System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
Ptrace()	The ptrace() system call provides a means by which one process (the "tracer") may observe and control the execution of another process (the "tracee"), and examine and change the tracee's memory and registers. It is primarily used to implement breakpoint debugging and system call tracing.	Linux kernel built with the User Namespaces (CONFIG_USER_NS) support is vulnerable to a potential privilege escalation flaw. It could occur when a root owned process tries to enter a user namespace, wherein a user attempts to attach the entering process via ptrace(1). A privileged name space user could use this flaw to potentially escalate their privileges on the system.	A local privileged namespace user can obtain elevated privileges outside of the original namespace on the target system	Linux kernel versions 4.4.1 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 4.4.1.	https://knowledge.windriver.com/en-us/000_Products/000/010/000/040/020/000_SCP7-247_%3A_Security_Advisory_-_linux_-_CVE-2015-8709	
shmat()	The shmat(2) function maps a shared memory segment, previously created with the shmget(2) function, into the address space of the calling process. The shmat(2) function first increases the reference count on the underlying vm_object and then attempts to insert the vm_object into the process address space.	The vulnerability occurs because the shmat(2) function forgets to decrease the reference count if the vm_map_find function returns failure. an unprivileged user may have write access to an unreferenced piece of kernel memory, which could later become part of a privileged object.	Serious. Local users can elevate thier priviliges	FreeBSD <= 8.2.0	• Administrators are advised to apply the necessary patches by upgrading to versions > 8.2. • If shared memory is not being used, the kernel can be recompiled so as not to support SYSVSHM. • Administrators can stop all use of shared memory segments on a system by running the following commands as root: # sysctl -w kern.ipc.shmmax=0 # echo 'kern.ipc.shmmax=0' >> /etc/sysctl.conf # ipcs | awk '/^m/ { print $2 }' | xargs -n 1 ipcrm -m	•http://www.linuxsecurity.com/content/view/105767/170/ •http://www.securitytracker.com/id/1008951 •https://tools.cisco.com/security/center/viewAlert.x?alertId=7237	
vmsplice()	With pipes on linux, one could use the splice functionality to get zero-copy moves of the data from one process to the other. For example, the sending process uses vmsplice() with SPICE_F_GIFT to gift the data into a pipe, then the receiving processes uses splice() with SPLICE_F_MOVE to move the data straight from the pipe into the disk file without touching it.	Changes to the vmsplice() code had caused the omission of a couple of important permissions checks. In particular, if the application had requested that vmsplice() move the contents of a pipe into a range of memory, the kernel didn't check whether that application had the right to write to that memory. So the exploit could simply write a code snippet of its choice into a pipe, then ask the kernel to copy it into a piece of kernel memory. Think of it as a quick-and-easy rootkit installation mechanism.	Taking advance of Linux vulnerabilities can allow local privilege escalation. This means you login as a normal unprivileged user, but you run some program, and you end up as a root user.	Linux kernel versions3.18.24.1 and before	1. Administrators are advised to apply the necessary patches by upgrading to versions > 3.18.24.1.	•http://stackoverflow.com/questions/1350361/what-is-the-most-efficient-way-to-exchange-high-volume-data-between-2-process/1350550#1350550 •http://lwn.net/Articles/268783/	attack explained: http://colesec.inve
splice()	With pipes on linux, one could use the splice functionality to get zero-copy moves of the data from one process to the other. For example, the sending process uses vmsplice() with SPICE_F_GIFT to gift the data into a pipe, then the receiving processes uses splice() with SPLICE_F_MOVE to move the data straight from the pipe into the disk file without touching it.	splicing a memory range into a pipe, the kernel must, first, read in one or more iovec structures describing that memory range. The 2.6.23 vmsplice() changes omitted a check on whether the purported iovec structures were in readable memory. This looks more like an information disclosure vulnerability than anything else - though, as we will see, it can be hard to tell sometimes.	Taking advance of Linux vulnerabilities can allow local privilege escalation. This means you login as a normal unprivileged user, but you run some program, and you end up as a root user.	Linux kernel versions 2.6.24.1 and before	1. Administrators are advised to apply the necessary patches by upgrading to versions > 2.6.24.1.	http://lwn.net/Articles/268783/	attack explained: http://colesec.inve

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm

perf_even_open()	A call to perf_event_open() creates a file descriptor that allows measuring performance information. Each file descriptor corresponds to one event that is measured; these can be grouped together to measure multiple events simultaneously.	On systems compiled with PERF_EVENTS support, a local user can supply a specially crafted perf_event_open() call to execute arbitrary code on the target system with root privileges. The vulnerability resides in the perf_swevent_init() function in 'kernel/events/core.c'.	A local user can obtain root privileges on the target system.	linux 3.10.9 and before and Red Hat Enterprise Linux 6	Save the script with .stp extension
					%{
					#include <linux/perf_event.h>
					%}
					function sanitize_config:long (event:lo
					struct perf_event *event;
					#if STAP_COMPAT_VERSION >= ST
					event = (struct perf_event *) (uns
					#else
					event = (struct perf_event *) (uns
					#endif
					event->attr.config &= INT_MAX;
					%}
					
					probe kernel.function("perf_swevent_i
					sanitize_config($event);
					}
					• Install the systemtap package and ar
					• Run the stap -g [filename-from-step-
						https://access.redhat.com/solutions/37
						http://www.securitytracker.com/id/1028	

Mremap()	A call to mremap( ) expands or shrinks mapping in the region[addr, addr+old_size) to the new size new_size. The kernel can potentially move the mapping at the same time, depending on the availability of space in the process' address space and the value of flags. On success, mremap( ) returns a pointer to the newly resized memory mapping.	It was discovered and reported that there is a missing return value check within the mremap(2) system call. When resizing or moving virtual memory areas, the function reportedly does not test the return value of the do_munmap() function. Cases where the function fails (for example, due to the number of virtual memory areas being exceeded by the calling process) will not be properly detected, according to the report. As a result, the kernel may move memory belonging to one process into memory space that is allocated to another process.	Execution of arbitrary code via local system.	Linux version 2.6.2 and less	explains the attack in detail and its patch. https://www.giac.org/paper/gcih/574/linux-kernel-mremap-vulnerability/106035	http://www.securitytracker.com/id/1009130	
Munmap()	The munmap() system call deletes the mappings for the specified address range, and causes further references to addresses within the range to generate invalid memory references. The region is also automatically unmapped when the process is terminated. On the other hand, closing the file descriptor does not unmap the region.	A vulnerability was reported in PaX. A local user can obtain root privileges on the target system. A local user can exploit a flaw in expand_stack() to obtain root privileges on the target system.	Root access via local system	Linux 2.4.18	Fix not available	•http://securitytracker.com/id?1017509 •http://www.securityfocus.com/bid/22014/info	Block
Do_brk()	The do_brk() is an internal kernel function that is called indirectly to manage process's memory heap (brk), growing or shrinking it accordingly.	The function lacks of bound checks of its parameters and may be exploited to create arbitrary large virtual memory area, exceeding user accessible memory limit. Thus, the kernel memory above this limit may become part of user process's memory as visible to the kernel memory manager.	Successful exploitation of do_brk() leads to full compromise of vulnerable system, including gaining full uid 0 privileges, possibility of kernel code and data structures modification as well as kernel-level (ring0) code execution.	Linux version 2.6 and below	Limiting maximum size of user process's data segment with ulimit –d command provides some workaround for exploit on brk system call.	http://isec.pl/vulnerabilities/isec-0012-do_brk.txt	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comı
modify_ldt()	modify_ldt() reads or writes the local descriptor table (LDT) for a process. The LDT is an array of segment descriptors that can be referenced by user code. Linux allows processes to configure a per-process (actually per-mm) LDT.	An analyze of the uselib function load_elf_library() from binfmt_elf.c revealed a flaw in the handling of the library's brk segment (VMA). That segment is created with the current->mm->mmap_sem semaphore NOT held while modifying the memory layout of the calling process. This can be used to disturb the memory management and gain elevated privileges. Also the binfmt_aout binary format loader code is affected in the same way.	Unprivileged local users can gain elevated (root) privileges. Since the vulnerability permits privilege 0 ring code execution, users may also break out from virtual machines like UML (user mode Linux).	Linux 2.9 and below	1. Administrators are advised to apply the necessary patches by upgrading to linux versions > 3.	http://www.isec.pl/vulnerabilities/isec-0021-uselib.txt	
		memory leak by not freeing them					
umount() and umount2()	• umount() and umount2() remove the attachment of the (topmost) filesystem mounted on target.Appropriate privilege (Linux: the CAP_SYS_ADMIN capability) is required to unmount filesystems. • Linux 2.1.116 added the umount2() system call, which, like umount(), unmounts a target, but allows additional flags controlling the behavior of the operation	•A flaw was found in the Linux kernel which is related to the user namespace lazily unmounting file systems. The fs_pin struct has two members (m_list and s_list) which are usually initialized on use in the pin_insert_group. However, these members might go unmodified; in this case, the system panics when it attempts to destroy or free them. This flaw could be used to launch a denial-of-service attack. •The fs_pin implementation in the Linux kernel before 4.0.5 does not ensure the internal consistency of a certain list data structure, which allows local users to cause a denial of service (system crash) by leveraging user-namespace root access for an MNT_DETACH umount2 system call, related to fs/fs_pin.c and include/linux/fs_pin.h	local users can cause a denial of service (system crash) attack.	Linux version 4.0.5 and before	Linux version 4.0.5 and before	•https://lwn.net/Alerts/660410/ •https://security-tracker.debian.org/tracker/CVE-2015-4178 •http://man7.org/linux/man-pages/man2/umount.2.html	
		Null pointer deference					
ioctl()	An ioctl , which means "input-output control" is a kind of device-specific system call. There are only a few system calls in Linux (300-400), which are not enough to express all the unique functions devices may have. So a driver can define an ioctlwhich allows a userspace application to send it orders.	A DoS flaw was found for a Linux kernel built for the x86 architecture which had the KVM virtualization support(CONFIG_KVM) enabled. The kernel would be vulnerable to a NULL pointer dereference flaw in Linux kernel's kvm_apic_has_events() function while doing an ioctl. An unprivileged user able to access the "/dev/kvm" device could use this flaw to crash the system kernel.	local users can cause a denial of service (system crash) attack.	Linux version 4.1.3 and before	Linux version 4.1.3 and before	https://lwn.net/Alerts/660410/	
		no proper permission check					
ioctl()	An ioctl , which means "input-output control" is a kind of device-specific system call. There are only a few system calls in Linux (300-400), which are not enough to express all the unique functions devices may have. So a driver can define an ioctlwhich allows a userspace application to send it orders.	The get_bitmap_file function in drivers/md/md.c in the Linux kernel before 4.1.6 does not initialize a certain bitmap data structure, which allows local users to obtain sensitive information from kernel memory via a GET_BITMAP_FILE ioctl call	The get_bitmap_file function in drivers/md/md.c in the Linux kernel before 4.1.6 does not initialize a certain bitmap data structure, which allows local users to obtain sensitive information from kernel memory via a GET_BITMAP_FILE ioctl call	Linux kernel versions 4.1.6 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 4.1.6.	https://lwn.net/Alerts/660410/	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
write()	write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd	It was discovered that an integer overflow error existed in the SCSIgeneric (sg) driver in the Linux kernel. A local attacker with writepermission to a SCSI generic device could use this to cause a denial of service (system crash) or potentially escalate their privileges	local users can cause a denial of service (system crash) attack.	Linux version 4.1 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 4.1.	https://lwn.net/Alerts/660410/	
		memory leak by not freeing them					
add_key()	add_key() asks the kernel to create or update a key of the given type and description, instantiate it with the payload of length plen, and to attach it to the nominated keyring and to return its serial number.	It was found that the Linux kernel's keyring implementation would leak memory when adding a key to a keyring via the add_key() function. A local attacker could use this flaw to exhaust all available memory on the system.	local users can cause a denial of service (system crash) attack.	Linux version 4.1.4 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 4.1.4	https://lwn.net/Alerts/660410/	
recvmsg()	The recvfrom() and recvmsg() calls are used to receive messages from a socket, and may be used to receive data on a socket whether or not it is connection-oriented.	The skb_copy_and_csum_datagram_iovec function in net/core/datagram.c in the Linux kernel did not accept a length argument, which allowed local users to cause a denial of service (memory corruption) or possibly have unspecified other impact via a write system call followed by a recvmsg system call	local users can cause a denial of service (system crash) attack.	Linux version 3.18.22 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 3.18.22	• https://lists.opensuse.org/opensuse-security-announce/2016-08/msg00003.html	
		no proper permission check					
setsockopt()	setsockopt() manipulate options for the socket referred to by the file descriptor sockfd. Options may exist at multiple protocol levels; they are always present at the uppermost socket level.	The netfilter subsystem in the Linux kernel did not validate certain offset fields, which allowed local users to gain privileges or cause a denial of service (heap memory corruption) via an IPT_SO_SET_REPLACE setsockopt call	Taking advance of Linux vulnerabilities can allow local privilege escalation. This means you login as a normal unprivileged user, but you run some program, and you end up as a root user.	netfilter subsystem in the Linux kernel through 4.5.2	This field should be bounds checked prior to writing to a counter value at the supplied offset.	https://access.redhat.com/security/cve/cve-2016-3134	
		buffer overflow					
getaddrinfo()	getaddrinfo() returns one or more addrinfo structures, each of which contains an Internet address that can be specified in a call to bind(2) or connect (2). The getaddrinfo() function combines the functionality provided by the gethostbyname(3) and getservbyname (3) functions into a single interface, but unlike the latter functions,getaddrinfo() is reentrant and allows programs to eliminate IPv4-versus-IPv6 dependencies	a serious defect in the getaddrinfo() library call in glibc exists. This defect allows an attacker to cause buffer overflow to occur, creating the possibility of remote code execution in some circumstances.	Taking advance of Linux vulnerabilities can allow local privilege escalation. This means you login as a normal unprivileged user, but you run some program, and you end up as a root user.	Linux version 4.22 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 4.22	https://www.isc.org/blogs/a-few-words-about-the-glibc-vulnerability-cve-2015-7547/	
		no permission check anyone can use it					
mknod()	The system call mknod() creates a filesystem node (file, device special file, or named pipe) named pathname, with attributes specified by mode and dev.	normally setuid root during installation when built from source. The action of setting a binary on a UNIX system setuid root allows any local user on the system to execute that binary as the root or administrative user.	Gain privilege access	Linux version 4.22 and before	https://blogs.akamai.com/2016/01/delegate-v9913-setuid-binary-vulnerability.html	https://blogs.akamai.com/2016/01/delegate-v9913-setuid-binary-vulnerability.html	
		null pointer deference					

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comı
fcntl()	• The fcntl system call is the access point for several advanced operations on file descriptors. The first argument to fcntl is an open file descriptor, and the second is a value that indicates which operation is to be performed. For some operations, fcntl takes an additional argument. • The fcntl system call allows a program to place a read lock or a write lock on a file, somewhat analogous to the mutex locks	fcntl calls on directories, which allows local users to cause a denial of service by NULL pointer dereference and crashes the system via standard filesystem operations	local users can cause a denial of service (system crash) attack.	Linux version 3.16 and before	Administrators are advised to apply the necessary patches by upgrading to linux versions > 3.16	http://www.informit.com/articles/article.aspx?p=23618&seqNum=4	
sock_dgram()	socket() creates an endpoint for communication and returns a descriptor.The domain argument specifies a communication domain; this selects the protocol family which will be used for communication.	• socket() creates an endpoint for communication and returns a descriptor. The domain argument specifies a communication domain; this selects the protocol family which will be used for communication. • If an operation on a particular socket is unimplemented, they are expected to point the associated function pointer to predefined stubs, for example if the "accept" operation is undefined it would point to sock_no_accept(). However, we have found that this is not always the case and some of these pointers are left uninitialized. •This issue is easily exploitable for local privilege escalation. In order to exploit this, an attacker would create a mapping at address zero containing code to be executed with privileges of the kernel, and then trigger a vulnerable operation using a sequence	Gain Privilege escalation	Linux version 4.2 and below	Recent kernels with mmap_min_addr support may prevent exploitation if the sysctl vm.mmap_min_addr is set above zero. However, administrators should be aware that LSM based mandatory access control systems, such as SELinux, may alter this functionality.	https://lwn.net/Articles/347006/	
mkdirat()	mkdir() attempts to create a directory named pathname. The mkdirat() system call operates in exactly the same way as mkdir(), except for the differences described here.If the pathname given in pathname is relative, then it is interpreted relative to the directory referred to by the file descriptor dirfd (rather than relative to the current working directory of the calling process, as is done by mkdir() for a relative pathname).	NA					
mknodat()	If the pathname given in pathname is relative, then it is interpreted relative to the directory referred to by the file descriptor dirfd (rather than relative to the current working directory of the calling process, as is done by mknod(2) for a relative pathname).	NA					

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comı
mmap() and mmap2()	mmap() creates a new mapping in the virtual address space of the calling process. The starting address for the new mapping is specified in addr. The length argument specifies the length of the mapping.	• Linux Kernel versions 2.2.23 and prior contain a locally exploitable system denial of service (DoS) vulnerability when a malformed call to the memory is requested. •The vulnerability exists in the /proc/pid/mem interface, which enables one application to access another application under certain circumstances. System memory can be accessed by directly mapping pages using the mmap() function. Linux kernels incorrectly validate mmap() requests when accessing memory pages that are non-readable. •The mmap() function on /proc/pid/mem for Linux Kernel versions 2.4 and later is not supported and does not contain the vulnerability. Future stable versions should correct the problem	There is no impact as the vulnerability is in version that does not affect docker.			https://tools.cisco.com/security/center/viewAlert.x?alertId=5193	
Mprotect()	changes protection for the calling process's memory page(s) containing any part of the address range in the interval [addr, addr+len-1]. addr must be aligned to a page boundary. If the calling process tries to access memory in a manner that violates the protection, then the kernel generates a SIGSEGV signal for the process.	The Linux Kernel versions 2.4.32 and prior and 2.6.16.5 and prior contain a vulnerability that could allow an unauthenticated, local attacker to bypass certain security restrictions. The vulnerability is due to improper validation of shared memory permissions in the mprotect()function. This function can grant write permissions to read-only attachments of shared memory regardless of the Interprocess Communications (IPC) permissions already set on the segment.	There is no impact as the vulnerability is in version that does not affect docker.				
mq_timedreceive, mq_timedsend , mq_unlink msgctl , msgrcv , munlock , munlockall , newfstatat, _newselect, pause, pread64, preadv, pselect6, pwrite64, pwritev, read, readlink, readlinkat, readv, recv, recvfrom, remap_file_pages, renameat, renameat2, restart_syscall, rmdir, rt_sigaction, rt_sigpending, rt_sigqueueinfo, rt_sigreturn, rt_sigtimedwait	NA	NA	NA	NA	NA	NA	
Mq_notify()	It allows the calling process to register or unregister for delivery of an asynchronous notification when a new message arrives on the empty message queue referred to by the message queue descriptor mqdes.	Calling the signal() function in a multithreaded program is undefined behavior. This compliant solution uses an object of type atomic_bool to indicate when the child thread should terminate its loop.	System crash	Linux 4.2	This compliant solution uses an object of type atomic_bool to indicate when the child thread should terminate its loop. mqueue.h has mq_notify(), which can deliver a notification either by calling a signal handler or by creating a new thread. This sounds promising: a "main" thread could launch all receiving threads, and then callmq_notify() which will launch a signalling thread when a message arrives.	https://www.securecoding.cert.org/confluence/display/c/CON37-C.+Do+not+call+signal()+in+a+multithreaded+program	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
mq_open()	It creates a new POSIX message queue or opens an existing queue. The queue is identified by name.	Linux kernel 'mq_open()' system call is prone to a local denial-of-service vulnerability	There is no impact as the vulnerability is in version that does not affect docker.			http://www.securityfocus.com/bid/16283/info	
Msgget, msgsnd	The msgget() system call returns the System V message queue identifier associated with the value of the key argument. A new message queue is created if key has the value IPC_PRIVATE or key isn't IPC_PRIVATE, no message queue with the given key key exists,and IPC_CREAT is specified in msgflg	The bug is caused by a reference leak in the keyrings facility. Even though the bug itself can directly cause a memory leak, it has far more serious consequences. After a quick examination of the relevant code flow, we found that the usage field used to store the reference count for the object is of type atomic_t, which under the hood, is basically an int – meaning 32-bit on both 32-bit and 64-bit architectures. While every integer is theoretically possible to overflow, this particular observation makes practical exploitation of this bug as a way to overflow the reference count seem feasible. And it turns out no checks are performed to prevent overflowing the usage field from wrapping around to 0.	Memory leak	Linux 3.8 – 3.17	Linux version higher than 3.17	http://perception-point.io/2016/01/14/analysis-and-exploitation-of-a-linux-kernel-vulnerability-cve-2016-0728/	
msync	msync() flushes changes made to the in-core copy of a file that was mapped into memory using mmap(2) back to the filesystem. Without use of this call, there is no guarantee that changes are written back before munmap(2) is called	Multiple race conditions in the Advanced Union Filesystem (aufs) aufs3-mmap.patch and aufs4-mmap.patch patches for the Linux kernel 3.x and 4.x allow local users to cause a denial of service (use-after-free and BUG) or possibly gain privileges via a (1) madvise or (2) msync system call, related to mm/madvise.c and mm/msync.c.	Causes Denial of Service	Linux version 4.2.1 and before	Any linux version > 4.2.1	https://security-tracker.debian.org/tracker/CVE-2015-7312	
Poll(), ppoll()	poll() performs a similar task to select (2): it waits for one of a set of file descriptors to become ready to perform I/O.	SIG_SETMASK argument was incorrectly replaced by a SIG_BLOCK argument after the poll() system call. Consequently, the SIGCHLD signal could be permanently blocked, which caused signal masks to not return to their original values and defunct processes to be generated. With this update, the original signal masks are restored and defunct processes are no longer generated.	System Crash	Red hat linux 7	Upgrade to higher version	https://www.tenable.com/plugins/index.php?view=single&id=76684	
prlimit64	The Linux-specific prlimit() system call combines and extends the functionality of setrlimit() andgetrlimit(). It can be used to both set and get the resource limits of an arbitrary process.The resource argument has the same meaning as for setrlimit() and getrlimit().	Stack (frame) overflow in getaddrinfo() when called with AF_INET6	There is no impact as the vulnerability is in version that does not affect docker.				
removexattr	removexattr() removes the extended attribute identified by name and associated with the given path in the filesystem.	It was found that a regular user could remove xattr permissions on files by using the chown or write system calls. A local attacker could use this flaw to deny elevated permissions from valid users, services, or applications, potentially resulting in a denial of service	Privilege escalation	Linux version 3.19 – 4.2	Version > 4.2	https://bugzilla.redhat.com/show_bug.cgi?id=1185139	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
rename	rename will rename the specified files by replacing the first occurrence of expression in their name by replacement.	If the file referenced by dest_file exists prior to calling rename(), the behavior is implementation-defined. On POSIX systems, the destination file is removed.	Undefined behavior	Linux version 4.2	Upgrade to higher version of Linux	https://www.securecoding.cert.org/confluence/display/c/FIO10-C.+Take+care+when+using+the+rename()+function	
truncate()	truncate a file to a specified length. Cause the regular file named by the path or referenced by the fd to be truncated to a size of precisely 'length' bytes.	None	N/A	N/A	N/A		
fork()	fork() creates a new process by duplicating the calling process. The new process, referred to as the child, is an exact duplicate of the calling process, referred to as the parent, except for a few a things	The fork implementation in the Linux kernel before 4.5 on s390 platforms mishandles the case of four page-table levels, which allows local users to cause a denial of service (system crash) or possibly have unspecified other impact via a crafted application, related to arch/s390/include/asm/mmu_context.h and arch/s390/include/asm/pgalloc.h.	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of servic	kernel before 4.5 on s390 platforms	Move to higher version or a different architecture	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2143	
		arch/x86/kernel/entry_64.S in the Linux kernel before 3.19.2 does not prevent the TS_COMPAT flag from reaching a user-mode task, which might allow local users to bypass the seccomp or audit protection mechanism via a crafted application that uses the (1) fork or (2) close system call, as demonstrated by an attack against seccomp before 3.16.	This exploit could bypass seccomp	kernels older than 3.19.2	https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=956421fbb74c3a6261903f3836c0740187cf038b	http://www.openwall.com/lists/oss-security/2015/04/02/1 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2830	
		The Linux kernel before 3.15.4 on Intel processors does not properly restrict use of a non-canonical value for the saved RIP address in the case of a system call that does not use IRET, which allows local users to leverage a race condition and gain privileges, or cause a denial of service (double fault), via a crafted application that makes ptrace and fork system calls.	Allow local users to gain privileges	Kernel before 3.15.4	Move to higher kernel version	https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-4699	
execve()	execve() executes the program pointed to by filename. execve() does not return on success, and the text, data, bss, and stack of the calling process are overwritten by that of the program loaded	The (1) execve and (2) fexecve system calls in the FreeBSD kernel 8.4 before p11, 9.1 before p14, 9.2 before p7, and 10.0 before p4 destroys the virtual memory address space and mappings for a process before all threads have terminated, which allows local users to cause a denial of service (triple-fault and system reboot) via a crafted system call, which triggers an invalid page table pointer dereference.	Allows disruption of service	FreeBSD kernel 8.4 before p11, 9.1 before p14, 9.2 before p7, and 10.0 before p4	Move to higher versions	http://www.cvedetails.com/cve/CVE-2014-3880/	
		FreeBSD 5.1 for the Alpha processor allows local users to cause a denial of service (crash) via an execve system call with an unaligned memory address as an argument.	Allows disruption of service	FreeBSD 5.1	Move to higher versions	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2004-0811	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
fstat()	int fstat(int fd, struct stat *buf): stats the file pointed to by the file descriptor fd and fills in buf	The kernel in Sun Solaris 9 allows local users to cause a denial of service (panic) by calling fstat with a first argument of AT_FDCWD.	Allows disruption of service	Sun Solaris 9		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1673	
		The 64-bit versions of Microsoft Visual C++ 8.0 standard library (MSVCR80.DLL) time functions, including (1) localtime, (2) localtime_s, (3) gmtime, (4) gmtime_s, (5) ctime, (6) ctime_s, (7) wctime, (8) wctime_s, and (9) fstat, trigger an assertion error instead of a NULL pointer or EINVAL when processing a time argument later than Jan 1, 3000, which might allow context-dependent attackers to cause a denial of service (application exit) via large time values. NOTE: it could be argued that this is a design limitation of the functions, and the vulnerability lies with any application that does not validate arguments to these functions. However, this behavior is inconsistent with documentation, which does not list assertions as a possible result of an error condition.	Denial of Service of the system	Microsoft Visual C++ 8.0		https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-0842	
		Format string vulnerability in OpenBSD fstat program (and possibly other BSD-based operating systems) allows local users to gain root privileges via the PWD environmental variable.	Local user can gain root privilege	below OpenBSD 2.7	http://ftp.openbsd.org/pub/OpenBSD/patches/2.7/common/028_format_strings.patch	http://www.openbsd.org/errata27.html https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0842	
socketpair	The socketpair() call creates an unnamed pair of connected sockets in the specified domain, of the specified type, and using the optionally specified protocol.	FreeBSD, NetBSD, and OpenBSD allow an attacker to cause a denial of service by creating a large number of socket pairs using the socketpair function, setting a large buffer size via setsockopt, then writing large buffers.	Allows disruption of service			https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2000-0489	
write	write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd.						
dup(),dup2,dup3	The dup() family system call creates a copy of the file descriptor oldfd with some variations						
socket	int socket(int domain, int type, int protocol): creates an endpoint for communication and returns a descriptor. The domain argument specifies a communication domain; this selects the protocol family which will be used for communication						

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comm
truncate()	Truncate a file to a specified length. Cause the regular file named by the path or referenced by the fd to be truncated to a size of precisely 'length' bytes.	None	-	-	-	http://man7.org/linux/man-pages/man2/truncate.2.html	-
tee()	Duplicating pipe content. Duplicates upto 'len' bytes of data from the pipe referred to by the file descriptor fd_in to the pipe referred to by the file descriptor fd_out.	None	-	-	-	http://man7.org/linux/man-pages/man2/tee.2.html	-
socketcall()	Socket system calls. Common kernel entry point for the socket system calls. Call determines which socket function to invoke. args points to a block containing the actual arguments, which are passed through to the appropriate call.	A vulnerability has been found in tcp_read_sock function in net/ipv4/tcp.c in the linux kernel before 2.6.34 which can affect the proper working of socketcall. This function could allow an unauthenticated, local attacker to cause a denial of service (DoS) condition on a targeted system. The vulnerability is due to improper handling of socket buffers by the affected kernel versions. An attacker could exploit this vulnerability by sending crafted system calls to the targeted system. An exploit could allow the attacker to cause a kernel panic, resulting in a DoS condition.	There will be impact as far as availability of resources are concerned due to DoS attacks. Confidentiality and integrity will remain unimpacted.	Before linux kernel 2.6.34	A patch has been developed post linux kernel versions 2.6.34.	https://www.cvedetails.com/cve/CVE-2013-2128/ , http://man7.org/linux/man-pages/man2/socketcall.2.html	In order to ensure backward compatibility of docker, a patch needs to be added in every linux kernel version atleast after 2.6.0
tkill()/tgkill()	Send a signal to a thread. tgkill() sends the signal sig to the thread with the thread ID tid in the thread group tgid. tkill() is an obsolete predecessor to tgkill(). It allows only the target thread ID to be specified, which may result in the wrong thread being signaled if a thread terminates and its thread ID is recycled.	kernel/signal.c in the Linux kernel before 2.6.39 allows local users to spoof the uid and pid of a signal sender via a sigqueueinfo system call. This is in turn might impact the working of tkill.	Integrity and availibility will be partially impacted due to spoofing by malicious user of the signals.	Before Linux kernel 2.6.39	Patch has been issued for later versions of kernel after 2.6.39	http://man7.org/linux/man-pages/man2/tkill.2.html , https://www.cvedetails.com/cve/CVE-2011-1182/	For backward compatibility with earlier version of linux, workaround is needed
unlink()/unlinkat()	Deletes the name and possibly the file it refers to. If that name was the last link to a file and no processes have the file open, the file is deleted and the space it was using is made available for reuse. If the name was the last link to a file but any processes still have the file open, the file will remain in existence until the last file descriptor referring to it is closed.	None	-	-	-	http://man7.org/linux/man-pages/man2/unlink.2.html	
umask()	Set file mode creation mask. Sets the calling process's file mode creation mask (umask) to mask & 0777 (i.e., only the file permission bits of mask are used), and returns the previous value of the mask.	xinetd 2.1.8 and earlier runs with a default umask of 0, which could allow local users to read or modify files that are created by an application that runs under xinetd but does not set its own safe umask.	The default files created becomes world writable due to which modification of data by 3rd party becomes highly probable.	For versions of xinetd before 2.1.8	Patch has been released for later versions of xinetd	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-1322 , http://man7.org/linux/man-pages/man2/umask.2.html	This can be mitigated by hardening measures as well

d	Description:	Vulnerability	Impact Type	Impact	Version	Patch	References:	Additional Comments
fork()	fork() creates a new process by duplicating the calling process. The new process, referred to as the child, is an exact duplicate of the calling process, referred to as the parent, except for a few a things	The fork implementation in the Linux kernel before 4.5 on s390 platforms mishandles the case of four page-table levels, which allows local users to cause a denial of service (system crash) or possibly have unspecified other impact via a crafted application, related to arch/s390/include/asm/mmu_context.h and arch/s390/include/asm/pgalloc.h.	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of servic		kernel before 4.5 on s390 platforms	Move to higher version or a different architecture	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2143	
		arch/x86/kernel/entry_64.S in the Linux kernel before 3.19.2 does not prevent the TS_COMPAT flag from reaching a user-mode task, which might allow local users to bypass the seccomp or audit protection mechanism via a crafted application that uses the (1) fork or (2) close system call, as demonstrated by an attack against seccomp before 3.16.	This exploit could bypass seccomp		kernels older than 3.19.2	https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=956421fbb74c3a6261903f3836c0740187cf038b	http://www.openwall.com/lists/oss-security/2015/04/02/1 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2830	
		The Linux kernel before 3.15.4 on Intel processors does not properly restrict use of a non-canonical value for the saved RIP address in the case of a system call that does not use IRET, which allows local users to leverage a race condition and gain privileges, or cause a denial of service (double fault), via a crafted application that makes ptrace and fork system calls.	Allow local users to gain privileges		Kernel before 3.15.4	Move to higher kernel version	https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-4699	
execve()	execve() executes the program pointed to by filename. execve() does not return on success, and the text, data, bss, and stack of the calling process are overwritten by that of the program loaded	The (1) execve and (2) fexecve system calls in the FreeBSD kernel 8.4 before p11, 9.1 before p14, 9.2 before p7, and 10.0 before p4 destroys the virtual memory address space and mappings for a process before all threads have terminated, which allows local users to cause a denial of service (triple-fault and system reboot) via a crafted system call, which triggers an invalid page table pointer dereference.	Allows disruption of service		FreeBSD kernel 8.4 before p11, 9.1 before p14, 9.2 before p7, and 10.0 before p4	Move to higher versions	http://www.cvedetails.com/cve/CVE-2014-3880/	
		FreeBSD 5.1 for the Alpha processor allows local users to cause a denial of service (crash) via an execve system call with an unaligned memory address as an argument.	Allows disruption of service		FreeBSD 5.1	Move to higher versions	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2004-0811	
fstat()	int fstat(int fd, struct stat *buf): stats the file pointed to by the file descriptor fd and fills in buf	The kernel in Sun Solaris 9 allows local users to cause a denial of service (panic) by calling fstat with a first argument of AT_FDCWD.	Allows disruption of service		Sun Solaris 9		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1673	
		The 64-bit versions of Microsoft Visual C++ 8.0 standard library (MSVCR80.DLL) time functions, including (1) localtime, (2) localtime_s, (3) gmtime, (4) gmtime_s, (5) ctime, (6) ctime_s, (7) wctime, (8) wctime_s, and (9) fstat, trigger an assertion error instead of a NULL pointer or EINVAL when processing a time argument later than Jan 1, 3000, which might allow context-dependent attackers to cause a denial of service (application exit) via large time values. NOTE: it could be argued that this is a design limitation of the functions, and the vulnerability lies with any application that does not validate arguments to these functions. However, this behavior is inconsistent with documentation, which does not list assertions as a possible result of an error condition.	Denial of Service of the system		Microsoft Visual C++ 8.0		https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2007-0842	
		Format string vulnerability in OpenBSD fstat program (and possibly other BSD-based operating systems) allows local users to gain root privileges via the PWD environmental variable.	Local user can gain root privilege		below OpenBSD 2.7	http://ftp.openbsd.org/pub/OpenBSD/patches/2.7/common/028_format_strings.patch	http://www.openbsd.org/errata27.html https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0842	

socketpair	The socketpair() call creates an unnamed pair of connected sockets in the specified domain, of the specified type, and using the optionally specified protocol.	FreeBSD, NetBSD, and OpenBSD allow an attacker to cause a denial of service by creating a large number of socket pairs using the socketpair function, setting a large buffer size via setsockopt, then writing large buffers.	Allows disruption of service				https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2000-0489	
write	write() writes up to count bytes from the buffer pointed buf to the file referred to by the file descriptor fd.	N/A	N/A		N/A	N/A	N/A	
dup(),dup2, dup3	The dup() family system call creates a copy of the file descriptor oldfd with some variations	N/A	N/A		N/A	N/A	N/A	
socket	int socket(int domain, int type, int protocol): creates an endpoint for communication and returns a descriptor. The domain argument specifies a communication domain; this selects the protocol family which will be used for communication	N/A	N/A		N/A	N/A	N/A	
open	Given a pathname for a file, open() returns a file descriptor, a small, nonnegative integer for use in subsequent system calls (read(2), write (2), lseek(2), fcntl(2), etc.). The file descriptor returned by a successful call will be the lowest-numbered file descriptor not currently open for the process.	The open() function in FreeBSD allows local attackers to write to arbitrary files.	Allows unauthorized modification		FreeBSD2.1.0 and 2.2	Use higher versions no explicit patch found	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-1999-0322	
close	close() closes a file descriptor, so that it no longer refers to any file and may be reused	arch/x86/kernel/entry_64.S in the Linux kernel before 3.19.2 does not prevent the TS_COMPAT flag from reaching a user-mode task, which might allow local users to bypass the seccomp or audit protection mechanism via a crafted application that uses the (1) fork or (2) close system call, as demonstrated by an attack against seccomp before 3.16.	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of servic		kernel before 4.5 on s390 platforms	Move to higher version or a different architecture	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2143	
waitpid	The waitpid() system call suspends execution of the calling process until a child specified by pid argument has changed state. By default, waitpid() waits only for terminated children, but this behavior is modifiable via the options argument	N/A	N/A		N/A	N/A		
chmod	chmod changes the file mode bits of each given file according to mod	The capabilities implementation in the Linux kernel before 3.14.8 does not properly consider that namespaces are inapplicable to inodes, which allows local users to bypass intended chmod restrictions by first creating a user namespace, as demonstrated by setting the setgid bit on a file with group ownership of root.	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service		linux kernel before 3.14.8	move to higher versions	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-4014	
kill	kill is used to send a signal to a process	Format string vulnerability in top program allows local attackers to gain root privileges via the "kill" or "renice" function.	Provides administrator access, Allows complete confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service		FreeBSD 3.5.1 below	Move to FreeBSD 3.5.1 or above	http://www.securityfocus.com/bid/1895/solution	
pipe	pipe() creates a pipe, a unidirectional data channel that can be used for interprocess communication. The array pipefd is used to return two file descriptors referring to the ends of the pipe. pipefd[0] refers to the read end of the pipe. pipefd[1] refers to the write end of the pipe. Data written to the write end of the pipe is buffered by the kernel until it is read from the read end of the pipe.	fs/pipe.c in the Linux kernel before 4.5 does not limit the amount of unread data in pipes, which allows local users to cause a denial of service (memory consumption) by creating many pipes with non-default sizes.	Allows disruption of service		Linux kernel before 4.5	https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=759c01142a5d0f364a462346168a56de28a80f52	https://bugzilla.redhat.com/show_bug.cgi?id=1313428	
							https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2016-2847	

		The (1) pipe_read and (2) pipe_write implementations in fs/pipe.c in the Linux kernel before 3.16 do not properly consider the side effects of failed __copy_to_user_inatomic and __copy_from_user_inatomic calls, which allows local users to cause a denial of service (system crash) or possibly gain privileges via a crafted application, aka an "I/O vector array overrun."	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service		Linux kernel before 3.16	http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=f0d1bec9d58d4c038d0ac958c9af82be6eb18045	https://bugzilla.redhat.com/show_bug.cgi?id=1202855	
						http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=637b58c2887e5e57850865839cc75f59184b23d1	http://www.openwall.com/lists/oss-security/2015/06/06/2	
							https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-1805	
ipc		Race condition in the Pipe (IPC) close function in FreeBSD 6.3 and 6.4 allows local users to cause a denial of service (crash) or gain privileges via vectors related to kqueues, which triggers a use after free, leading to a NULL pointer dereference or memory corruption.	Provides administrator access, Allows complete confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service		FreeBSD 6.3 and 6.4	1) Upgrade your vulnerable system to 6-STABLE, or to the RELENG_6_4, or RELENG_6_3 security branch dated after the correction date. 2) To patch your present system: The following patches have been verified to apply to FreeBSD 6.3 and 6.4. a) Download the relevant patch from the location below, and verify the detached PGP signature using your PGP utility. # fetch http://security.FreeBSD.org/patches/SA-09:13/pipe.patch # fetch http://security.FreeBSD.org/patches/SA-09:13/pipe.patch.asc b) Apply the patch. # cd /usr/src # patch < /path/to/patch c) Recompile your kernel as described in <URL:http://www.FreeBSD.org/handbook/kernelconfig.html> and reboot the system.	http://www.securitytracker.com/id?1022982	
							https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2015-1805	
mlock,mlock2, mlockall		N/A	N/A		N/A	N/A		
accept	The accept() system call is used with connection-based socket types (SOCK_STREAM, SOCK_SEQPACKET). It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. The newly created socket is noin the listening state. The original socket sockfd is unaffected by this call.	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6653	Allows disruption of service	Low	NetBSD-current before 20061023, NetBSD 3.0 and 3.0.1 before 20061024, and NetBSD 2.x before 20061029	http://securitytracker.com/id?1017293		
		nvd.nist.gov/nvd.cfm?cvename=CVE-2002-0973	Provides unauthorized access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	Low	FreeBSD 4.6.1 RELEASE-p10 and earlier	http://www.securityfocus.com/bid/5493/solution		
access	access() checks whether the calling process can access the file pathname. If pathname is a symbolic link, it is dereferenced	N/A	N/A	N/A	N/A	N/A		
accept4	may have the issues of accept function							
alarm	alarm() arranges for a SIGALRM signal to be delivered to the calling process in seconds seconds.	N/A						
brk	brk() sets the end of the data segment to the value specified by addr, when that value is reasonable, the system has enough memory, and the process does not exceed its maximum data size	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2003-0961	Provides administrator access, Allows complete confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	Low	Linux kernel 2.4.22 and earlier			

bind	When a socket is created with socket(2), it exists in a name space (address family) but has no address assigned to it. bind() assigns the address specified by addr to the socket referred to by the file descriptor sockfd. addrlen specifies the size, in bytes, of the address structure pointed to by addr. Traditionally, this operation is called “assigning a name to a socket”.	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-8956	Allows unauthorized disclosure of information; Allows disruption of service	Low	Linux kernel before 4.2			
		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-9644	Allows unauthorized modification	2.9	Linux kernel before 3.18.5			
		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-2678	Allows disruption of service	6.9	Linux kernel through 3.14			
		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-4062	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service	10	FreeBSD 7.3	https://www.freebsd.org/security/advisories/FreeBSD-SA-11:05.unix.asc		
		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3118	Allows disruption of service	2.9	not a kernel issue, TOCTOU vuln	https://bugs.debian.org/cgi-bin/bugreport.cgi?att=1;bug=375617;filename=spread-3.17.2_tmpfile.patch;msg=12		
getgid32	getegid() returns the effective group ID of the calling process. Subsequently, Linux 2.4 added getgid32() and getegid32(), supporting 32-bit IDs	N/A						
getgroups, getgroups32	getgroups() returns the supplementary group IDs of the calling process in list. The argument size should be set to the maximum number of items that can be stored in the buffer pointed to by list. If the calling process is a member of more than size supplementary groups, then an error results. It is unspecified whether the effective group ID of the calling process is included in the returned list. (Thus, an application should also call getegid(2) and add or remove the resulting value.)	N/A						
getitimer	These system calls provide access to interval timers, that is, timers that initially expire at some point in the future, and (optionally) at regular intervals after that. When a timer expires, a signal is generated for the calling process, and the timer is reset to the specified interval (if the interval is nonzero).	https://tools.cisco.com/security/center/viewAlert.x?alertId=4766	Denial of service		OpenBSD versions 3.0 and 3.1	ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.0/common/032_kerntime.patch		
getpeername	getpeername() returns the address of the peer connected to the socket sockfd, in the buffer pointed to by addr. The addrlen argument should be initialized to indicate the amount of space pointed to by addr. On return it contains the actual size of the name returned (in bytes). The name is truncated if the buffer provided is too small.	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0973	Provides unauthorized access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	6.4	FreeBSD 4.6.1 RELEASE-p10 and earlier	http://www.securityfocus.com/bid/5493/solution	not good to be used with python and perl	
		https://www.cvedetails.com/cve/CVE-2010-3493/	Denial of service		Python before 3.2			
		https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-5011						
		https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-0761			Perl 5.10	http://www.securityfocus.com/bid/47766/solution		
getpgid	getpgid() returns the PGID of the process specified by pid. If pid is zero, the process ID of the calling process is used.	N/A						
getpriority	The getpriority() function shall obtain the nice value of a process, process group, or user	N/A						
getrandom()	The getrandom() system call fills the buffer pointed to by buf with up to buflen random bytes. These bytes can be used to seed user-space random number generators or for cryptographic purposes.	N/A						

getresgid, getresguid32	getresgid() returns the real UID, the effective UID, and the saved set-user-ID of the calling process group, in the arguments ruid, euid, and suid, respectively.Subsequently, Linux 2.4 added getresuid32() and getresgid32(), supporting 32-bit IDs	N/a						
getresuid, getresuid32	getresuid() returns the real UID, the effective UID, and the saved set-user-ID of the calling process, in the arguments ruid, euid, and suid, respectively.Subsequently, Linux 2.4 added getresuid32() and getresgid32(), supporting 32-bit IDs	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3215	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service	10	policycoreutils 2.2.5	http://www.securityfocus.com/bid/67341/solution		
getrlimit	The getrlimit() system call gets resource limits	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-2188	Allows disruption of service	6.9	OpenBSD before 3.2	http://www.securityfocus.com/bid/6124/solution		
get_robust_list	The get_robust_list() system call returns the head of the robust futex list of the thread whose thread ID is specified in pid. If pid is 0, the head of the list for the calling thread is returned. The list head is stored in the location pointed to by head_ptr. The size of the object pointed to by **head_ptr is stored in len_ptr.	N/A						
getrusage	getrusage() returns resource usage measures	N/A						
getsid	getsid() returns the session ID of the process with process ID pid. If pid is 0, getsid() returns the session ID of the calling process.	N/A						
getsockname	getsockname() returns the current address to which the socket sockfd is bound, in the buffer pointed to by addr. The addrlen argument should be initialized to indicate the amount of space (in bytes) pointed to by addr. On return it contains the actual size of the socket address.	nvd.nist.gov/nvd.cfm?cvename=CVE-2002-0973	Provides unauthorized access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	Low	FreeBSD 4.6.1 RELEASE-p10 and earlier	http://www.securityfocus.com/bid/5493/solution		
getsockopt	get and set options on sockets	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-8612	Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service	6.4	FreeBSD 10.1 before p5, 10.0 before p17, 9.3 before p9, and 8.4 before p23	http://www.securityfocus.com/archive/1/archive/1/534563/100/0/threaded		
get_thread_area	get_thread_area() reads the GDT entry indicated by u_info->entry_number and fills in the rest of the fields in u_info.	N/A						
gettid	gettid() returns the caller's thread ID (TID). In a single-threaded process, the thread ID is equal to the process ID (PID, as returned by getpid(2)). In a multithreaded process, all threads have the same PID, but each one has a unique TID.	N/A						
gettimeofday	get time	N/A						
getuid,getuid32	get user identity. Subsequently, Linux 2.4 added getuid32() and geteuid32(), supporting 32-bit IDs.	N/A						
getxattr	getxattr() retrieves the value of the extended attribute identified by name and associated with the given path in the filesystem. The attribute value is placed in the buffer pointed to by value; size specifies the size of that buffer. The return value of the call is the number of bytes placed in value.	N/A						

inotify_add_watch	inotify_add_watch() adds a new watch, or modifies an existing watch, for the file whose location is specified in pathname; the caller must have read permission for this file. The fd argument is a file descriptor referring to the inotify instance whose watch list is to be modified. The events to be monitored for pathnameare specified in the mask bit-mask argument	N/A						
inotify_init, inotify_init1	initialize an inotify instance	N/A						
inotify_rm_watch	remove an existing watch from an inotify instance	N/A						
io_cancel	cancel an outstanding asynchronous I/O operation	N/A						
io_destroy	The io_destroy() system call will attempt to cancel all outstanding asynchronous I/O operations against ctx_id, will block on the completion of all operations that could not be canceled, and will destroy the ctx_id.	N/A						
io_getevents	read asynchronous I/O events from the completion queue	https://patchwork.kernel.org/patch/4412421/			Linux kernel 3.10	patched in higher versions		
ioprio_get	The ioprio_get() system call gets the I/O scheduling class and priority of one or more threads.	N/A						
ioprio_set	The ioprio_set() system call sets the I/O scheduling class and priority of one or more threads.	N/A						
io_setup	create an asynchronous I/O context	N/A						
Io_submit	submit asynchronous I/O blocks for processing	http://vulisehlt.blogspot.com/2016/03/linux-iosubmit-l2tp-sendmsg-integer.html			Linux kernel 3.10 or 3.18	patched in higher versions like 4.4		
lgetxattr	retrieve an extended attribute value	N/A						
link	make a new name for a file	N/A						
linkat	create a file link relative to directory file descriptors	N/A						
listen	listen for connections on a socket	N/A						
listxattr	listxattr() retrieves the list of extended attribute names associated with the given path in the filesystem.	N/A						
llistxattr	llistxattr() is identical to listxattr(), except in the case of a symbolic link, where the list of names of extended attributes associated with the link itself is retrieved, not the file that it refers to.	N/A						
_llseek	reposition read/write file offset	N/A						
lremovexattr	lremovexattr() removes the extended attribute identified by name and associated with the given path in the filesystem., except in the case of a symbolic link, where the extended attribute is removed from the link itself, not the file that it refers to.	N/A						

lseek	The lseek() function repositions the file offset of the open file description associated with the file descriptor fd to the argument offset according to the directive	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2006-0145	Provides user account access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	6.4	NetBSD 1.6 through 2.1, and OpenBSD 3.8	ftp://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2006-001.tx t.asc		
		https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2003-1234	Allows unauthorized modification; Allows disruption of service	4.9	FreeBSD before 4.2 through 5.0	http://www.securityfocus.com/bid/6524/solution		
lsetxattr	sets the value of the extended attribute identified by name and associated with the given path in the filesyste, except in the case of a symbolic link, where the extended attribute is set on the link itself, not the file that it refers to. The size argument specifies the size (in bytes) of value; a zero-length value is permitted,	N/A						
memfd_create	create an anonymous file	N/A						
mkdir	make directories	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2002-1633	Provides user account access, Allows partial confidentiality, integrity, and availability violation; Allows unauthorized disclosure of information; Allows disruption of service	6.4	QNX 4.25	May not have a solution found in 2002 no soln till 2014		
utimes	change file last access and modification times							
set_thread_area	set a GDT entry for thread-local storage	https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-8133	Allows unauthorized modification	2.9	Linux kernel through 3.18.1	http://www.securityfocus.com/bid/71684/solution		
set_tid_add	set pointer to thread ID	N/A						
shmctl	System V shared memory control	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-7026	Allows disruption of service	6.9	Linux kernel before 3.12.2	move to higher version		
		https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4342	Allows disruption of service	6.9	Red Hat Enterprise Linux 3	https://bugzilla.redhat.com/show_bug.cgi?id=205618		
shmdt	detaches the shared memory segment located at the address specified by shmaddr from the address space of the calling process.	N/A						
setuid32	setuid() sets the effective user ID of the calling process. If the calling process is privileged (more precisely: if the process has the CAP_SETUIDcapability in its user namespace), the real UID and saved set-user-ID are also set. setuid32() supports 32-bit IDs	It is leveraged by a lot many other syscalls						

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
truncate()/ftruncate()	Truncate a file to a specified length. Cause the regular file named by the path or referenced by the fd to be truncated to a size of precisely 'length' bytes.	None	-	-	-	http://man7.org/linux/man-pages/man2/truncate.2.html	-
tee()	Duplicating pipe content. Duplicates upto 'len' bytes of data from the pipe referred to by the file descriptor fd_in to the pipe referred to by the file descriptor fd_out.	None	-	-	-	http://man7.org/linux/man-pages/man2/tee.2.html	-
socketcall()	Socket system calls. Common kernel entry point for the socket system calls. Call determines which socket function to invoke. args points to a block containing the actual arguments, which are passed through to the appropriate call.	A vulnerability has been found in tcp_read_sock function in net/ipv4/tcp.c in the linux kernel before 2.6.34 which can affect the proper working of socketcall. This function could allow an unauthenticated, local attacker to cause a denial of service (DoS) condition on a targeted system. The vulnerability is due to improper handling of socket buffers by the affected kernel versions. An attacker could exploit this vulnerability by sending crafted system calls to the targeted system. An exploit could allow the attacker to cause a kernel panic, resulting in a DoS condition.	There will be impact as far as availability of resources are concerned due to DoS attacks. Confidentiality and integrity will remain unimpacted.	Before linux kernel 2.6.34	A patch has been developed post linux kernel versions 2.6.34.	https://www.cvedetails.com/cve/CVE-2013-2128/ , http://man7.org/linux/man-pages/man2/socketcall.2.html	In order to ensure backward compatibility of docker, a patch needs to be added in every linux kernel version atleast after 2.6.0
tkill()/tgkill()	Send a signal to a thread. tgkill() sends the signal sig to the thread with the thread ID tid in the thread group tgid. tkill() is an obsolete predecessor to tgkill(). It allows only the target thread ID to be specified, which may result in the wrong thread being signaled if a thread terminates and its thread ID is recycled.	kernel/signal.c in the Linux kernel before 2.6.39 allows local users to spoof the uid and pid of a signal sender via a sigqueueinfo system call. This is in turn might impact the working of tkill.	Integrity and availibility will be partially impacted due to spoofing by malicious user of the signals.	Before Linux kernel 2.6.39	Patch has been issued for later versions of kernel after 2.6.39	http://man7.org/linux/man-pages/man2/tkill.2.html , https://www.cvedetails.com/cve/CVE-2011-1182/	For backward compatibility with earlier version of linux, workaround is needed

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
unlink()/unlinkat()	Deletes the name and possibly the file it refers to. If that name was the last link to a file and no processes have the file open, the file is deleted and the space it was using is made available for reuse. If the name was the last link to a file but any processes still have the file open, the file will remain in existence until the last file descriptor referring to it is closed.	None	-	-	-	http://man7.org/linux/man-pages/man2/unlink.2.html	
umask()	Set file mode creation mask. Sets the calling process's file mode creation mask (umask) to mask & 0777 (i.e., only the file permission bits of mask are used), and returns the previous value of the mask.	xinetd 2.1.8 and earlier runs with a default umask of 0, which could allow local users to read or modify files that are created by an application that runs under xinetd but does not set its own safe umask.	The default files created becomes world writable due to which modification of data by 3rd party becomes highly probable.	For versions of xinetd before 2.1.8	Patch has been released for later versions of xinetd	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-1322 , http://man7.org/linux/man-pages/man2/umask.2.html	This can be mitigated by hardening measures as well
ioctl()	Manipulates underlying device parameters of special files. In particular, many operating characteristics of character special files may be controlled with ioctl() requests. The argument d must be an open file descriptor. The 2nd argument is a device dependent request code. Usually on success, 0 is returned.	The btrfs_ioctl_clone function in fs/btrfs/ioctl.c in the btrfs functionality in the Linux kernel 2.6.29 through 2.6.32, and possibly other versions, does not ensure that a cloned file descriptor has been opened for reading, which allows local users to read sensitive information from a write-only file descriptor.	Sensitive information is leaked	Linux kernel 2.6.29 through 2.6.32	Developers and users are advised to install linux kernel beyond 2.6.32	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1636	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
madvise()	give advice about the use of memory. This system call is used to give advice or directions to the kernel about the address range beginning at address addr and with size length bytes. Initially the system call supported a set of "conventional" advice values, which are also available on several other implementations. Subsequently, a number of linux specific advice values have been added.	Multiple race conditions in the madvise_remove function in mm/madvise.c in the Linux kernel before 3.4.5 allow local users to cause a denial of service (use-after-free and system crash) via vectors involving a (1) munmap or (2) close system call.	DoS attacks would ensure that system would be depleted of its resources.	Linux kernel before version 3.4.5	Upgrade to linux kernels with version greater than 3.4.5	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-3511 , http://man7.org/linux/man-pages/man2/madvise.2.html	
syslog()/klogctl()	read and/or clear kernel message ring buffer; set console_loglevel. This system call is used to control the kernel printk() buffer; the glibc wrapper function for the system call is called klogctl().	None	-	-	-	http://man7.org/linux/man-pages/man2/syslog.2.html	
ipc()	System V IPC calls. ipc() is a common kernel point entry point for the System V IPC calls for messages, semaphores and shared memory. call determines which IPC function to invoke; the other arguments are passed through to the appropriate call.	The System V IPC implementation in the kernel in Android before 6.0 2016-01-01 allows attackers to cause a denial of service (global kernel resource consumption) by leveraging improper interaction between IPC resource allocation and the memory manager	DoS attacks will deplete system resources thus reducing it for genuine processes or system calls	All android versions prior to 6.0	Upgrade to Android v6 or later	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6646 , https://linux.die.net/man/2/ipc	
swapoff()	enable/disable devices and files for paging and swapping. Used to specify devices on which paging and swapping are to take place.	None	-	-	-	https://linux.die.net/man/8/swapoff	
	The device or file used is given by the specialfile parameter. It may be of the form -L label or -U uuid to indicate a device by label or uuid						

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
sigreturn()	return from signal handler and cleanup stack frame. If the Linux kernel determines that an unblocked signal is pending for a process, then, at the next transition back to user mode in that process (e.g., upon return from a system call or when the process is rescheduled onto the CPU), it saves various pieces of process context (processor status word, registers, signal mask, and signal stack settings) into the user-space stack	Hitachi Super-H architecture in NetBSD 1.5 and 1.4.1 allows a local user to gain privileges via modified Status Register contents, which are not properly handled by (1) the sigreturn system call or (2) the process_write_regs kernel routine	Sensitive information can be compromised due to unauthorized privileges granted	Version 1.5 and 1.4.1	Use versions later than 1.5	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0734 , http://man7.org/linux/man-pages/man2/sigreturn.2.html	
adjtimex()	tune kernel clock. Linux uses David L. Mill's clock adjustment algorithm. The system call reads and optionally sets adjustment parameters for this algorithm. It takes a pointer to a timex structure, updates kernel parameters from (selected) field values, and returns the same structure updated with the current kernel values.	None	-	-	-	http://man7.org/linux/man-pages/man2/adjtimex.2.html	
sysctl()	modify kernel parameters at run time. Parameters are those listed under /proc/sys.	net/rds/sysctl.c in the Linux kernel before 3.19 uses an incorrect data type in a sysctl table, which allows local users to obtain potentially sensitive information from kernel memory or possibly have unspecified other impact by accessing a sysctl entry	Compromise of sensitive information	Prior to linux kernel 3.19	Use version released after 3.19	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2042 , https://linux.die.net/man/8/sysctl	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
quotactl()	manipulate disk quotas. The quota system can be used to set per-user, per-group and per-project limits on the amount of disk space used on a filesystem. For each user and/or group, a soft limit and a hard limit can be set for each filesystem. The hard limit can't be exceeded. The soft limit can be exceeded, but warnings will ensue. Moreover, the user can't exceed the soft limit for more than grace period duration (one week by default) at a time; after this, the soft limit counts as a hard limit.	vzkernel before 042stab080.2 in the OpenVZ modification for the Linux kernel 2.6.32 does not initialize certain length variables, which allows local users to obtain sensitive information from kernel stack memory via (1) a crafted ploop driver ioctl call, related to the ploop_getdevice_ioc function in drivers/block/ploop/dev.c, or (2) a crafted quotactl system call, related to the compat_quotactl function in fs/quota/quota.c	leakage of sensitive info	vzkernel before 042stab080.2	It has been fixed in version 042stab080.2 and after	http://man7.org/linux/man-pages/man2/quotactl.2.html , http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2013-2239	This impacted only openVZ and as of now nothing has been found on docker but still I felt important to mark this as one of the vulnerable sys calls
sigprocmask()	examine and change blocked signals. Used to fetch and/or change the signal mask of the calling thread. The signal mask is the set of signals whose delivery is currently blocked for the caller	NA	NA	NA	NA	http://man7.org/linux/man-pages/man2/sigprocmask.2.html	
nanosleep()	high-resolution sleep. It suspends the execution of the calling thread until either at least the time specified in *req has elapsed, or the delivery of a signal that triggers the invocation of a handler in the calling thread or that terminates the process	NA	NA	NA	NA	http://man7.org/linux/man-pages/man2/nanosleep.2.html	
prctl()	operations on a process. prctl() is called with a first argument describing what to do (with values defined in <linux/prctl.h>), and further arguments with a significance depending on the first one.	The suid_dumpable support in Linux kernel 2.6.13 up to versions before 2.6.17.4, and 2.6.16 before 2.6.16.24, allows a local user to cause a denial of service (disk consumption) and possibly gain privileges via the PR_SET_DUMPABLE argument of the prctl function and a program that causes a core dump file to be created in a directory for which the user does not have permissions	With DoS attack, significant memory and cpu consumption takes place	Upto linux version 2.6.17	Use versions after 2.6.17	http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2451 , http://man7.org/linux/man-pages/man2/prctl.2.html	PR_SET_DUMPABLE : Set the state of the "dumpable" flag, which determines whether core dumps are produced for the calling process upon delivery of a signal whose default behavior is to produce a core dump.

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
sigsuspend()	wait for a signal. It temporarily replaces the signal mask of the calling process with the mask given by mask and then suspends the process until delivery of a signal whose action is to invoke a signal handler or to terminate a process	NA	NA	NA	NA	http://man7.org/linux/man-pages/man2/rt_sigsuspend.2.html	
mincore()	determine whether pages are resident in memory. mincore() returns a vector that indicates whether pages of the calling process's virtual memory are resident in core (RAM), and so will not cause a disk access (page fault) if referenced. The kernel returns residency information about the pages starting at the address addr, and continuing for length bytes.	The mincore function in the Linux kernel before 2.4.33.6 does not properly lock access to user space, which has unspecified impact and attack vectors, possibly related to a deadlock.	Frequently leads to deadlock conditions	Before linux kernel 2.4.33.6	Use versions after kernel 2.4.33.6	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4814 , http://man7.org/linux/man-pages/man2/mincore.2.html	
setxattr()	set an extended attribute value. Extended attributes are name:value pairs associated with inodes (files, directories, symbolic links, etc.). They are extensions to the normal attributes which are associated with all inodes in the system (i.e., the stat(2) data)	NA	NA	NA	NA	http://man7.org/linux/man-pages/man2/setxattr.2.html	
sendfile()	transfer data between file descriptors. It copies data between one file descriptor and another. Because this copying is done within the kernel, sendfile() is more efficient than the combination of read(2) and write(2), which would require transferring data to and from user space	The sendfile system-call implementation in sys/kern/uipc_syscalls.c in the kernel in FreeBSD 9.2-RC1 and 9.2-RC2 does not properly pad transmissions, which allows local users to obtain sensitive information (kernel memory) via a length greater than the length of the file.	Obtain sensitive information from kernel memory	FreeBSD 9.2-RC1 and 9.2-RC2	Use versions after FreeBSD 9.2	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-5666 , http://man7.org/linux/man-pages/man2/sendfile64.2.html	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
readahead()	initiate file readahead into page cache. Initiates readahead on a file so that subsequent reads from that file will be satisfied from the cache, and not block on disk I/O (assuming the readahead was initiated early enough and that other activity on the system did not in the meantime flush pages from the cache).	NA	NA	NA	NA	http://man7.org/linux/man-pages/man2/readahead.2.html	
chown(), fchown(), lchown(), fchownat()	change ownership of a file, chown() changes the ownership of the file specified by pathname, which is dereferenced if it is a symbolic link, fchown() changes the ownership of the file referred to by the open file descriptor fd, lchown() is like chown(), but does not dereference symbolic links	NA	-	-	-	http://man7.org/linux/man-pages/man2/lchown.2.html	
clock_getres()	clock and time functions, The function clock_getres() finds the resolution (precision) of the specified clock clk_id	NA	-	-	-	http://man7.org/linux/man-pages/man2/clock_gettime.2.html	
clock_gettime()	clock and time functions, The function clock_getres() finds the resolution (precision) of the specified clock clk_id	The div_long_long_rem implementation in include/asm-x86/div64.h in the Linux kernel before 2.6.26 on the x86 platform allows local users to cause a denial of service (Divide Error Fault and panic) via a clock_gettime system call.	Depletion of resources as a result of DoS attacks	Before 2.6.26	Use linux version from 2.6.26 onwards	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3209	
clock_nanosleep()	high-resolution sleep with specifiable clock. It differs in allowing the caller to select the clock against which the sleep interval is to be measured, and in allowing the sleep interval to be specified as either an absolute or a relative value.	The init_posix_timers function in kernel/posix-timers.c in the Linux kernel before 2.6.31-rc6 allows local users to cause a denial of service (OOPS) or possibly gain privileges via a CLOCK_MONOTONIC_RAW clock_nanosleep call that triggers a NULL pointer dereference.	Sensitive info might be lost due to privileges gained	Before 2.6.31-rc6	Use linux version beyond 2.6.31	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-2767	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
connect()	initiate a connection on a socket	The __rds_conn_create function in net/rds/connection.c in the Linux kernel through 4.2.3 allows local users to cause a denial of service (NULL pointer dereference and system crash) or possibly have unspecified other impact by using a socket that was not properly bound.	Resource unavailability due to DoS attacks	Before 4.2.3	Use linux kernel version after 4.2.3	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6937	
copy_file_range()	Copy a range of data from one file to another. The copy_file_range() system call performs an in-kernel copy between two file descriptors without the additional cost of transferring data from the kernel to user space and then back into the kernel.	NA	-	-	-	http://man7.org/linux/man-pages/man2/copy_file_range.2.html	
creat()	open and possibly create a file or device	NA	-	-	-	https://linux.die.net/man/2/creat	
epoll_create()/ epoll_create1()	open an epoll file descriptor	fs/eventpoll.c in the Linux kernel before 2.6.38 places epoll file descriptors within other epoll data structures without properly checking for (1) closed loops or (2) deep chains, which allows local users to cause a denial of service (deadlock or stack memory consumption) via a crafted application that makes epoll_create and epoll_ctl system calls.	Resource unavailability due to DoS attacks	Before 2.6.38	Use linux version beyond 2.6.38	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-1082	
epoll_ctl()	control interface for an epoll file descriptor	Use-after-free vulnerability in net/unix/af_unix.c in the Linux kernel before 4.3.3 allows local users to bypass intended AF_UNIX socket permissions or cause a denial of service (panic) via crafted epoll_ctl calls.	Resource unavailability due to DoS attacks	Before linux kernel 4.3.3	Use linux from 4.3.3 onwards	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-7446	
epoll_wait(), epoll_pwait()	wait for an I/O event on an epoll file descriptor	NA	-	-	-	http://man7.org/linux/man-pages/man2/epoll_wait.2.html	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
eventfd()/eventfd2()	create a file descriptor for event notification	The mem_cgroup_usage_unregister_event function in mm/memcontrol.c in the Linux kernel before 3.2.10 does not properly handle multiple events that are attached to the same eventfd, which allows local users to cause a denial of service (NULL pointer dereference and system crash) or possibly have unspecified other impact by registering memory threshold events.	Resource unavailability due to DoS attacks	Before linux 3.2.10	Use linux version 3.2.10 onwards	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1146	
execveat()	execute program relative to a directory file descriptor	NA	-	-	-	http://man7.org/linux/man-pages/man2/execveat.2.html	
exit()	terminate the calling process						
faccessat()	check user's permissions of a file relative to a directory file descriptor	NA	-	-	-	https://linux.die.net/man/2/faccessat	
posix_fadvise()	predeclare an access pattern for file data. Programs can use posix_fadvise() to announce an intention to access file data in a specific pattern in the future, thus allowing the kernel to perform appropriate optimizations.	NA	-	-	-	https://linux.die.net/man/2/fadvise64	
fallocate()	manipulate file space. This is a nonportable, Linux-specific system call.	The fallocate implementation in the GFS2 filesystem in the Linux kernel before 3.2 relies on the page cache, which might allow local users to cause a denial of service by preallocating blocks in certain situations involving insufficient memory.	DoS attacks resulting in resource unavailability	Before linux version 3.2	Use linux version beyond 3.2	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-4098	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
fanotify_mark()	add, remove, or modify an fanotify mark on a filesystem object	The fill_event_metadata function in fs/notify/fanotify/fanotify_user.c in the Linux kernel through 3.9.4 does not initialize a certain structure member, which allows local users to obtain sensitive information from kernel memory via a read operation on the fanotify descriptor.	Leakage of sensitive information	Linux kernel 3.9.4	Use linux kernel version beyond 3.9.4	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2148	
fchdir()	change working directory	browser/renderer_host/database_dispatcher_host.cc in Google Chrome before 5.0.375.70 on Linux does not properly handle ViewHostMsg_DatabaseOpenFile messages in chroot-based sandboxing, which allows remote attackers to bypass intended sandbox restrictions via vectors involving fchdir and chdir calls	Impacts sandboxing features	Before google chrome 5.0.375.70	Use later versions of google chrome	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2298	
exit_group()	exit all threads in a process	NA	-	-	-	http://man7.org/linux/man-pages/man2/exit_group.2.html	
fchmod()/chmod()	change permissions of a file	Stack-based buffer overflow in the site chmod command in Serv-U FTP Server before 4.2 allows remote attackers to execute arbitrary code via a long filename.	Unauthorized locations can be accessed	Before serv U FTP 4.2	Use Serv U FTP server after version 4.2	cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2111	
fdatasync()/fsync()	synchronize a file's in-core data with that on disk. flushes all data buffers of a file to disk (before the system call returns). It resembles fsync() but is not required to update the metadata such as access time.	Due to a bug in the ext4 code, the fdatasync() system call did not force the inode size change to be written to the disk if it was the only metadata change in the file. This could result in the wrong inode size and possible data loss if the system terminated unexpectedly	Loss of data	Before RHEL 5	Using latest versions of RHEL	https://www.tenable.com/plugins/index.php?view=single&id=77518	Not necessarily a bug in fdatasync, but this sys call has been used to carry out exploitation
fgetxattr()	retrieve an extended attribute value	Buffer overflow in the __nfs4_get_acl_uncached function in fs/nfs/nfs4proc.c in the Linux kernel before 3.7.2 allows local users to cause a denial of service (memory corruption and system crash) or possibly have unspecified other impact via a getxattr system call for the system. nfs4_acl extended attribute of a pathname on an NFSv4 filesystem	Memory corruption	Before linux kernel 3.7.2	Use versions later than 3.7.2	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-4591	

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
flistxattr()	list extended attribute names	Unspecified vulnerability in the listxattr system call in Linux kernel, when a "bad inode" is present, allows local users to cause a denial of service (data corruption) and possibly gain privileges via unknown vectors.	Data corruption			http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5753	
flock()	apply or remove an advisory lock on an open file	dump 0.4 b10 through b29 allows local users to cause a denial of service (execution prevention) by using flock() to lock the /etc/dumpdates file	Unavailability of resources as a result of DoS attacks	dump 0.4	Using later versions of dump	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1914	
fremoveattr()	remove an extended attribute	-	-	-	-	https://linux.die.net/man/2/removexattr	
fstat64()	get file status	Format string vulnerability in OpenBSD fstat program (and possibly other BSD-based operating systems) allows local users to gain root privileges via the PWD environmental variable.	Confidentiality lost			http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0994	
fstatat()	get file status relative to a directory file descriptor	-	-	-	-	https://linux.die.net/man/2/fstatat64	
fstatfs()	get file system statistics	procfs on FreeBSD before 4.5 allows local users to cause a denial of service (kernel panic) by removing a file that the fstatfs function refers to	Unavailability of resources as a result of DoS attacks	FreeBSD 4.5	Use later versions	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1674	
futimesat()	change timestamps of a file relative to a directory file descriptor.	-	-	-	-	http://man7.org/linux/man-pages/man2/futimesat.2.html	
getcpu()	determine CPU and NUMA node on which the calling thread is running	-	-	-	-	http://man7.org/linux/man-pages/man2/getcpu.2.html	
getcwd()	get current working directory	getcwd() file descriptor leak in FTP	Loss of data		Upgrade libc C library to the current version	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0083	Not much information given on CVE regarding this vulnerability

System Call()	Description:	Vulnerability	Impact	Version	Patch	References:	Additional Comments
getdents()	get directory entries	Multiple integer overflows in the next_pidmap function in kernel/pid. c in the Linux kernel before 2.6.38.4 allow local users to cause a denial of service (system crash) via a crafted getdents syscall	Data Corruption	Linux kernel 2.6.38.4	Use later versions	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-1593	
getgid()	get group identity	-	-	-	-	http://man7.org/linux/man-pages/man2/getgid.2.html	
getuid()	get user identity	Stack-based buffer overflow in the GetUID function in src-IL/src/il_dicom.c in DevIL 1.7.8 allows remote attackers to cause a denial of service (application crash) or execute arbitrary code via a crafted DICOM file	Data can be overwritten or leaked	DevIL 1.7.8	Use later versions of DevIL	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3994	
uname()	get name and information about current kernel	Multiple cross-site scripting (XSS) vulnerabilities in the Report Viewer in Ericsson Drutt Mobile Service Delivery Platform (MSDP) 4.x, 5.x, and 6.x allow remote attackers to inject arbitrary web script or HTML via uname	Execution of malicious code	6.x	Use versions beyond 6.0	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-2165	
ugetrlimit()	get resource limits	OpenBSD before 3.2 allows local users to cause a denial of service (kernel crash) via a call to getrlimit (2) with invalid arguments, possibly due to an integer signedness error.	Unavailability of resources as a result of DoS attacks	OpenBSD 3.2	Use versions beyond OpenBSD 3.2	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-2188	
utime()/utimes()	change file last access and modification times	-	-	-	-	http://man7.org/linux/man-pages/man2/utime.2.html	
utimensat()	change file timestamps with nanosecond precision	The utimensat system call (sys_utimensat) in Linux kernel 2.6.22 and other versions before 2.6.25.3 does not check file permissions when certain UTIME_NOW and UTIME_OMIT combinations are used, which allows local users to modify file times of arbitrary files, possibly leading to a denial of service.	Unavailability of resources as a result of DoS attacks	Linux kernel 2.6.25	Use version beyond 2.6.25	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2148	

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
Timer_create	create a POSIX per-process timer	• If a user creates a timer without specifying a sigevent info object, the kernel creates one itself, using a stack local variable. Particularly, it will use the timer ID as sival_int. But as sigev_value is a union containing a pointer and an int, that assignment will only partially initialize sigev_value on systems where the size of a pointer is bigger than the size of an int. On such systems we'll copy the uninitialized stack bytes from the timer_create() call to userland when the timer actually fires and we're going to deliver the signal. • A race condition can occur when processing posix timers in the kernel to cause local DoS. A process can create a POSIX timer using timer_create(2) and set it to send an alarm to the process using timer_settimer(2). If a process is exiting when one of the timers fires, the action can trigger BUG_ON in the kernel, leading to a kernel OOPS error and a DoS condition.	• It can lead to data leakage from the local stack. • An unauthenticated, local attacker can cause a denial of service (DoS) condition.	• Linux kernel versions 3.12 and before. • Linux kernel versions 2.6.16.20 and before.	• Low • Medium	• Administrators are advised to apply the necessary patches by upgrading to linux versions > 3.13 • Administrators are advised to apply the necessary patches by upgrading to linux versions > 2.6.16.21	•https://patchwork.ozlabs.org/patch/407797/ •https://tools.cisco.com/security/center/viewAlert.x?alertId=11156	• This race condition is somewhat difficult to beat. Any local attacker can compile code to create POSIX timers and set them to fire. However, an attacker must interrupt the malicious program while it is exiting to trigger this vulnerability. In addition, there must be a POSIX timer that is ready to fire but has not yet been delivered. Using multiple timers and automated multiple attempts, an attacker could trigger BUG_ON over time; however, this is difficult to achieve on a busy system.
Timer_delete	delete a POSIX per-process timer	None	N/A	N/A	N/A	N/A	N/A	N/A
Timerfd_create	timerfd_create() creates a new timer object, and returns a file descriptor that refers to that timer. They provide an alternative to the use of setitimer(2) or timer_create (2), with the advantage that the file descriptor may be monitored by select(2), poll(2), and epoll(7)	A file descriptor in the Unix/POSIX world has lots of state associated with it. One bit of information determines whether the file descriptor is automatically closed when the process executes an exec call to start executing another program. This is useful, for instance, to establish pipelines. Traditionally, when a file descriptor is created (e.g., with the default open() mode) this close-on-exec flag is not set and a programmer has to explicitly set it using TFD_CLOEXEC flag. Closing the descriptor is a good idea for two main reasons: • The new program's file descriptor table might fill up. For every open file descriptor resources are consumed leading to a DoS, or, • More importantly, information might be leaked to the second program. That program might get access to information it normally wouldn't have access to.	If the file descriptor is not closed, an attacker could potentially read the satus of a timer that he should ideally, have no access to.	• Linux kernel versions 2.6.26 and before.	• Low	• Administrators are advised to apply the necessary patches by upgrading to linux versions > 2.6.26	http://udrepper.livejournal.com/20407.html	
Timerfd_gettime	These system calls create and operate on a timer that delivers timer expiration notifications via a file descriptor.	None	N/A	N/A	N/A	N/A	N/A	N/A
Timer_getoverrun	An application can use the overrun count to accurately calculate the number of timer expirations that would have occurred over a given time interval. Timer overruns can occur both when receiving expiration notifications via signals.	None	N/A	N/A	N/A	N/A	N/A	N/A
Timer_gettime	timer_gettime() returns the time until next expiration, and the interval, for the timer specified by timerid, in the buffer pointed to by curr_value	None	N/A	N/A	N/A	N/A	N/A	N/A
Timer_settime	timer_settime() arms or disarms the timer identified by timerid.	None	N/A	N/A	N/A	N/A	N/A	N/A
Times	Gets process times. times() returns the number of clock ticks that have elapsed since an arbitrary point in the past.	None	N/A	N/A	N/A	N/A	N/A	N/A
Capget	As of Linux 2.2, the power of the superuser (root) has been partitioned into a set of discrete capabilities. Each thread has a set of effective capabilities identifying which capabilities (if any) it may currently exercise. This syscall, lists all such capabilities of the thread(s).	None	N/A	N/A	N/A	N/A	N/A	N/A
Capset	As of Linux 2.2, the power of the superuser (root) has been partitioned into a set of discrete capabilities. Each thread has a set of effective capabilities identifying which capabilities (if any) it may currently exercise. This syscall, sets the capabilities of the thread(s).	None	N/A	N/A	N/A	N/A	N/A	Even though there isn't a vulnerability in this function. It would still be a good idea to block off this call to harden the container. This could be used by another process to elevate capabilities of another process.

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
Chdir	Changes the current working directory of the calling process to the directory specified in path.	Create folders /A, /A/B, /C, /D inside the namespace. Bind-mount the /A inside the namespace to /D. Let a process chdir to /D/B.Move /D/B over into /C. The process which chdir'ed to /D/B is now in /C/B, but at the same time it is in a bind mount with /D as root.It can then traverse upwards, past what looks like / inside the namespace.	• Can lead to a Privilege escalation	• Linux Kernel before 4.2.4.	High	Administrators are advised to apply the necessary patches by upgrading to linux version to 4.2.4 or above.	http://www.openwall.com/lists/oss-security/2015/04/03/7	Vulnerable to TOCTOU attacks. - https://www.usenix.org/legacy/event/fast05/tech/full_papers/wei/wei.pdf
		The autofs module provides support for the automount filesystem, as well as the interface between the kernel and the automountd daemon, which is responsible for the actual mounting. Calls such as chdir() executed in the automount directory are handled by the module, and if the desired directory is defined in the configuration files, automountd then mounts that directory/device. When a chdir() or similar function is called in the autofs directory, by a user doing something along the lines of "cd xxxx", the function fs/autofs/root.c:autofs_root_lookup() is called. The autofs kernel module does not check the size of the directory names it receives. It is passed the name and the names length through dentry->d_name.name and dentry->d_name.len respectively. Later on it memcpy()'s the name into a 256 byte buffer, using dentry->d_name.len as the number of bytes to copy, without checking its size. A nonprivilaged user may attempt to cd to a directory name exceeding 255 characters. This overwrites memory, probably the kernel stack and anything beyond it, and causes kernel errors or makes the machine reboot .	• Boundary condition error. Fails the Kernel.	• Linux Kernel 2.2 and before.	High	Administrators are advised to apply the necessary patches by upgrading to linux versions > 2.2	http://www.securityfocus.com/bid/312/info	CVE-1999-0460 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-2925)
Fchown	These system calls change the owner and group of a file.	Linux kernel could allow a local attacker to mount a remote file system from a vulnerable system and modify files' group IDs, caused by a missing check in the fchown function.	• File manipulation	• Linux Kernel 2.x is vulnerable.	Low	Administrators are advised to apply the necessary patches by upgrading to linux versions > 2.x	https://exchange.xforce.ibmcloud.com/vulnerabilities/16599	CVE-2004-0497(http://www.cvedetails.com/cve/CVE-2004-0497/)
Futex	The futex() system call provides a method for waiting until a certain condition becomes true. It is typically used as a blocking construct in the context of shared-memory synchronization.	The vulnerability is due to improper handling of user-supplied input when processing the futex syscall used by the affected software. A local attacker could exploit this vulnerability by executing a crafted futex_requeue command on a targeted device. A successful exploit could allow the attacker to gain elevated privileges on the targeted device, which could be leveraged to conduct further attacks.	• Privilege escalation	• Linux Kernel 3.14.5 and prior are vulnerable.	Medium	Administrators are advised to apply the necessary patches by upgrading to linux versions > 3.14.5	https://tools.cisco.com/security/center/viewAlert.x?alertId=34570	(CVE-2014-3153) To exploit this vulnerability, an attacker would need access to the local system. This access requirement decreases the likelihood of a successful exploit.
		The vulnerability exists in futex functions that perform get_user() calls while holding mmap_sem for reading. A local attacker can exploit this vulnerability by calling a futex function and forcingget_user() to fail. Exploitation causes a Kernel deadlock and denies service to legitimate users.	• Denial of Service	• Linux Kernel 2.6.11.6 and prior are vulnerable.	Medium	Administrators are advised to apply the necessary patches by upgrading to linux versions > 2.6.11.6	https://tools.cisco.com/security/center/viewAlert.x?alertId=8985	CVE-2005-0937
Getdents64	The system call getdents() reads several linux_dirent structures from the directory referred to by the open file descriptor fd into the buffer pointed to by dirp.	None	N/A	N/A	N/A	N/A	N/A	N/A
Getpid	Returns the process id of the calling process	None	N/A	N/A	N/A	N/A	N/A	N/A
Setgid	Sets the effective group ID of the calling process	Under specific conditions where a setgid system call is made to temporarily elevate privileges and then drop the privileges, the current setgid system calls may fail to completely drop the privileges. When the permissions are not correctly dropped, a local attacker may be able to obtain root privileges by exploiting a weakness in the program with the elevated permissions.	• Privilege escalation	• All Linux kernels are vulnerable	Medium	Does not exist.	https://tools.cisco.com/security/center/viewAlert.x?alertId=4221 http://www.securiteam.com/unixfocus/5YP0N0K7QI.html	This vulnerability has not been proven to be exploitable using any specific programs; however, it does create a situation where a local attacker may be able to gain increased privileges. Linux users who have carefully removed setuid and setgid permissions to harden their systems could still be vulnerable.
Getppid	Returns the process ID of the parent of the current process	None	N/A	N/A	N/A	N/A	N/A	N/A
Lstat	lstat() returns the information about the file pointed to by path and fills in buf, or if path is a symbolic link, then the link itself is stat-ed, not the file that it refers to.	Suffers from TOCTOU (Time-of-check-time-of-use problems)	• Privilege escalation	• All Linux kernels are vulnerable	Medium	Does not exist.	https://cwe.mitre.org/data/definitions/367.html	This vulnerability is not in the syscall itself but it can occur in the usage of it.
Stat	Used to return information about the file pointed to by the pathname.	Suffers from TOCTOU (Time-of-check-time-of-use problems)	• Privilege escalation	• All Linux kernels are vulnerable	Medium	Does not exist.	http://www.cis.syr.edu/~wedu/Teaching/CompSec/LectureNotes_New/Race_Condition.pdf	This vulnerability is not in the syscall itself but it can occur in the usage of it.

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
OpenAt	The openat() system call operates in exactly the same way as open(2), except for the differences described in this manual page. If the pathname given in pathname is relative, then it is interpreted relative to the directory referred to by the file descriptor dirfd (rather than relative to the current working directory of the calling process, as is done by open(2) for a relative pathname). If pathname is relative and dirfd is the special value AT_FDCWD, then pathname is interpreted relative to the current working directory of the calling process (like open(2)). If pathname is absolute, then dirfd is ignored.	None	N/A	N/A	N/A	N/A	N/A	N/A
Prctl	Perform operations on a process	None	N/A	N/A	N/A	N/A	N/A	N/A
Setgroups	Sets the supplementary group IDs for the calling process	The manipulation with an unknown input leads to a privilege escalation vulnerability. The exploitation is known to be easy. Local access is required to approach this attack. No form of authentication is required for exploitation. The technical details are unknown and an exploit is not available.	• Privilege escalation	• All versions of BSD	Medium	Does not exist	https://vuldb.com/?id.80318 https://tools.cisco.com/security/center/viewAlert.x?alertId=43030	An authenticated, local attacker could exploit this vulnerability by initiating the system call on a targeted system, the processing of which could lead to arbitrary results. This could allow an attacker to overwrite certain kernel memory locations. A successful exploit could allow an attacker to bypass security restrictions and gain elevated privileges.
Setuid								
rt_tgsigqueueinfo()	The rt_sigqueueinfo() and rt_tgsigqueueinfo() system calls are the low-level interfaces used to send a signal plus data to a process or thread. The receiver of the signal can obtain the accompanying data by establishing a signal handler with the sigaction(2) SA_SIGINFO flag.	The kill() system call sends a signal to a process. The tkill() system call sends a signal to a thread. The tgkill() system call sends a signal to a thread member of a group. The rt_sigqueueinfo() system call sends a signal to a process, using the detailled siginfo_t structure. The rt_tgsigqueueinfo() call is for threads. The si_code field of the siginfo_t structure indicates the signal sender: - SI_USER : when the user calls kill(), the kernel sends the signal to rt_sigqueueinfo() with SI_USER - SI_USER : when the user calls tkill() or tgkill(), the kernel sends the signal to rt_tgsigqueueinfo() with SI_TKILL - SI_QUEUE : sigqueue(), implemented in the glibc - SI_KERNEL : the signal comes from the kernel, with no identified origin - SI_ASYNCIO, SI_ASYNCNL : implemented in the glibc A user program is only allowed to use SI_QUEUE, SI_ASYNCIO or SI_ASYNCNL. However, rt_sigqueueinfo() and rt_tgsigqueueinfo() only forbid SI_USER (0) and SI_KERNEL(0x80). The case SI_TKILL (-6) was forgotten. A local attacker can therefore use rt_tgsigqueueinfo(), to send a signal with a SI_TKILL sender.	• Unintended use	Linux kernel 2.6.39 and prior are vulnerable.	Low	This vulnerability has been patched in Linux version 2.6.39.	https://vigilance.fr/vulnerability/Linux-kernel-sending-signal-via-rt-tgsigqueueinfo-10487	
sched_getaffinity	get a process's CPU affinity mask	None	N/A	N/A	N/A	N/A	https://linux.die.net/man/2/sched_getaffinity	
sched_setaffinity	get a process's CPU affinity mask	None	N/A	N/A	N/A	N/A	https://linux.die.net/man/2/sched_getaffinity	
Sched_getattr	get scheduling policy and attributes	The sched_read_attr function in kernel/sched/core.c in the Linux kernel 3.14-rc before 3.14-rc4 uses an incorrect size, which allows local users to obtain sensitive information from kernel stack memory via a crafted sched_getattr system call. We're copying the on-stack structure to userspace, but forgot to give the right number of bytes to copy. This allows the calling process to obtain up to PAGE_SIZE bytes from the stack (and possibly adjacent kernel memory).	• Information leakage	Linux kernel 3.14-rc and prior are vulnerable	Medium	This vulnerability has been patched in Linux 3.14-rc	http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=4efbc454ba68def5ef285b26ebfcfdb605b52755 https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-9903	

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
sched_setattr	set scheduling policy and attributes	None	N/A	N/A	N/A	N/A		
sched_getparam	get scheduling parameters	None	N/A	N/A	N/A	N/A		
sched_setparam	set scheduling parameters	None	N/A	N/A	N/A	N/A		
sched_get_priority_max	get static priority range	None	N/A	N/A	N/A	N/A		
sched_get_priority_min	get static priority range	None	N/A	N/A	N/A	N/A		
sched_getscheduler	get scheduling policy/parameters							

sched_setscheduler	set scheduling policy/parameters	Not only admin users, but any users are allowed to set SCHED_FIFO with no restriction. All that's needed is a program like this one:
		
		#include <sched.h>
		int main() {
		struct sched_param param;
		param.sched_priority = sched_get_priority_max (SCHED_FIFO);
		sched_setscheduler(0,SCHED_FIFO,&param);
		while(1);
		}
		
		It will crash the system with no hope of recovering, even if ran by a normal user with no special rights (not even desktop-type rights). I know that it's hard to provide security against a user crashing the machine (e.g. swapping like hell or fork bomb), but this is a bit extreme.
		
		I assume the best way to fix the problem is to
		1) Limit the use of SCHED_FIFO to members of the audio group; and especially
		2) Make sure that all non-root RT tasks are limited to e.g. 70% CPU over a time window and have a lower maximum RT priority (e.g. 40).
		AFAIK, all these are already supported by the kernel and PAM, which is why I'm surprised Dapper got it so wrong.
			• Denial-of-service	linux-source-2.6.15 (Ubuntu)	High	Fixed	http://people.xiph.org/~jm/sched_fifo.pdf	

sched_yield	Causes the calling thread to relinquish the CPU. The thread is moved to the end of the queue for its static priority and a new thread gets to run.	None	N/A	N/A	N/A	N/A		
seccomp	operate on Secure Computing state of the process	Linux Kernel versions 2.6.28.7 and prior contain a vulnerability that could allow a local attacker to bypass security restrictions. The vulnerability is in the seccomp subsystem due to an error when handling syscall filtering. An attacker on an x86_64 system could exploit the vulnerability by issuing a malicious system call. This action could allow the attacker to bypass access restrictions and perform actions with elevated privileges.	• Privlege escalation	Linux Kernel versions 2.6.28.7 and prior are vulnerable.	Medium	Linux 2.6.28.8 or later	https://tools.cisco.com/security/center/viewAlert.x?alertId=17807	
select	Allow a program to monitor multiple file descriptors, waiting until one or more of the file descriptors become "ready" for some class of I/O operation	None	N/A	N/A	N/A	N/A		
semctl	System V semaphore control operations	None	N/A	N/A	N/A	N/A		
semget	get a System V semaphore set identifier	None	N/A	N/A	N/A	N/A		
semop	System V semaphore operations	None	N/A	N/A	N/A	N/A		

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
semtimedop	semaphore operations	Integer overflow in the sys_oabi_semtimedop function in arch/arm/kernel/sys_oabi-compat.c in the Linux kernel before 2.6.39 on the ARM platform, when CONFIG_OABI_COMPAT is enabled, allows local users to gain privileges or cause a denial of service (heap memory corruption) by providing a crafted argument and leveraging a race condition.	• Privilege escalation • Denial of Service	Linux kernel 2.6.37 and prior	Medium	Linux kernel 2.6.38 and later	http://www.securiteam.com/cves/2011/CVE-2011-1759.html	
send	send a message on a socket	None	N/A	N/A	N/A	N/A		
sendfile64	transfer data between file descriptors	None	N/A	N/A	N/A	N/A		
sendmmsg	send multiple messages on a socket							
sendmsg	send a message on a socket	Linux kernel is prone to a local buffer-overflow vulnerability. The vulnerability affects 'sendmsg()' when malformed user-supplied data is copied from userland to kernel memory. A successful attack can allow a local attacker to trigger an overflow, which may lead to a denial-of-service condition due to memory corruption. Arbitrary code execution resulting in privilege escalation is possible as well.	• Denial of Service	Linux kernel 2.6 before 2.6.13.1	Low	Linux kernel 2.7 and later	https://people.canonical.com/~ubuntu-security/cve/2005/CVE-2005-2490.html	
send	send a message on a socket	None	N/A	N/A	N/A	N/A		
sendto	send a message on a socket	None	N/A	N/A	N/A	N/A		
setfsgid	set group identity used for files	None	N/A	N/A	N/A	N/A		
setfsgid32	set group identity used for files	None	N/A	N/A	N/A	N/A		
setfsuid	set user identity used for filesy	None	N/A	N/A	N/A	N/A		
setfsuid	set user identity used for filesy	None	N/A	N/A	N/A	N/A		
setgid32	sets the effective group ID of th	None	N/A	N/A	N/A	N/A		
setgroups	Sets the supplementary group	In current kernels, using setgroups() to change a pr	• Privilege escalation	Linux kernel thr	Low	Linux kernel 3.17.4 or	https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-8989 https://lwn.net/Articles/626665/	
setitimer	Set value of an interval timer	None	N/A	N/A	N/A	N/A		
setpgid	Set process group	None	N/A	N/A	N/A	N/A		
setregid	Set real or effective group id of	None	N/A	N/A	N/A	N/A		
setregid32	Set real or effective group id of	None	N/A	N/A	N/A	N/A		
setresgid	set real, effective and saved gr	None	N/A	N/A	N/A	N/A		
setresgid32	set real, effective and saved gr	None	N/A	N/A	N/A	N/A		
setresuid	set real, effective and saved us	None	N/A	N/A	N/A	N/A		
setresuid32	set real, effective and saved us	None	N/A	N/A	N/A	N/A		
setrlimit	Set resource limits	None	N/A	N/A	N/A	N/A		
set_robust_list	set list of robust futexes							

System Call()	Description:	Vulnerability	Impact	Version	Severity	Patch	References:	Additional Comments
setsid	creates a session and sets the process group ID							