Automatically Backup Module against Ransomware Attack

2018/06/26

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

- Encrypt/Erase files on Network mounted Drive
- Encrypt/Erase files On USB Drive
- Encrypt/Erase files On Local Drive

Any files on connected Device:

- Encrypt
- Erase randomly

Time left before the price goes up: 41:18:14
Price for decryption: -0.05

© SIOS Technology, Inc. All rights Reserved.
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

0-day?
No Antivirus HW?
Workaround

Prepare for Ransomware infection.

File Server

Antivirus

0-day?
No Antivirus HW?
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).

<table>
<thead>
<tr>
<th>Client</th>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Choose target file</td>
<td>- Write Encrypted file</td>
<td>• Request to Delete file.</td>
</tr>
<tr>
<td></td>
<td>- Encrypt file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© SIOS Technology, Inc. All rights Reserved.
## File Server:

### How Ransomware work (from File Server).

<table>
<thead>
<tr>
<th>Client</th>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Choose target file</td>
<td>- Encrypt file</td>
<td>- Write Encrypted file</td>
<td>- Request to Delete file.</td>
</tr>
</tbody>
</table>

**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).

<table>
<thead>
<tr>
<th>Client</th>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Choose target file</td>
<td>- Write Encrypted file</td>
<td>• Request to Delete file.</td>
<td></td>
</tr>
<tr>
<td>- Encrypt file</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Server</th>
<th>Usual behavior as File Server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• sys_open/sys_read through smbd etc.</td>
<td>• Write some file on shared volume through smbd.</td>
</tr>
<tr>
<td>• sys_close to target file(?)</td>
<td>• Delete file on shared Volume through smbd.</td>
</tr>
</tbody>
</table>

© SIOS Technology, Inc. All rights Reserved.
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.**
So, How we can do?

<table>
<thead>
<tr>
<th>Client</th>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Choose target file</td>
<td>- Write Encrypted file</td>
<td>- Request to Delete file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Server</th>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Encrypt file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• sys_open/sys_read through smbd etc.</td>
<td>• Write unknown file on shared volume through smbd.</td>
<td>• Delete file on shared Volume through smbd.</td>
</tr>
<tr>
<td></td>
<td>• sys_close to target file(?)</td>
<td></td>
<td>Let’s think about here.</td>
</tr>
</tbody>
</table>
Linux+Samba

Config Recycle bin on Samba 3/4.

“vfs objects = recycle” to enable Recycle bin.

[Share]
comment = Public Stuff
path = /Share/
browsable = yes
writable = yes
printable = no

ufs objects = recycle
guest ok = yes
read only = no
recycle:repository = .recycle
recycle:keeptree = yes
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”?

<table>
<thead>
<tr>
<th>Encrypt File</th>
<th>Write Encrypted File</th>
<th>Delete Original file</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Choose target file</td>
<td>- Write Encrypted file</td>
<td>• Request to Delete file.</td>
</tr>
<tr>
<td>- Encrypt file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.

What about this situation?
From File Server:

Server: Can’t find User or Ransomware!!
From File Server: (Auto Backup)

PC/User

User

System Call

fopen()

fread()

fwrite()

sys_open()

sys_read()

sys_write()

Kernel

Backup

© SIOS Technology, Inc. All rights Reserved.
Create backup “before writing” by using Fanotify/inotify

Fanonotify is hooking “after writing”.

- > Can’t take backup “before writing”.

User

PC/User

Kernel

SystemCall

fopen()

fread()

fwrite()

sys_open()

sys_read()

sys_write()
Create backup "before writing" by using LSM.

System Call
- sys_open()
- sys_read()
- sys_write()

Kernel
- file_backup(LSM)

User
- fopen()
- fread()
- fwrite()
From File Server: (LSM solution)

PC/User

User

PC/User

Performance.....

Kernel

System Call

fopen() sys_open()

fread() sys_read()

fwrite() sys_write()

file_backup(LSM)

Backup
From File Server: (LSM solution)

Limit to backup “Labeled” file.

PC/User

User

Kernel

System Call

fopen()
fread()
fwrite()

sys_open()
sys_read()
sys_write()

file_backup(LSM)

Backup

© SIOS Technology, Inc. All rights Reserved.
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...

Userland Program

sys_open(), sys_read(), sys_write(), ....

Ism_tmp (Temp name)

Kernel
Usually...

Userland Program

sys_open(), sys_read(), sys_write(), ....

Ism_tmp (Temp name)

Kernel
Labeled Case

Userland Program

sys_open(), sys_read(), sys_write(), ....

Ismt_temp (Temp name)

Wait

user.backup_label

Kernel
Labeled Case

Userland Program

sys_open(), sys_read(), sys_write(), ….

Ism_tmp (Temp name)

Kernel

Copy

Wait

user.backup_label
Labeled Case

Userland Program

sys_open(), sys_read(), sys_write(), ....

Isms_tmp (Temp name)

Kernel

user.backup_label
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_msgAlloc_security)
LSM_HOOK_INIT(msg_msgFree_security)
LSM_HOOK_INIT(msg_queue_alloc_security)
So far..

Userland Program

Kernel

sys_open(), sys_read(), sys_write(), ....

Ism_tmp (Temp name)

Wait

user.backup_label

Done(demo)
5. Demo
6. Conclusion
Conclusion

Not only for taking backup....
Conclusion

Not only for taking backup....

Module Use case
ex.)

- Encrypt file
- AntiVirus
- Clustering
- Disaster Recovery

.... and so on.

ASAP, Publish Module as OSS.
Question?
Thanks!!