CSE545 Software Security

Prof. Xiao

Group 12

Briana Rajan

Lushaank Kancherla

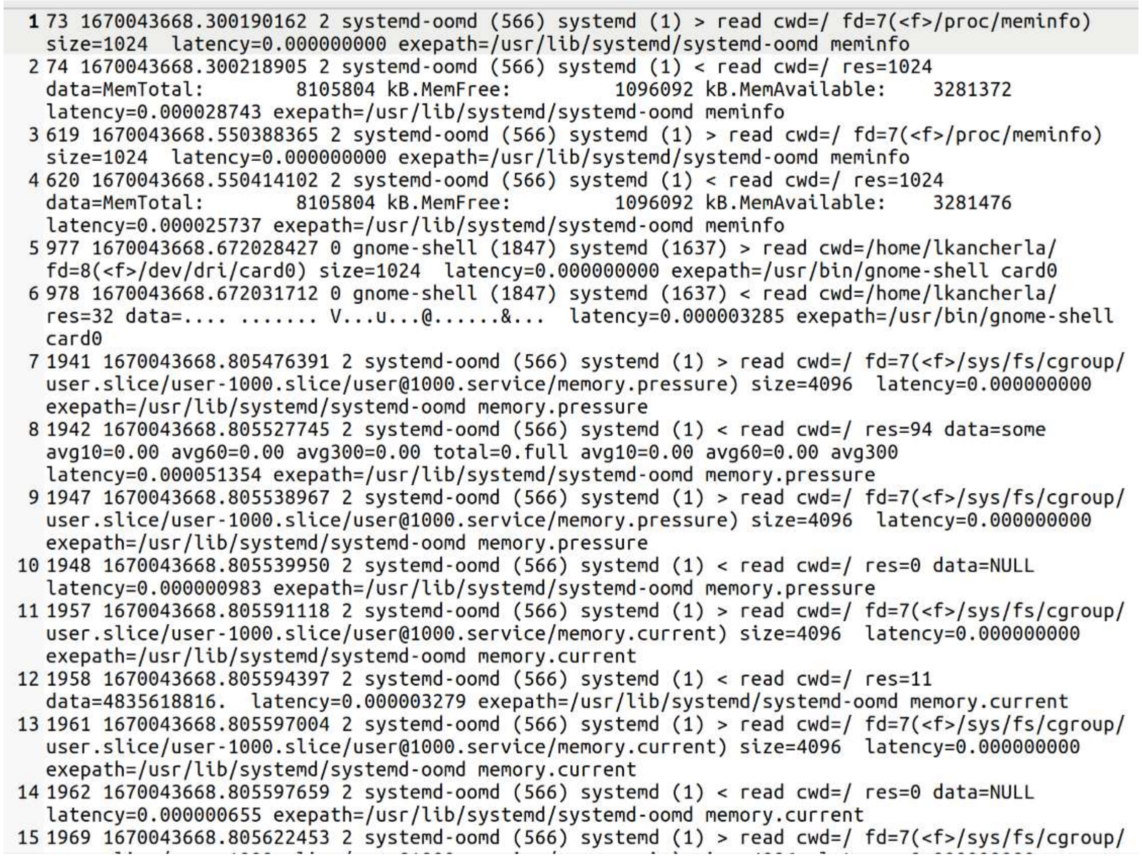
Teja Reddy Nagireddy

Write a parser for parsing sysdig output logs, and output the correctly
parsed information line by line in the report .

Sysdig code commands used:

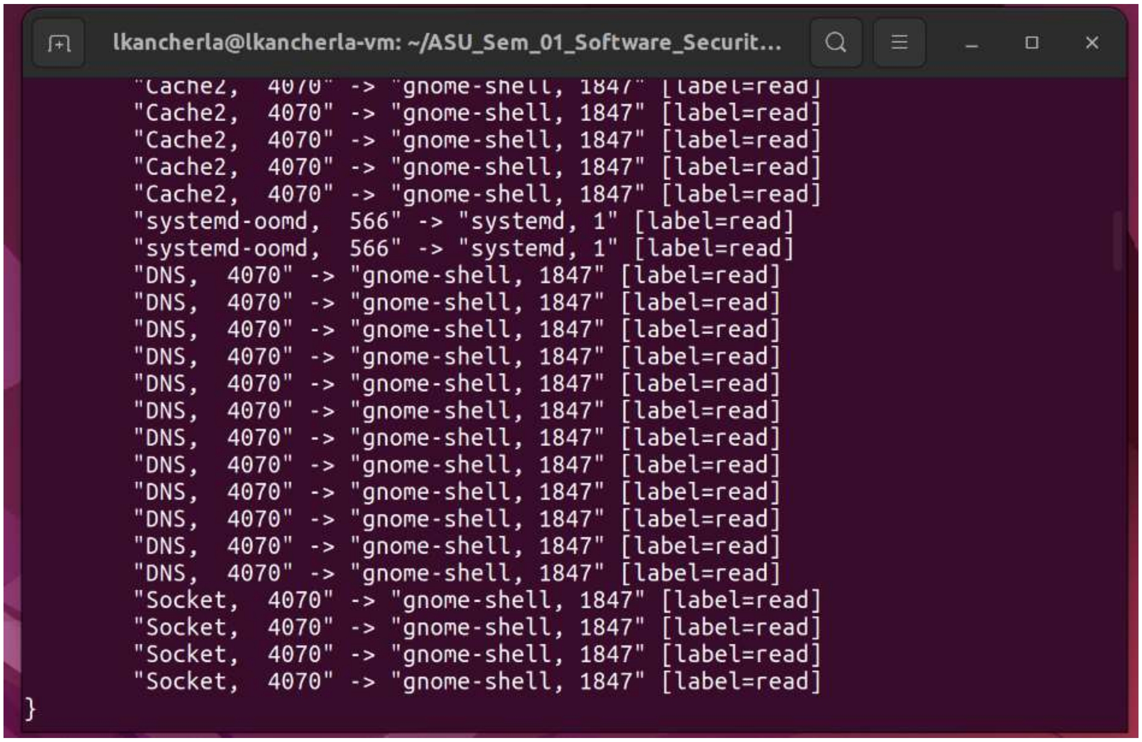
```
sudo sysdig -p "%evt.num %evt.rawtime.s.%evt.rawtime.ns %evt.cpu %proc.name (%proc.pid)
%proc.pname (%proc.ppid) %evt.dir %evt.type cwd=%proc.cwd %evt.args
latency=%evt.latency.s.%evt.latency.ns exepath=%proc.exepath %fd.filename" "(evt.type=read or
evt.type=readv or evt.type=write or evt.type=writev) and proc.name!=sysdig and proc.name!=tmux
and fd.type=file" -n 1000 > file8.txt
```

```
sudo sysdig -p "%evt.num %evt.rawtime.s.%evt.rawtime.ns %evt.cpu %proc.name (%proc.pid)
%proc.pname (%proc.ppid) %evt.dir %evt.type cwd=%proc.cwd %evt.args latency=%evt.latency
exepath=%proc.exepath %fd.filename %fd.cip %fd.sip %fd.cip.name %fd.sip.name"(evt.type=read or
evt.type=readv or evt.type=write or evt.type=writev) and proc.name!=sysdig and proc.name!=tmux
and fd.type=file" -n 1000 > file7.txt
```



```
73 1670043668.300190162 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/proc/meminfo)
size=1024  latency=0.000000000 exepath=/usr/lib/systemd/systemd-oomd meminfo
74 1670043668.300218905 2 systemd-oomd (566) systemd (1) < read cwd=/ res=1024
data=MemTotal:        8105804 kB.MemFree:         1096092 kB.MemAvailable:    3281372
latency=0.000028743 exepath=/usr/lib/systemd/systemd-oomd meminfo
619 1670043668.550388365 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/proc/meminfo)
size=1024  latency=0.000000000 exepath=/usr/lib/systemd/systemd-oomd meminfo
620 1670043668.550414102 2 systemd-oomd (566) systemd (1) < read cwd=/ res=1024
data=MemTotal:        8105804 kB.MemFree:         1096092 kB.MemAvailable:    3281476
latency=0.000025737 exepath=/usr/lib/systemd/systemd-oomd meminfo
977 1670043668.672028427 0 gnome-shell (1847) systemd (1637) > read cwd=/home/lkancherla/
fd=8(<f>/dev/dri/card0) size=1024  latency=0.000000000 exepath=/usr/bin/gnome-shell card0
978 1670043668.672031712 0 gnome-shell (1847) systemd (1637) < read cwd=/home/lkancherla/
res=32 data=.... ....... V...u...@......&...  latency=0.000003285 exepath=/usr/bin/gnome-shell
card0
1941 1670043668.805476391 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/sys/fs/cgroup/
user.slice/user-1000.slice/user@1000.service/memory.pressure) size=4096  latency=0.000000000
exepath=/usr/lib/systemd/systemd-oomd memory.pressure
1942 1670043668.805527745 2 systemd-oomd (566) systemd (1) < read cwd=/ res=94 data=some
avg10=0.00 avg60=0.00 avg300=0.00 total=0.full avg10=0.00 avg60=0.00 avg300
latency=0.000051354 exepath=/usr/lib/systemd/systemd-oomd memory.pressure
1947 1670043668.805538967 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/sys/fs/cgroup/
user.slice/user-1000.slice/user@1000.service/memory.pressure) size=4096  latency=0.000000000
exepath=/usr/lib/systemd/systemd-oomd memory.pressure
1948 1670043668.805539950 2 systemd-oomd (566) systemd (1) < read cwd=/ res=0 data=NULL
latency=0.000000983 exepath=/usr/lib/systemd/systemd-oomd memory.pressure
1957 1670043668.805591118 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/sys/fs/cgroup/
user.slice/user-1000.slice/user@1000.service/memory.current) size=4096  latency=0.000000000
exepath=/usr/lib/systemd/systemd-oomd memory.current
1958 1670043668.805594397 2 systemd-oomd (566) systemd (1) < read cwd=/ res=11
data=4835618816.  latency=0.000003279 exepath=/usr/lib/systemd/systemd-oomd memory.current
1961 1670043668.805597004 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/sys/fs/cgroup/
user.slice/user-1000.slice/user@1000.service/memory.current) size=4096  latency=0.000000000
exepath=/usr/lib/systemd/systemd-oomd memory.current
1962 1670043668.805597659 2 systemd-oomd (566) systemd (1) < read cwd=/ res=0 data=NULL
latency=0.000000655 exepath=/usr/lib/systemd/systemd-oomd memory.current
1969 1670043668.805622453 2 systemd-oomd (566) systemd (1) > read cwd=/ fd=7(<f>/sys/fs/cgroup/
```

The above image shows values of different fields from the log entries extracted by the parser.

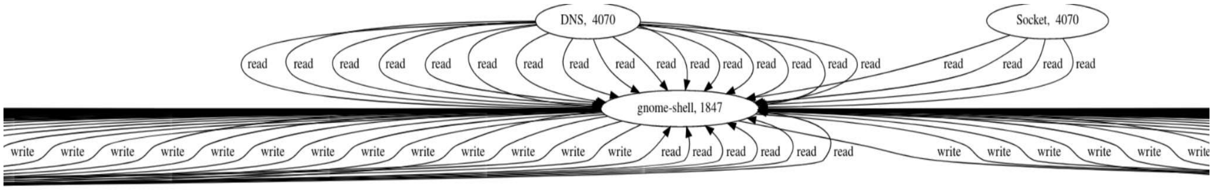


```
        "Cache2,  4070" -> "gnome-shell, 1847" [label=read]
        "Cache2,  4070" -> "gnome-shell, 1847" [label=read]
        "Cache2,  4070" -> "gnome-shell, 1847" [label=read]
        "Cache2,  4070" -> "gnome-shell, 1847" [label=read]
        "Cache2,  4070" -> "gnome-shell, 1847" [label=read]
        "systemd-oomd,  566" -> "systemd, 1" [label=read]
        "systemd-oomd,  566" -> "systemd, 1" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "DNS,  4070" -> "gnome-shell, 1847" [label=read]
        "Socket,  4070" -> "gnome-shell, 1847" [label=read]
        "Socket,  4070" -> "gnome-shell, 1847" [label=read]
        "Socket,  4070" -> "gnome-shell, 1847" [label=read]
        "Socket,  4070" -> "gnome-shell, 1847" [label=read]
}
```

The above shot shows the tuples that we created using the sysdig output described above. It shows the tuple (concatenated) consisting of process ID and process name in index 0, the event type shown in index 1, and the event action/arguments shown in index 2.

Project 2 Part 2

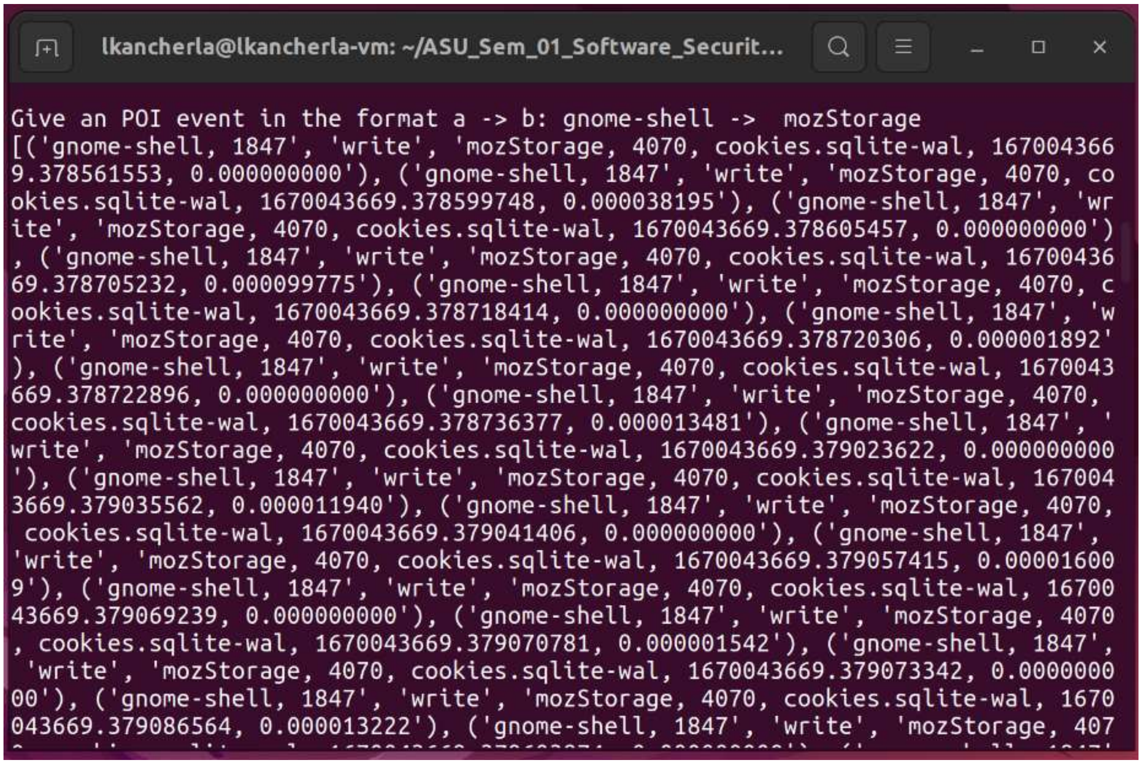
Using these events stored as with the tuple format, we were able to construct the graph using the Graphviz library.



Interpretation of the graph: The tuples constructed in the question 1 are connected via matching the entities.

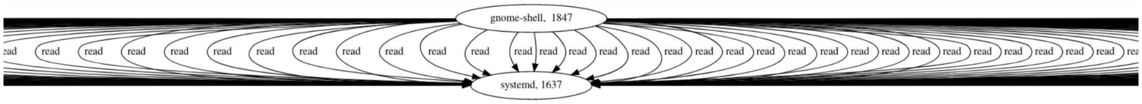
Project 2 Part 3

In section 3, we backtracked via the equal graphs so as to detect time primarily based events and the interplay of a point of interest. The backtrack algorithm carried out begins off locating the given point of interest within the graph on the latest point available. As soon as that is located, it'll discover a corresponding edge where the starting place of the POI fits the destination of some other edge within the graph that occurred before this event. It's going to filter out thru till it finds the earlier starting point and repeat this procedure till no matching events are discovered.



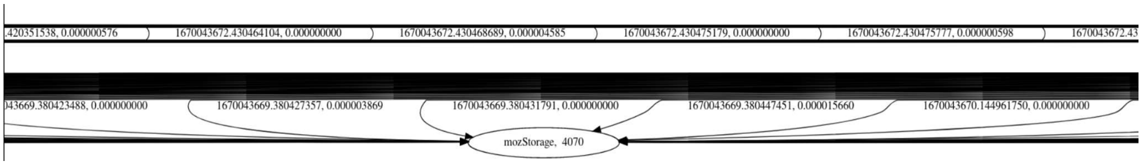
```
Give an POI event in the format a -> b: gnome-shell ->  mozStorage
[('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 167004366
9.378561553, 0.000000000'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, co
okies.sqlite-wal, 1670043669.378599748, 0.000038195'), ('gnome-shell, 1847', 'wr
ite', 'mozStorage, 4070, cookies.sqlite-wal, 1670043669.378605457, 0.000000000')
, ('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 16700436
69.378705232, 0.000099775'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, c
ookies.sqlite-wal, 1670043669.378718414, 0.000000000'), ('gnome-shell, 1847', 'w
rite', 'mozStorage, 4070, cookies.sqlite-wal, 1670043669.378720306, 0.000001892'
), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 1670043
669.378722896, 0.000000000'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070,
cookies.sqlite-wal, 1670043669.378736377, 0.000013481'), ('gnome-shell, 1847', '
write', 'mozStorage, 4070, cookies.sqlite-wal, 1670043669.379023622, 0.000000000
'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 167004
3669.379035562, 0.000011940'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070,
 cookies.sqlite-wal, 1670043669.379041406, 0.000000000'), ('gnome-shell, 1847',
'write', 'mozStorage, 4070, cookies.sqlite-wal, 1670043669.379057415, 0.00001600
9'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 16700
43669.379069239, 0.000000000'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070
, cookies.sqlite-wal, 1670043669.379070781, 0.000001542'), ('gnome-shell, 1847',
 'write', 'mozStorage, 4070, cookies.sqlite-wal, 1670043669.379073342, 0.0000000
00'), ('gnome-shell, 1847', 'write', 'mozStorage, 4070, cookies.sqlite-wal, 1670
043669.379086564, 0.000013222'), ('gnome-shell, 1847', 'write', 'mozStorage, 407
```

Point-of-interest (POI) event Graph:



The processes are shown in squares with the event items shown in ovals. the edges are shown, named by using their respective event identification. every event id has an edge connecting the process and event item. The direction of the edge is dictated by how the event call route, with read/readv/recvmesg all directing towards the event item and write/writev/sendmsg all directing trowards the process node.

Backward traceability graph:



Contribution:
The project was done in collaboration with equal contribution by all the team members.