Automatically Backup Module against Ransomware Attack

2018/06/26

Vice President/OSS/Security Evangelist
Kazuki Omo(面　和毅)

Agenda

1. Ransomware?
2. Prepare for Ransomware infection.
3. Server Side Solution
4. Implementation
5. (+Demo)

Who am I?

- Security Researcher/Engineer (18 years)
- SELinux/MAC Evangelist (14 years)
- Antivirus Engineer (3 years)
- SIEM Engineer (3 years)
- CISSP (#366942)
- 120kg Bench Press Max
- Member of Secure OSS-Sig

1. Ransomware

What is Ransomware?

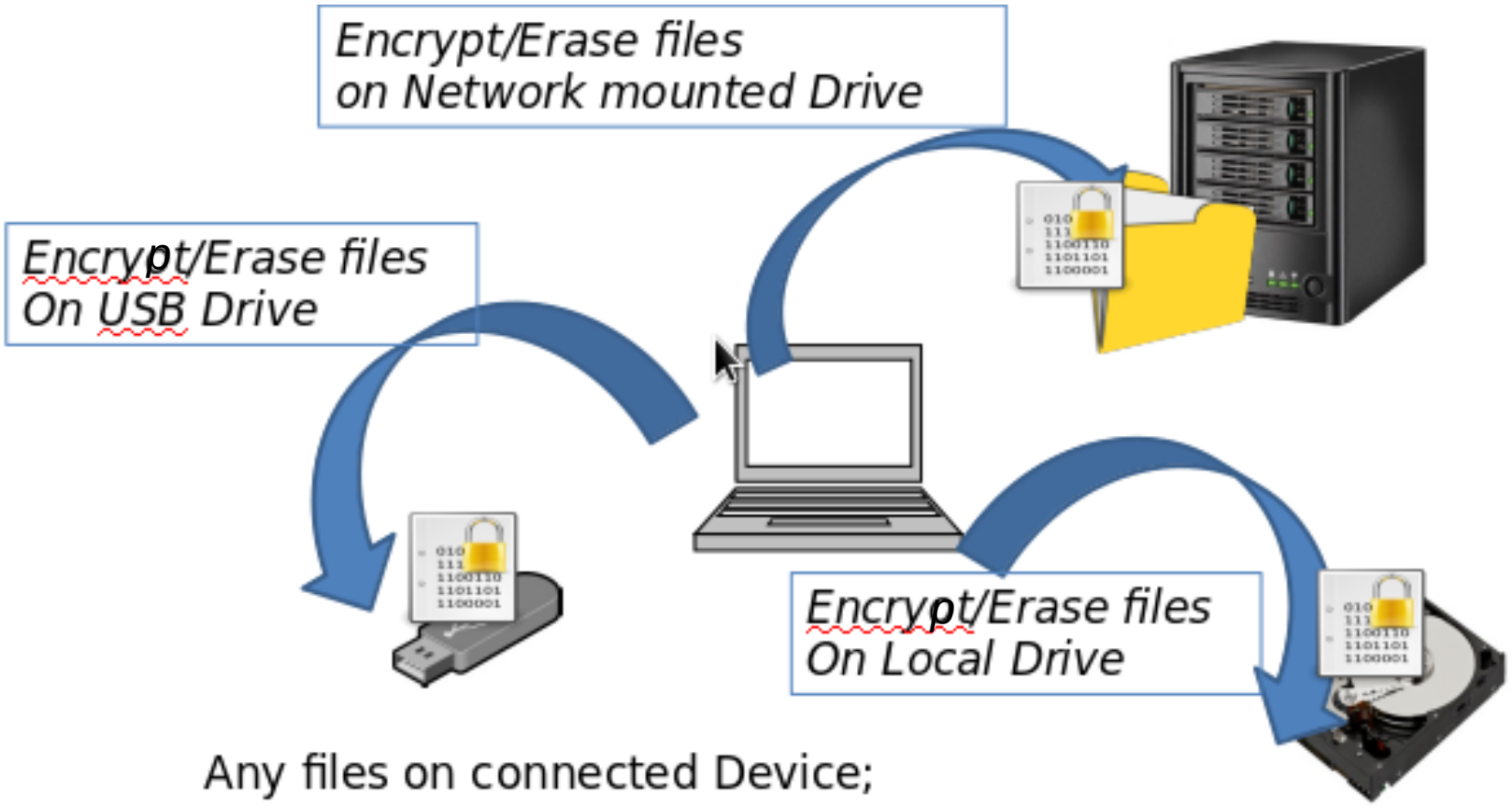
Ransomware

[TrendMicro definition]

Ransomware is a **type of malware** that prevents or limits users from accessing their system, either by locking the system's screen or **by locking the users' files** unless a ransom is paid.

(Ref: https://www.trendmicro.com/vinfo/us/security/definition/ransomware)

So, what Ransomware do?



Any files on connected Device;

- Encrypt ->
- Erase randomly ->

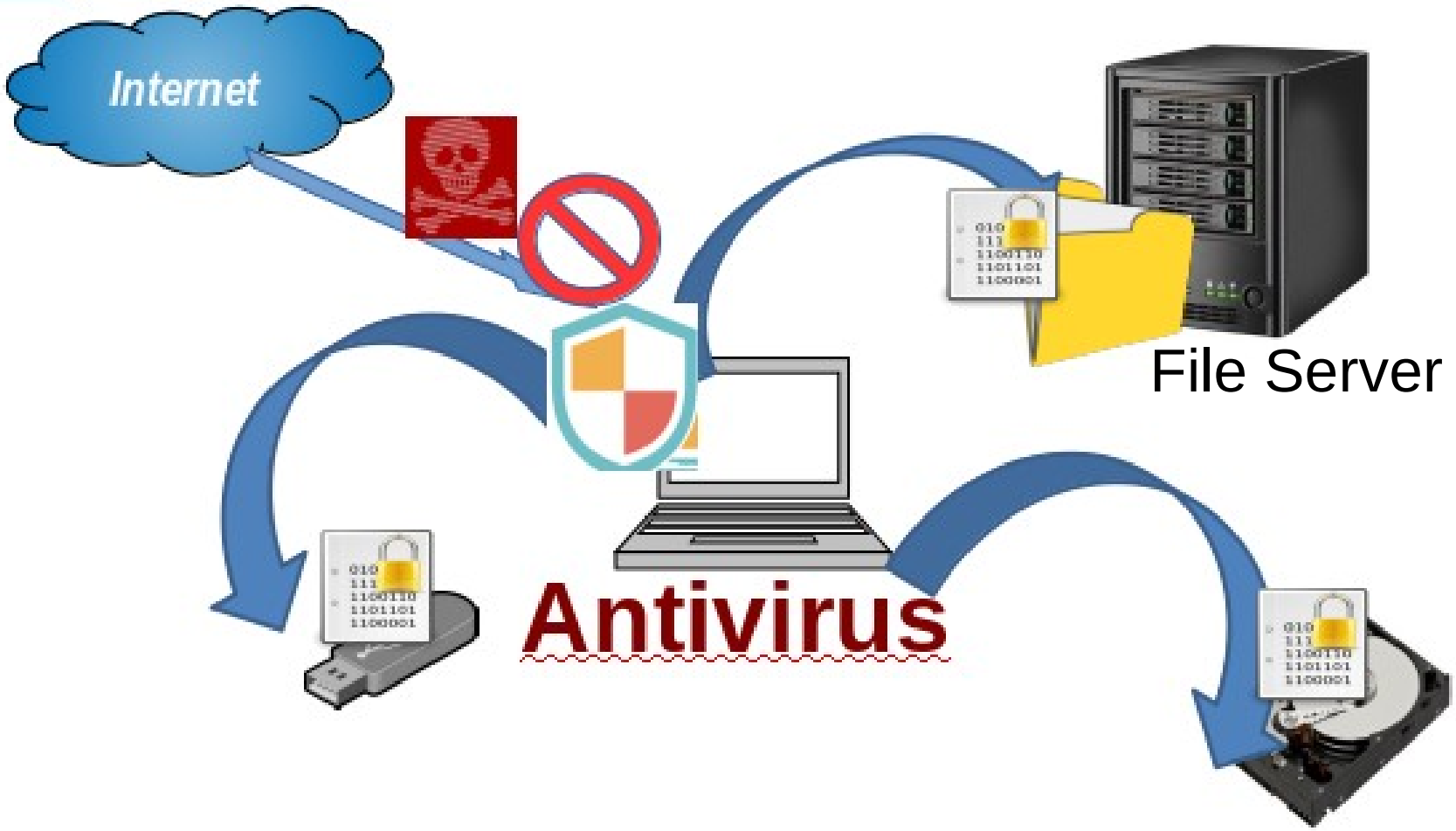
Workaround?

“Ransomware is a **type of malware**”

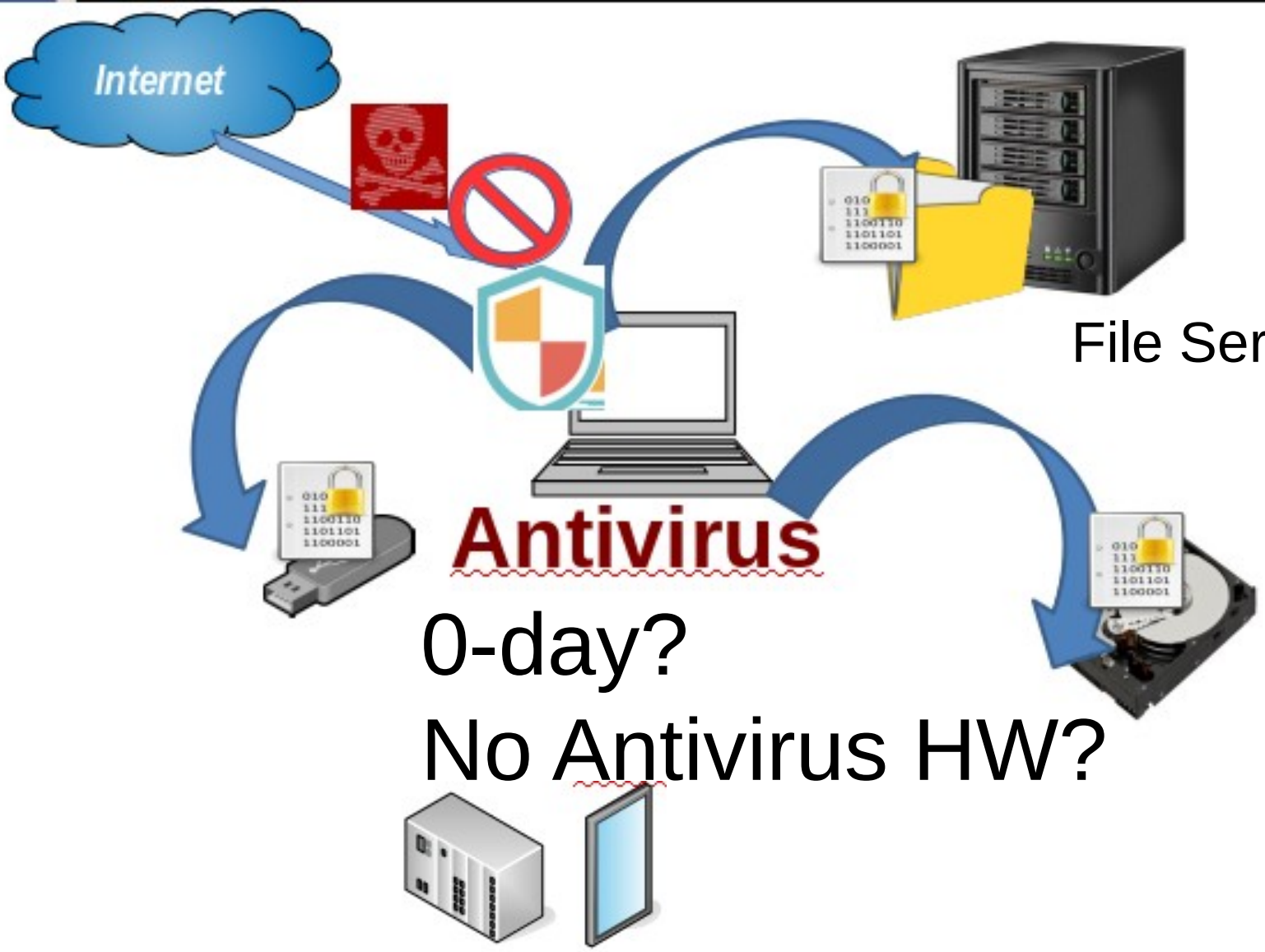
Let’s think about

- Detect/Reject Ransomware. → Antivirus
- Prepare for Ransomware infection. → ???

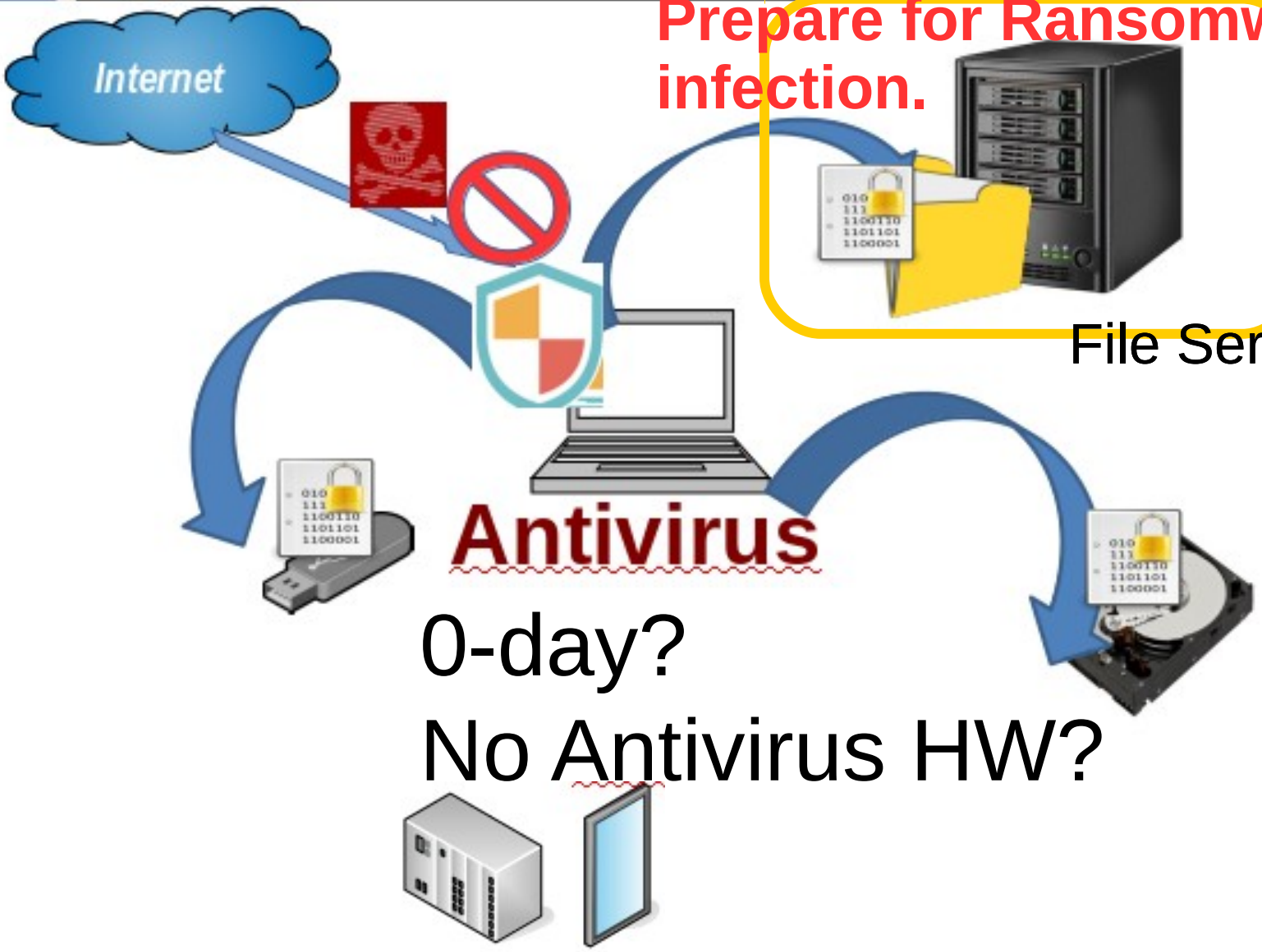
Workaround 1. Antivirus



Workaround 1. Antivirus



Workaround



2. Prepare for Ransomware infection.

File Server Side:

“Prepare for Ransomware” (From Server point of view)

- Fight with Ransomware (AntiVirus, etc.)

File Server Side:

“Prepare for Ransomware” (From Server point of view)

- Fight with Ransomware (AntiVirus, etc.) → **Difficult**

Client Side:

How Ransomware work (from Client Side).

	Encrypt File	Write Encrypted File	Delete Original file
Client	- Choose target file - Encrypt file	- Write Encrypted file	• Request to Delete file.

File Server:

How Ransomware work (from File Server).

	Encrypt File	Write Encrypted File	Delete Original file
Client	- Choose target file - Encrypt file	- Write Encrypted file	• Request to Delete file.
File Server	• sys_open/sys_read through smbd etc. • sys_close to target file(?)	• Write some file on shared volume through smbd.	• Delete file on shared Volume through smbd.

File Server:

How Ransomware work (from File Server).

	Encrypt File	→ Write Encrypted File	→ Delete Original file
Client	- Choose target file - Encrypt file	- Write Encrypted file	• Request to Delete file.
File Server	• sys_open/sys_read through smbd etc. • sys_close to target file(?)	• Write some file on shared volume through smbd.	• Delete file on shared Volume through smbd.

Usual behavior as File Server.

File Server Side:

“Prepare for Ransomware” (From Server point of view)

- Fight with Ransomware (AntiVirus, etc.) → **Difficult**
- **Prepare a way to restore the file. → Take Backup.**

File Server:

So, How we can do?

	Encrypt File	Write Encrypted File	Delete Original file
Client	- Choose target file - Encrypt file	- Write Encrypted file	• Request to Delete file.
File Server	• sys_open/sys_read through smbd etc. • sys_close to target file(?)	• Write unknown file on shared volume through smbd.	• Delete file on shared Volume through smbd.

Let's think about here.

SIOS

Config Recycle bin on Samba 3/4.

“vfs objects = recycle” to enable Recycle bin.

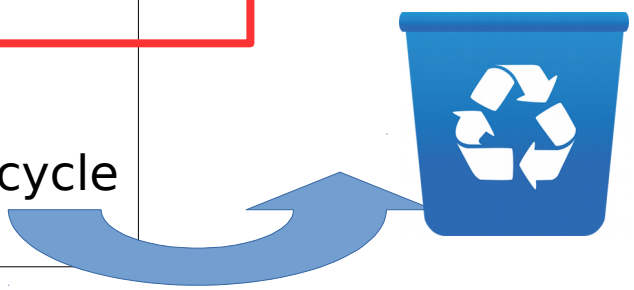
```
[Share]
        comment = Public Stuff
        path = /Share/
        browseable = yes
        writable = yes
        printable = no
        vfs objects = recycle
        guest ok = yes
        read only = no
        recycle:repository = .recycle
        recycle:keeptree = yes
```

Linux+Samba

Config Recycle bin on Samba 3/4.

“vfs objects = recycle” to enable Recycle bin.

```
[Share]
        comment = Public Stuff
        path = /Share/
        browseable = yes
        writable = yes
        printable = no
        vfs objects = recycle
        guest ok = yes
        read only = no
        recycle:repository = .recycle
        recycle:keeptree = yes
```



.recycle/

How about Modify?

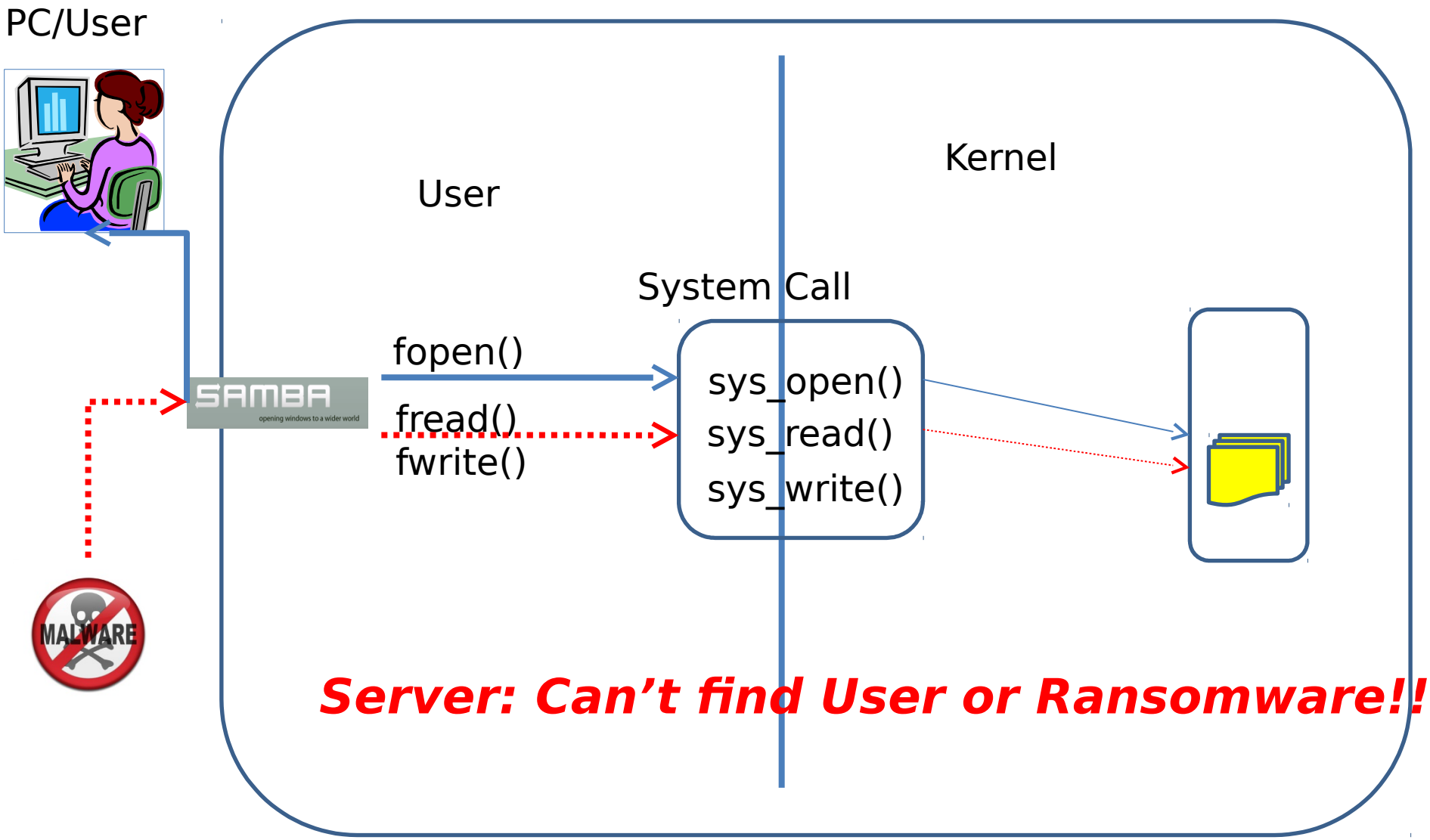
What about “**Modify**”, not “Delete”?

	Encrypt File	Write Encrypted File	Delete **Modify** Client Original file
	- Choose target file - Encrypt file	- Write Encrypted file	• Request to Delete file.
File Server	• sys_open/sys_read through smbd etc. • sys_close to target file(?)	• Write unknown file on shared volume through smbd.	• Delete **Modify** file on shared Volume through smbd.

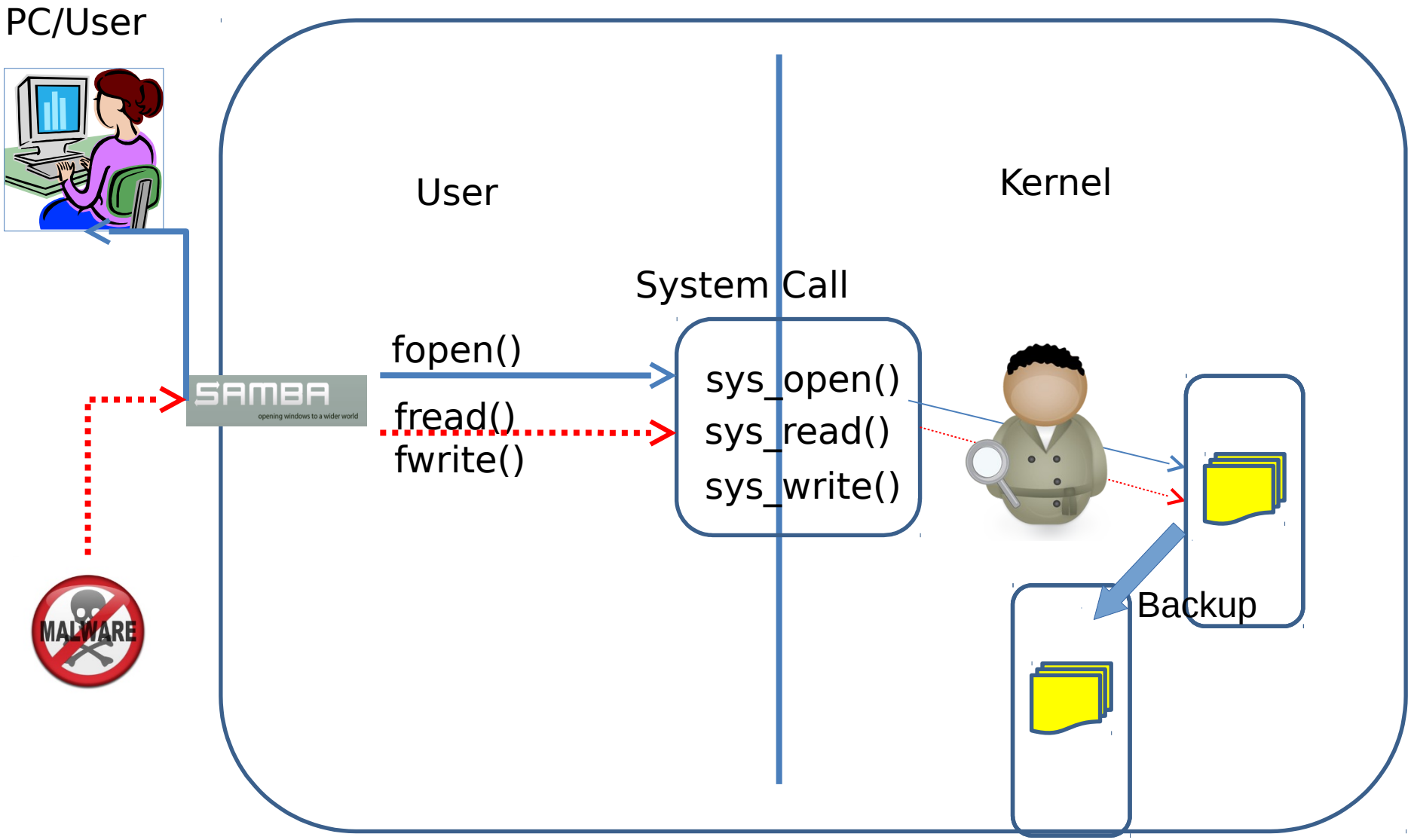
What about this situation?

3. Server side Solution.

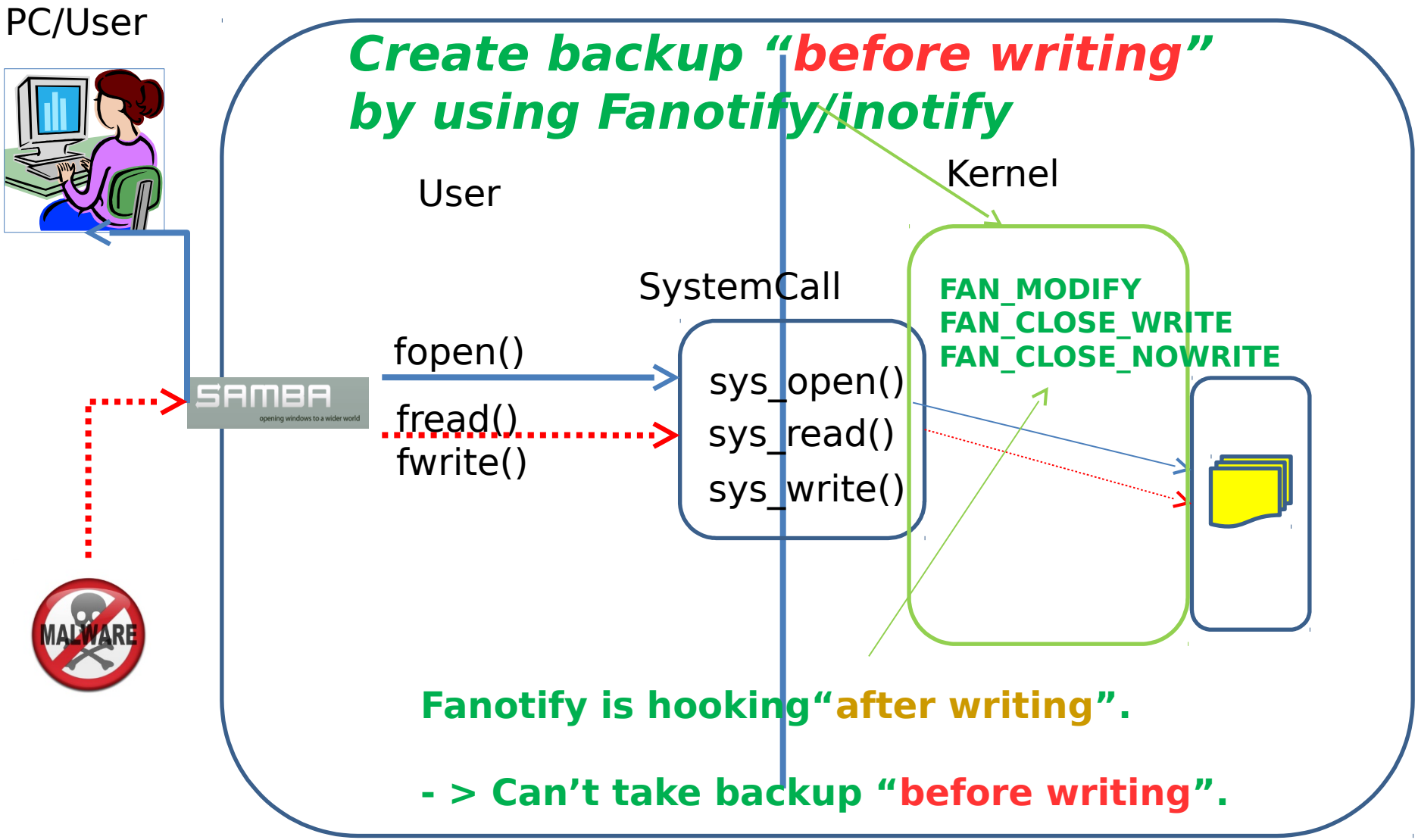
From File Server:



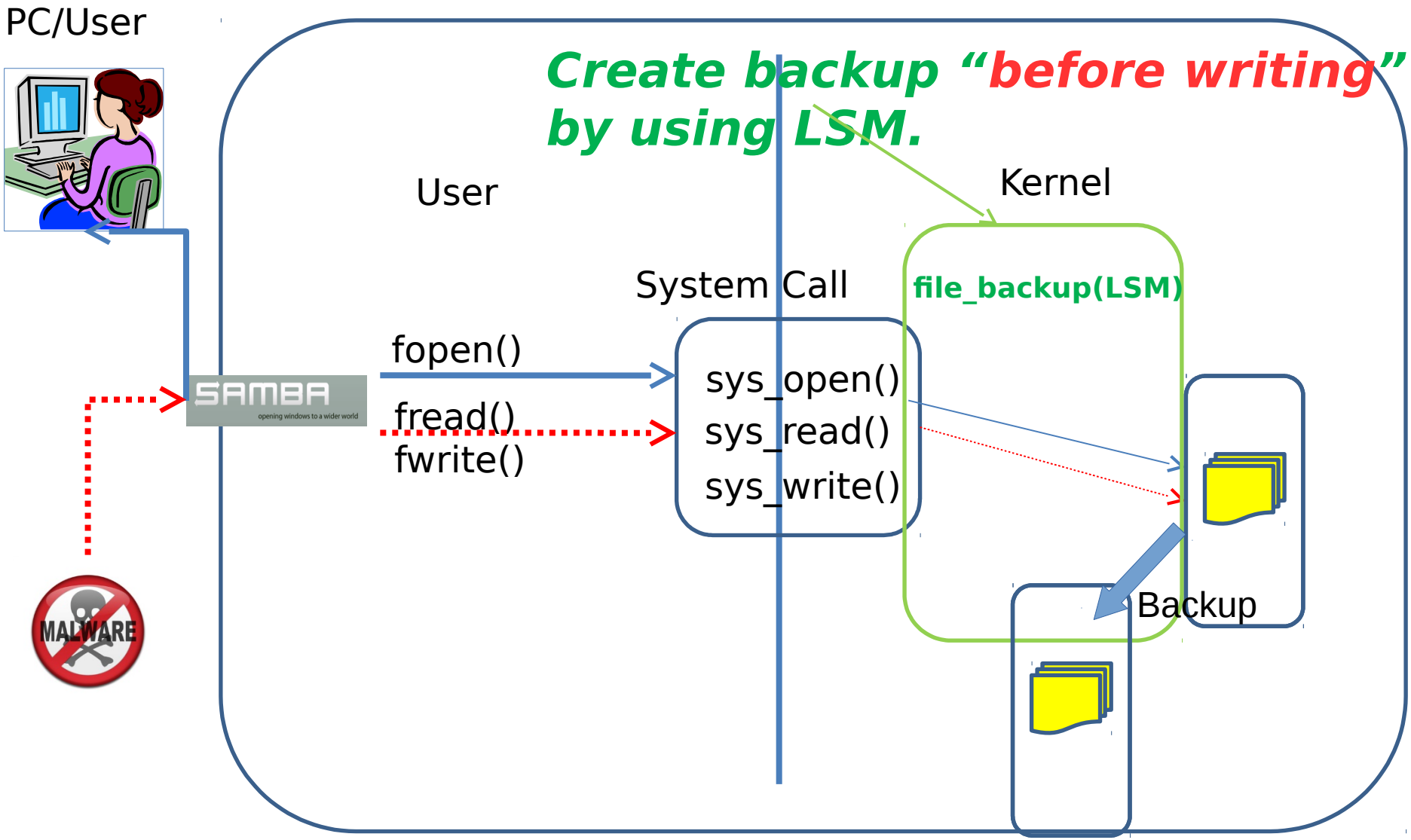
From File Server: (Auto Backup)



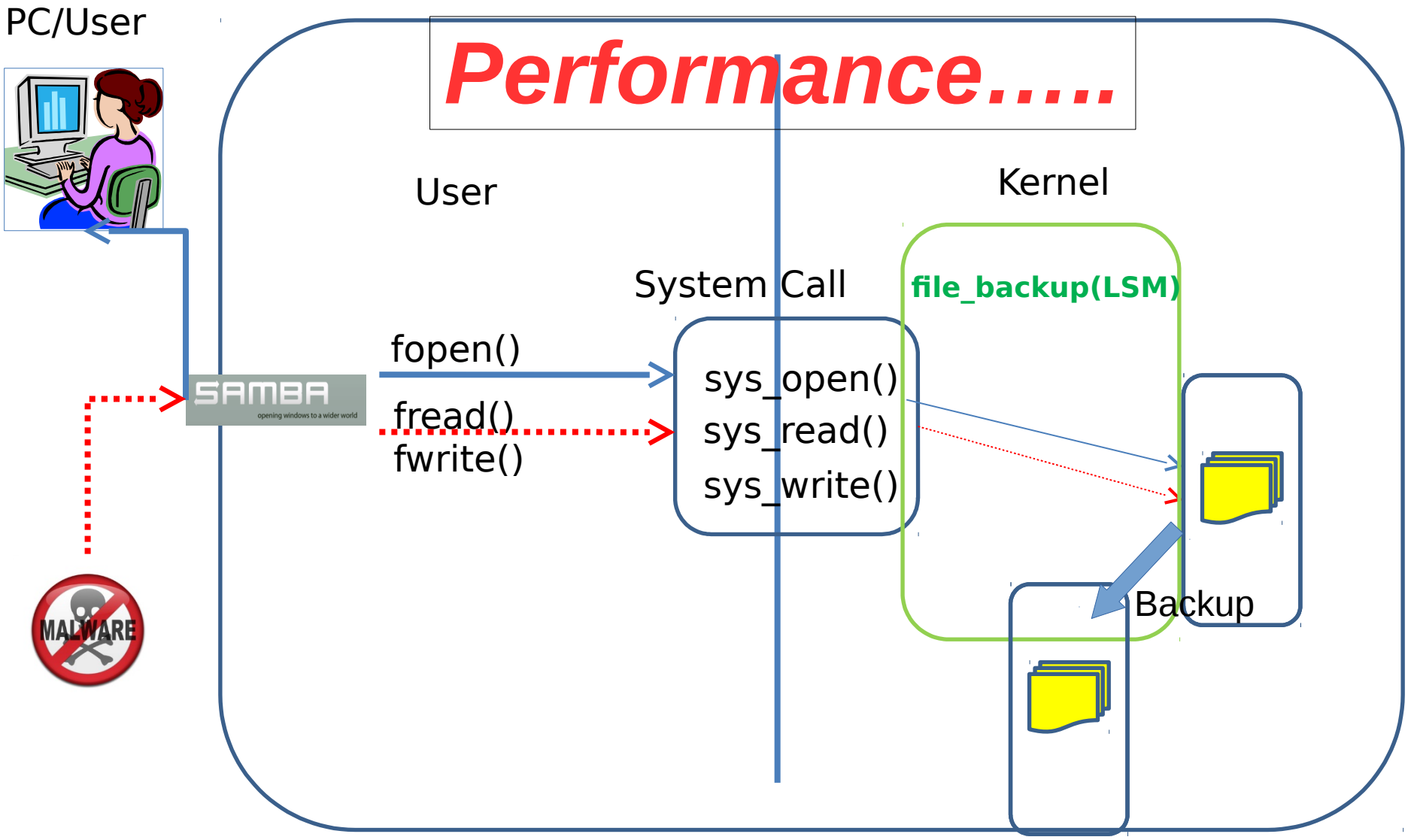
From File Server: (fanotify solution)



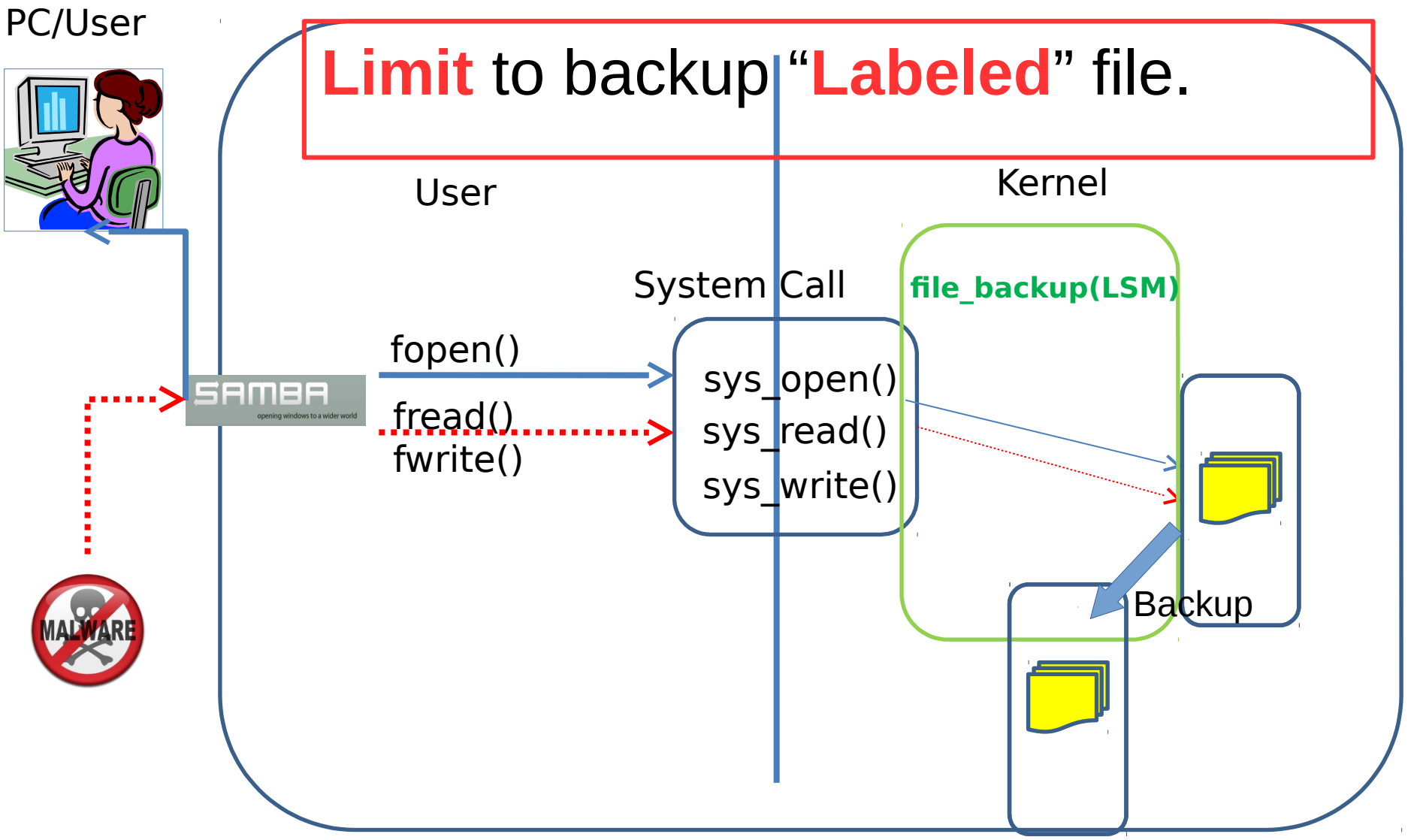
From File Server: (LSM solution)



From File Server: (LSM solution)



From File Server: (LSM solution)

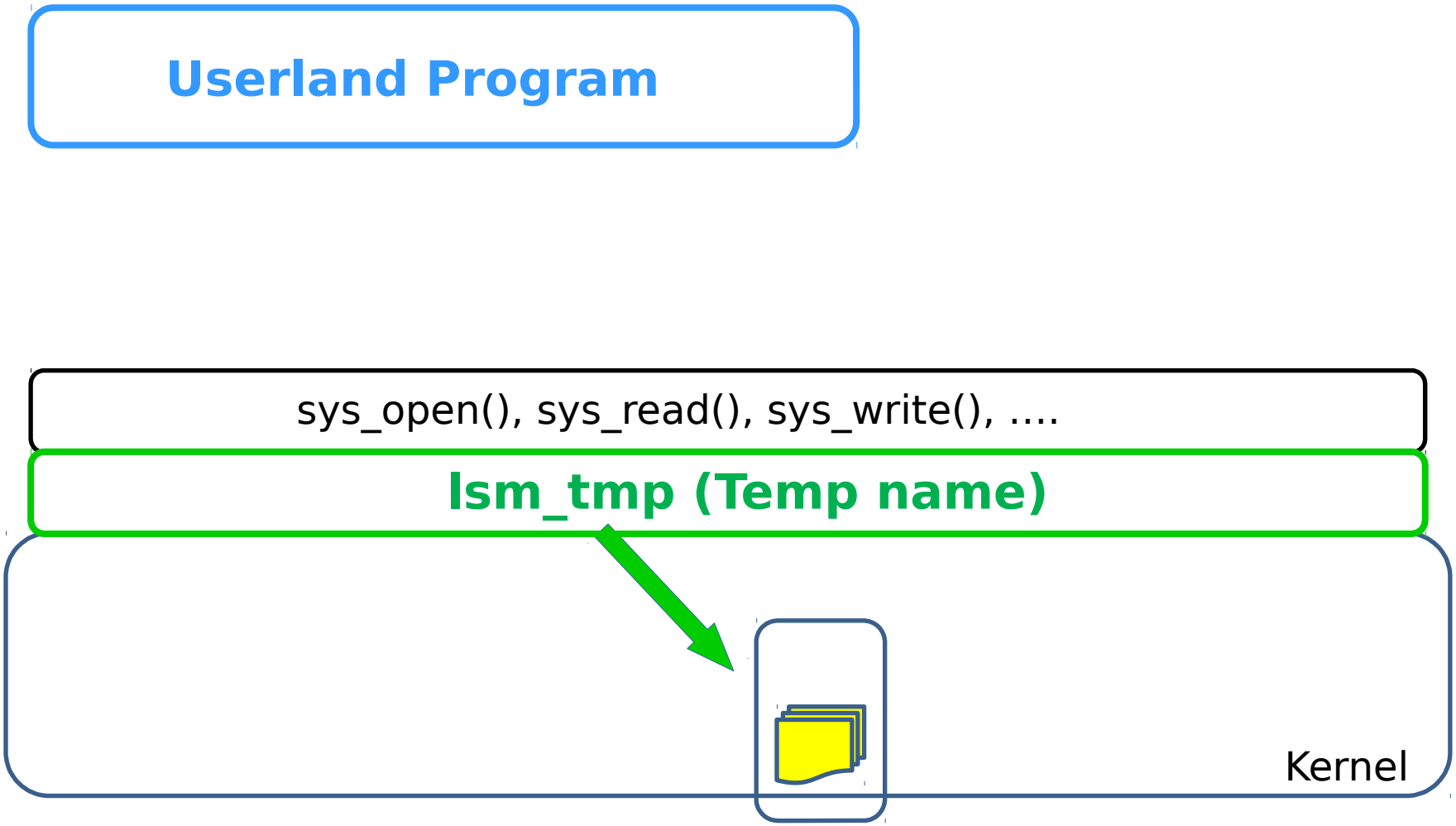


File Label

```
user@local:~/testdir$ ls -lh
total 4.0K
-rw-r--r-- 1 user user 148 Jun  1 10:43 h123
user@local:~/testdir$ getfattr h123
# file: h123
user.backup_label

user@local:~/testdir$
```

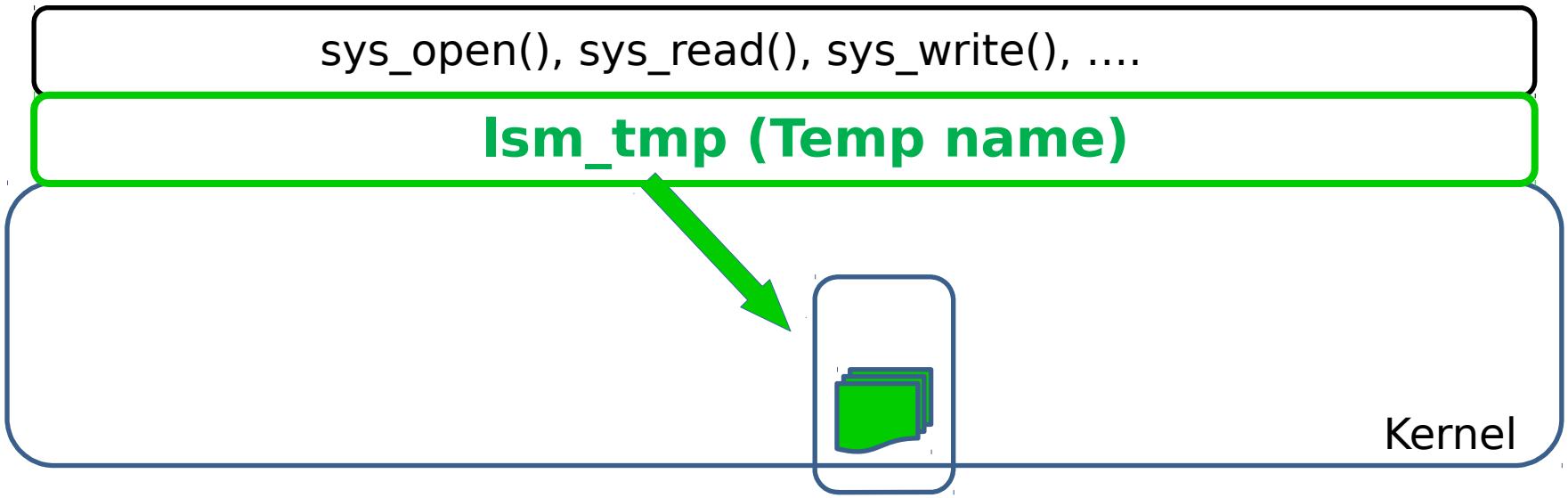
4. Implementation

Usually...

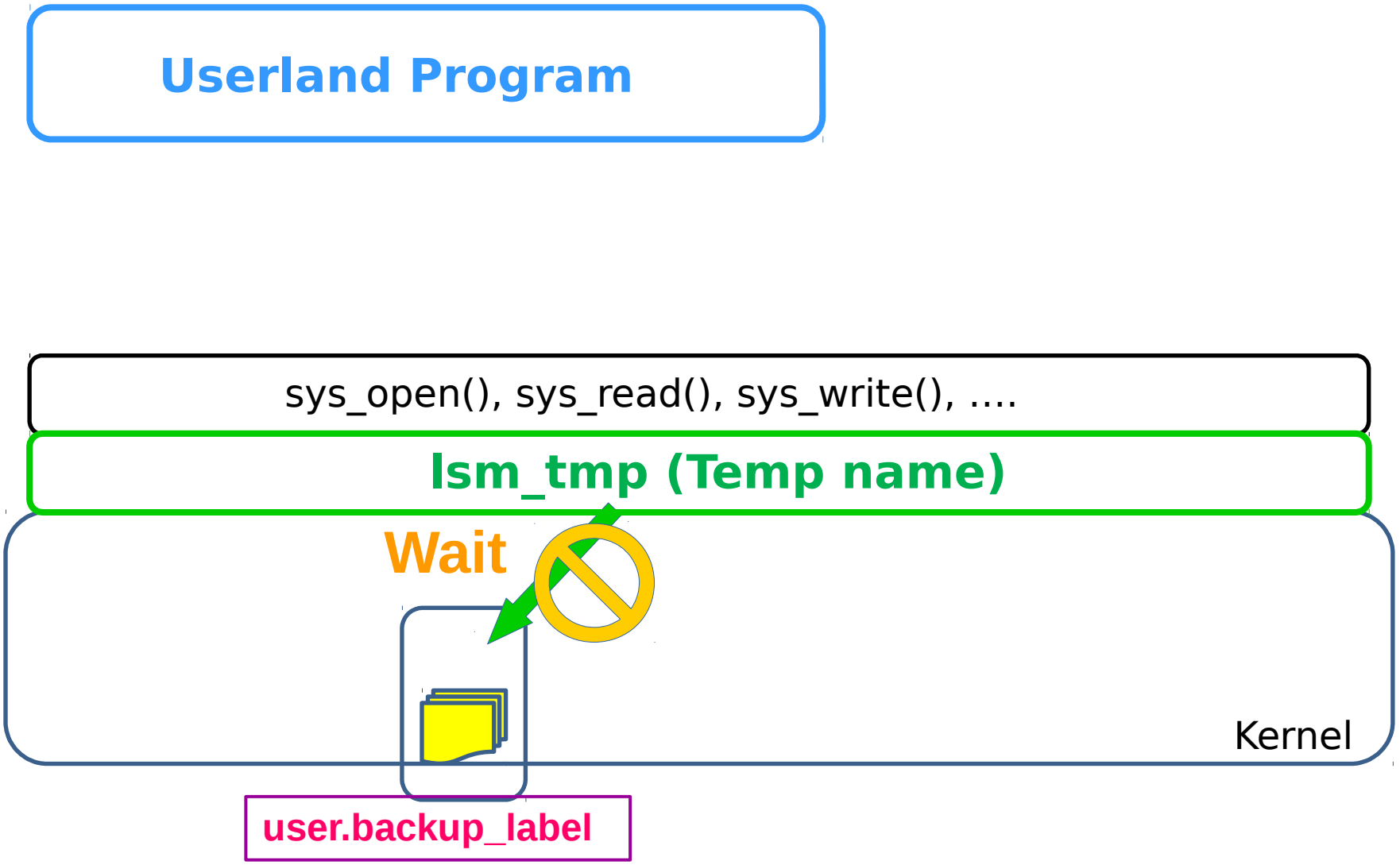


Usually...

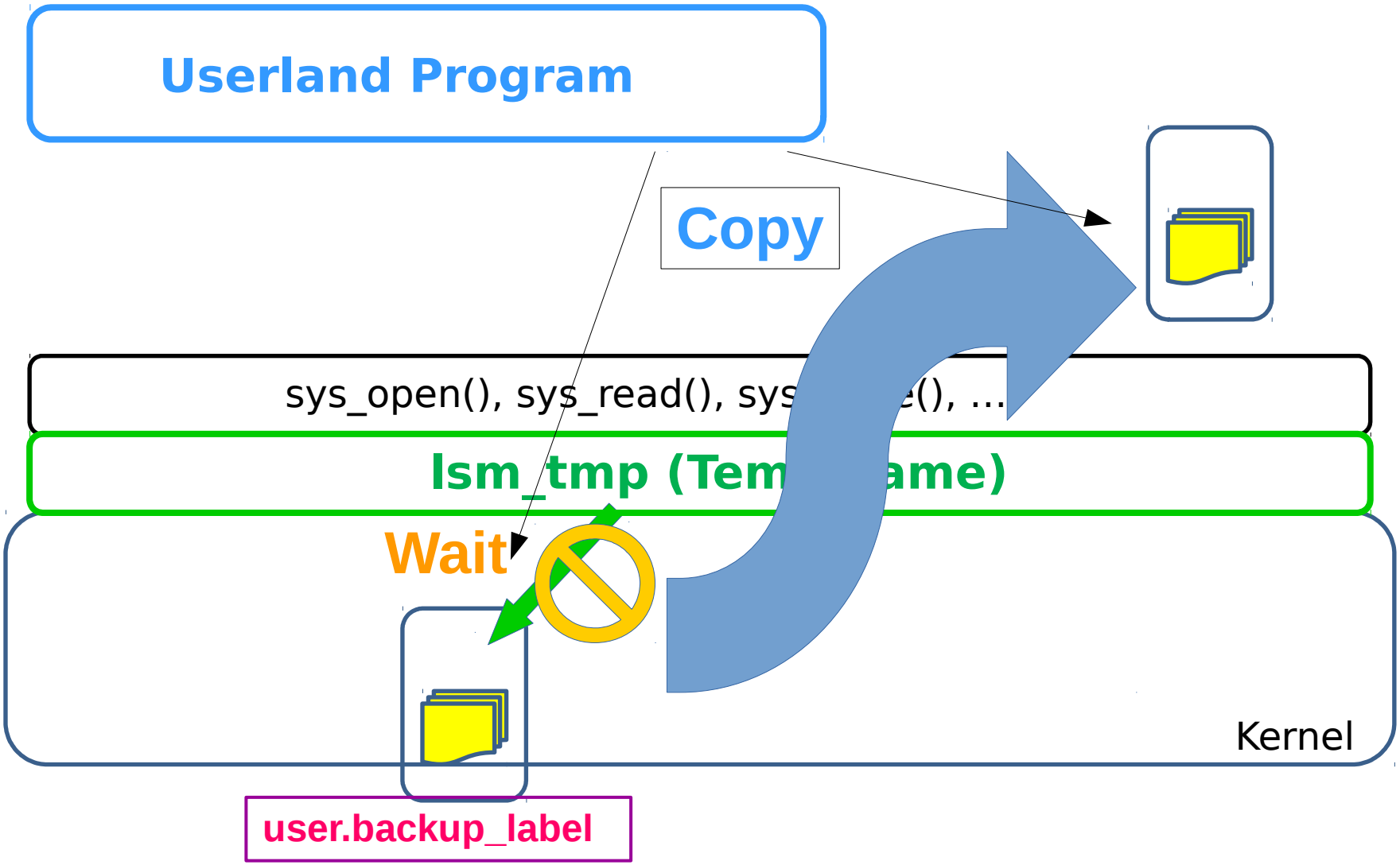
Userland Program



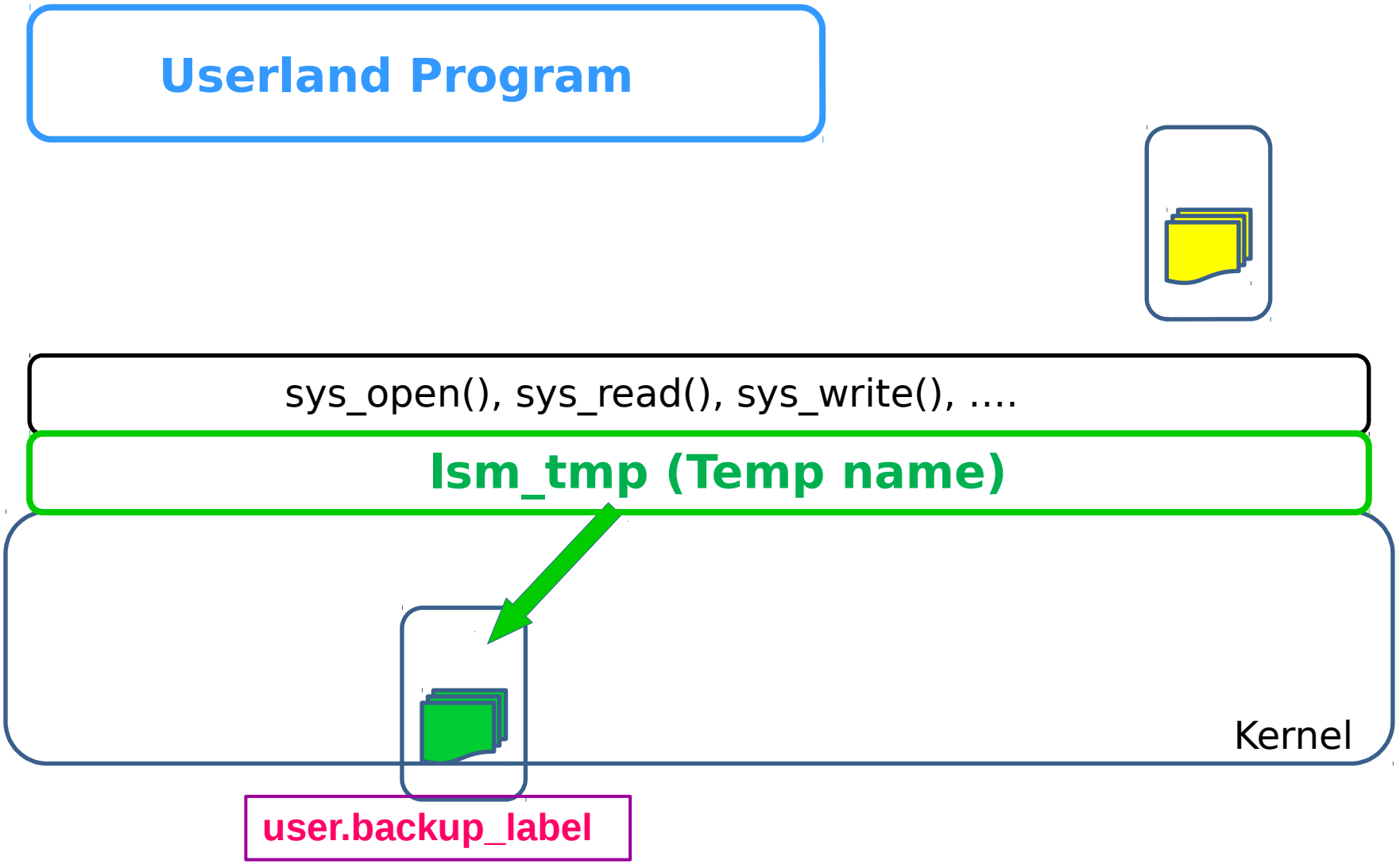
Labeled Case



Labeled Case



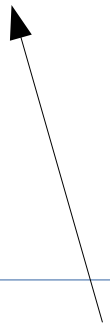
Labeled Case



LSM Hooks...

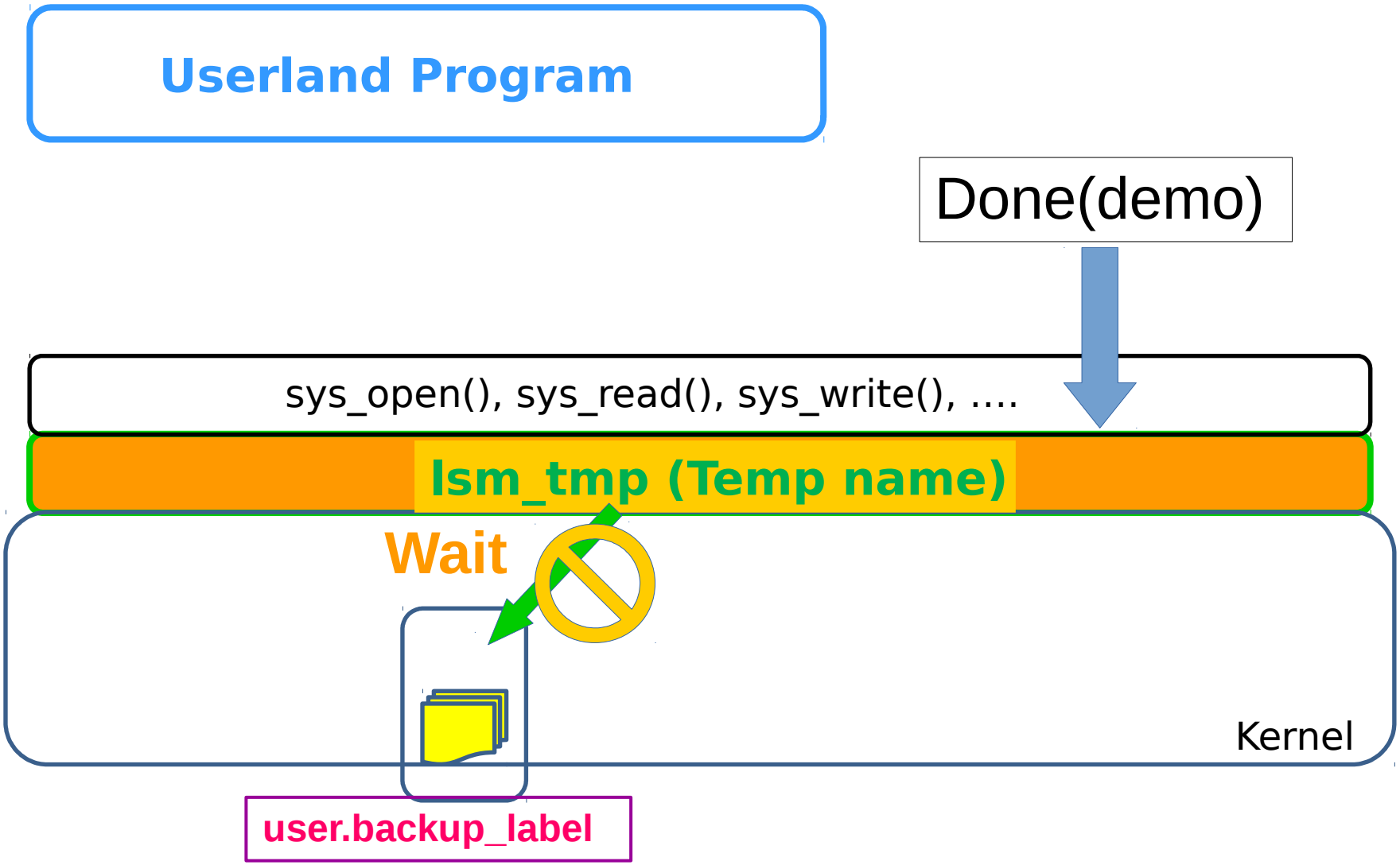
More than 150 LSM_HOOK

```
        LSM_HOOK_INIT(binder_transfer_file)
        LSM_HOOK_INIT(ptrace_access_check)
        LSM_HOOK_INIT(ptrace_traceme)
        LSM_HOOK_INIT(capget)
        LSM_HOOK_INIT(capset)
        LSM_HOOK_INIT(capable)

....
        LSM_HOOK_INIT(inode_follow_link)
        LSM_HOOK_INIT(inode_permission)
        LSM_HOOK_INIT(inode_setattr)

        LSM_HOOK_INIT(msg_msg_alloc_security)
        LSM_HOOK_INIT(msg_msg_free_security)
        LSM_HOOK_INIT(msg_queue_alloc_security)
```



Here

So far..



5. Demo

6. Conclusion

Conclusion

Not only for taking backup….

Conclusion

Not only for taking backup….

Module Usecase
ex.)

- Encrypt file
- AntiVirus
- Clustering
- Disaster Recovery

…. and so on.

ASAP, Publish Module as OSS.

Question?

Thanks!!