardening Solaris

In addition to installing a minimized Solaris, the configuration of many software packages allows additional "hardening" against attacks. A significant first step can also be to run `netservices limited`, effectively disabling all network services except ssh. Note that this behavior is already the default in Solaris 11. Additional measures are described in the Beginner's Guide to LDoms [2] and other Oracle VM Server for SPARC documentation and include installing SUNWjass on Solaris 10 and its tailored LDoms driver. More information about how to secure Solaris can be found at the Solaris Security Page[6].

Counter Measure #12: Role Separation & Application Isolation

In a consolidation scenario, the Execuction Environment should be viewed much like a system controller. Although Solaris is installed on all of its domains, production applications should only be deployed in regular guest domains without further privileges, never in a domain that is part of the execution environment. (This is the same recommendation as for the system controllers of the Sun E25k servers, which also run Solaris.) Production applications are necessarily connected to other systems and are much more exposed to outside attacks. They should be installed only on guest

[6] http://www.oracle.com/us/products/servers-storage/solaris/security/index.html

systems. The execution environment should only provide the necessary infrastructure for these guests. Separating the two allows for granularity in administration privileges. Administrators of the production guests need not have access to the execution environment, and vice versa. Furthermore, the different roles of the execution environment (control domain, IO domain, etc.) should be assigned to different domains if possible. This reduces the amount of damage that can be done, should any one of these domains be compromised.

On the other hand, when configured for Hardware Partitioning, separation and isolation are usually already achieved through the architecture of the overall system. The role of the execution environment is very much limited to system configuration, as guest systems with virtual IO typically don't exist and all domains, typically running as root domains, own their exclusive IO hardware. There are no dependencies between these domains, and isolation is typically higher than in the consolidation scenario. The only exception to this is the control domain. While also usually a root domain that owns IO hardware, it also owns the privilege to control the hypervisor and thus change the overall system configuration. This higher level of privilege should be taken into account when deploying production applications on this domain. Access to the `ldm` command should be restricted using RBAC and other means. Furthermore, if possible, applications running on this domain should be deployed in Solaris Zones[7]. This allows complete control of the resource consumption of the application and fully isolates the application and its administrators (as well as any intruder) from the global zone with its hyperprivileged access. With these measures in place, the control domain in a hardware partitioned system is well protected and isolated from the application environment. Depending on the thread and risk assessment described in section "Grouping Servers into Security Classes and Zones" on page 31, this may be sufficient to allow the deployment of applications in the control domain.

Furthermore, you can extend the concept of role separation to the network environment used to connect your different servers. This leads to

[7] Solaris Zones were previously called Solaris Containers. There is no technical difference between the two terms.

Counter Measure #13: Dedicated Management Network

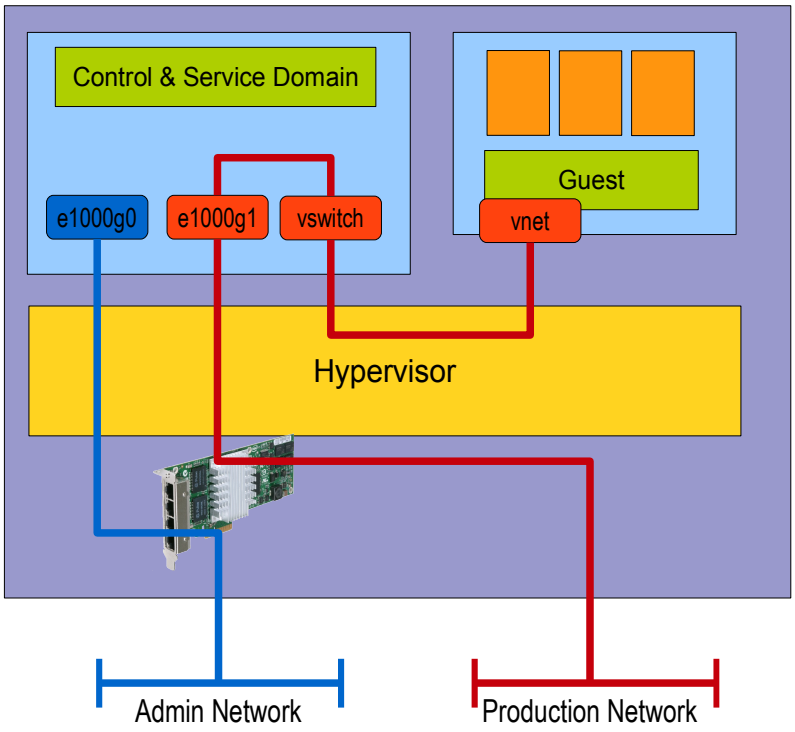

Figure 7: Dedicated Management Network

It has been a recommended best practice for all servers equipped with service processors of any kind to connect these to a dedicated management network. The same reasons now also apply to the domains of the execution environment in a consolidation scenario. If networked at all, these domains should be hosted on their own, dedicated network, with no direct connection to those networks assigned to production. Of course, it is also technically possible not to connect them to any network at all and do all administrative work through the single console connection made available by the ILOM service processor. However, while possible, this makes administration sufficiently cumbersome to be impracticable in most cases. Similar to role separation, separating production and administration networks protects against both malicious and unintentional eavesdropping and manipulations. It also eliminates the possibility of an attack on the execution environment from the guest systems over the shared network.

With the above counter measures, which apply to all components, the Execution Environment is well protected against manipulation attempts. In the next few sections, we will have a closer look at individual parts of the Execution Environment and how they can be further protected.

The System ILOM

All current Oracle Sun systems, be they SPARC or x86 based, come with a built-in system controller (ILOM) that controls basic environmentals like fan speed and chassis power. It is also used for firmware upgrades and provides the system console for the primary domain. The ILOM is accessible through a serial connection and through a network port. Here, the user has a choice of ssh, HTTP, HTTPS, SNMP and IPMI. The enterprise class family of servers has a similar device called XSCF,

which offers similar functions. Many of the considerations for the ILOM also apply to the XSCF of these larger machines.

Threat #5: Complete System Denial-of-Service

If an attacker can gain control of the system ILOM, he or she can remove power from all running guests, or install manipulated firmware to gain access to at least one of the guests. This scenario is applicable to all systems that come with such a controller device, which is common with all servers today. It has, at first sight, nothing to do with virtualization as such. However, in the virtualized world, the damage that can be done in such a case is much larger, since more than one server system – potentially several dozen – are at risk, all housed in the same system enclosure.

Likewise, if an attacker gains control over the control and/or IO domain, he or she can easily disable all dependent guest systems by shutting down the corresponding IO services.

Evaluation

The ILOM of Oracle's volume servers is usually networked, connected to an administration network. As is with any networked device, it is susceptible to attacks over the network. Access to the ILOM is further possible from the control domain of the system, using IPMI with the BMC access module. Since it controls the very basic functions of the server, it should be well protected and isolated from normal production networks. Likewise, service domains could be attacked either over the network, or through an error in the virtualization stack. A blocking of guest IO or a system shutdown would then be possible. The damage, although of course depending on the applications running on the guest systems, is limited, as data is neither lost nor compromised. On the other hand, the number of guests affected can be large, multiplying the individual guest's damage done. Therefore, protecting against the possibility of this thread and limiting potential damage is important.

Counter Measures

There are several measures that can be taken beyond those that directly affect the configuration of the ILOM. They will be discussed as we discuss the securing of the control, service and IO domains.

Counter Measure #14: Securing the ILOM

The ILOM, as the system service processor, has full control over chassis power, Oracle VM Server for SPARC startup configurations, console access to the primary domain and more. Therefore, protecting it is vitally important for overall system security. Naturally, placing the ILOM's network port in a separate network segment is a requirement. Ideally, this is separate from the administrative network used for the domains of the execution environment. Other measures to secure the ILOM include:

- Disabling all services that are not required for operation. Some of the more obvious are
  - http
  - IPMI

  - SNMP
- Others might be
  - https
  - ssh
- Configuring dedicated accounts for administrators granting them only the rights they need. This is especially important for console access, firmware upgrades and setting of startup configurations.

The Hypervisor

The hypervisor is the layer of software and supporting hardware that implements and controls the virtualization of real hardware. (See Figure 5 on page 8.) The components in question are:

- The actual hypervisor, which is implemented in firmware and supported by the CPU of the systems.
- The kernel modules running in the control domain to configure the hypervisor
- The kernel modules and daemons running in IO and service domains to provide virtualized IO.
- The kernel modules and device drivers running in the guests to access virtualized IO devices.
- The latter two include the kernel modules needed to communicate via Logical Domain Cannels (LDC).

All of the functionality of the execution environment and its domains is based on functionality implemented in the hypervisor. From a security point of view, they act as layers of protection around the hypervisor.

Threat #6: Breaking the Isolation

As already mentioned in the introduction of this chapter, the primary goal of an attacker is to break out of the isolated runtime environment provided by the hypervisor and hijack other guests or the whole system. This would be achieved by exploiting one or more bugs in the implementation or design of any part of the overall execution environment. This is the most general threat possible, with the potential for the most severe damage to a system. The attacker could, in the worst case, take over not just one or more guests, but eventually control the complete system, hypervisor and all attached hardware.

Evaluation

As with all software, it is impossible to preclude the existence of errors. However, the system is designed in a very modular way, with different levels of privileges granted to guests, the hypervisor and the control domain. Each functional module is implemented in a separate kernel module, device driver or daemon, which can be configured in many different ways. This modularity requires clean APIs and simple communication protocols, reducing the overall risk for error.

Even if an exploit for such a potential error seems rather unlikely, if it does exist, the damage that can be done in such a case is enormous. It means that the whole system is under the control of the attacker, he or she has complete control over

- the platform
- the control domain
- all disks and network ports attached to IO domains
- all disk and network services configured to guest domains
- all guest systems

This is the worst possible case thinkable in virtualized environments. Therefore, unlikely as it might be, this case needs to be considered in any and all deployments of virtualized environments.

Counter Measures

The measures described in this section focus on protecting the core of the hypervisor itself. Of course, they apply to surrounding layers of software in the domains of the execution environment just as well. Likewise, all measures suitable to protect these also add to the protection of the hypervisor, as these domains can be viewed as layers of protection around the hypervisor itself.

Counter Measure #15: Validating Firmware and Software Signatures

System firmware and OS patches are usually downloaded from Oracle's websites and could possibly be manipulated. To avoid the infection with trojan versions of patches or system firmware, these software packages should be checked against their MD5 checksums, which are always published by Oracle, before being installed. This guarantees the integrity of these updates.

Counter Measure #16: Validating Kernel Modules

Oracle VM Server for SPARC uses several drivers and kernel modules in both the guest domain and the domains of the execution environment to implement to overall virtualization system. Similar to the integrity checks of software and firmware updates, checking the integrity of these kernel modules gives additional protection against manipulated versions that an attacker might want to install in a compromised guest system. All kernel modules and most binaries distributed with Solaris carry a digital signature. This signature can be checked using the elfsign utility to verify their genuine origin. Of course, the integrity of the elfsign utility itself needs to be established first. To automate this process, consider configuring BART (Basic Audit and Reporting Tool) to check on any changes in Solaris binary and configuration files.

The Blueprint "Integrating BART and the Solaris Fingerprint Database in the Solaris 10 Operating System" [5] describes how to combine BART and the Solaris Fingerprint Database to automatically perform similar integrity checks. Although this database has been discontinued, the concepts

described in this Blueprint can be carried over to utilize elfsign and BART in a similar manner. For Solaris 11, the IPS packaging system itself offers much of the same functionality[8].

The Control Domain

The control domain is the one domain that can change the configuration of the hypervisor, and thus has control over all attached hardware resources – CPU threads, memory and IO resources. Although it is only needed for configuration tasks and could theoretically be shut down once the system is configured, keeping this domain safe is important. Since in most practical implementations, the control domain also has the role of an IO- and service domain, the additional considerations for these components of the execution environment also apply.

Threat #7: Control Domain Denial-of-Service

This threat describes the possible shutdown of the control domain, or at least the software components required to reconfigure the hypervisor. Both result in a denial of service of the configuration tools that constitute this service. As the control domain is only needed for configuration changes, this does not immediately affect any of the guest domains – provided that they access their network and disk resources through other service domains.

Evaluation

Attacking the control domain over the network is equivalent to attacking any other properly protected Solaris instance, success is just as (un)likely. The likelihood of a successful penetration of the virtualization stack, as mentioned above, also seems low, but needs to be taken into consideration. The damage of a shut-down or similar denial of service of the control domain is relatively low, as all guests will continue to run. Of course, if the control domain also acts as a service domain for other guests, this will impact these guests.

Counter Measures

Protection for the control domain consists of several methods for access control, both for the domain itself, as well as for the configuration utility, LDoms Manager. All of these are normal security best practices also recommended in other circumstances. Of course, the more general measures described for the Execution Environment in general also apply.

Counter Measure #17: Console Access

In cases where highest security requirements need to be met, not configuring any administrative network access to the execution environment's domains is an option. In such cases, all administration is done using the ILOM's console service to the control domain which allows console access to all other domains through the vntsd service. This scenario allows to completely disable sshd and other

[8] https://blogs.oracle.com/cmt/entry/solaris_fingerprint_database_how_it

network services for the execution environment, reducing the required network ports to those that are needed for production guests.

While this option reduces the risk of being attacked over the administrative network, it should be carefully considered, since this limits the number of concurrent administrative logins to one, and will be seen as an annoying restriction by the administrators.

LDoms Manager

LDoms Manager is the configuration utility used to configure the hypervisor. It runs exclusively in the control domain and is used to create and configure all domains and their hardware resources. Naturally, access to this utility needs to be protected. Depending on security needs, its use should also be logged and monitored. This is discussed in this section.

Threat #8: Unauthorized Use of Configuration Utilities

It doesn't always have to be a bribed or dissatisfied administrator that misuses his or her system access. It could just as well be an outside attacker who took control of an administrator's user ID or an administrator from a different group that has access to a system even though it is not in his scope. In any case, there are several possibilities for someone to have unauthorized access to the configuration utilities of the execution environment.

Evaluation

Essentially, there are two basic scenarios here. One is an administrator that has access to a system he or she doesn't need to access. This can easily be address by well maintained identity management and other technical means. The other involves criminal interest, either by a hacker or by an employee. The latter is a fact of life that we can not deny, thus we have to be prepared for it. On the damage side, once someone has access to the configuration utility of the execution environment, he or she essentially "owns" the platform. Damage can be substantial and includes loss of service as well as disclosure, manipulation and loss of data.

Counter Measures

Measures suited to counter unauthorized access of course rely on a well maintained identity management system being in place. The additional measures listed below address both scenarios from above. Implementing strict and fine-grained access control helps protect against intentional and unintentional access. Other measures, like the two-person-rule, also help against the more criminal approaches.

Counter Measure #18: Applying the Two-Person Rule

In very sensitive environments, you should consider implementing a two-person rule for LDom Manager and other administrative tools. This can be enforced using RBAC. How this is done is described in a Blueprint: "Enforcing the Two-Person Rule Via Role-Based Access Control in the

Solaris 10 Operating System" [6]. This protects not only against many social engineering attacks, but also against compromised administrative accounts and normal human error.

Counter Measure #19: RBAC for LDoms Manager

As already mentioned above, using RBAC for the ldm command allows for fine grain access control while maintaining complete retraceability. Configuring RBAC is described in the Oracle VM Server for SPARC Admin Guide [4]. Using RBAC helps safeguard against human errors, since not all features of the ldm command are available to all administrators. It also reduces the damage involved with a compromised administrator's access.

Counter Measure #20: Hardening LDoms Manager

LDoms Manager features several network services for access, monitoring and domain migration. Not all of these are needed in all cases. It is good security practice to disable these network services if not used. They are:

- LDom Migration Service on TCP ports 4983 and 8101
- XMPPSupport on TCP port 6482
- SNMP on UDP port 161
- LDom Discovery Service on multicast address 239.129.9.27, port 64535

How to disable these services is documented in the Oracle VM Server for SPARC Administration Guide [4]. The exception here is the LDom Discovery Service, which can not be disabled while ldmd is still running. However, using Solaris IP filtering, access to this service can be blocked.

This reduces the attack surface of LDoms Manager to the minimum required to operate it normally. Denial of Service attacks and other attempts to misuse these network services can be countered effectively in this way, preventing unauthorized use of the utility.

Counter Measure #21: Auditing for LDoms Manager

LDoms Manager has been integrated into the Solaris 10 Auditing system. As the configuration of the execution environment and the guest domains is vital to the security of the overall system, any changes to this configuration need to be logged to allow for later retracing of possibly hostile actions. The audit logs should be scanned regularly, and possibly copied to a separate system for secure archival. Configuration of BSM for LDoms Manager is described in the Oracle VM Server for SPARC Admin Guide [4] on page 30.

The Service Domain

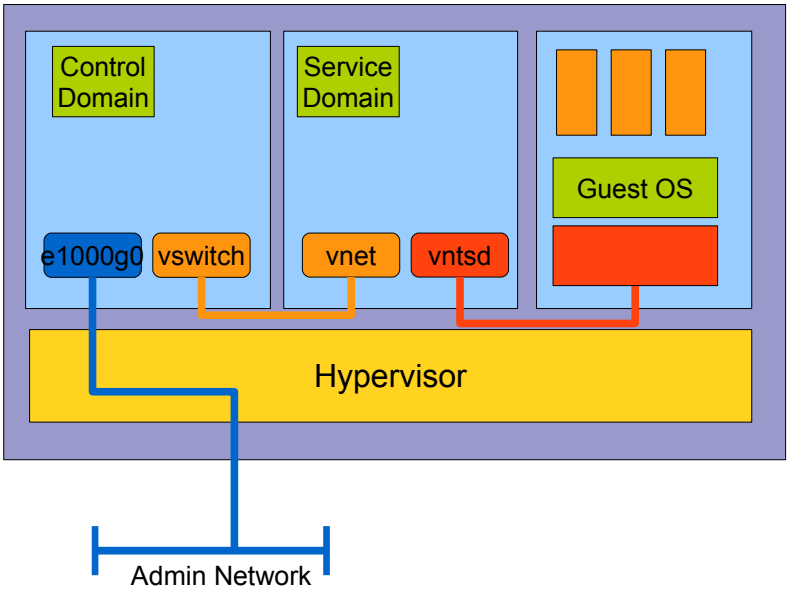

Figure 8: An Example for a Service Domain

A service domain is a domain that provides some virtual service to other guest domains of the system. Typically, this would be a virtual switch, virtual disk or virtual console service. Figure 7 shows an example for a service domain offering console services. Note that usually, the control domain is also the one used to host the console services, and is thus also a service domain. Typically, the domains of the execution environment combine the functions of control domain, IO domain and service domain in one or two domains. For the sake of discussing threat situations that are related to each of these functions, they are discussed separately here.

Threat #9: Manipulation of a Service Domain

If an attacker gains control over a service domain, he or she can listen in on any communication happening through the services offered by this domain, or manipulate data. Typically, this might be console access to guests or network or disk services. Note that the latter two also require this domain to be an IO domain.

Evaluation

While the attack strategies and the likelihood for success are the same as for an attack on the control domain, the possible damage is substantially less since he or she can not change the system configuration. Possible damage does include the theft or manipulation of data being offered by the service domain. However, the attacker is limited to manipulating the service, he or she has no access to any data sources. Depending on the service, actually making malicious use of this requires the exchange of kernel modules.

Counter Measures

Counter Measure #22: Granular Service Domains

If possible, each service domain should only offer one service to its clients. Should one of these service domains be compromised, only this one service would fail and guests relying on other services would continue to run. Of course, this needs to be weighed against the additional complexity. In case of IO services, having redundant IO domains as described below is recommended.

Counter Measure #23: No Network Connection between Service Domains and Guests

The default behavior for domains running a vSwitch service (a service providing a virtual switch) is to disallow traffic between the service domain and any guest domain connected to the vSwitch. Keeping this default behavior eliminates the risk of the service domain being attacked from a compromised guest via this connection and is thus highly recommended. In other, less modular virtualization environments, the equivalent measure would be to disallow any network connection between guest systems and the execution environment.

Counter Measure #24: Restricted Access to Virtual Consoles

Similar to the ldm command, access to individual virtual consoles can be limited to those users that should have access to them. Again, the benefit here is that no one administrator has access to all consoles, preventing both inadvertent access to the wrong system and malicious access to consoles other than those assigned to the potentially compromised account. Configuration details can be found in the Oracle VM Server for SPARC Admin Guide [4].

The IO Domain

Any domain that has direct access to physical IO devices like network ports or disks is an IO domain in the terminology of Oracle VM Server for SPARC. The details of configuring IO domains are discussed in the Oracle VM Server for SPARC Admin Guide [4]. In a consolidation scenario, ownership of hardware alone is of no value and any IO domain will also be a service domain. It will provide IO services of some sort to guests, providing them access to some or all of the hardware available to this IO domain. Therefore, the discussion regarding service domains also applies to IO domains when running in a consolidation scenario. Additionally, there are several other scenarios worth considering:

Threat #10: Denial-of-Service of IO Domain or Service Domain

If an attacker somehow succeeds in blocking the IO services of an IO domain, all dependent guests will equally be blocked. A successful DoS attack on such a domain could be achieved by overloading the backend network or disk infrastructure, or by injecting some fault into the domain, forcing it to hang or panic. Likewise, if the attacker suspends the services of a service domain, any guest systems depending on these services will immediately hang, waiting for the IO services to resume. Note that by default, these domains will not crash, but rather hang indefinitely, until the IO service is resumed.

Evaluation

Denial-of-Service attacks on any domain over the network are a common type of attack. They can be successful simply because network ports by nature are open for communication and thus can be overwhelmed with traffic. A resulting loss of service to guests would block these guests, but nothing more. A similar attack on disk resources is more difficult. It could either be conducted via some SAN infrastructure or against the IO domain itself. In all cases, the only damage that can be done is a temporary halt of all guest systems depending on the attacked IO domain. Of course, depending on the nature of the service delivered by these guests, the real world impact can be substantial. However, data is neither compromised nor lost, and the system configuration remains intact. If the IO domain is deployed in a hardware partitioning scenario, there are no dependent guests. Thus, the only domain affected would be the IO domain itself, and the attack would be identical to an attack on any normal system and its IO subsystem.

Counter Measures

Counter Measure #25: Granular IO Domains

Configuring multiple IO domains reduces the impact of one domain failing or being compromised. Until Oracle VM Server for SPARC (Logical Domains) version 1.3, you could configure one IO domain per PCIe Bus. With version 2.0, it is possible to assign individual PCIe slots to a guest, giving it IO domain-like capabilities. The limitation here is that if the domain which actually owns the PCIe bus crashes, it will reset that bus, leading to a subsequent crash of the domain that was assigned the individual slot. This feature, while definitely an improvement, does not fully eliminate the need for two true IO domains, each owning a separate PCIe bus. Taking granularity one step further, you can configure redundancy:

Counter Measure #26: Redundant Hardware and IO-Domains

High Availability is always also a concept of enhanced security, since it helps to make services more robust against denial-of-service attacks. In the case of Oracle VM Server for SPARC, this means extending the concept of redundant disk and network resources into redundant IO domains. This configuration option not only allows rolling upgrades of the IO domains, it also protects against the impact of a failed IO domain due to a successful denial-of-service attack. With the advent of Hybrid IO, redundant IO domains will eventually be less important, since guest systems can then have direct access to individual IO devices. However, for installations with many guest systems where Hybrid IO is not an option, redundant IO domains should be considered. Most other virtualization platforms lack this feature. Here, avoiding guest failure due to failed hardware access needs to be accomplished in other ways, if possible at all.

Threat #11: Manipulation in an IO Domain

On first sight, this threat looks similar to the manipulation of a service domain. The difference is subtle, but worth noting. An IO domain has direct access to the backend devices, usually disks, which

it virtualizes in some form and then offers to guests in its role as a service domain. In case of a successful attack, the attacker has full access to these devices. He or she can thus read sensitive data or manipulate software on the boot disks of guests, striving to add them to the list of his or her victims. The attack strategy is again the same as for attacks on any other domain – either over the network, or through some error in the isolation of individual LDoms.

Evaluation

This scenario is just as likely as a successful attack on a service domain or the control domain. However, the IO domain is a very attractive target, given the potential access to a large number of disk devices. It is therefore important to have this threat in mind when dealing with sensitive data in a guest running on virtualized disks.

Counter Measures

Counter Measure #27: Protecting Virtual Disks

As described above, once an IO domain has been compromised, the attacker has full access to the guest's virtual disks. Protecting their contents against unauthorized access or manipulation can be accomplished in two ways:

1. Encrypting the contents of the virtual disks. This has to be done by the guest using the virtual disks. Currently, there is no volume manager or file system available for Solaris 10 that would allow this transparently for applications. However, there are many applications that are able to encrypt their own data, pgp/gpg or Oracle 11g encrypted tablespaces being two examples. Solaris 11 includes ZFS encrypted datasets, which also provide transparent encryption of all data stored in the filesystem.
2. Distributing the data over several virtual disks serviced by different IO domains. For example, a guest could create a striped (RAID 1/RAID 5) volume, striping over several virtual disks obtained from two IO domains. If one of these IO domains was compromised, the attacker would have a very hard time making use of the portion of data available to him. This is an example of using granular IO domains mentioned above.

The Guest Domains

The guest systems are not part of the execution environment. However, as they usually require some sort of network connectivity to the outside world, they are the most likely target for a generic attack. If this is successful, the attacker could then discover that he or she has hacked into a virtualized system, and begin to launch attacks on the execution environment.

Counter Measure #28: Securing the Guest OS

As mentioned, the guest operating system is usually the first line of defense against any attack. With the exception of attacks coming from within the datacenter, for example from a already compromised

system, an attacker will need to break into a guest system with connections to the outside world before he or she can attempt to break guest isolation and capture the complete environment. Therefore, just like in any other environment, hardening the guest OS is highly recommended to make an attacker's life as hard as possible.

As an extension to the concept of a hardened guest OS, you can deploy your application in Solaris Zones. Solaris Zones add an additional layer of isolation between the application's network service and the operating system of the guest OS. A successful attack on the service would compromise only the zone and not the underlying operating system. This prevents the attacker from expanding his or her control beyond the resources allocated to the zone, making it substantially harder to eventually break guest isolation. Solaris 11 further improved the security features of Solaris Containes (called Solaris Zones in Solaris 11) with the concept of Immutable Zones. For more information please consult the Solaris 11 Administration Guide about Immutable Zones[9]. It is a recommended best practice to deploy all applications in Solaris Zones because of the additional layer of security as well as the comprehensive set of resource controls available for Zones. This is especially true in a scenario where LDoms are used for hardware partitioning. Here, Zones are the ideal way to introduce a layer of isolation between the hardware platform, which in this case is always also part of the execution environment, and the application environment.

There are numerous ways to secure the guest OS, depending on the individual requirements. These measures are beyond the scope of this paper. A good starting point is the Solaris Security Page at http://www.oracle.com/us/products/servers-storage/solaris/security/index.html or the book "Solaris 10 Security Essentials" [7].

Summary

Since this paper focuses on security implications of virtualization using Oracle VM Server for SPARC, it should be no surprise that most of the threats discussed above are related to some sort of manipulation of the execution environment. Essentially, overcoming guest isolation and manipulating either the control domain or a service domain are at the center of our attention. And basically, there are two main attack vectors to consider: Exploiting a bug in the hypervisor to break out of a guest system, or traditional attacks on the components of the execution environment. While they can never be totally precluded, breaches of guest isolation require a high level of skill and criminal effort to achieve, and are therefore less likely to be attempted than the more traditional types of attacks on the operating systems of control domain or service domain. These, on the other hand, are well known not only to a potential attacker, but also with regards to available counter measures. Damage control by containment of guests in security groups or zones and adequate defenses against traditional attacks are therefore the nature of the counter measures that will need to be deployed in order to obtain best possible security.

[9] http://docs.oracle.com/cd/E23824_01/html/821-1460/glhep.html#scrolltoc

Recommended Deployment Options

In the previous chapter, we discussed threats and counter measures applicable to virtualized environments and Oracle VM Server for SPARC. Essentially, this is all you need to develop your own security guideline. Evaluate yourself which threats apply to your servers, and which counter measures you deem appropriate in your particular case. There is no "one size fits all" solution. Most of the counter measures discussed above come at a price: They make everyday administration harder, they require more complex configurations or additional work, for example regular checking of logfiles. Likewise, not all threats apply equally to all situations: Some servers are more exposed than others, while others need extra protection because of the sensitivity of the data they process. Therefore, just like there are no two identical server situations, the ideal security solution would be a custom, tailored combination of counter measures for each situation.

In this chapter, we will discuss some general recommendations for any Oracle VM Server for SPARC deployment. We will further discuss three recommended sets of counter measures for three theoretical, but typical threat situations, all based on a **consolidation scenario**. Based on these threats, some configuration options will be recommended, which implement a suitable set of counter measures. Finally, we'll discuss some recommendations when using LDoms for hardware partitioning. With these examples, you will have an easier time deciding how to configure your own servers.

For all the recommendations below, keep in mind that they only address the threats specific to Oracle VM Server for SPARC or virtualization. Of course, normal security measures that apply to any server today, also apply here – in addition to these recommendations.

General Recommendations

Some of the counter measures discussed in the previous chapter are fundamental enough to be recommended for all Oracle VM Server for SPARC deployments. Having a **dedicated management network** for ILOM and the domains of the execution environment is one such fundamental recommendation. Maintaining **synchronized time** on all of these devices and to configure **logging**, possibly to a dedicated and separate log server, is another. Both of these measures are basic enough to expect them in most operational environments today, even without the additional challenges of virtualization. Just as vital is a well managed, centralized **access and identity management system**. New roles like LDom Manager access can simply be plugged into an existing infrastructure. Having a well established **change management** is a recommendation not only for virtualized environments, but for all larger datacenters. Nevertheless, the additional complexity of virtual environments is a challenge that can be countered by good change management. Finally, general best practices for any server should be to make sure you use encrypted channels like ssh to access all your systems' components, and to check the integrity of any system patch or firmware upgrade against their published fingerprints or MD5 checksums before applying them.

Grouping Servers into Security Classes and Zones

Threat levels are defined by two factors. First, the value or sensitivity of the data processed by the server in question. The higher the value of this data or the potential damage in case of a breach, the higher the security needs for this server. For example, a SunRay server that hosts internal user's desktop sessions would be less valuable than a database server processing a bank's customer accounts. Second, the level of exposure to potential attackers. If a server is located close to the internet, it is far more likely to be targeted by an outside attacker than a SunRay server located on internal networks which are protected by several levels of firewalls and other gateways. Always keep in mind that not all attacks are launched from the outside. The database hosting you customer's data should be well protected against unauthorized access from internal networks just as well!

Both of these factors need to be considered when defining the security requirements for a set of servers. In an abstract formula, you would have:

$$Security\ Needs = Value\ of\ Data * Probability\ of\ Breach$$

The higher either of the contributing factors, the higher the overall security needs. Since it is very difficult to calculate the exact probability of a breach, and almost as difficult to put a number to the value of your data or the cost or damage in case of a breach, you can work with good estimates. For the purpose of this paper, we will use the values shown in Figure 9.

Security Class	Security Need	Value of Data	Probability of Breach
Low	1,000	10,000	10.00%
Low	2,500	50,000	5.00%
Medium	25,000	500,000	5.00%
Medium	10,000	10,000,000	0.10%
High	100,000	100,000,000	0.10%

Figure 9: Security Classes based on Threat Level and Data Value

In addition to these security classes, you must also consider the additional concept of security zones. You might have several groups of servers with similar or identical security needs, according to your assessment like in Figure 9. However, these groups of servers might each serve very different purposes and live on different network segments within your enterprise. You will want to preserve the separation of these groups when migrating them to a virtualized environment, mainly to address the risks associated with domain migration.

Consolidation Scenario 1 – Low Security Class

In this first scenario, we will assume a group of servers with low security requirements. This means that they are located in networks well protected against outside intruders, or host data that is not very valuable to attackers and easily recoverable. Examples could be SunRay servers for internal users that service internal networks only, or Webservers behind a firewall and load balancer that have no local data. While the SunRay servers are very hard to attack from the outside since they are far away from the outer lines of defense, the webservers are rather close and a probable target for an attack. However, since they do not themselves store any valuable data, they might still be considered low risk servers.

Recommendations for servers in this security class are:

- Only deploy guest systems of the same security class and of only one security zone on one physical platform. This means that you can deploy several webservers on one platform, and several SunRay servers on another. But do not deploy SunRay servers and webservers on the same platform. Likewise, do not deploy servers with **higher** security requirements on this same platform.
- No network connection between the administration of the execution environment and your guest domains. This is essential to protect your platform against attacks over the network of a compromised guest system.
- One vSwitch for each network segment of your production network. This increases the level of isolation between these networks.
- Only assign required resources to guest systems.
- Deploy production applications in guest systems only, not in any domain of the execution environment.
- Apply normal Solaris hardening and security best practices to your guest systems like you would in a non virtualized environment.

All of these counter measures are aimed at protecting the isolation of both the execution environment and the individual guest systems. Most represent general best practices that also help to keep your overall configuration simple and easy to understand. They have no significant impact on administration, keeping the "security pain" at a minimum for administrators. The limitation of deploying only systems of the same security class and zone on one platform is needed for damage control and to protect guests of other security classes against possible breaches of guest isolation.

Consolidation Scenario 2 – Medium Security Class

In this second scenario, lets assume some servers that are either more exposed to attacks or carry more valuable data, although they don't yet require highest possible protection. Examples could be a corporate mail server or a database hosting the content of several websites. Both servers host data of considerable value, and the mail server needs to be available both on the internal and external network to accept and deliver emails.

In addition to the recommendations for low security requirements, the recommendations for servers in this security class are:

- Configure redundant IO domains. This doesn't only protect against the impact of losing one IO domain to an attacker, but also increases the overall availability of the platform, a usual requirement for services of this kind.
- Harden Solaris on all domains, the guests and the execution environment. Do this by installing SUNWjass as described in the Oracle VM Server for SPARC Admin Guide [4] or the Beginner's

Guide to LDoms [2]. For Solaris 11, review the Solaris 11 Security Guidelines[10]. In addition, install only a minimal set of required software packages (OS minimization).

- Deactivate all ports and network services of the control domain that you don't require. For example, LDom migration and discovery are not necessary in all cases.

- Configure RBAC to control access to the ldm command and the guest systems' consoles. This is described in the Oracle VM Server for SPARC Admin Guide [4].

These counter measures add a significant level of security, but also availability and serviceability to your platform. Granularity reduces the impact of a single successful attack. Redundancy reduces the impact of individual faults. It also allows for patching and other maintenance tasks in the execution environment without impacting your guest systems. A higher level of control over the access to the execution environment reduces both the attack surface of the administration interface and the probability of human error. It also requires a higher level of attention from the administrators. This might be seen as a drawback at first, but will eventually lead to higher levels of qualification.

Consolidation Scenario 3 – High Security Class

Finally, there are those servers that require administrators with some extra paranoia training. Besides those servers located at high security datacenters of military or government agencies, those are typically servers that host highly valuable and highly sensitive data. For example, a database server hosting a bank's customer accounts or the medical history of a hospital's patients. Those servers are not typically located close to the outside world, as direct access from the internet would be too risky. Nevertheless, they might be subject to attacks from the inside, either through disloyal employees or through a previous breach where the attacker has already captured other internal systems.

Additional recommendations for servers in this highest security class are:

- Do not use VLANs to consolidate Ethernet segments onto one wire. At this security level, we need to avoid even the slightest risk. As an alternative – or an additional precaution – configure IPsec for all network connections to this server.

- Configure RBAC to require the two-person rule for all ldm commands in the control domain. How to do this is described in the Blueprint "Enforcing the Two-Person Rule Via Role-Based Access Control in the Solaris 10 Operating System" [6].

- Regularly check the integrity of OS binaries and kernel modules. This can be done using the elfsign utility and by using BART to compare against a verified baseline. Check the Blueprint "The Solaris Fingerprint Database - A Security Validation Tool for Solaris Environment System Files"[8] for methods on how to automate this. Although the Fingerprint Database is no longer available, the methods mentioned there can be applied to elfsign in a similar manner. For Solaris 11, this is well supported by the packaging system[11].

[10] http://docs.oracle.com/cd/E23824_01/html/819-3195/index.html

[11] https://blogs.oracle.com/cmt/entry/solaris_fingerprint_database_how_it

- Configure BART to quickly detect any modification of any OS binary or configuration file. The Blueprint "Integrating BART and the Solaris Fingerprint Database in the Solaris 10 Operating System" [5] describes this in detail.
- Configure Auditing for the LDoms Manager. This allows you to reconstruct in detail any changes made to the configuration. This is described in the Oracle VM Server for SPARC Admin Guide [4] on page 53.

To meet high security requirements, attention to every detail is needed. Auditing of system activity, and the regular checking of the audit logs, will reveal unusual events. Validating software before installation protects against trojans of all sorts. IPsec allows for tight control of system communication – only authorized systems will be able to send and receive network traffic.

Avoiding the use of VLAN tagging is a measure that protects system isolation. As such, it could also have been part of a more basic set of recommendations. However, the probability of a VLAN breach is considered low, and the recommendation to use dedicated vSwitches per network segment is part of these recommendations.

Scenario with Hardware Partitioning

In this section, we will discuss the differences to the more common consolidation scenario that arise because of the different deployment model. As already described in "Deployment Models" on page 5, the overall role of the Execution Environment is very different when Oracle VM Server for SPARC is used to create only a small number of guests, each of which exclusively own their own part of the hardware. In this case, the domains are fully independent of each other and the Execution Environment is reduced to the control domain which usually also hosts the virtual console service, the only real service running to sustain the overall system. These root domains correspond more to domains as we know them from Dynamic System Domains in the M-Series Servers. The only difference is that while the M-Series uses crossbar reconfiguration to isolate the domains, LDoms use the embedded hypervisor to achieve this. Because of this difference, many of the threats discussed in this paper do not apply to these root domains. Of course, normal Solaris security principles still apply, as with any other Solaris system.

Special consideration should be given to the role of the control domain in this case. Unlike in the consolidation scenario, it usually doesn't serve as a service domain for other guests, except for the console service. It doesn't share IO resources with or for other domains and could even be shut down once the configuration of the system has been completed. However, due to technical requirements, the control domain must own at least one PCIe root complex and a minimum of CPU and memory. On a SPARC T4-4 server, this would be $1/4^{th}$ of the available IO resources. It is therefore often desirable to use the control domain for production applications. On first sight, this is in conflict with the principle of role separation. However, there are several well documented ways to protect the role of the control domain – essentially write access to the hypervisor through the ldm command or its kernel modules using RBAC and general Solaris security mechanisms from unauthorized access. Further protection for the control domain can be achieved by isolating production applications in

Solaris Zones. It is therefore conceivable to host production applications in the control domain in a Hardware Partitioning Scenario, in a

- Low Security Scenario, if
  - Access to the ldm command is protected using RBAC
  - Access to the root account is protected using RBAC – "root as a role".
- Medium Security Scenario, if
  - In addition, applications are deployed in Solaris Zones only.

For the highest security requirements, the principle of role separation will need to be upheld. In these scenarios, the cost of a completely separate control domain (1 PCIe root complex, 1-2 strands of CPU and about 2 GB of RAM) are easily outweighed by the strict security requirements of such an environment.

Summary

The above recommendations showed you how to apply the counter measures described in "Attack & Defense" on page 8 to secure your servers, depending on their individual security requirements. Not all of the counter measures mentioned there were actually applied – some, like encryption of virtual disks are hard to implement using current software, others, like operational guidelines, should be a given in any larger datacenter. However, by seeing how to apply these measures to three theoretical scenarios, it should be easy to create your own set of security guidelines, allowing for your specific security needs. This was the intention of this paper.

Conclusion

Studies and papers about security typically leave the impression on the reader that the subject of the study – Oracle VM Server for SPARC in our case – is completely insecure and can only be operated securely after numerous configuration changes and security fixes have been applied to it. I'm sure, if you would read a paper about the security features of your house, how these can be put to their best use and what you could do to further enhance them, you would eventually have a similar impression. Nevertheless, our houses are usually quite safe to live in. They are significantly more secure than the tent you might have had in mind after finishing the said paper. And not all of us need or want to live in high security environments like Fort Knox or Alcatraz.

Just like your house provides a high level of security "by default", a typical Oracle VM Server for SPARC installation based on best practices is already well secured against unauthorized use. In many cases this level of security turns out to be sufficient. Nevertheless, there is an attack surface that remains. There are risks, how unlikely they might be. This paper tried to list and discuss these risks. Once we know about the risks, we can come up with means to address them, either by reducing the chance of them actually happening, or by minimizing the potential damage. Real danger lies in those risks we fail to see, not in those we know about and are prepared for.

In the chapter "Attack & Defense" on page 8, we discussed a long list of possible counter measures. Not all of them are required to deploy LDoms securely. Look at them as a toolbox available to you to pick and choose those tools that best suit your specific requirements. How this could be done was shown the chapter "Recommended Deployment Options" on page 30. For more generic deployments, you might be able to implement the recommendations given there. However, the intention of this chapter was also to show how the individual counter measures can be combined to create deployment guidelines based on your own risk assessment.

Finally, I would like to reiterate that security is more than a number of configuration options. This paper only covered those directly related to Oracle VM Server for SPARC and virtualization. For a complete picture, you need to consider how to secure your operating system and your application. Here, using Solaris Zones is highly recommended. Furthermore, have a look at the broader environment. Secure your network components. Implement Identity Management. Consider using intrusion detection products. Prepare against attacks on your employee's loyalty with techniques of social engineering. Have battle plans at hand in case something does go wrong – don't leave damage control to intuition and chance.

Appendix

Bibliography

[1] : Glenn Brunette, SPARC SuperCluster Security Principles and Capabilities, 2012,http://www.oracle.com/technetwork/articles/servers-storage-admin/supercluster-security-1723872.html
[2] : Tony Shoumack, Beginner's Guide to LDoms, 2007,http://www.oracle.com/technetwork/articles/systems-hardware-architecture/beginners-vm-server-sparc-256946.pdf
[3] : Oracle Corporation, Oracle VM Server for SPARC - Enabling a Flexible, Efficient IT Infrastructure, ,http://www.oracle.com/us/oraclevm-sparc-wp-073442.pdf
[4] : Oracle Corporation, Oracle VM for SPARC Administration Guide 2.2, 2012,http://docs.oracle.com/cd/E35434_01/pdf/E23807.pdf
[5] : Glenn Brunette, Integrating BART and the Solaris Fingerprint Database in the Solaris 10 Operating System, 2005,(no longer available) http://www.sun.com/blueprints/0405/819-2260.pdf
[6] : Glenn Brunette, Enforcing the Two-Person Rule Via Role-Based Access Control in the Solaris 10 Operating System, 2005,http://www.oracle.com/technetwork/articles/systems-hardware-architecture/two-person-rule-solaris-277014.pdf
7: Sun Microsystems Security Engineers, Solaris 10 Security Essentials, 2009
[8] : Vasanthan Dasan, Alex Noordergraaf, Lou Ordorica, and Glenn Brunette, The Solaris Fingerprint Database - A Security Validation Tool for Solaris Environment System Files, 2006,http://www.oracle.com/technetwork/articles/systems-hardware-architecture/solaris-fingerprint-db-277032.pdf

List of Figures

List of Threats

List of Counter Measures

Secure Deployment of Oracle VM Server for SPARC
Second Edition, September 2012

Author: Stefan Hinker

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together