Tales of a System Call Spelunker

An Introduction to System Call Analysis with Stratoshark

https://github.com/je-clark/sharkfest-25-us-stratoshark

Josh Clark | www.jeclark.net | #sf25us

Agenda

- Introduction
- What are system calls?
- How do Wireshark and Stratoshark capture data?
- What does a Stratoshark capture look like?
- Demo – HTTP Analysis
- BREAK
- Practice – HTTPS Analysis
- Demo – SCP Analysis
- Practice – SFTP Analysis
- Practice – Distributed System Troubleshooting

https://github.com/je-clark/sharkfest-25-us-stratoshark

Introduction

- B.S. and M.S. in Computer Engineering
  - Spent several semesters breaking Linux trying to optimize networking
- Principal Performance Engineer
  - Uses expertise in network protocols and Linux internals to identify bottlenecks in distributed systems
- Wireshark Certified Analyst

https://github.com/je-clark/sharkfest-25-us-stratoshark

What are System Calls?

Opening and reading a file in Python

```
With open(r’./file.txt’) as file:
    contents = read(file)
```

What are System Calls?

Opening and reading a file in C

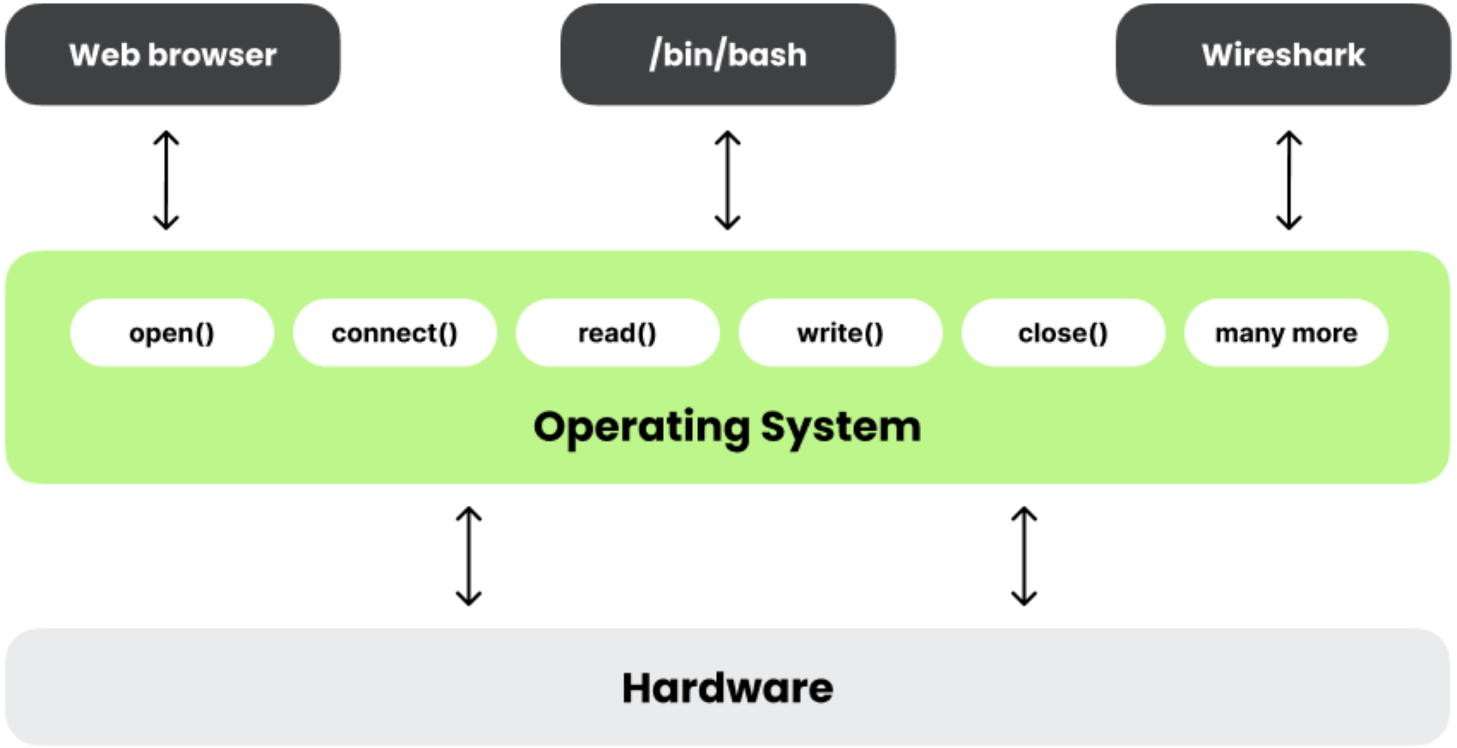
```
#include <stdio.h>
#include <stdlib.h>

int main() {
    FILE *file = fopen("./file.txt", "r");
    char *contents = malloc(filesize + 1);

    fread(contents, 1, filesize, file);
    contents[filesize] = '\0';

    fclose(file);
    return 0;
}
```

What are System Calls?

Playing Music in MATLAB

```
[y, Fs] = audioread('success.wav’);
sound(y, Fs);
```

What are System Calls?

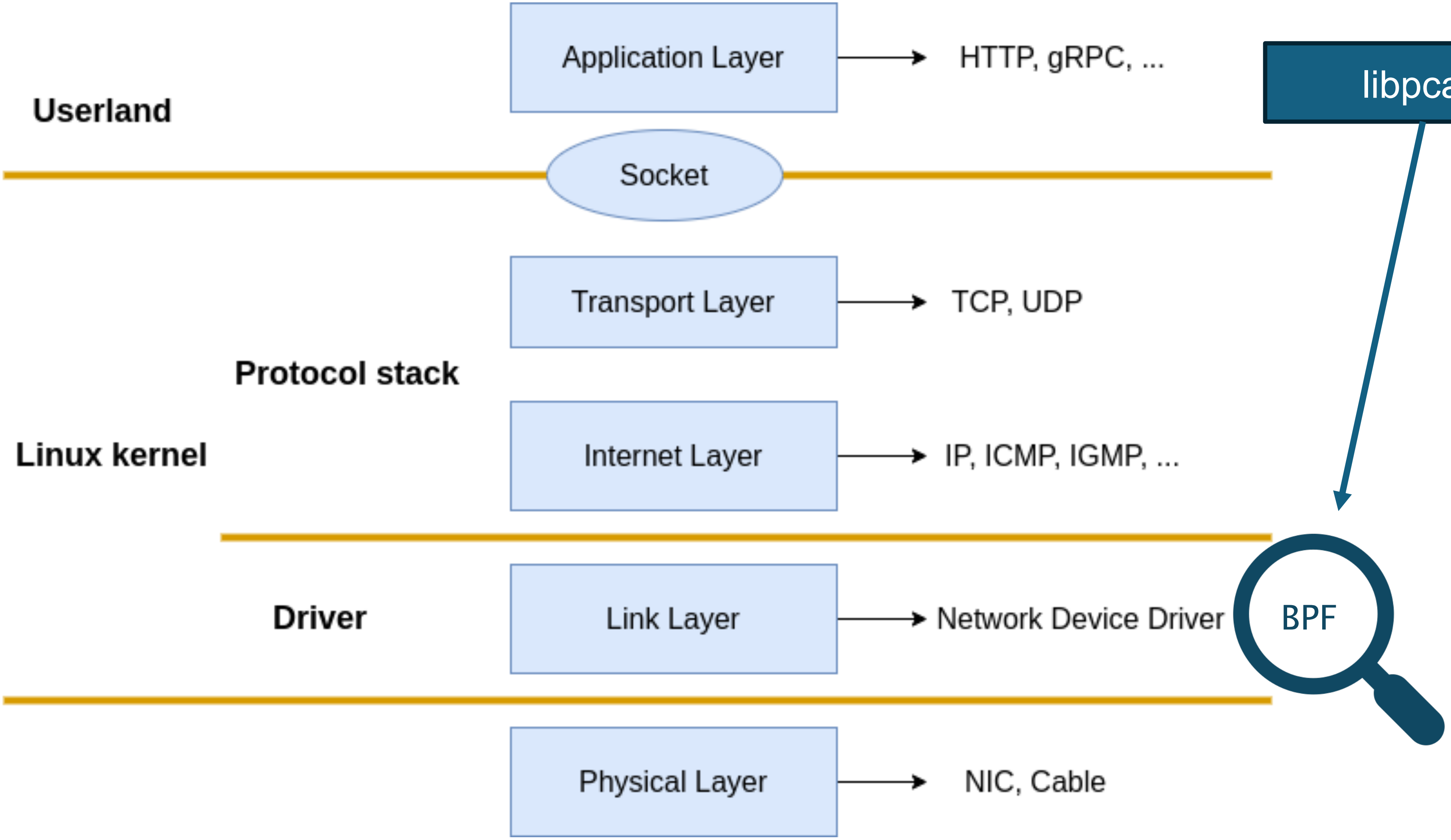


https://blog.wireshark.org/2025/01/those-arent-packets/

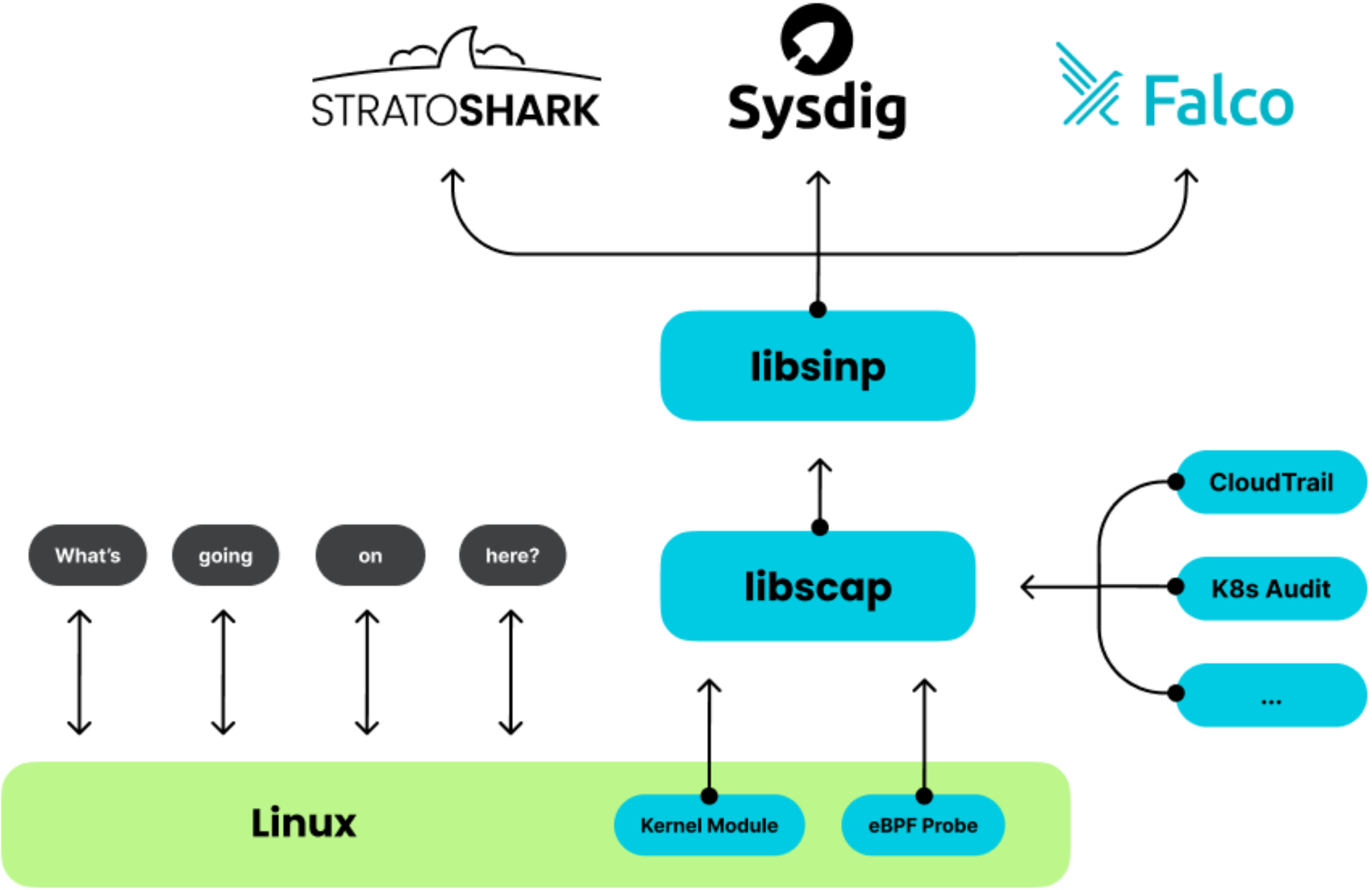
Where Can I Learn About System Calls?

https://filippo.io/linux-syscall-table/

How Does Wireshark Capture Packets?



How Can I Capture System Calls?



https://blog.wireshark.org/2025/01/those-arent-packets/

Getting Started with sysdig – Installing sysdig

https://github.com/draios/sysdig/wiki/How-to-Install-Sysdig-for-Linux

```
curl -s https://download.sysdig.com/stable/install-sysdig
| sudo bash
```

Getting Started with sysdig – Basic Capture

```
Sudo sysdig -w capture.scap
```

Getting Started with sysdig – Get more event data

```
Sudo sysdig -s 1000 -w capture.scap
```

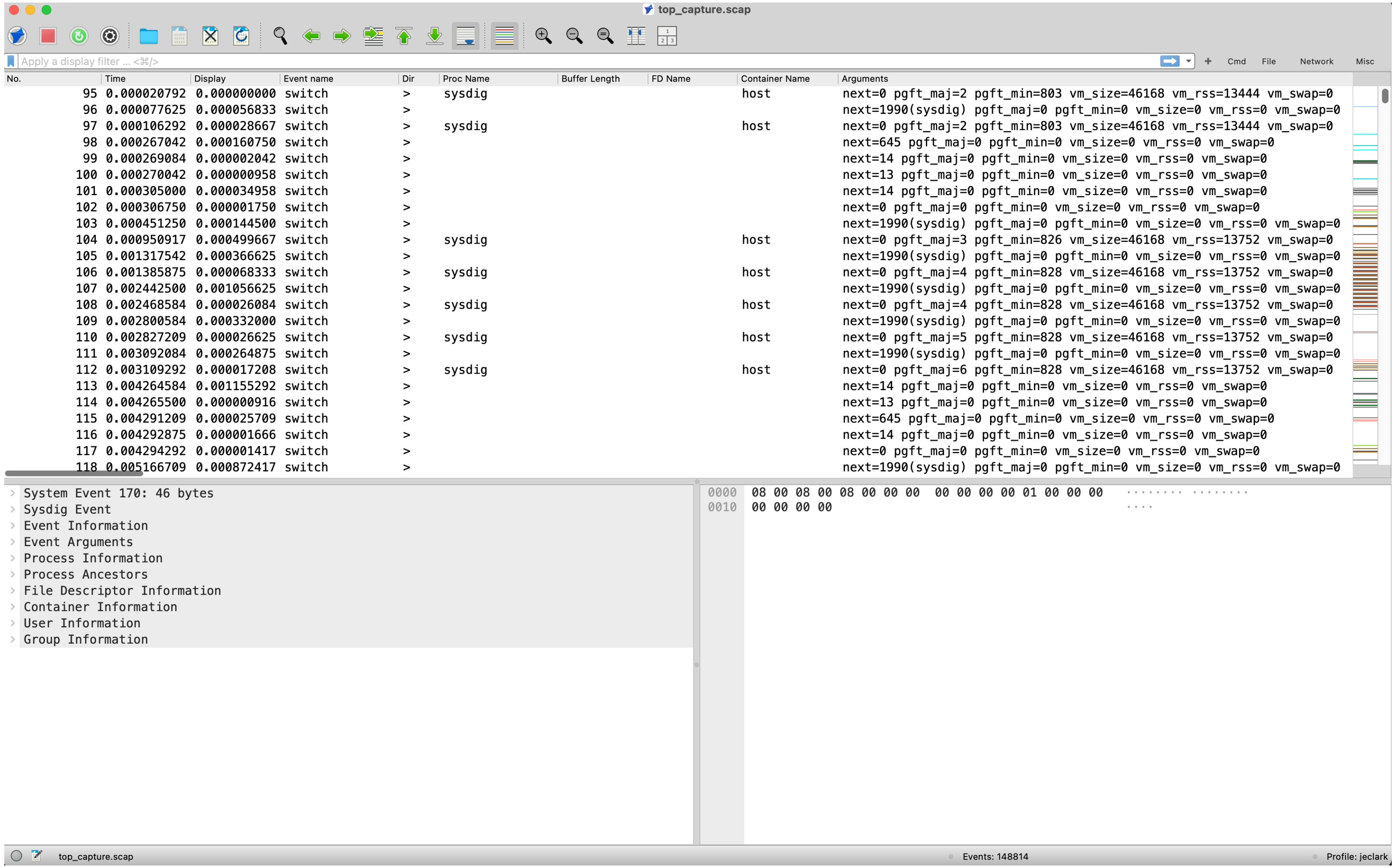
Getting Started with sysdig – Filter on Processes

```
Sudo sysdig -w capture.scap proc.name=nginx or
proc.name=python3
```

Getting Started with sysdig – Filter on Call Types

```
Sudo sysdig -w capture.scap evt.type=open
```

Stratoshark Interface



top_capture.scap

Apply a display filter ... <⌘/>

No.	Time	Display	Event name	Dir	Proc Name	Buffer Length	FD Name	Container Name	Arguments
95	0.000020792	0.000000000	switch	>	sysdig			host	next=0 pgft_maj=2 pgft_min=803 vm_size=46168 vm_rss=13444 vm_swap=0
96	0.000077625	0.000056833	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
97	0.000106292	0.000028667	switch	>	sysdig			host	next=0 pgft_maj=2 pgft_min=803 vm_size=46168 vm_rss=13444 vm_swap=0
98	0.000267042	0.000160750	switch	>					next=645 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
99	0.000269084	0.000002042	switch	>					next=14 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
100	0.000270042	0.000000958	switch	>					next=13 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
101	0.000305000	0.000034958	switch	>					next=14 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
102	0.000306750	0.000001750	switch	>					next=0 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
103	0.000451250	0.000144500	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
104	0.000950917	0.000499667	switch	>	sysdig			host	next=0 pgft_maj=3 pgft_min=826 vm_size=46168 vm_rss=13752 vm_swap=0
105	0.001317542	0.000366625	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
106	0.001385875	0.000068333	switch	>	sysdig			host	next=0 pgft_maj=4 pgft_min=828 vm_size=46168 vm_rss=13752 vm_swap=0
107	0.002442500	0.001056625	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
108	0.002468584	0.000026084	switch	>	sysdig			host	next=0 pgft_maj=4 pgft_min=828 vm_size=46168 vm_rss=13752 vm_swap=0
109	0.002800584	0.000332000	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
110	0.002827209	0.000026625	switch	>	sysdig			host	next=0 pgft_maj=5 pgft_min=828 vm_size=46168 vm_rss=13752 vm_swap=0
111	0.003092084	0.000264875	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
112	0.003109292	0.000017208	switch	>	sysdig			host	next=0 pgft_maj=6 pgft_min=828 vm_size=46168 vm_rss=13752 vm_swap=0
113	0.004264584	0.001155292	switch	>					next=14 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
114	0.004265500	0.000000916	switch	>					next=13 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
115	0.004291209	0.000025709	switch	>					next=645 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
116	0.004292875	0.000001666	switch	>					next=14 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
117	0.004294292	0.000001417	switch	>					next=0 pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0
118	0.005166709	0.000872417	switch	>					next=1990(sysdig) pgft_maj=0 pgft_min=0 vm_size=0 vm_rss=0 vm_swap=0

```
> System Event 170: 46 bytes
> Sysdig Event
> Event Information
> Event Arguments
> Process Information
> Process Ancestors
> File Descriptor Information
> Container Information
> User Information
> Group Information
```

```
0000  08 00 08 00 08 00 00 00  00 00 00 00 01 00 00 00   ........ ........
0010  00 00 00 00                                        ....
```

top_capture.scap

Events: 148814

Profile: jeclark

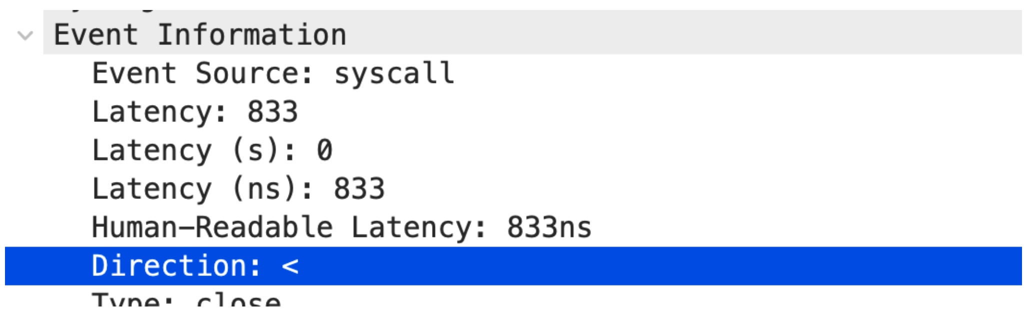
Event list

Event details

Event bytes

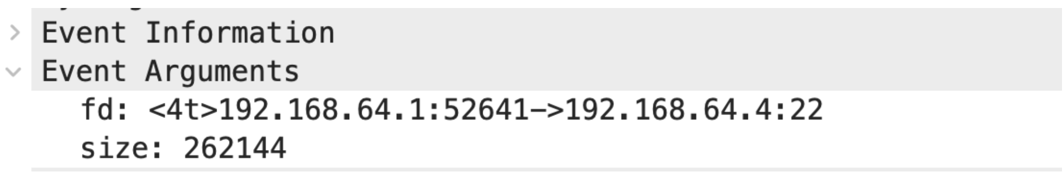
Stratoshark Interface – New Fields

Direction:
> Enters a syscall
< Exits a syscall

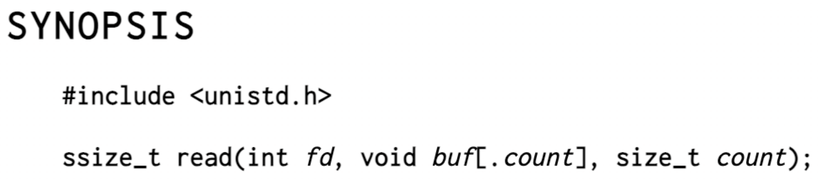


```
⌄ Event Information
    Event Source: syscall
    Latency: 833
    Latency (s): 0
    Latency (ns): 833
    Human-Readable Latency: 833ns
    Direction: <
    Type: close
```

Arguments:
Syscall-specific
Discoverable in reference docs



```
› Event Information
⌄ Event Arguments
    fd: <4t>192.168.64.1:52641->192.168.64.4:22
    size: 262144
```



```
SYNOPSIS

    #include <unistd.h>

    ssize_t read(int fd, void buf[.count], size_t count);
```

Capture 1

- 1_top.scap

Capture 1 Findings

- Use the ‘in {}’ filter syntax as a first choice
- execve() returns to the new process, but is called by the parent process
- /proc is a directory where the kernel writes process and system statistics
- getdents64() writes information about contents of a directory to a memory map created with the mmap() call
- The terminal window uses the file descriptor /dev/pts/0

Capture 2

- 2_http_packets.pcapng
- 2_http_syscalls.scap

Capture 3

- 3_https_packets.pcapng
- 3_https_syscalls.scap
- 3_https_worksheet.pdf

Capture 4

- 4_scp_packets.pcapng
- 4_scp _syscalls.scap

Capture 5

- 5_sftp_packets.pcapng
- 5_sftp_syscalls.scap
- 5_sftp_worksheet.pdf

Distributed Systems Troubleshooting

- Distributed_system_worksheet.pdf
  - If you feel confident, try using only page 1
  - If you want a guided tour, use the entire worksheet
- Josh_web_store.har
- Web_packets.pcapng
- Web_syscalls.scap
- Nas_packets.pcapng
- Nas_syscalls.scap
- Db_packets.pcapng
- Db_syscalls.scap

Current Limitations

- Unable to see some inter-process and kernel communication
  - Statically linked libraries (e.g. nginx and openssl)
  - Shared memory locations (e.g. getdents64())
- sysdig missing features that other syscall tools include
  - Strace gives you more data for getdents64() than sysdig
- No Windows or MacOS support
  - Gitlab issue is open for Windows procmon support

Conclusion

- In this session, we
  - Learned what system calls are and how to capture them with sysdig
  - Learned the basic Stratoshark interface
  - Examined several common Linux applications in both Wireshark and Stratoshark
  - Found the bottlenecks in a distributed system using both Wireshark and Stratoshark

Feedback