# Sentinel – Ransomware Detection in File Storage

Cornel Constantinescu and Sangeetha Seshadri
IBM Research, Almaden

## The Problem:

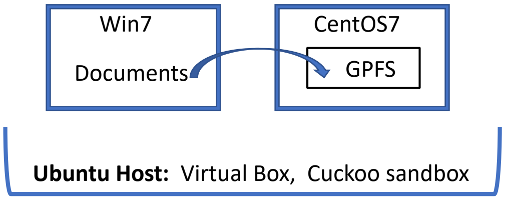
- Ransomware is malware that encrypts victims' files, extorting a ransom to be paid.
- In 2020 ransomware attacks increased by an alarming 715%, the global damage due to ransomware attacks, was $20 billions in 2020 and $11 billions in 2019
- Existing solutions to detect ransomware attacks proposed to run on individual Windows desktops/laptops use kernel modules or network sniffers to capture / analyze file system traffic; they are *intrusive and do not scale* to enterprise scale distributed file systems.
- This work focuses on zero-day (unknown) ransomware attack detection and recovery nonintrusive, real time, light weight and scalable to distributed file systems with many nodes.

## Our Method:

- Our ransomware detection and recovery is based on analyzing file access patterns, using available file system facilities, such as audit logs, filter drivers (Windows), stacked file systems (Unix), and/or light weight (live) events (Linux and IBM Spectrum Scale). Access is monitored at two levels of detail:
  - high detail - for ransomware detection, including information about individual read and write operations (offset, length, entropy), and
  - low detail - for recovery, including just enough information to keep track of files being updated or deleted by a process. Here, we also capture snapshot events and backup events. These enable us to make recovery more precise (find the most recent uncorrupted version of a file).

Periodic, high detail samples of file system activity are taken and analyzed to detect ransomware. Low-detail access log (audit log) is used to identify the files that might have been corrupted and need to be restored from backup or snapshot. The detection app can run either directly on the client or on the file server, monitoring the SMB (or NFS) client shares. All the live ransomware attacks from open-source repositories we tested (including WannaCry detailed below) were successfully detected.

## Test-bed for ransomware detection


Win7
Documents
CentOS7
GPFS
**Ubuntu Host:** Virtual Box, Cuckoo sandbox

## Example of WannaCry ransomware attack file access pattern:

| Event | processId | Inode # | pathName | fileSize | pathNewName | bytesRead | bytesWritten | minROff | maxROff | minWOff | maxWOff |
|---|---|---|---|---|---|---|---|---|---|---|---|
| open | 17704 | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| create | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 0 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| open | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 0 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| open | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 1001800 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| close | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 1001800 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| close | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 1001800 | NULL | 0 | 1001800 | 0 | 0 | 0 | 1001799 |
| rename | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRYT | 1001800 | 22_Zaharia.pdf.WNCRY | 0 | 0 | 0 | 0 | 0 | 0 |
| open | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRY | 1001800 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| close | 17704 | 59151 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf.WNCRY | 1001800 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| close | 17704 | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 1001508 | 0 | 0 | 1001507 | 0 | 0 |
| open | 17704 | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| close | 17704 | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 1001508 | 0 | 0 | 1001507 | 0 | 0 |
| unlink | 17704 | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |
| destroy | NA | 61191 | /gpfs/fs0/sharegpfs/DAY2/22_Zaharia.pdf | 1001508 | NULL | 0 | 0 | 0 | 0 | 0 | 0 |