clamav on access scan with fanotify

by eqmcc

Description

1. about fanofity

fanotify: the fscking all notification system, is officially merged into linux kernel in version 2.6.36(http://kernelnewbies.org/Linux_2_6_36), it will report interested file/file system changes/actions, full details can be found in following links:

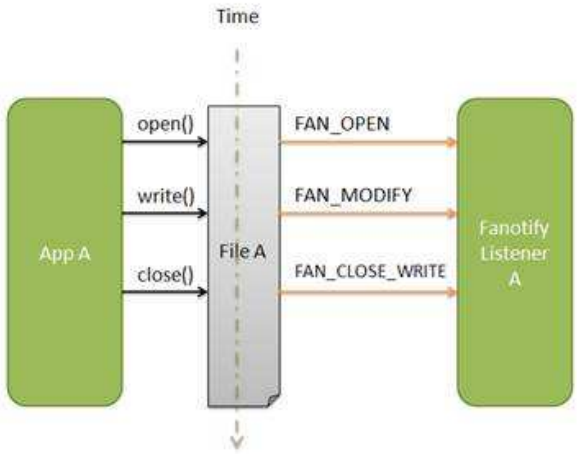
link1 http://lwn.net/Articles/339253/

and updates on **link1**:

http://stackoverflow.com/questions/1835947/how-do-i-program-for-linuxs-new-fanotify-file-system-monitoring-feature

link2 https://www.ibm.com/developerworks/cn/linux/l-cn-fanotify/

fanotify works in following procedure as shown in **link2**, basic, in an application interested in watching any file system changes should register a listener into fanotify and later upon any file operation in watching scope from any other application, event should be sent to the register application and furthermore, access or deny decision can be made by register application and hand back to kernel to final grant or deny the file operation.



the APIs for fanotify are described at http://lwn.net/Articles/339399/

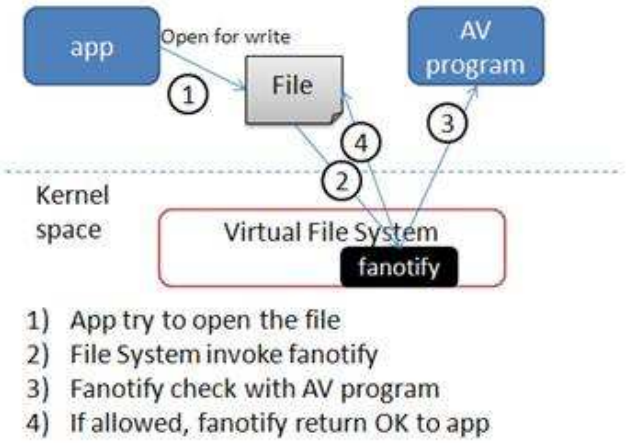
the manual page in progress: http://thread.gmane.org/gmane.linux.man/2375

kernel sources for x86/ia64 in 2.6.38 touched by fanotify are as bellow:

```
arch/ia64/include/asm/unistd.h
arch/ia64/kernel/entry.S
arch/x86/ia32/ia32entry.S
arch/x86/ia32/sys_ia32.c
arch/x86/include/asm/sys_ia32.h
arch/x86/include/asm/unistd_32.h
arch/x86/include/asm/unistd_64.h
arch/x86/kernel/asm-offsets.s
arch/x86/kernel/syscall_table_32.S
fs/notify/Kconfig
```

```
fs/notify/Makefile
fs/notify/fanotify/Kconfig
fs/notify/fanotify/Makefile
fs/notify/fanotify/fanotify.c
fs/notify/fanotify/fanotify_user.c
include/asm-generic/unistd.h
include/config/auto.conf
include/generated/autoconf.h
include/linux/Kbuild
include/linux/fanotify.h
include/linux/fs.h
include/linux/fsnotify_backend.h
include/linux/sched.h
include/linux/syscalls.h
kernel/sys_ni.c
```

2. the user case for clamav can be demonstrated as bellow(via **link2**):



3. dazuko and DazukoFS is not used in clamav any more and fanotify is coming

http://lists.nongnu.org/archive/html/dazuko-help/2010-10/msg00002.html

Re: [Dazuko-help] It is dead?

From: Frans de Boer
Subject: Re: [Dazuko-help] It is dead?
Date: Wed, 13 Oct 2010 20:22:05 +0200
User-agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.1.12)

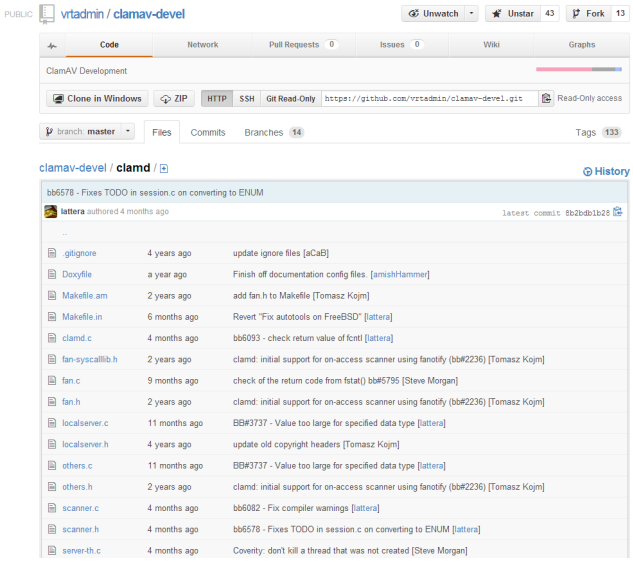
On 10/13/2010 10:41 AM, John Ogness wrote:
> *Note: This email is being cross-posted to dazuko-devel since it*
> *contains relevant development information.*
>
> *On 2010-10-13, Frans de Boer <address@hidden> wrote:*
>
>> *No progress, no resolution for the 2.6.35+ kernel....is there a*
>> *change it will spark back to life or am I missing something?*
>>
> *The 2.6.36 kernel is about to be released. I don't plan on integrating*
> *the already publicly available DazukoFS patches into a new version*
> *until then. At that point I would also fix any new issues with 2.6.36.*
>
> *2.6.36 is actually a very interesting kernel release because it will*
> *include (for the first time) the fanotify interface. Although the*
> *implementation is quite different, fanotify could be used as a*
> *replacement for DazukoFS. And since it is now mainline Linux, one must*
> *ask if DazukoFS should continue to exist.*
>
> *I plan on writing test applications using fanotify to demonstrate that*
> *fanotify can be a replacement for DazukoFS. If the applications indeed*
> *show this (which I assume they will), then I will officially orphan*
> *(or pass on maintenance of) DazukoFS (after fixing 2.6.36 issues).*
>
> *John Ogness*
>
>
I know that ClamAV in the 0.97 release plan to use Fanotify instead of dazukoFS. When a product has made it main stream, then why try to continue if dazukoFS already has offered itself for inclusion in the main stream. As far as I know, it did not even made it into the staging area.

Anyhow, thanks sofar for given us dazuko and dazukoFS in the past.

Regards, Frans.

Note:

the support of dazuko/dazukoFS is lifted from version 0.98 using fanotify(fan.c):



in this article, we use clamav-0.98 as an example with fanotify helping doing on access scan

Data structures

TBD

Test case

1. test fanotify
2. test clamav on access scan with fanotify

test fanotify

environment:

OS

```
Linux ubuntu 2.6.38-8-generic #42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011 i686 athlon i386 GNU/Linux
```

check gcc version

```
gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/i386-linux-gnu/gcc/i686-linux-gnu/4.5.2/lto-wrapper
Target: i686-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.5.2-8ubuntu4' --with-bugurl=file:///usr/share/doc/gcc-4.5/README.Bugs --enable-languages=c,c++,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.5 --enable-shared --enable-multiarch --with-multiarch-defaults=i386-linux-gnu --enable-linker-build-id --with-system-zlib --libexecdir=/usr/lib/i386-linux-gnu --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/include/c++/4.5 --libdir=/usr/lib/i386-linux-gnu --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-plugin --enable-gold --enable-ld=default --with-plugin-ld=ld.gold --enable-objc-gc --enable-targets=all --disable-werror --with-arch-32=i686 --with-tune=generic --enable-checking=release --build=i686-linux-gnu --host=i686-linux-gnu --target=i686-linux-gnu
Thread model: posix
gcc version 4.5.2 (Ubuntu/Linaro 4.5.2-8ubuntu4)
```

check kernel version and complier

```
cat /proc/version
Linux version 2.6.38-8-generic (buildd@vernadsky) (gcc version 4.5.2 (Ubuntu/Linaro 4.5.2-8ubuntu3) ) #42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011
```

compile and run

the test code is at:

http://www.lanedo.com/~aleksander/fanotify/fanotify-example.c

this code can monitoring file operation on specified target dir.

compile:

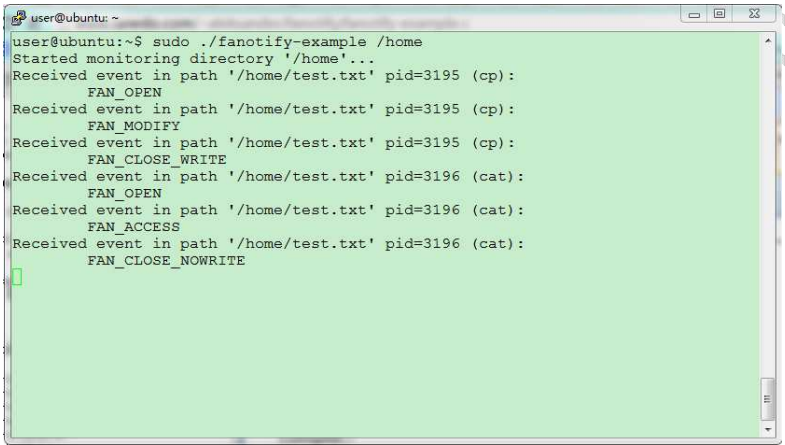
```
gcc -o fanotify-example fanotify-example.c
```

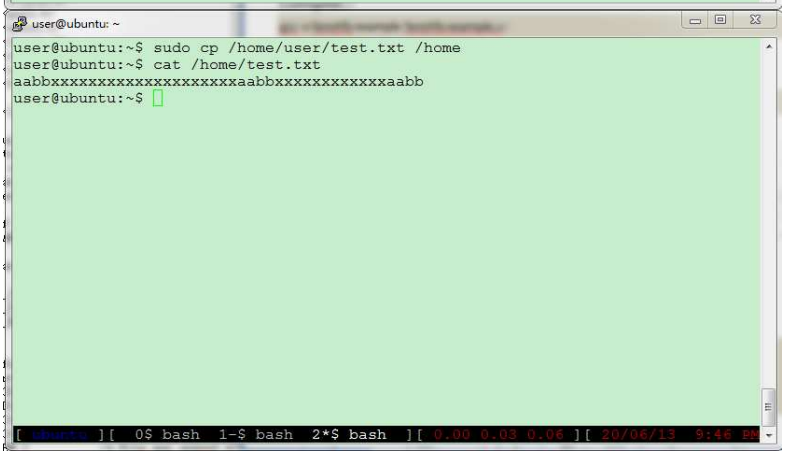
run to monitor /home dir:

```
sudo ./fanotify-example /home
```

file operation:

1. copy a file to /home
2. cat this file





test clamav on access scan with fanotify

this test will using a self defined bytecode database for clamav daemon to play with

and later will operate on a file that matches the bytecode for clamav to scan and catch.

preparation - bytecode

source code of bytecode

test_bytecode_on_access.c

```
VIRUSNAME_PREFIX("test_bytecode_on_access.c")
VIRUSNAMES("A","B")
TARGET(7)
SIGNATURES_DECL_BEGIN
DECLARE_SIGNATURE(magic)
SIGNATURES_DECL_END

SIGNATURES_DEF_BEGIN
DEFINE_SIGNATURE(magic,"61616262")        // the pattern as "aabb" in hex
SIGNATURES_END
bool logical_trigger (void)
{
        // @ clamav-bytecode-compiler/obj/Release/lib/clang/1.1/include/bytecode_local.h
        return count_match(Signatures.magic) != 1; // if "aabb" match count is '1', it's not a virus
}

int entrypoint (void)
{
        int count = count_match(Signatures.magic);
        if ( count == 3) foundVirus("B"); // 3 matches of "aabb", find virus B
        else if ( count != 0) foundVirus ("A"); // have match but no 3 times, find virus A
        return 0;
}
```

Note:

in production mode, clamav will not accept unsigned bytecode, so in order to make this test case work, following patch should be made to clamav code(also, an alternative solution is use normal signature rather than bytecode):

```
--- a/clamd/clamd.c
+++ b/clamd/clamd.c
@@ -462,6 +462,8 @@ int main(int argc, char **argv)
                dboptions |= CL_DB_BYTECODE_UNSIGNED;
                logg("#Bytecode: Enabled support for unsigned bytecode.\n");
        }
+       dboptions |= CL_DB_BYTECODE_UNSIGNED; //CHR always enable loading unsigned bytecode
+
        if((opt = optget(opts,"BytecodeMode"))->enabled) {
```

```
                enum bytecode_mode mode;
                if (!strcmp(opt->strarg, "ForceJIT"))
diff --git a/clamscan/manager.c b/clamscan/manager.c
```

compile and put in clamav virus db

```
sudo cp test_bytecode_on_access.cbc /var/lib/clamav
```

test file

test.txt <= virus match

```
aabbxxxxxxxxxxxxxxxxxxxxaabbxxxxxxxxxxxxaabb
```

preparation – clamd.conf

edit as:

```
User root # run as root
ScanOnAccess yes  # enable on access scan
OnAccessIncludePath /home # protect /home dir
```

Note:

currently, fanotify can only monitor 2 level of FS changes, e.g.: in this case change is in /home dir and subdir in /home(e.g.: /home/user) will be catch and other deep level change(e.g.: /home/user/anotherdir) will not be catch

on access scan test

start clamav and expect following logs:

```
user@ubuntu:~$ sudo cat /tmp/clamd.log
Thu Jun 20 22:32:40 2013 -> +++ Started at Thu Jun 20 22:32:40 2013
Thu Jun 20 22:32:40 2013 -> clamd daemon devel-clamav-0.97-826-gfc53951 (OS: linux-gnu, ARCH: i386, CPU: i686)
Thu Jun 20 22:32:40 2013 -> Running as user root (UID 0, GID 0)
Thu Jun 20 22:32:40 2013 -> Log file size limited to 2097152 bytes.
Thu Jun 20 22:32:40 2013 -> Reading databases from /var/lib/clamav
Thu Jun 20 22:32:40 2013 -> Not loading PUA signatures.
Thu Jun 20 22:32:40 2013 -> Bytecode: Security mode set to "TrustSigned".
Thu Jun 20 22:32:45 2013 -> Loaded 1727016 signatures.
Thu Jun 20 22:32:46 2013 -> TCP: Bound to address 127.0.0.1 on port 3310
Thu Jun 20 22:32:46 2013 -> TCP: Setting connection queue length to 30
Thu Jun 20 22:32:46 2013 -> LOCAL: Unix socket file /tmp/clamd.socket
Thu Jun 20 22:32:46 2013 -> LOCAL: Setting connection queue length to 30
Thu Jun 20 22:32:46 2013 -> Limits: Global size limit set to 104857600 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: File size limit set to 26214400 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: Recursion level limit set to 16.
Thu Jun 20 22:32:46 2013 -> Limits: Files limit set to 10000.
Thu Jun 20 22:32:46 2013 -> Limits: Core-dump limit is 0.
Thu Jun 20 22:32:46 2013 -> Limits: MaxEmbeddedPE limit set to 10485760 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: MaxHTMLNormalize limit set to 10485760 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: MaxHTMLNoTags limit set to 2097152 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: MaxScriptNormalize limit set to 5242880 bytes.
Thu Jun 20 22:32:46 2013 -> Limits: MaxZipTypeRcg limit set to 1048576 bytes.
Thu Jun 20 22:32:46 2013 -> Archive support enabled.
Thu Jun 20 22:32:46 2013 -> Algorithmic detection enabled.
Thu Jun 20 22:32:46 2013 -> Portable Executable support enabled.
Thu Jun 20 22:32:46 2013 -> ELF support enabled.
Thu Jun 20 22:32:46 2013 -> Mail files support enabled.
Thu Jun 20 22:32:46 2013 -> OLE2 support enabled.
Thu Jun 20 22:32:46 2013 -> PDF support enabled.
Thu Jun 20 22:32:46 2013 -> SWF support enabled.
Thu Jun 20 22:32:46 2013 -> HTML support enabled.
Thu Jun 20 22:32:46 2013 -> Self checking every 600 seconds.
Thu Jun 20 22:32:46 2013 -> Listening daemon: PID: 3327
Thu Jun 20 22:32:46 2013 -> MaxQueue set to: 100
Thu Jun 20 22:32:46 2013 -> ScanOnAccess: Protecting directory '/home'
Thu Jun 20 22:32:46 2013 -> ScanOnAccess: Max file size limited to 5242880 bytes
user@ubuntu:~$
```

file operations:

1. create /home/test.txt as bellow:

   aabbxxxxxxxxxxxxxxxxxxxxxxaabbxxxxxxxxxxxxaabb

2. cat it
3. have fanotify-example monitoring as well

results:

fanotify-example output:

```
user@ubuntu:~$ sudo ./fanotify-example /home

[sudo] password for user:

Started monitoring directory '/home'...

# sudo vi /home/test.txt

Received event in path '/home/user' pid=3528 (vi):

        FAN_OPEN

Received event in path '/home/user' pid=3528 (vi):

        FAN_CLOSE_NOWRITE

Received event in path '/home/user' pid=3528 (vi):

        FAN_OPEN

Received event in path '/home/user' pid=3528 (vi):

        FAN_CLOSE_NOWRITE

Received event in path '/home/user' pid=3528 (vi):

        FAN_OPEN

Received event in path '/home/user' pid=3528 (vi):

        FAN_CLOSE_NOWRITE

Received event in path '/home/user' pid=3528 (vi):

        FAN_OPEN
```

```
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
```

```
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/.test.txt.swx' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/.test.txt.swx' pid=3528 (vi):
        FAN_CLOSE_WRITE
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_CLOSE_WRITE
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_MODIFY
Received event in path '/home/user' pid=3528 (vi):
        FAN_OPEN
Received event in path '/home/user' pid=3528 (vi):
        FAN_CLOSE_NOWRITE
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_MODIFY
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_MODIFY
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_MODIFY
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_MODIFY
Received event in path '/home/test.txt' pid=3528 (vi):
        FAN_OPEN
        FAN_MODIFY
Received event in path '/home/test.txt' pid=3528 (vi):
        FAN_MODIFY
        FAN_CLOSE_WRITE
```

```
Received event in path '/home/.test.txt.swp' pid=3528 (vi):
        FAN_CLOSE_WRITE
# cat /home/test.txt
Received event in path '/home/test.txt' pid=3537 (cat):
        FAN_OPEN
Received event in path '/home/test.txt' pid=3537 (cat):
        FAN_ACCESS
Received event in path '/home/test.txt' pid=3537 (cat):
        FAN_CLOSE_NOWRITE
```

clamav output:

```
user@ubuntu:~$ sudo tail -n 1 /tmp/clamd.log
Thu Jun 20 22:37:38 2013 -> ScanOnAccess: /home/test.txt: test_bytecode_on_acces
s.c.B(cb5132ddcb50f30d0185a19841249bad:45) FOUND
user@ubuntu:~$
```

another test using clamav's test virus:
command:

```
user@ubuntu:~$ sudo cp /home/user/clamav-devel/clamav-devel/test/clam.exe /home
user@ubuntu:~$ cat /home/clam.exe
瑒 fp? 殄 ? 纊P笸!碙? bf瑧  1x
pvN/谔磊鬕ERNEL32.DLLExitProcessUSER32.DLLCLAMessageBoxA? ???PELaCaB郯@@

  [CLAMAV]纔ser@ubuntu:~$
```

clamd.log:

```
user@ubuntu:~$ sudo tail -n 2 /tmp/clamd.log
Thu Jun 20 22:42:53 2013 -> ScanOnAccess: Max file size limited to 5242880 bytes
Thu Jun 20 22:45:04 2013 -> ScanOnAccess: /home/clam.exe: ClamAV-Test-File(aa15b
cf478d165efd2065190eb473bcb:544) FOUND
user@ubuntu:~$
```

Note:
though identified the signature, but clamav still does not block the access to the file.

The call flow