|  |  |  |  |
| --- | --- | --- | --- |
| **《操作系统原理》实验报告** | | | |
| 实验名称 | 沙箱分析恶意软件 | 实验序号 | 8 |
| 实验日期 | 2023/5/23 | 实验人 |  |
| 一、实验题目   * 使用已有动态分析工具（沙箱）对恶意软件的**操作系统资源使用情况**进行分析   注意：   * + 恶意软件不限制平台（Windows、Android、Linux），可从 Virusshare 等平台下载；   + 动态分析工具不作要求，可使用（Windows - Cuckoo Sandbox，Android - Appmon，Linux - Dtrace）；   + 动态分析要求运行环境满足恶意软件执行要求，可能存在缺少运行环境无法执行的情况，选择可以成功运行的恶意软件分析即可；   + 动态分许需要自动化测试工具，触发恶意软件的模拟执行；   + **！！！恶意软件具有传播性、破坏性，可能造成严重危害，请在隔离虚拟机环境中分析以避免对真实系统造成危害！！！** | | | |
| 二、相关原理与知识  1. strace  strace是一个用于跟踪和调试Linux系统中进程执行的实用工具。它可以监视进程执行期间发出的系统调用，并提供有关系统调用的详细信息，如调用类型、参数和返回值。  下面是strace的一些常用特性和用法：  跟踪系统调用: strace可以记录进程发出的系统调用和相关参数，例如文件操作、进程控制、内存管理等。它可以帮助了解程序与操作系统的交互。  输出系统调用信息: strace可以输出系统调用的详细信息，包括调用号、参数和返回值。这对于调试和分析问题非常有用。  跟踪子进程: strace可以跟踪由目标进程创建的子进程，这在调试多进程应用程序时非常有用。  在调用strace时，通常为以下操作：   1. strace -o output.txt ./my\_program   这将启动my\_program并将strace的输出重定向到output.txt文件中。然后，可以分析输出文件以获取有关系统调用的信息。  2. 恶意软件下载平台virusshare  VirusShare是一个公共的恶意软件样本共享平台，旨在帮助安全研究人员、安全工程师和安全爱好者共享、分析和研究恶意软件样本。它提供了一个集中的资源库，包含了大量的恶意软件样本，涵盖了各种类型的恶意软件，如病毒、蠕虫、特洛伊木马、间谍软件等。 | | | |
| 三、实验过程  1. 恶意软件的下载  我通过virusshare网站下载对应的linux系统适用的恶意软件。对于linux系统，其可执行文件为二进制文件，即POSIX shell script，从网站上寻找相应的格式的，size并不大的文件，方便我们使用strace对恶意软件进行跟踪时对结果进行分析。  最后，我选择了如下恶意软件，其基本信息如下：    该软件为适配linux系统的二进制可执行文件，且大小仅为269kb，异常明显（Detections 27/42），非常适合对它的资源使用情况进行分析。  通过对恶意软件的具体分析，可以看到该恶意软件的主要行为，如下图所示：    可以看到，该恶意软件主要操作为读取和修改系统关键设置，并发送消息进而泄露系统重要信息，造成隐私泄露；对关键内容的修改也会造成系统运行的错误。  2. 恶意软件的分析  将恶意软件顺利下载完成以后，我对其在终端上使用strace命令进行运行和跟踪，并将运行的结果进行输出到一个名为output.txt的文档中，方便进行分析。具体实验操作如下：    运行结果为：    四、实验结果  对于strace的结果，其具体内容如下所示：  execve("./virus", ["./virus"], 0x7ffe91f62770 /\* 60 vars \*/) = 0  brk(NULL) = 0x55e7ac42f000  arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd903e1120) = -1 EINVAL (无效的参数)  access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (没有那个文件或目录)  openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3  fstat(3, {st\_mode=S\_IFREG|0644, st\_size=105266, ...}) = 0  mmap(NULL, 105266, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3aebd98000  close(3) = 0  openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libkysec.so.0", O\_RDONLY|O\_CLOEXEC) = 3  read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0&\0\0\0\0\0\0"..., 832) = 832  fstat(3, {st\_mode=S\_IFREG|0644, st\_size=27160, ...}) = 0  mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3aebd96000  mmap(NULL, 30552, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3aebd8e000  mmap(0x7f3aebd90000, 12288, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3aebd90000  mmap(0x7f3aebd93000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x5000) = 0x7f3aebd93000  mmap(0x7f3aebd94000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x5000) = 0x7f3aebd94000  close(3) = 0  openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3  read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832  pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784  pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32  pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0T\354 ]\252R\353S\274q\317Gv\233\276\341"..., 68, 880) = 68  fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029224, ...}) = 0  pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784  pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32  pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0T\354 ]\252R\353S\274q\317Gv\233\276\341"..., 68, 880) = 68  mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3aebb9c000  mprotect(0x7f3aebbc1000, 1847296, PROT\_NONE) = 0  mmap(0x7f3aebbc1000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f3aebbc1000  mmap(0x7f3aebd39000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f3aebd39000  mmap(0x7f3aebd84000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f3aebd84000  mmap(0x7f3aebd8a000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3aebd8a000  close(3) = 0  openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libkysec\_log.so.0", O\_RDONLY|O\_CLOEXEC) = 3  read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\23\0\0\0\0\0\0"..., 832) = 832  fstat(3, {st\_mode=S\_IFREG|0644, st\_size=14792, ...}) = 0  mmap(NULL, 16792, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3aebb97000  mmap(0x7f3aebb98000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f3aebb98000  mmap(0x7f3aebb99000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3aebb99000  mmap(0x7f3aebb9a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3aebb9a000  close(3) = 0  mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3aebb94000  arch\_prctl(ARCH\_SET\_FS, 0x7f3aebb94740) = 0  mprotect(0x7f3aebd84000, 12288, PROT\_READ) = 0  mprotect(0x7f3aebb9a000, 4096, PROT\_READ) = 0  mprotect(0x7f3aebd94000, 4096, PROT\_READ) = 0  mprotect(0x55e7ac0ae000, 8192, PROT\_READ) = 0  mprotect(0x7f3aebddf000, 4096, PROT\_READ) = 0  munmap(0x7f3aebd98000, 105266) = 0  getuid() = 1000  getgid() = 1000  getpid() = 66565  rt\_sigaction(SIGCHLD, {sa\_handler=0x55e7ac0a3ca0, sa\_mask=~[RTMIN RT\_1], sa\_flags=SA\_RESTORER, sa\_restorer=0x7f3aebbe2210}, NULL, 8) = 0  geteuid() = 1000  brk(NULL) = 0x55e7ac42f000  brk(0x55e7ac450000) = 0x55e7ac450000  getppid() = 66562  stat("/home/jasper/\346\241\214\351\235\242", {st\_mode=S\_IFDIR|0755, st\_size=4096, ...}) = 0  stat(".", {st\_mode=S\_IFDIR|0755, st\_size=4096, ...}) = 0  openat(AT\_FDCWD, "./virus", O\_RDONLY) = 3  openat(AT\_FDCWD, "/sys/kernel/security/kysec/status", O\_RDONLY) = -1 ENOENT (没有那个文件或目录)  fcntl(3, F\_DUPFD, 10) = 10  close(3) = 0  fcntl(10, F\_SETFD, FD\_CLOEXEC) = 0  geteuid() = 1000  getegid() = 1000  rt\_sigaction(SIGINT, NULL, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=0}, 8) = 0  rt\_sigaction(SIGINT, {sa\_handler=0x55e7ac0a3ca0, sa\_mask=~[RTMIN RT\_1], sa\_flags=SA\_RESTORER, sa\_restorer=0x7f3aebbe2210}, NULL, 8) = 0  rt\_sigaction(SIGQUIT, NULL, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=0}, 8) = 0  rt\_sigaction(SIGQUIT, {sa\_handler=SIG\_DFL, sa\_mask=~[RTMIN RT\_1], sa\_flags=SA\_RESTORER, sa\_restorer=0x7f3aebbe2210}, NULL, 8) = 0  rt\_sigaction(SIGTERM, NULL, {sa\_handler=SIG\_DFL, sa\_mask=[], sa\_flags=0}, 8) = 0  rt\_sigaction(SIGTERM, {sa\_handler=SIG\_DFL, sa\_mask=~[RTMIN RT\_1], sa\_flags=SA\_RESTORER, sa\_restorer=0x7f3aebbe2210}, NULL, 8) = 0  read(10, "#!/bin/sh\necho '0wned'>>mail.sta"..., 8192) = 269  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  write(1, "0wned\n", 6) = 6  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  write(1, "HOST DET:\n", 10) = 10  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66566  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 66566  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66566, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66566  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  stat("/user/lib/ccache/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/sbin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/bin/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/games/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/games/finger", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  write(2, "./virus: 5: ", 12) = 12  write(2, "finger: not found", 17) = 17  write(2, "\n", 1) = 1  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  stat("/user/lib/ccache/w", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/w", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/w", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/w", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/w", {st\_mode=S\_IFREG|0755, st\_size=22760, ...}) = 0  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66567  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 66567  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66567, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66567  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  stat("/user/lib/ccache/ps", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/ps", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/ps", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/ps", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/ps", {st\_mode=S\_IFREG|0755, st\_size=137688, ...}) = 0  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66568  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 66568  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66568, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66568  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  write(1, ".\n", 2) = 2  wait4(-1, 0x7ffd903e0cec, WNOHANG, NULL) = -1 ECHILD (没有子进程)  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_WRONLY|O\_CREAT|O\_APPEND, 0666) = 3  fcntl(1, F\_DUPFD, 10) = 11  close(1) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 1) = 1  close(3) = 0  write(1, "\n", 1) = 1  dup2(11, 1) = 1  close(11) = 0  openat(AT\_FDCWD, "mail.stat", O\_RDONLY) = 3  fcntl(0, F\_DUPFD, 10) = 11  close(0) = 0  fcntl(11, F\_SETFD, FD\_CLOEXEC) = 0  dup2(3, 0) = 0  close(3) = 0  stat("/user/lib/ccache/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/sbin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/bin/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/games/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/games/mail", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  write(2, "./virus: 10: ", 13) = 13  write(2, "mail: not found", 15) = 15  write(2, "\n", 1) = 1  dup2(11, 0) = 0  close(11) = 0  stat("/user/lib/ccache/rm", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/rm", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/rm", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/rm", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/rm", {st\_mode=S\_IFREG|0755, st\_size=72056, ...}) = 0  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66569  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 66569  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66569, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66569  stat("/user/lib/ccache/mv", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/sbin/mv", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/local/bin/mv", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/sbin/mv", 0x7ffd903e0db0) = -1 ENOENT (没有那个文件或目录)  stat("/usr/bin/mv", {st\_mode=S\_IFREG|0755, st\_size=145792, ...}) = 0  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66570  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 1}], 0, NULL) = 66570  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66570, si\_uid=1000, si\_status=1, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66570  clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3aebb94a10) = 66571  wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 1}], 0, NULL) = 66571  --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=66571, si\_uid=1000, si\_status=1, si\_utime=0, si\_stime=0} ---  rt\_sigreturn({mask=[]}) = 66571  read(10, "", 8192) = 0  exit\_group(1) = ?  +++ exited with 1 +++  针对上述内容，分析如下：  execve（“./virus”，[“./virus]”，0x7ffe91f62770/\*60 vars\*/）：此系统调用执行名为“./vir病毒”的程序，参数为[“./virus”]。  openat（AT\_FDCWD，“/etc/ld.so.preload”，O\_RDONLY）：此系统调用试图打开文件“/etc/ld.O.ppreload”进行读取，但失败并返回错误“ENOENT”（没有这样的文件或目录）。  openat（AT\_FDCWD，“/lib/x86\_64-linux-gnu/libkysec.so.0”，O\_RDONLY）：此系统调用打开文件“/lib/x86 \_64-linux-gnu-libkysec.sec.so.0”进行读取。  openat（AT\_FDCWD，“/lib/x86\_64-linux-gnu/libc.so.6”，O\_RDONLY）：此系统调用打开文件“/lib/x86\_64-linux-gnu/libc.seo.6”进行读取。  openat（AT\_FDCWD，“/lib/x86\_64-linux-gnu/libkysec\_log.so.0”，O\_RDONLY）：此系统调用打开文件“/lib/x86\_64-linux-gnu/libkysec\_log.seo.0”进行读取。  getuid（）：此系统调用检索调用进程的真实用户ID。  getgid（）：此系统调用检索调用进程的实际组ID。  getpid（）：此系统调用检索调用进程的进程ID。  geteuid（）：此系统调用检索调用进程的有效用户ID。  brk（NULL）：此系统调用检索程序中断的当前值，该值表示进程数据段的结束。  stat（“/home/jasper/\346\241\214\351\235\242”，｛st\_mode=S\_IFDIR|0755，st\_size=4096，…｝）：此系统调用检索名为“/home/Jaster/…”的文件或目录的信息（字符采用非ASCII编码）。它检查它是否是一个目录。  stat（“.”，｛st\_mode=S\_IFDIR|0755，st\_size=4096，…｝）：此系统调用检索有关当前目录的信息。它检查它是否是一个目录。  openat（AT\_FDCWD，“./virus”，O\_RDONLY）：此系统调用打开文件“./virus”进行读取。  read（10，“#！/bin/sh\necho‘0wned'>>mail.sta”…，8192）：此系统调用将文件描述符10（以前从文件描述符3复制）的内容读取到缓冲区中。  openat（AT\_FDCWD，“mail.stat”，O\_WRONLY|O\_CREAT|O\_APPEND，0666）：此系统调用打开文件“mail.stat”进行写入，如果不存在则创建该文件，如果存在则附加到该文件。  write（1，“0wned\n”，6）：此系统调用将字符串“0wned \n”写入标准输出（文件描述符1）。  clone（child\_stack=NULL，flags=clone\_child\_CLEAARTID | clone\_child\_SETTID | SIGCHLD，child\_tidptr=0x7f3aebb94a10）：此系统调用通过复制调用进程来创建新的子进程。新进程在一个单独的线程中执行（如CLONE\_CHILD\_CLEARTID和CLONE\_CHILD\_SETTID标志所指定的），并且子线程的ID存储在CHILD\_tidptr所指定的内存位置。  futex（0x7f3aebb94a10，futex\_WAIT\_PRIVATE，2，NULL）：此系统调用将挂起调用线程，直到内存位置0x7f3adebb94a110处的值发生更改。它等待值为2的FUTEX\_WAKE操作。  openat（AT\_FDCWD，“/proc/sys/kernel/unprivileged\_userns\_clone”，O\_RDONLY）：此系统调用打开文件“/proc/sys/kernel/unsprivileged-userns\_clon”进行读取。  openat（AT\_FDCWD，“/proc/sys/kernel/randomize\_va\_saspace”，O\_RDONLY）：此系统调用打开文件“/proc/sys/kernel/randomize\_von\_saspace”进行读取。  uname（｛…｝）：此系统调用检索有关当前系统的信息，并将其存储在结构utsname数据结构中。  openat（AT\_FDCWD，“/etc/localtime”，O\_RDONLY）：此系统调用打开文件“/etc/localtime”进行读取。  gettimeofday（｛…｝）：此系统调用检索当前时间并将其存储在结构timeval数据结构中。  fcntl（1，F\_GETFD）：此系统调用检索文件描述符1（标准输出）的文件描述符标志。  fcntl（1，F\_DUFD\_CLOEXEC，0）：此系统调用复制设置了O\_CLOEXEC标志的文件描述符1（标准输出），并返回一个新的文件描述符。  fcntl（2，F\_GETFD）：此系统调用检索文件描述符2的文件描述符标志（标准错误）。  fcntl（2，F\_DUFD\_CLOEXEC，0）：此系统调用重复设置了O\_CLOEXEC标志的文件描述符2（标准错误），并返回一个新的文件描述符。  openat（AT\_FDCWD，“/home/jasper/.local/share/Trash/expunge.trashinfo”，O\_RDONLY）：此系统调用试图打开文件“/home/jasper/.local/sshare/Trash/exprunge.Trash info”进行读取，但失败，并返回错误“ENOENT”（没有这样的文件或目录）。  openat（AT\_FDCWD，“/usr/lib/loce/loce-archive”，O\_RDONLY|O\_CLOEXEC）：此系统调用打开文件“/usr/lib.loce/loce/archive”以便在设置O\_CLOEXC标志的情况下读取。  ioctl（0，TCGETS，｛B38400 opost isig icanon echo…｝）：此系统调用检索文件描述符0（标准输入）的终端设置，并将其存储在struct-termios数据结构中。  ioctl（1，TCGETS，｛B38400 opost isig icanon echo…｝）：此系统调用检索文件描述符1（标准输出）的终端设置，并将其存储在struct-termios数据结构中。  ioctl（2，TCGETS，｛B38400 opost isig icanon echo…｝）：此系统调用检索文件描述符2的终端设置（标准错误），并将其存储在struct-termios数据结构中。  getrlimit（RLIMIT\_STACK，｛rlim\_cur=8192\*1024，rlim\_max=RLIM64\_INFINITY｝）：此系统调用检索堆栈大小的资源限制，并将其存储在结构RLIMIT数据结构中。  通过上述分析，我们可以看到，这个恶意软件读取了包括用户ID、进程ID等重要信息，以及多次使用openat函数访问系统关键位置，对内容进行更改和读取，尤其是用户的隐私数据部分，对系统隐私安全和运行安全造成了威胁。  对于恶意软件的标签，checks-cpu-name，detect-debug-environment，direct-cpu-clock-access，sets-process-name，这些均无法直接从strace的跟踪中直接找到对应的语句，因此我们无法真正判断它的这些危害，但我们看到了它占用系统资源并读取隐私内容进行发送，危害了系统安全。 | | | |
| 五、问题总结  1. 在初次使用virusshare时，不知道如何使用  解决：向官方发送邮件，说明使用缘由，官方会回复，之后便可以注册账号并正常使用了。  2. 对于virusshare下载exe程序，无法使用  解决：linux系统无法直接运行exe程序，虽然可以通过使用wine来进行运行，但经过测试，使用strace + wine会导致跟踪结果过多而不好分析，因此原则寻找linux二进制可执行文件直接进行strace  3. strace时提示无运行权限  解决：virusshare中下载的病毒文件解压后，默认所有用户无法访问，需要手动更改设置才可以进行访问。 | | | |