

# Cache Side-Channel Attacks and the case of Rowhammer

**Daniel Gruss**  
**IAIK, Graz University of Technology**

April 28, 2016

1. Caches
2. Cache attacks
3. Cache Template Attacks
4. Rowhammer

# 1. Caches

## 2. Cache attacks

## 3. Cache Template Attacks

## 4. Rowhammer

# CPU Caches

- CPU speed increases
- Latency of DRAM is too high
- Problem: DRAM is a bottle neck

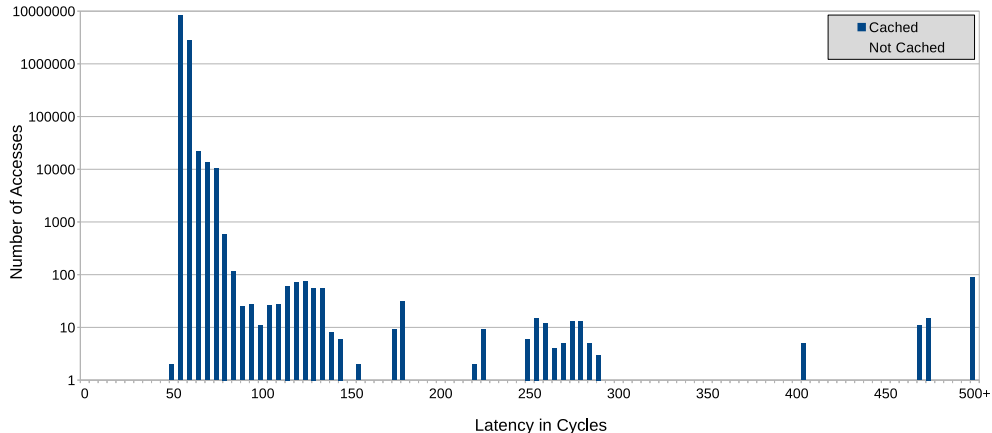
# CPU Caches

- CPU speed increases
- Latency of DRAM is too high
- Problem: DRAM is a bottle neck
- Solution: Caches - fast/small memory buffers

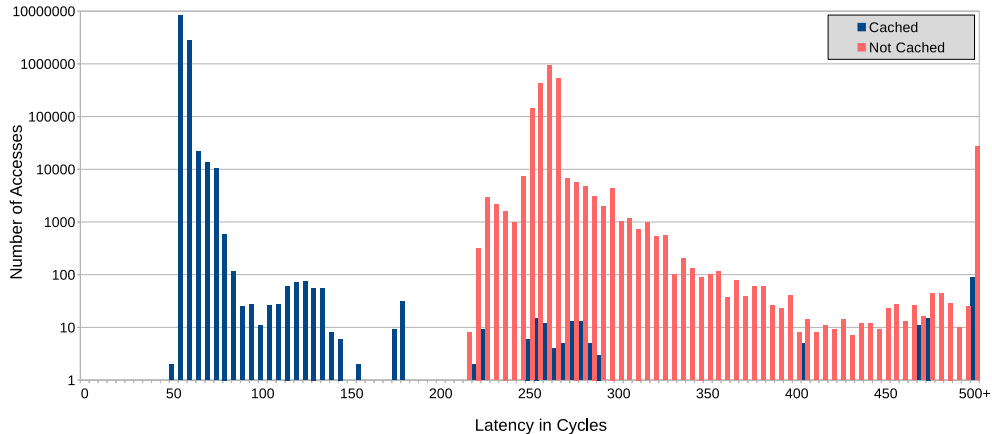
# Memory accesses are cached

- Every memory reference goes through the cache
- Transparent to OS and programs

# Memory Access Latency



# Memory Access Latency





# Directly mapped cache

Memory Address



# Directly mapped cache

Memory Address

|  |
|--|
|  |
|--|

Cache

|  |  |
|--|--|
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Directly mapped cache

Memory Address

|  |
|--|
|  |
|--|

Cache

| Tag | Data |
|-----|------|
|     |      |
|     |      |
|     |      |
|     |      |

# Directly mapped cache

Memory Address

|  |  |
|--|--|
|  |  |
|--|--|

Cache

| Tag | Data |
|-----|------|
|     |      |
|     |      |
|     |      |
|     |      |

# Directly mapped cache

Memory Address

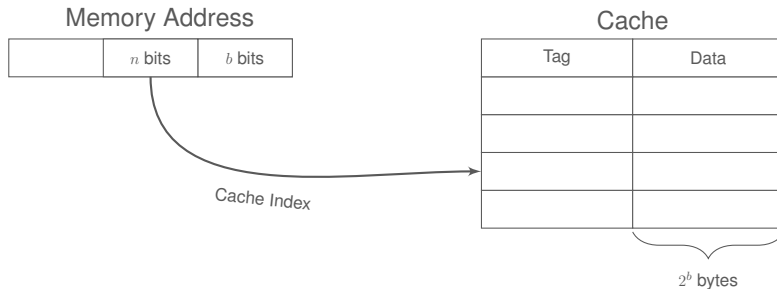
|  |          |
|--|----------|
|  | $b$ bits |
|--|----------|

Cache

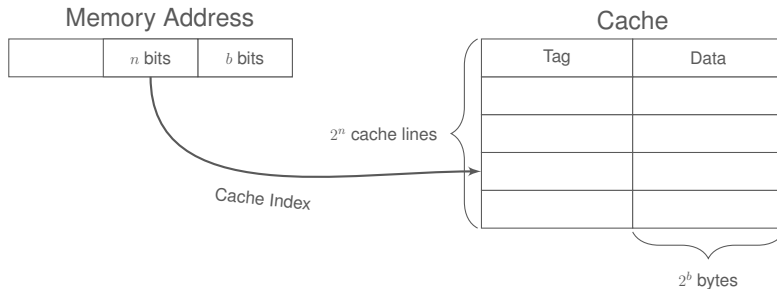
| Tag | Data |
|-----|------|
|     |      |
|     |      |
|     |      |
|     |      |

$2^b$  bytes

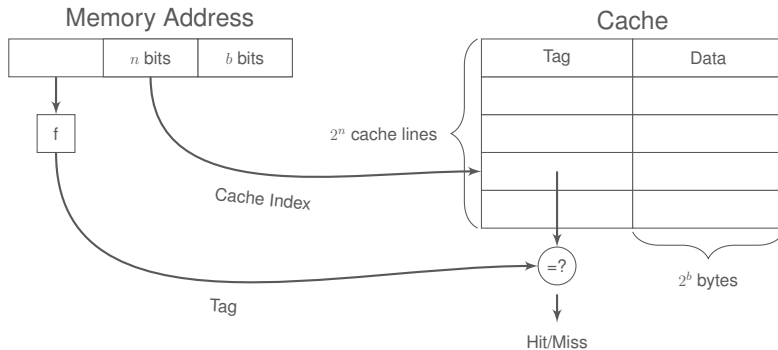
# Directly mapped cache



# Directly mapped cache

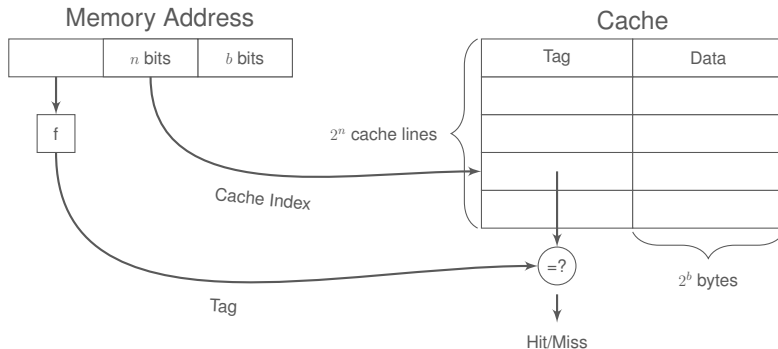


# Directly mapped cache



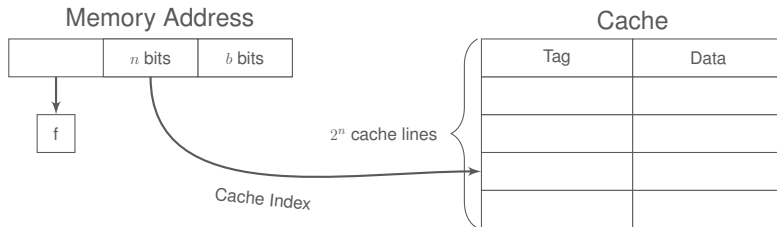


# Directly mapped cache

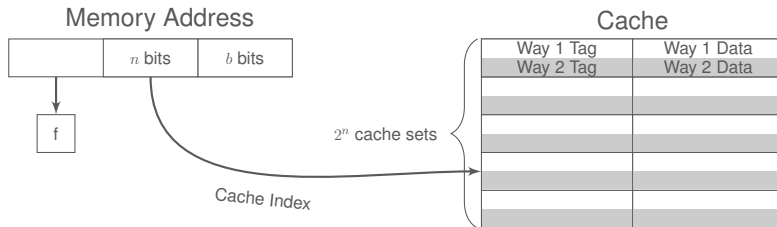


Problem: working on congruent addresses

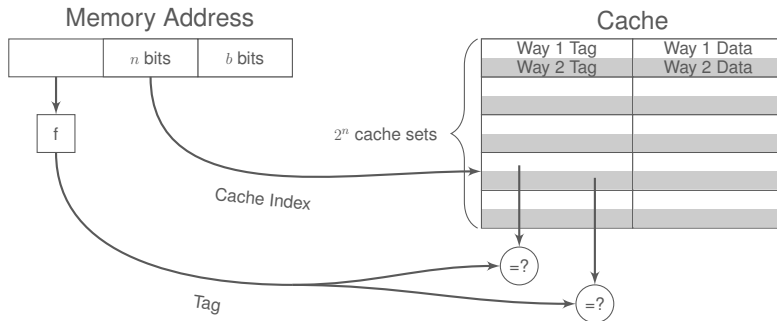
# 2-way set associativity



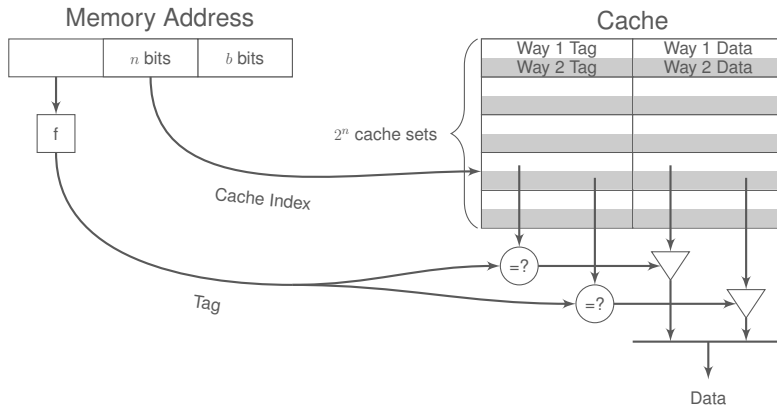
# 2-way set associativity



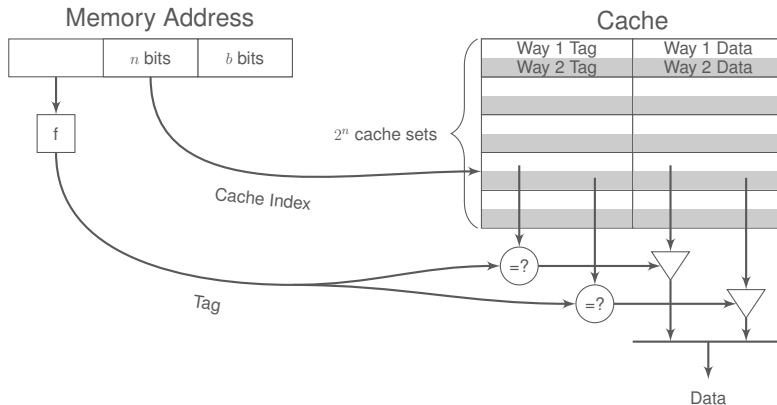
## 2-way set associativity



# 2-way set associativity



# 2-way set associativity



→ replacement policy is important

# Caches and Addresses

- Often use physical addresses
- Virtual-to-physical translation necessary

# TLB and Paging

- Paging: memory translated page-wise from virtual to physical
- TLB (translation lookaside buffer) caches virtual to physical mapping
- TLB has some latency



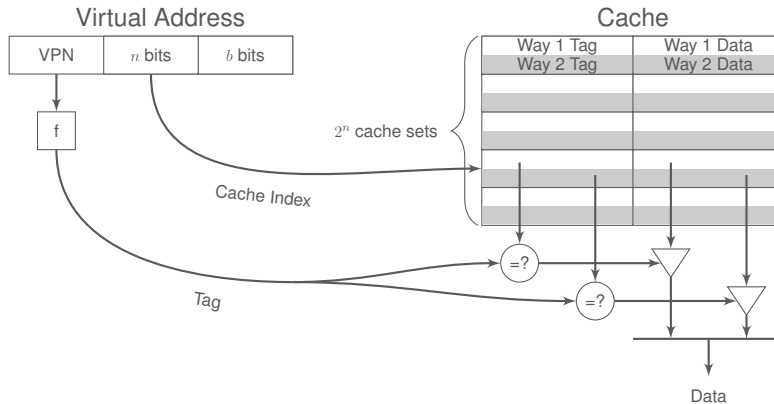
# TLB and Paging

- Paging: memory translated page-wise from virtual to physical
- TLB (translation lookaside buffer) caches virtual to physical mapping
- TLB has some latency
- Worst case for Cache: mapping not in TLB, need to load mapping from RAM
- Solution: Use virtual addresses instead of physical addresses

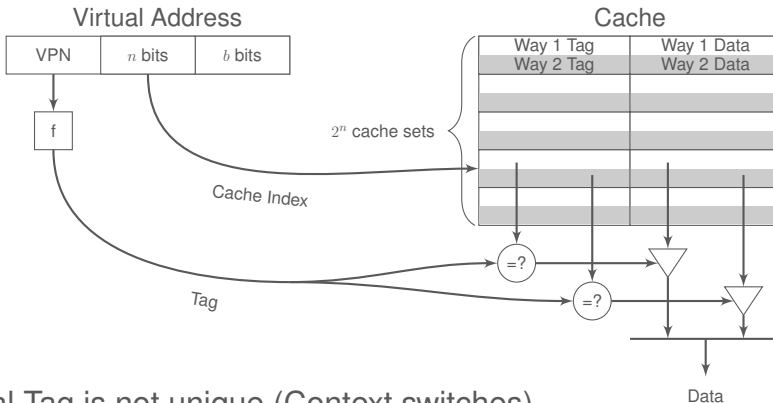
# Cache indexing methods

- VIVT: Virtually indexed, virtually tagged
- PIPT: Physically indexed, physically tagged
- PIVT: Physically indexed, virtually tagged
- VIPT: Virtually indexed, physically tagged

# VIVT

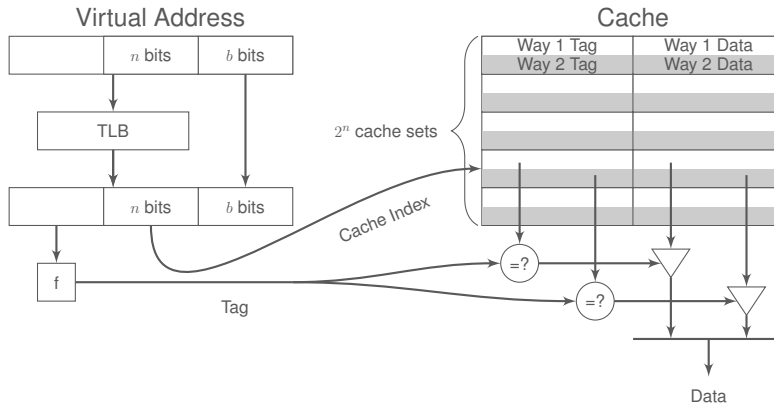


# VIVT

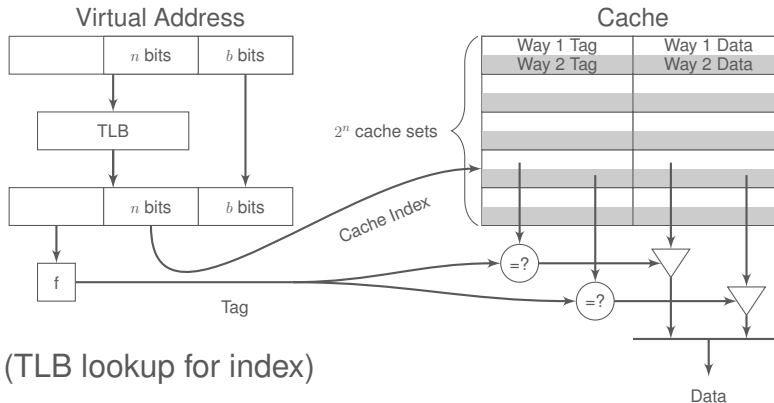


- Fast
- Virtual Tag is not unique (Context switches)
- Shared memory more than once in cache

# PIPT

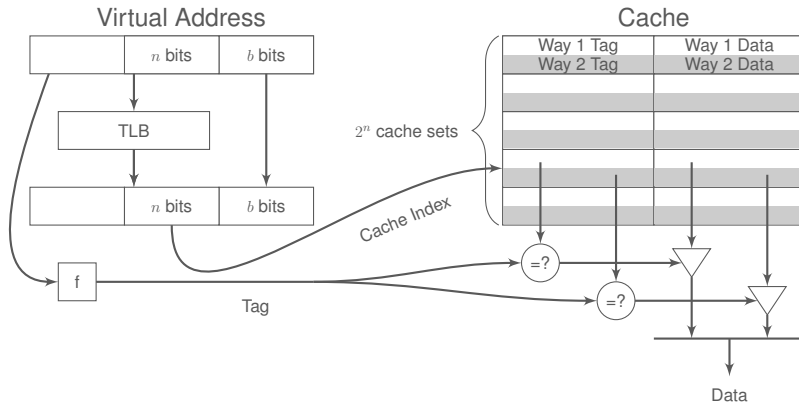


# PIPT

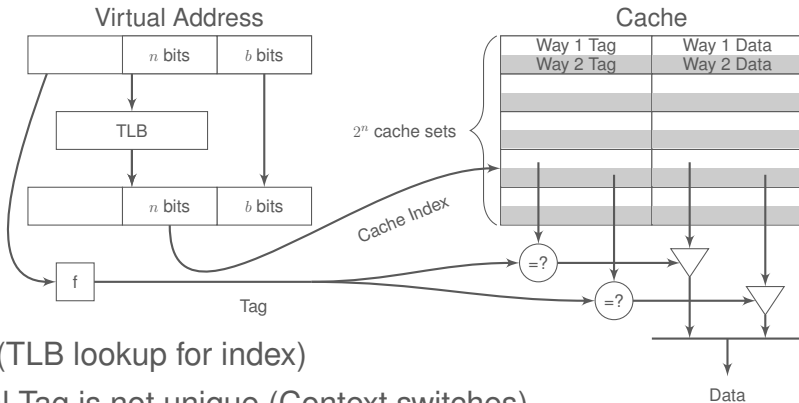


- Slow (TLB lookup for index)
- Shared memory only once in cache!

# (PIVT)



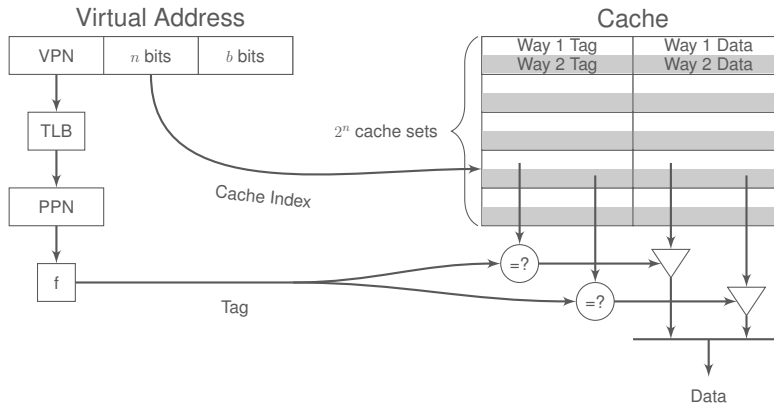
# (PIVT)



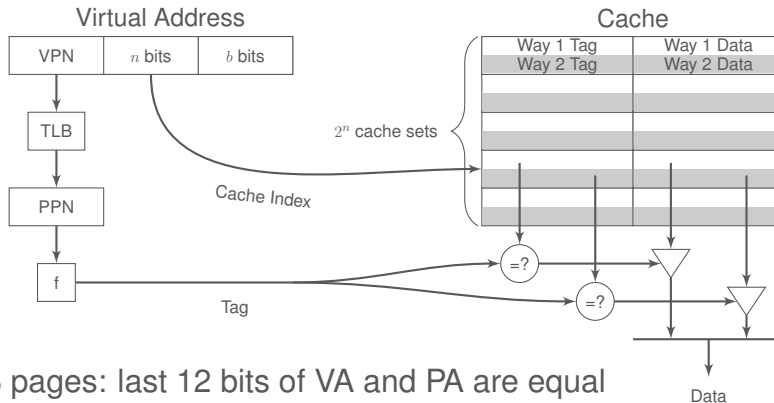
- Slow (TLB lookup for index)
- Virtual Tag is not unique (Context switches)
- Shared memory more than once in cache



# VIPT



# VIPT



- Fast
- 4 KiB pages: last 12 bits of VA and PA are equal
- Using more bits has disadvantages (like VIVT)

→ Cache size  $\leq$  # ways  $\cdot$  page size

# Remarks

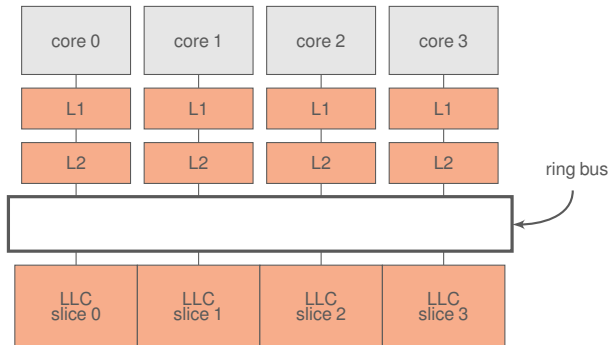
- L1 caches: VIVT or VIPT
- L2/L3 caches: PIPT

# Remarks

- L1 caches: VIVT or VIPT
- L2/L3 caches: PIPT

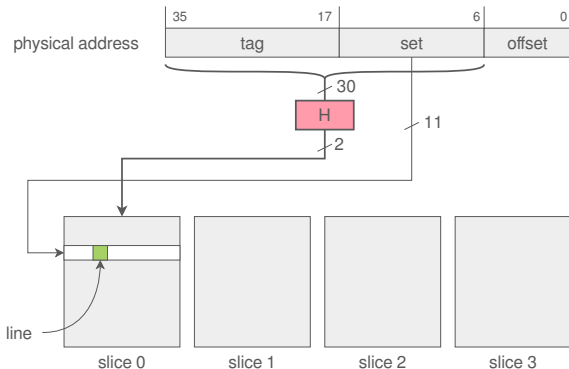
→ shared memory is shared in cache!

# Caches today



- L1 and L2 are private
  - last-level cache:
    - divided in **slices**
    - **shared** across cores
    - **inclusive**
- shared memory  
shared is in cache,  
across cores!

# Cache mapping



- Function  $H$  that maps slices is undocumented
- **Reverse-engineered** by [Hund et al., 2013, Maurice et al., 2015a, Inci et al., 2015, Yarom et al., 2015]

# Unprivileged cache maintainance

User programs can optimize cache usage:

- `prefetch`: suggest CPU to load data into cache
- `clflush`: throw out data from from all caches

... based on virtual addresses

1. Caches

2. Cache attacks

3. Cache Template Attacks

4. Rowhammer



# Cache attacks

- cache-based keylogging
- crypto key recovery
  - various implementations (AES, RSA, ECC, ...)
  - up to 97% of RSA key bits recovered after 1 encryption
- cross-VM, cross-core, even cross-CPU
- any architecture (Intel, AMD, ARM)

# Cross-core attacks?

- exploiting the **inclusive** property

# Cross-core attacks?

- exploiting the **inclusive** property
- last-level cache is a superset of L1 and L2

# Cross-core attacks?

- exploiting the **inclusive** property
- last-level cache is a superset of L1 and L2
- data evicted from last-level cache → evicted from L1 and L2

# Cross-core attacks?

- exploiting the **inclusive** property
- last-level cache is a superset of L1 and L2
- data evicted from last-level cache → evicted from L1 and L2
- a core can evict lines in the private L1 cache of another core

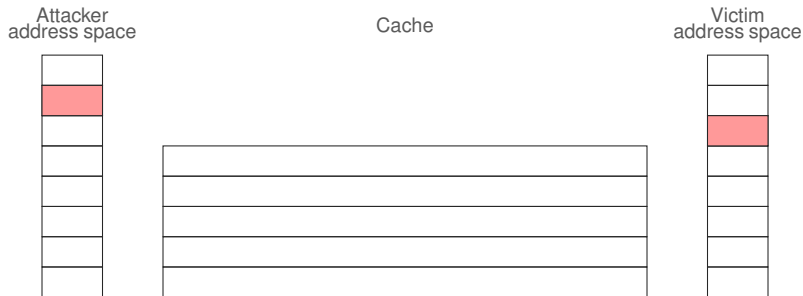
# Access-driven attacks

Attacker monitors **its own activity** to determine cache lines or sets accessed by victim.



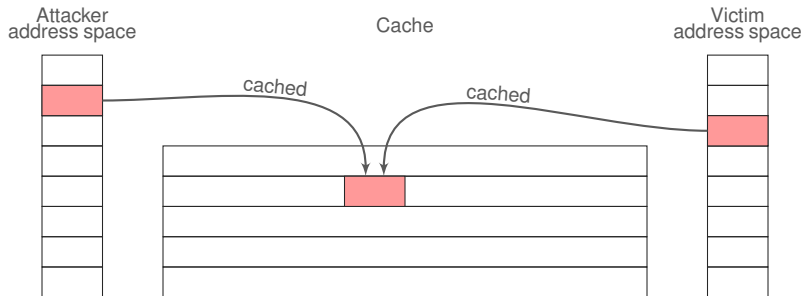
Same techniques for covert and side channels

# Flush+Reload



**step 0:** attacker maps shared library → shared memory, shared in cache

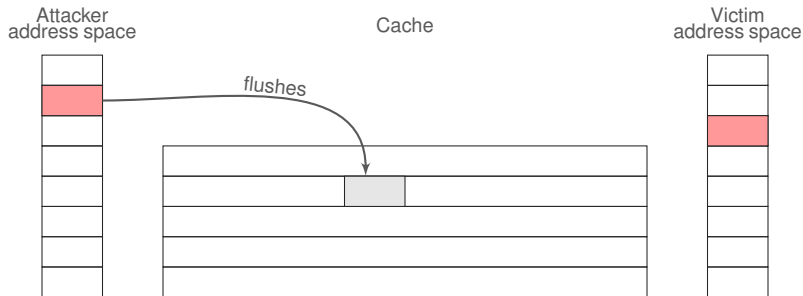
# Flush+Reload



**step 0:** attacker maps shared library → shared memory, shared in cache



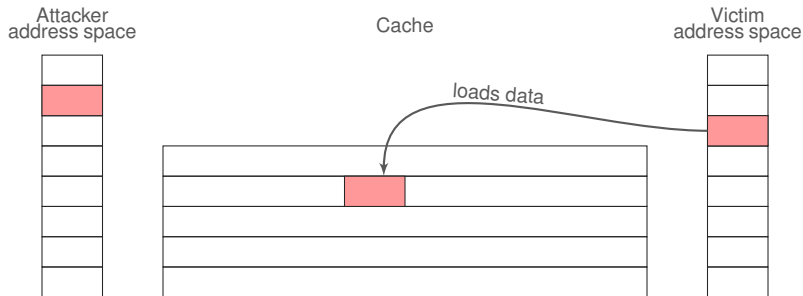
# Flush+Reload



**step 0:** attacker maps shared library → shared memory, shared in cache

**step 1:** attacker flushes the shared line

# Flush+Reload

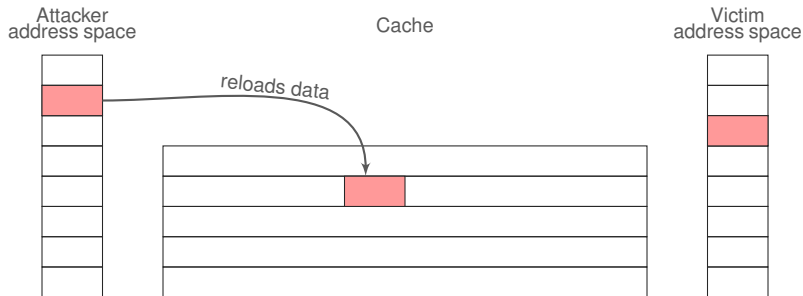


**step 0:** attacker maps shared library → shared memory, shared in cache

**step 1:** attacker flushes the shared line

**step 2:** victim loads data while performing encryption

# Flush+Reload



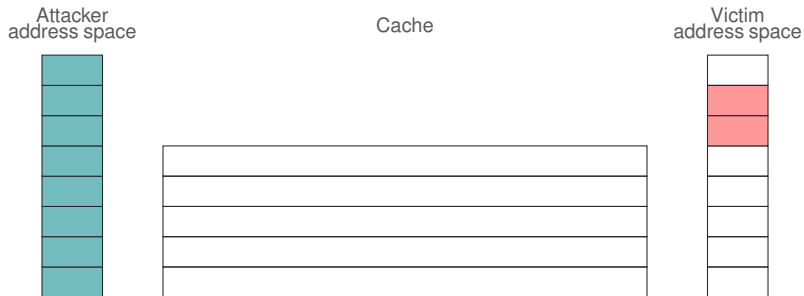
**step 0:** attacker maps shared library → shared memory, shared in cache

**step 1:** attacker flushes the shared line

**step 2:** victim loads data while performing encryption

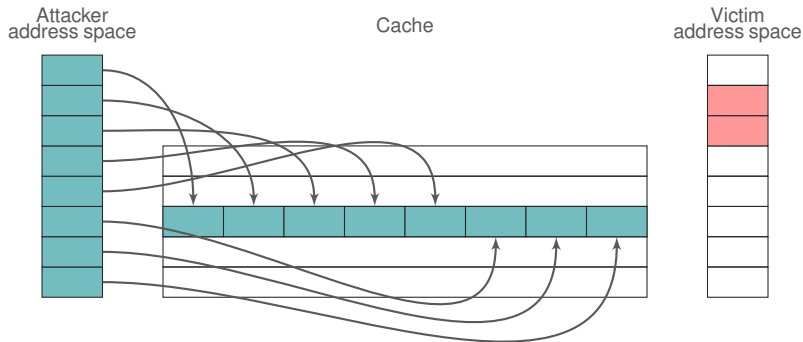
**step 3:** attacker reloads data → fast access if the victim loaded the line

# Prime+Probe



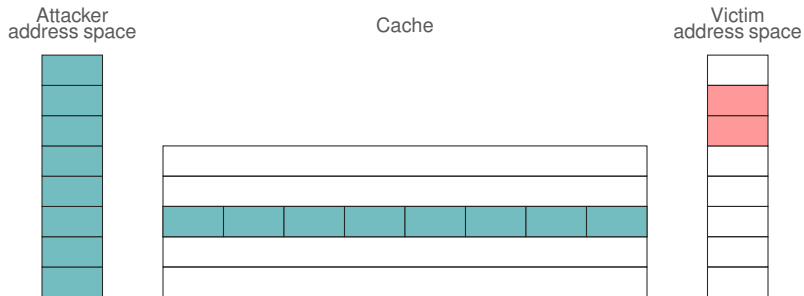
**step 0:** attacker fills the cache (prime)

# Prime+Probe



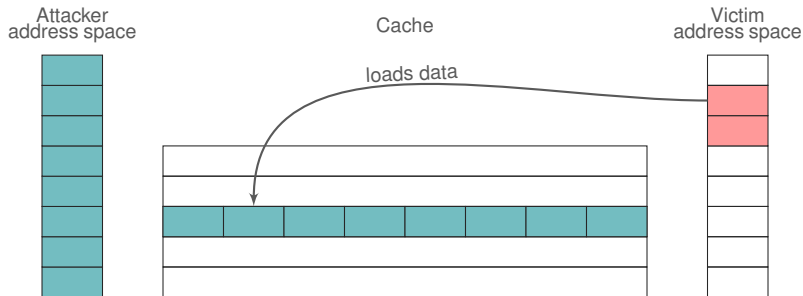
**step 0:** attacker fills the cache (prime)

# Prime+Probe



**step 0:** attacker fills the cache (prime)

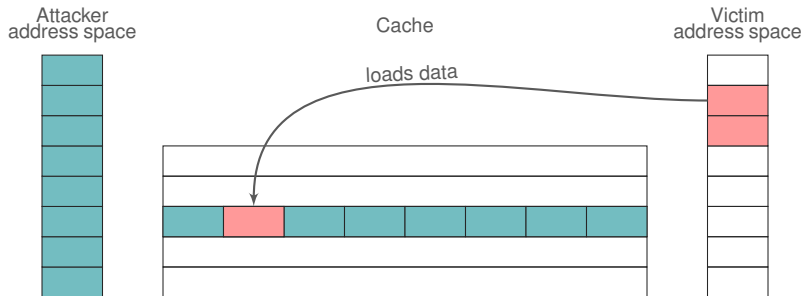
# Prime+Probe



**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

# Prime+Probe

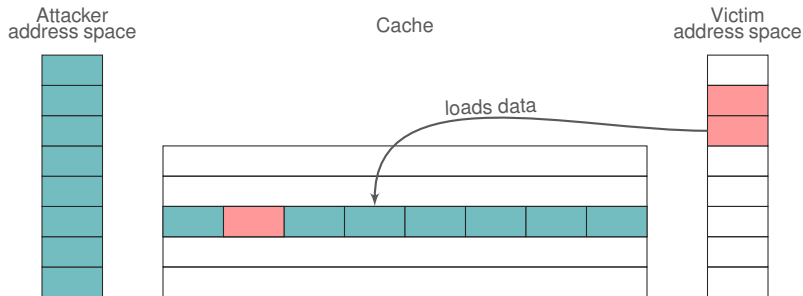


**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption



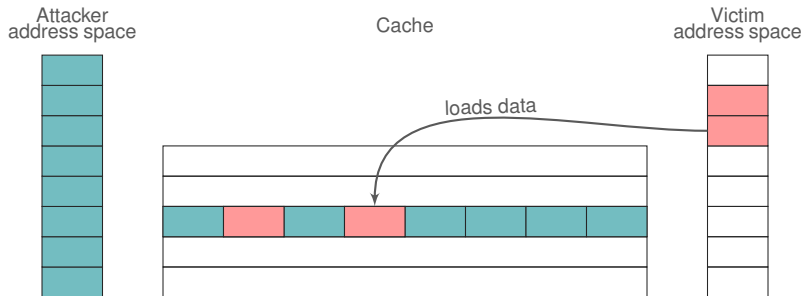
# Prime+Probe



**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

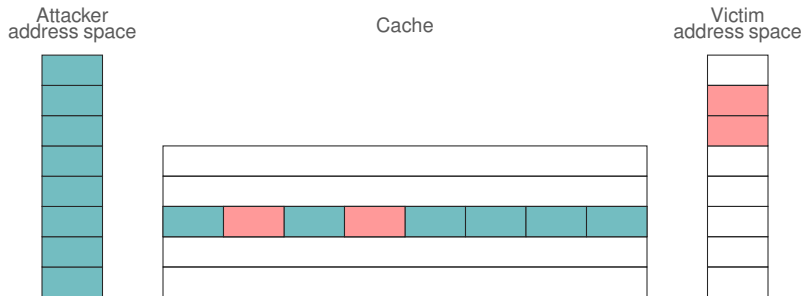
# Prime+Probe



**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

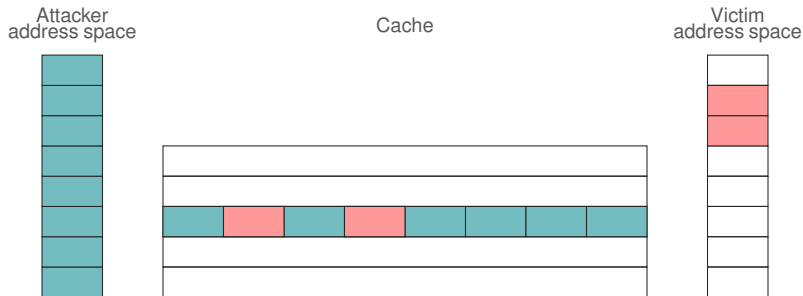
# Prime+Probe



**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

# Prime+Probe

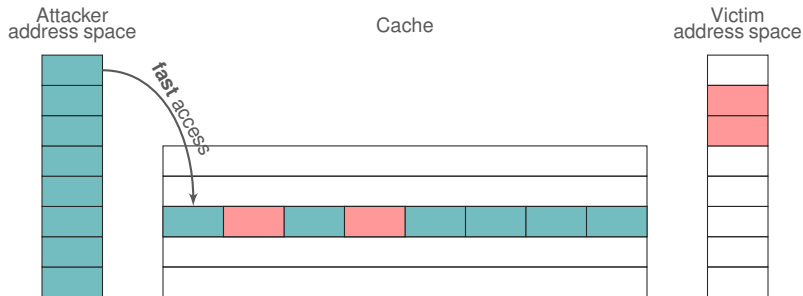


**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

**step 2:** attacker probes data to determine if the set was accessed

# Prime+Probe

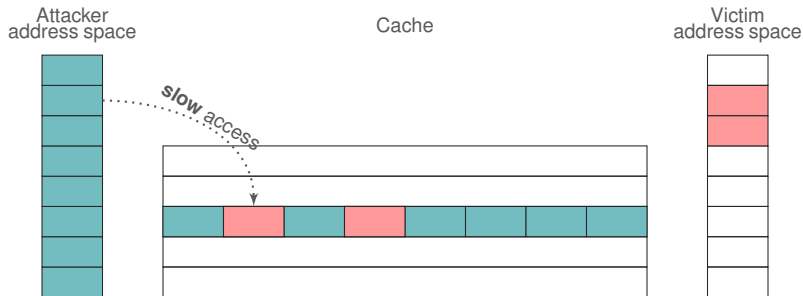


**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

**step 2:** attacker probes data to determine if the set was accessed

# Prime+Probe



**step 0:** attacker fills the cache (prime)

**step 1:** victim evicts cache lines while performing encryption

**step 2:** attacker probes data to determine if the set was accessed

1. Caches

2. Cache attacks

3. Cache Template Attacks

4. Rowhammer

# Can we spy on gedit?

gedit source code tells us `gedit_window_get_active_view()`; is used when a key is pressed, etc.



# Can we spy on gedit?

```
objdump -S /usr/bin/gedit
```

# Can we spy on gedit?

```
objdump -S /usr/bin/gedit
```

```
0000000000449ab0 <gedit_window_get_active_view>:
```

```
449ab0: 53                : push    %rbx
449ab1: 48 89 fb          : mov     %rdi,%rbx
449ab4: e8 87 df ff ff    : callq   447a40 <gedit_window_get_type>
449ab9: 48 85 db          : test    %rbx,%rbx
449abc: 74 1c             : je      449ada <gedit_window_get_active_vie
449abe: 48 8b 13          : mov     (%rbx),%rdx
449ac1: 48 85 d2          : test    %rdx,%rdx
449ac4: 74 05             : je      449acb <gedit_window_get_active_vie
```

# Can we spy on gedit?

```
objdump -S /usr/bin/gedit
```

```
0000000000449ab0 <gedit_window_get_active_view>:
```

```

449ab0: 53                : push    %rbx
449ab1: 48 89 fb          : mov     %rdi,%rbx
449ab4: e8 87 df ff ff    : callq   447a40 <gedit_window_get_type>
449ab9: 48 85 db          : test    %rbx,%rbx
449abc: 74 1c             : je      449ada <gedit_window_get_active_vie
449abe: 48 8b 13          : mov     (%rbx),%rdx
449ac1: 48 85 d2          : test    %rdx,%rdx
449ac4: 74 05             : je      449acb <gedit_window_get_active_vie

```

Round 449ab0 up to be cache-line-aligned: 449ac0

# Where are my functions!?

449ac0 is a virtual address! What now?

Where in the binary is this function?

# Where are my functions!?

```
readelf -a /usr/bin/gedit
```

# Where are my functions!?

```
readelf -a /usr/bin/gedit
```

Program Headers:

| Type | Offset   | VirtAddr | PhysAddr |        |
|------|----------|----------|----------|--------|
|      | FileSiz  | MemSiz   | Flags    | Align  |
| ...  |          |          |          |        |
| LOAD | 0x000000 | 0x400000 | 0x400000 |        |
|      | 0x089314 | 0x089314 | R E      | 200000 |
| ...  |          |          |          |        |

# Where are my functions!?

```
readelf -a /usr/bin/gedit
```

Program Headers:

| Type | Offset   | VirtAddr | PhysAddr |        |  |
|------|----------|----------|----------|--------|--|
|      | FileSiz  | MemSiz   | Flags    | Align  |  |
| ...  |          |          |          |        |  |
| LOAD | 0x000000 | 0x400000 | 0x400000 |        |  |
|      | 0x089314 | 0x089314 | R E      | 200000 |  |
| ...  |          |          |          |        |  |

0x400000-0x489314 is 0x0-0x89314 in the binary

# Where are my functions!?

```
readelf -a /usr/bin/gedit
```

Program Headers:

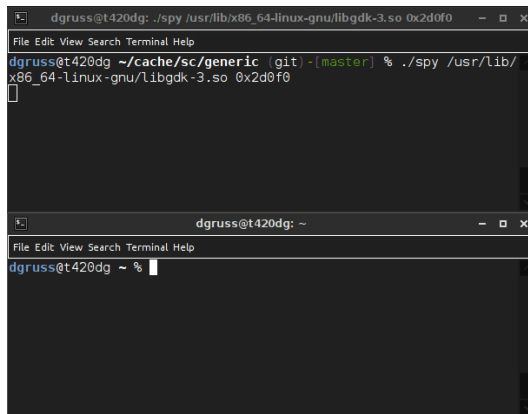
| Type | Offset   | VirtAddr | PhysAddr |        |  |
|------|----------|----------|----------|--------|--|
|      | FileSiz  | MemSiz   | Flags    | Align  |  |
| ...  |          |          |          |        |  |
| LOAD | 0x000000 | 0x400000 | 0x400000 |        |  |
|      | 0x089314 | 0x089314 | R E      | 200000 |  |
| ...  |          |          |          |        |  |

0x400000-0x489314 is 0x0-0x89314 in the binary

→ **we attack** 0x49ac0



# A cache-based keystroke logger?

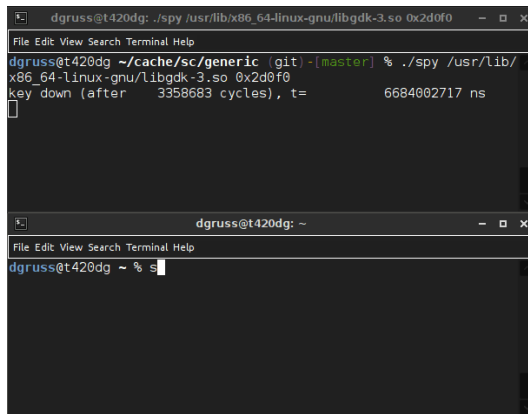


The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %' and the command entered is './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The bottom window has a title bar 'dgruss@t420dg: ~' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~ %' and the command entered is '%'. Both windows have a dark background and a light cursor.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
█

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % █
```

# A cache-based keystroke logger?

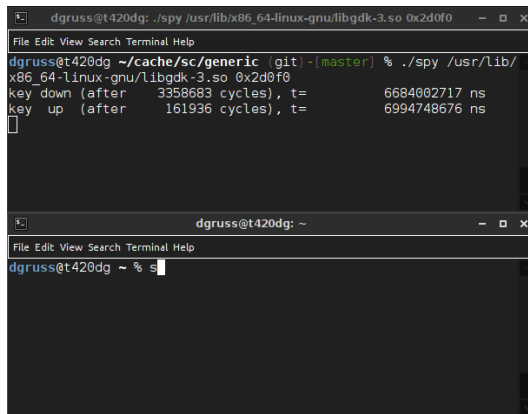


The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' has been executed, resulting in the output 'key down (after 3358683 cycles), t= 6684002717 ns'. The bottom window has a title bar 'dgruss@t420dg: ~' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~ %' and the command 's' has been entered, with a cursor following it.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
█

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % s█
```

# A cache-based keystroke logger?

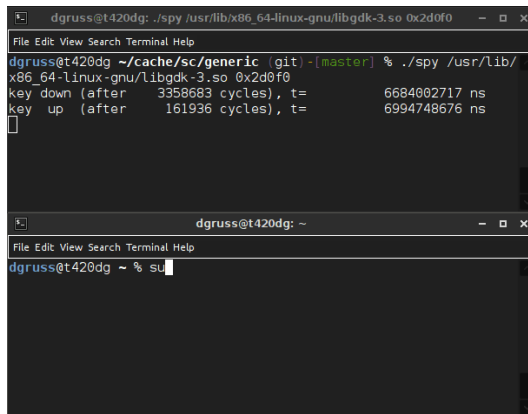


The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command executed is './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The output is:

```
key_down (after 3358683 cycles), t= 6684002717 ns
key_up (after 161936 cycles), t= 6994748676 ns
```

The bottom window has a title bar 'dgruss@t420dg: ~' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~ %'. The command executed is 's'.

# A cache-based keystroke logger?

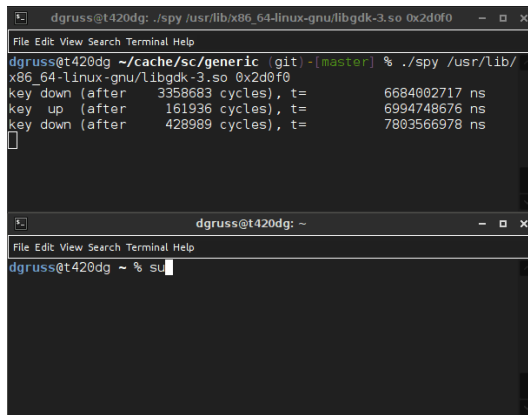


The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' has been executed, resulting in two lines of output: 'key down (after 3358683 cycles), t= 6684002717 ns' and 'key up (after 161936 cycles), t= 6994748676 ns'. The bottom window has a title bar 'dgruss@t420dg: ~' and a menu bar 'File Edit View Search Terminal Help'. The prompt is 'dgruss@t420dg ~ %' and the command 'su' has been entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
█

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % su
```

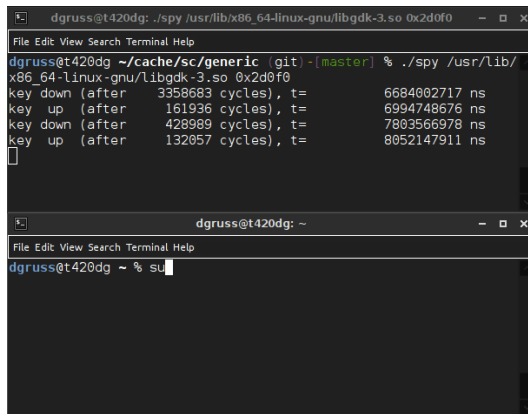
# A cache-based keystroke logger?



The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' has been executed, resulting in three lines of output: 'key\_down (after 3358683 cycles), t= 6684002717 ns', 'key\_up (after 161936 cycles), t= 6994748676 ns', and 'key\_down (after 428989 cycles), t= 7803566978 ns'. The bottom window has a title bar 'dgruss@t420dg: ~'. The prompt is 'dgruss@t420dg ~ %' and the command 'su' has been entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key_down (after 3358683 cycles), t= 6684002717 ns
key_up (after 161936 cycles), t= 6994748676 ns
key_down (after 428989 cycles), t= 7803566978 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % su
```

# A cache-based keystroke logger?

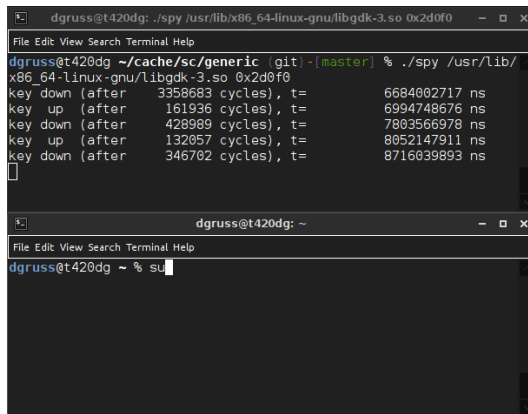


The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command executed is './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The output shows four key events with their cycle counts and timestamps:

| Event    | Cycles  | Timestamp (ns) |
|----------|---------|----------------|
| key down | 3358683 | 6684002717     |
| key up   | 161936  | 6994748676     |
| key down | 428989  | 7803566978     |
| key up   | 132057  | 8052147911     |

The bottom window has a title bar 'dgruss@t420dg: ~'. The prompt is 'dgruss@t420dg ~ %'. The command executed is 'su'.

# A cache-based keystroke logger?

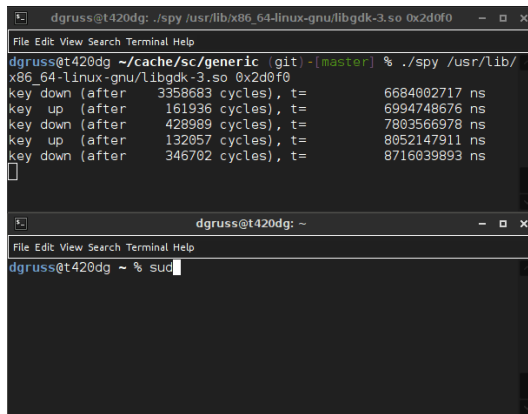


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays the output of the './spy' command, which logs key events. The output shows five key events with their respective cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'su' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key_down (after 3358683 cycles), t= 6684002717 ns
key_up (after 161936 cycles), t= 6994748676 ns
key_down (after 428989 cycles), t= 7803566978 ns
key_up (after 132057 cycles), t= 8052147911 ns
key_down (after 346702 cycles), t= 8716039893 ns
█

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % su
```

# A cache-based keystroke logger?



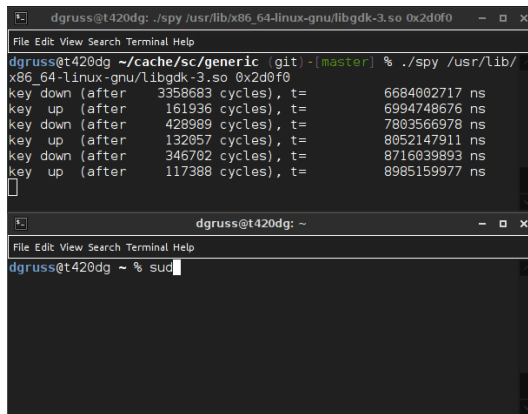
The image shows two terminal windows. The top window has a title bar 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The prompt is 'dgruss@t420dg ~/cache/sc/generic (git) -[master] %'. The command executed is './spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. The output shows five key events with their cycle counts and timestamps:

| Event    | Cycles  | Timestamp (ns) |
|----------|---------|----------------|
| key down | 3358683 | 6684002717     |
| key up   | 161936  | 6994748676     |
| key down | 428989  | 7803566978     |
| key up   | 132057  | 8052147911     |
| key down | 346702  | 8716039893     |

The bottom window has a title bar 'dgruss@t420dg: ~'. The prompt is 'dgruss@t420dg ~ %'. The command executed is 'sudo'.



# A cache-based keystroke logger?



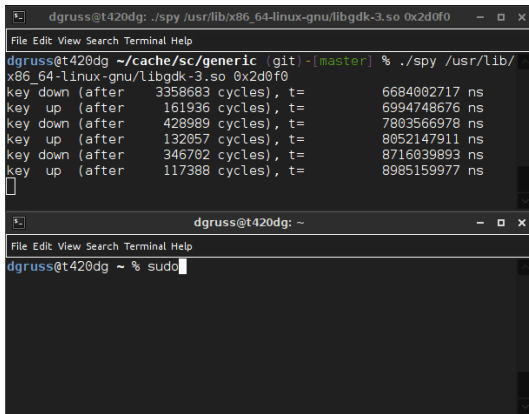
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays the output of the './spy' command, which logs key events. The output shows a sequence of key presses and releases with their corresponding cycle counts and timestamps in nanoseconds (ns). The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns

```

```
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo
```

# A cache-based keystroke logger?



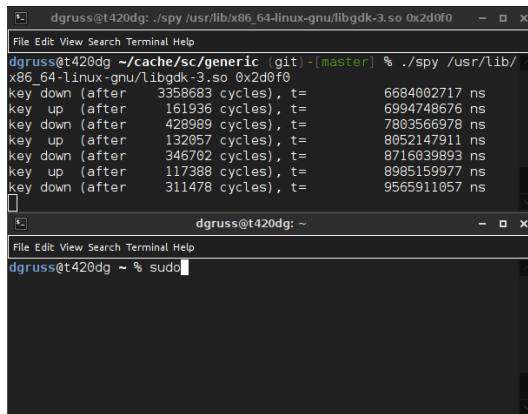
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays the output of a program that logs keystrokes. The output shows a sequence of key presses and releases with their corresponding timestamps in nanoseconds (ns). The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns

```

```
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo
```

# A cache-based keystroke logger?



The image displays two terminal windows. The top window shows the execution of a program that logs keystrokes by measuring CPU cycles. The output shows a sequence of key presses and releases with their corresponding cycle counts and timestamps. The bottom window shows the user entering the 'sudo' command.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
```

```
File Edit View Search Terminal Help
```

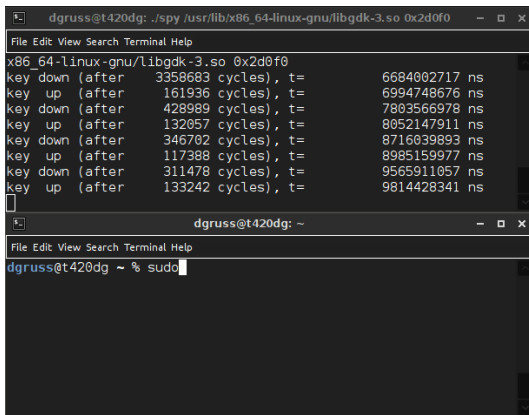
```
dgruss@t420dg ~/cache/sc/generic (git) -[master] % ./spy /usr/lib/
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
```

```
dgruss@t420dg: ~
```

```
File Edit View Search Terminal Help
```

```
dgruss@t420dg ~ % sudo
```

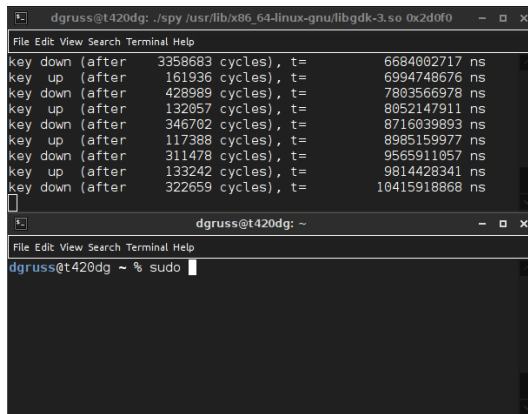
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
x86_64-linux-gnu/libgdk-3.so 0x2d0f0
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo
```

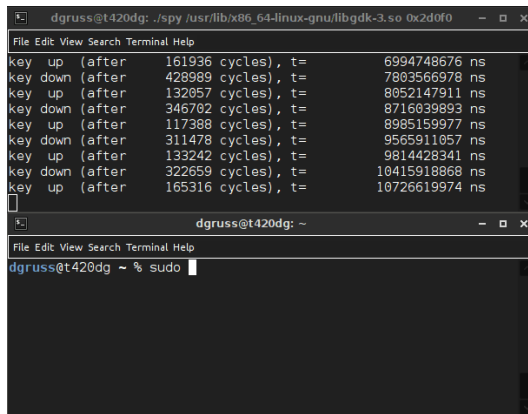
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 3358683 cycles), t= 6684002717 ns
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo
```

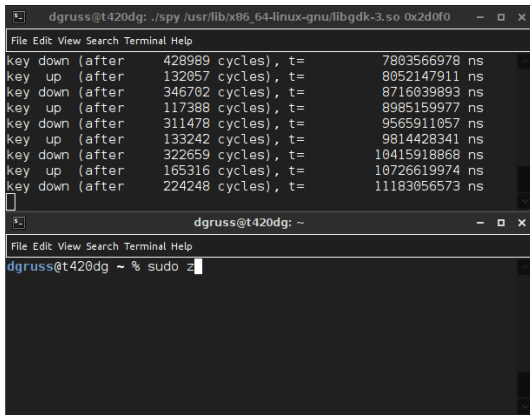
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 161936 cycles), t= 6994748676 ns
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo
```

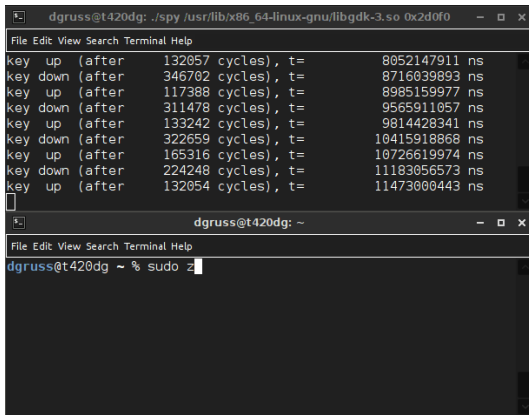
# A cache-based keystroke logger?



The image shows two terminal windows. The top window displays the output of a program running `./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0`. It shows a series of key events with their cycle counts and timestamps. The bottom window shows the user `dgruss` at the `t420dg` prompt, typing `sudo z`.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 428989 cycles), t= 7803566978 ns
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo z
```

# A cache-based keystroke logger?

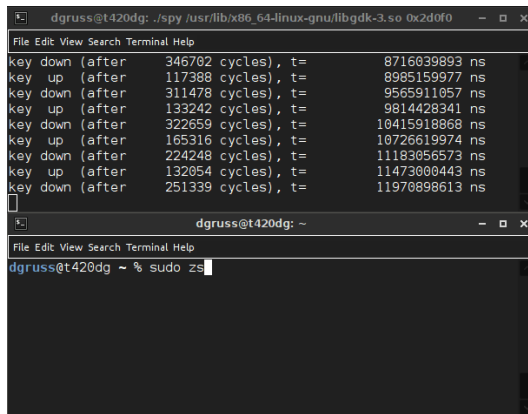


The image shows two terminal windows. The top window displays the output of a program running `./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0`. It shows a series of key events with their cycle counts and timestamps. The bottom window shows the user `dgruss` at the `t420dg` prompt, typing `sudo z`.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 132057 cycles), t= 8052147911 ns
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo z
```



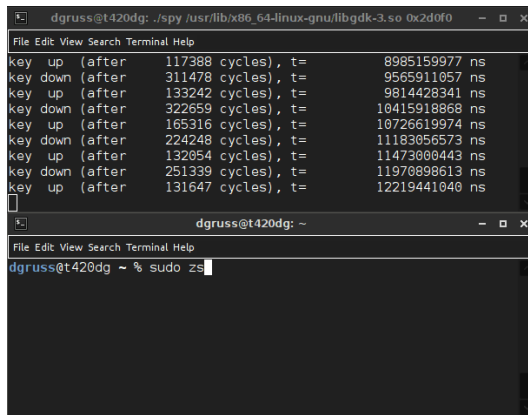
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo zs' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 346702 cycles), t= 8716039893 ns
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zs
```

# A cache-based keystroke logger?

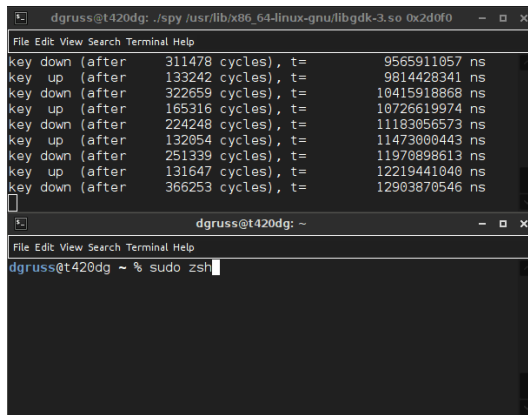


The image shows two terminal windows. The top window displays the output of a program running `./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0`. It shows a series of key events with timestamps in nanoseconds. The bottom window shows a shell prompt where the user has entered `sudo zs`.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 117388 cycles), t= 8985159977 ns
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zs
```

# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo zsh' being entered.

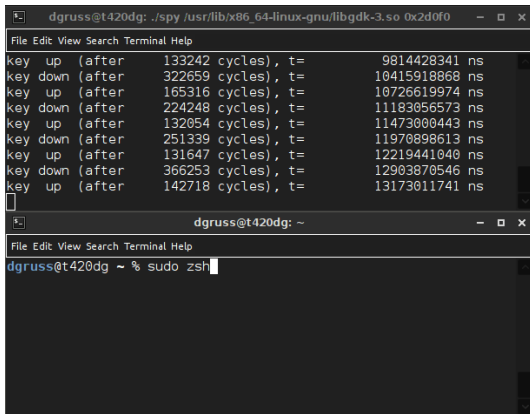
```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 311478 cycles), t= 9565911057 ns
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns

```

```
dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh

```

# A cache-based keystroke logger?

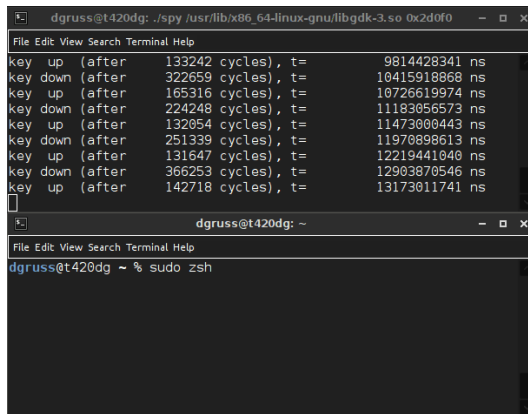


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo zsh' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
```

# A cache-based keystroke logger?

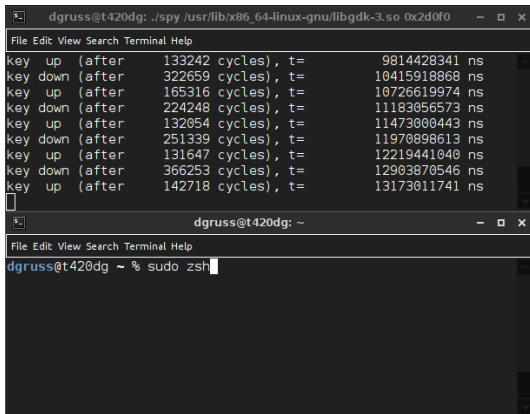


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo zsh' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
```

# A cache-based keystroke logger?

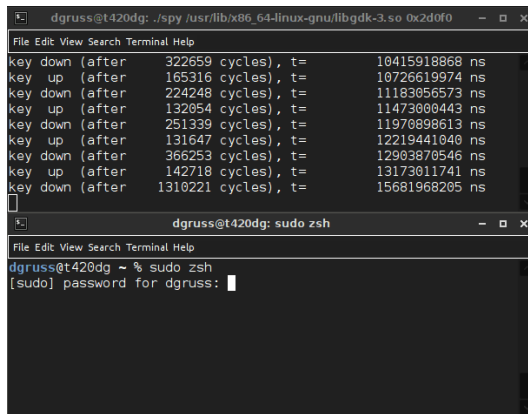


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: ~' and shows the command 'sudo zsh' being entered.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133242 cycles), t= 9814428341 ns
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns

dgruss@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
```

# A cache-based keystroke logger?



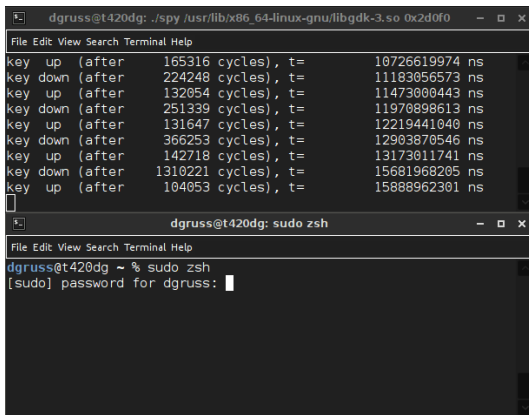
The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 322659 cycles), t= 10415918868 ns
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?



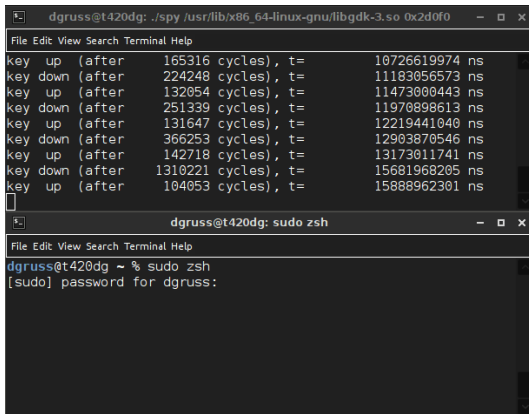
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```



# A cache-based keystroke logger?

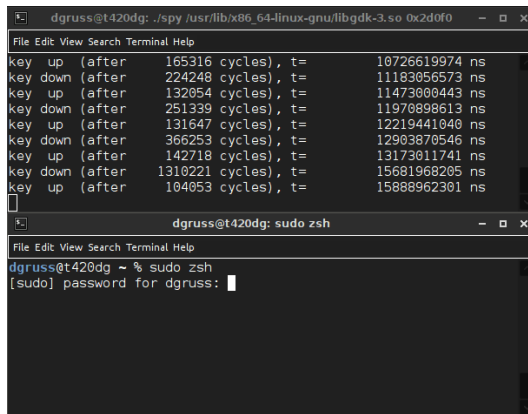


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the prompt for the sudo password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?



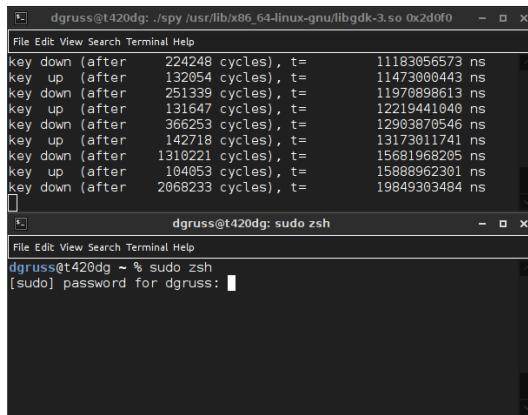
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 165316 cycles), t= 10726619974 ns
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?

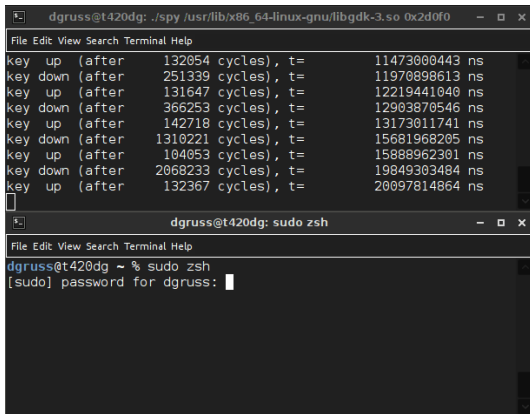


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 224248 cycles), t= 11183056573 ns
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
█

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: █
```

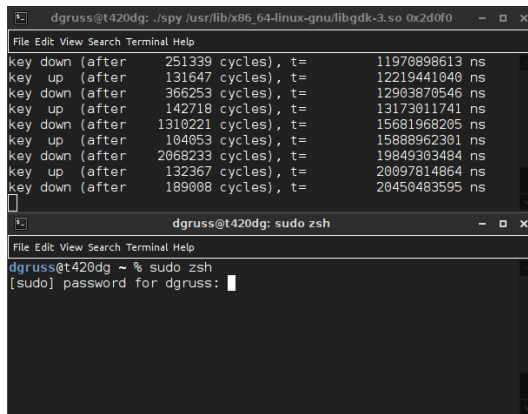
# A cache-based keystroke logger?



```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 132054 cycles), t= 11473000443 ns
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?

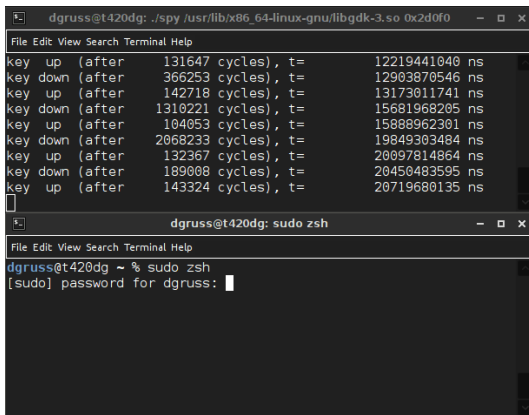


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user being prompted for their password to run 'sudo zsh'.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 251339 cycles), t= 11970898613 ns
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
█

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: █
```

# A cache-based keystroke logger?

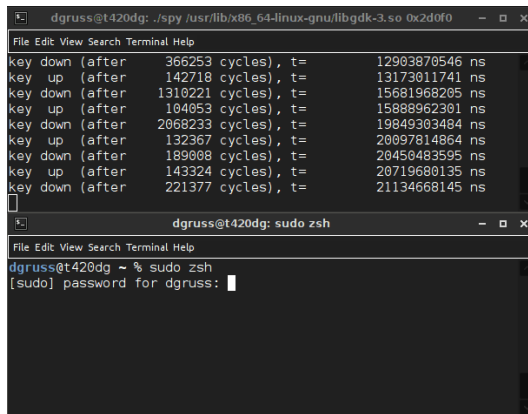


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 131647 cycles), t= 12219441040 ns
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
█

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: █
```

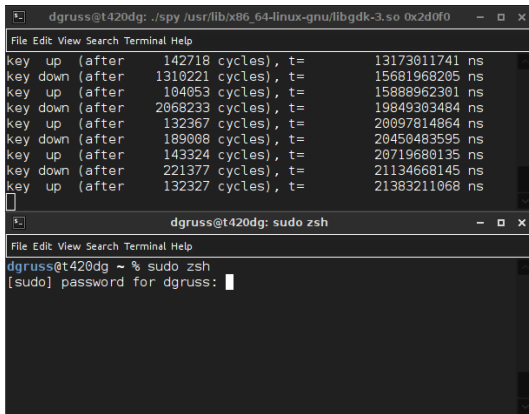
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 366253 cycles), t= 12903870546 ns
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

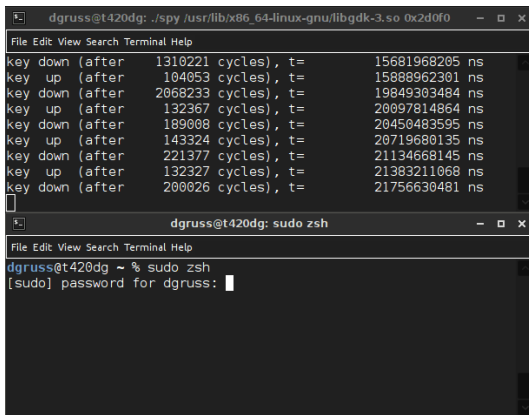
```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 142718 cycles), t= 13173011741 ns
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```



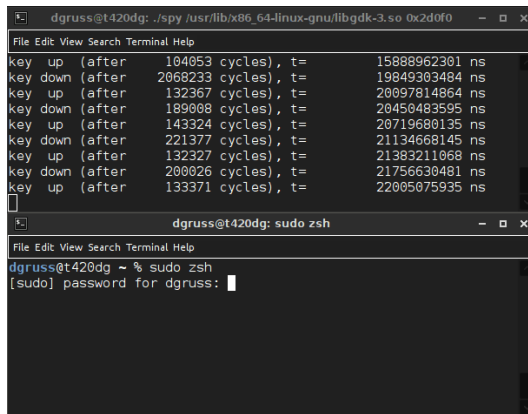
# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 1310221 cycles), t= 15681968205 ns
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?

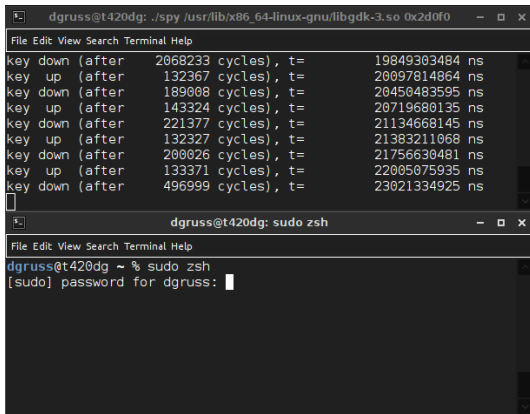


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 104053 cycles), t= 15888962301 ns
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?

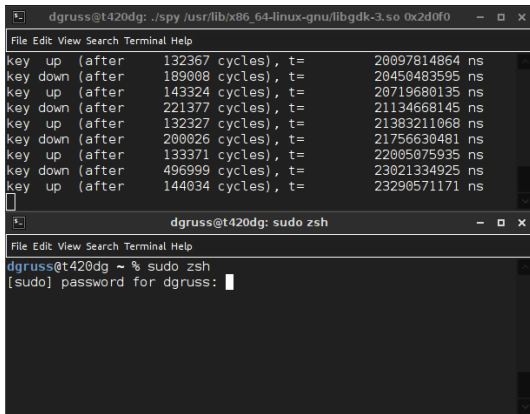


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 2068233 cycles), t= 19849303484 ns
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
█

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: █
```

# A cache-based keystroke logger?



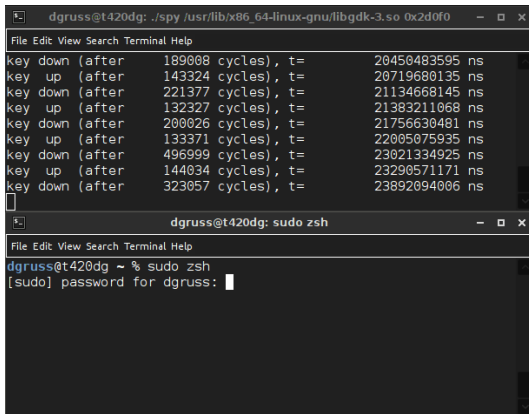
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 132367 cycles), t= 20097814864 ns
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?

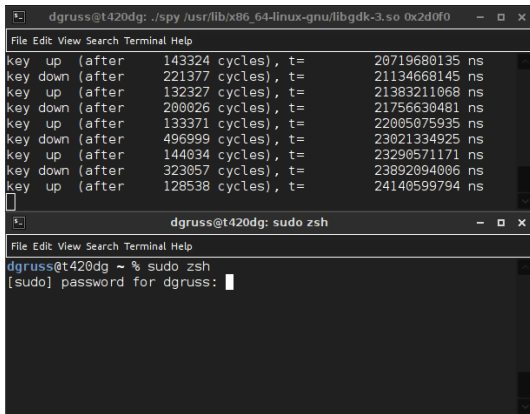


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user attempting to run 'sudo zsh' and being prompted for a password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 189008 cycles), t= 20450483595 ns
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?



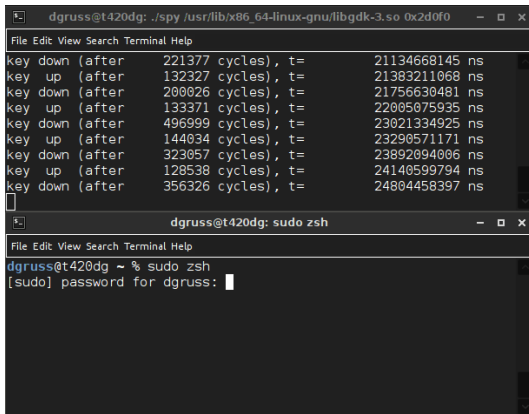
The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user being prompted for their password to run 'sudo zsh'.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 143324 cycles), t= 20719680135 ns
key down (after 221377 cycles), t= 21134668145 ns
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?



The image displays two terminal windows. The top window shows the execution of a program that logs keystrokes by measuring the time taken for memory accesses. The output shows a sequence of key presses and releases, each followed by the number of cycles and the time taken. The bottom window shows the user running 'sudo zsh' and being prompted for their password.

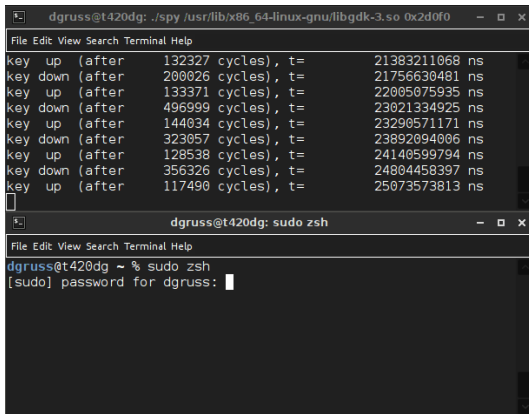
```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
```

| Event                              | Cycles | Time           |
|------------------------------------|--------|----------------|
| key down (after 221377 cycles), t= | 221377 | 21134668145 ns |
| key up (after 132327 cycles), t=   | 132327 | 21383211068 ns |
| key down (after 200026 cycles), t= | 200026 | 21756630481 ns |
| key up (after 133371 cycles), t=   | 133371 | 22005075935 ns |
| key down (after 496999 cycles), t= | 496999 | 23021334925 ns |
| key up (after 144034 cycles), t=   | 144034 | 23290571171 ns |
| key down (after 323057 cycles), t= | 323057 | 23892094006 ns |
| key up (after 128538 cycles), t=   | 128538 | 24140599794 ns |
| key down (after 356326 cycles), t= | 356326 | 24804458397 ns |

```
dgruss@t420dg: sudo zsh
```

```
[sudo] password for dgruss:
```

# A cache-based keystroke logger?

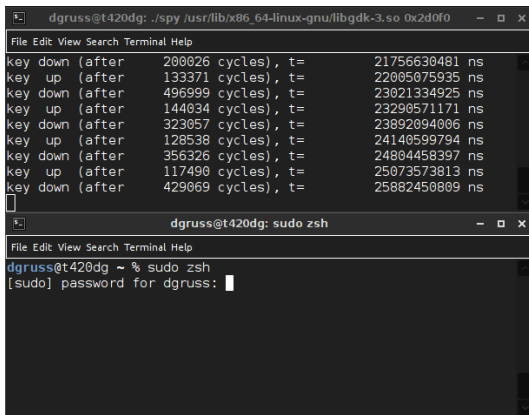


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 132327 cycles), t= 21383211068 ns
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```



# A cache-based keystroke logger?



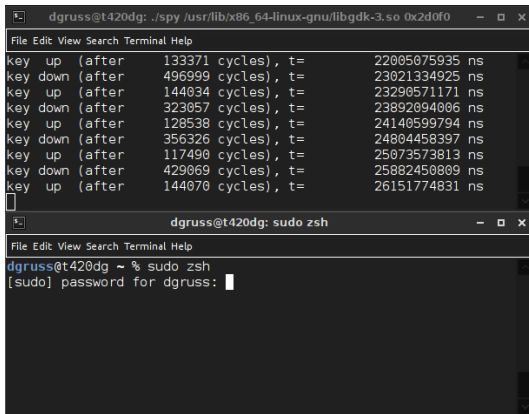
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 200026 cycles), t= 21756630481 ns
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?



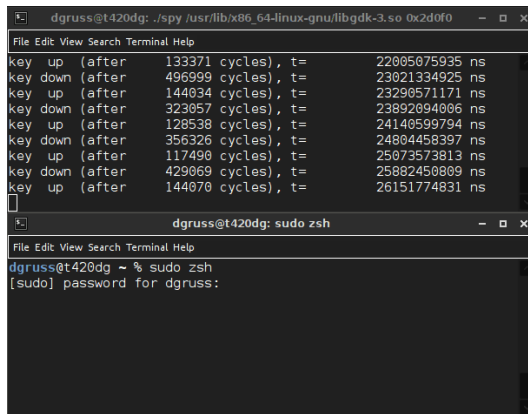
The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user attempting to run 'sudo zsh' and being prompted for a password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?



The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the prompt for a password.

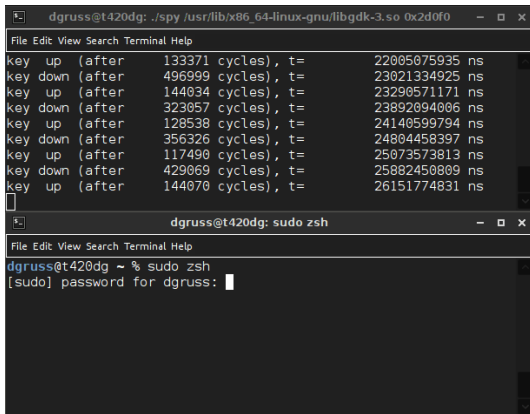
```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:

```

# A cache-based keystroke logger?



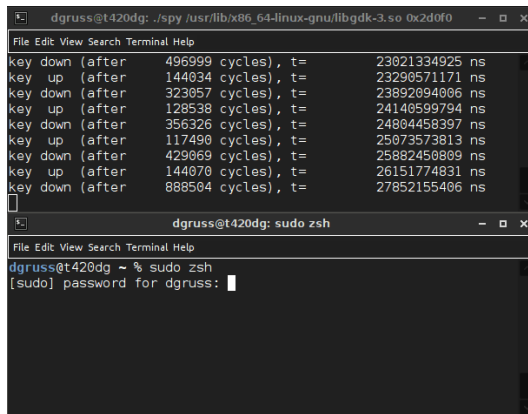
The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'dgruss@t420dg: sudo zsh', shows the user being prompted for their password to run 'sudo zsh'.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 133371 cycles), t= 22005075935 ns
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?

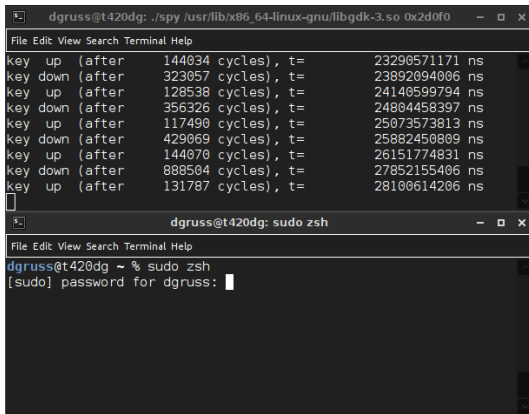


```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key down (after 496999 cycles), t= 23021334925 ns
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns

```

```
dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: 
```

# A cache-based keystroke logger?

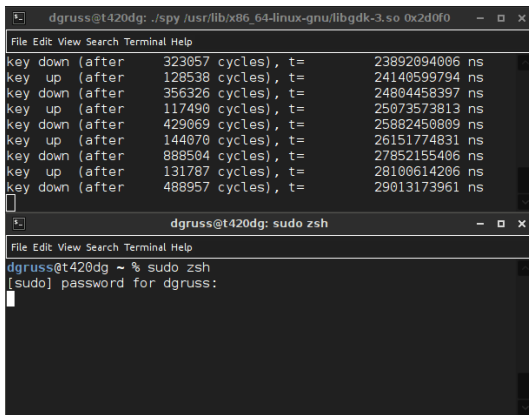


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0'. It displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 144034 cycles), t= 23290571171 ns
key down (after 323057 cycles), t= 23892094006 ns
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
█

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss: █
```

# A cache-based keystroke logger?

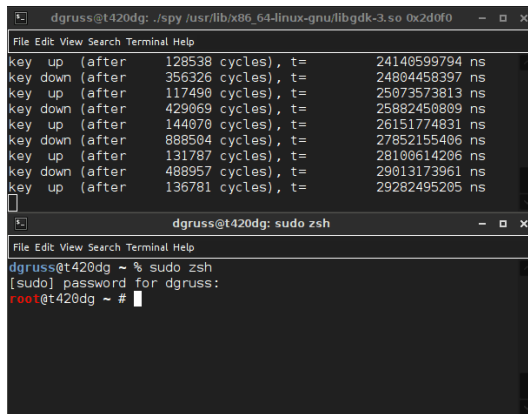


The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a table of key events. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for their password.

| Event           | Cycles | Time (ns)   |
|-----------------|--------|-------------|
| key down (after | 323057 | 23892094006 |
| key up (after   | 128538 | 24140599794 |
| key down (after | 356326 | 24804458397 |
| key up (after   | 117490 | 25073573813 |
| key down (after | 429069 | 25882450809 |
| key up (after   | 144070 | 26151774831 |
| key down (after | 888504 | 27852155406 |
| key up (after   | 131787 | 28100614206 |
| key down (after | 488957 | 29013173961 |

```
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
```

# A cache-based keystroke logger?



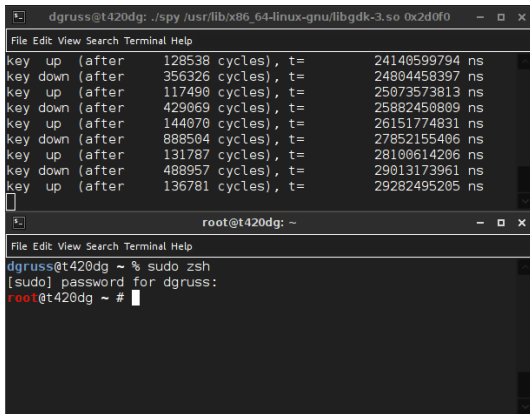
The image shows two terminal windows. The top window is titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0' and displays a list of key events with their cycle counts and timestamps. The bottom window is titled 'dgruss@t420dg: sudo zsh' and shows the user being prompted for a password and then becoming root.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
key down (after 488957 cycles), t= 29013173961 ns
key up (after 136781 cycles), t= 29282495205 ns

dgruss@t420dg: sudo zsh
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
root@t420dg ~ #
```



# A cache-based keystroke logger?

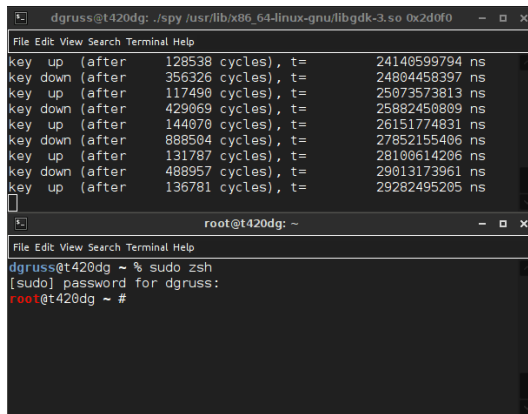


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'root@t420dg: ~', shows the user 'dgruss' running 'sudo zsh' and being prompted for a password, after which they become the root user.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
key down (after 488957 cycles), t= 29013173961 ns
key up (after 136781 cycles), t= 29282495205 ns

root@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
root@t420dg ~ #
```

# A cache-based keystroke logger?

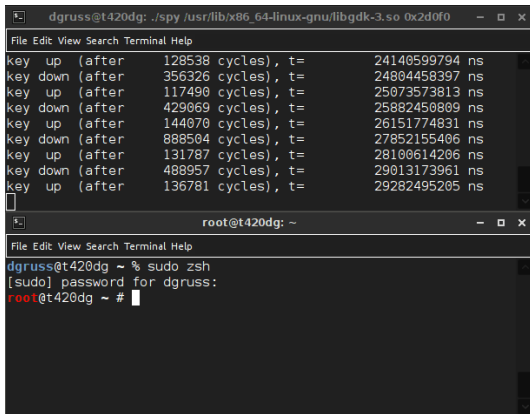


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'root@t420dg: ~', shows the user 'dgruss' running 'sudo zsh' and being prompted for a password, after which they become the root user.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
key down (after 488957 cycles), t= 29013173961 ns
key up (after 136781 cycles), t= 29282495205 ns

root@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
root@t420dg ~ #
```

# A cache-based keystroke logger?

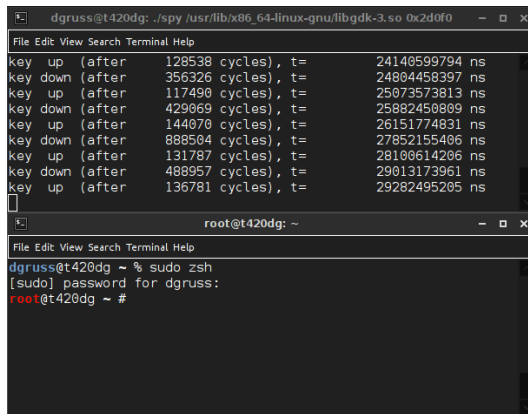


The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'root@t420dg: ~', shows the user 'dgruss' running 'sudo zsh' and being prompted for a password, after which they become the root user.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
key down (after 488957 cycles), t= 29013173961 ns
key up (after 136781 cycles), t= 29282495205 ns

root@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
root@t420dg ~ #
```

# A cache-based keystroke logger?



The image shows two terminal windows. The top window, titled 'dgruss@t420dg: ./spy /usr/lib/x86\_64-linux-gnu/libgdk-3.so 0x2d0f0', displays a list of key events with their cycle counts and timestamps. The bottom window, titled 'root@t420dg: ~', shows the user 'dgruss' running 'sudo zsh' and being prompted for a password, after which they become 'root'.

```
dgruss@t420dg: ./spy /usr/lib/x86_64-linux-gnu/libgdk-3.so 0x2d0f0
File Edit View Search Terminal Help
key up (after 128538 cycles), t= 24140599794 ns
key down (after 356326 cycles), t= 24804458397 ns
key up (after 117490 cycles), t= 25073573813 ns
key down (after 429069 cycles), t= 25882450809 ns
key up (after 144070 cycles), t= 26151774831 ns
key down (after 888504 cycles), t= 27852155406 ns
key up (after 131787 cycles), t= 28100614206 ns
key down (after 488957 cycles), t= 29013173961 ns
key up (after 136781 cycles), t= 29282495205 ns

root@t420dg: ~
File Edit View Search Terminal Help
dgruss@t420dg ~ % sudo zsh
[sudo] password for dgruss:
root@t420dg ~ #
```

# Challenges

- How to locate key-dependent memory accesses?

# Challenges

- How to locate key-dependent memory accesses?
- It's complicated:
  - Large binaries and libraries (third-party code)
  - Many libraries (gedit: 60MB)
  - Closed-source / unknown binaries
  - Self-compiled binaries

# Challenges

- How to locate key-dependent memory accesses?
- It's complicated:
  - Large binaries and libraries (third-party code)
  - Many libraries (gedit: 60MB)
  - Closed-source / unknown binaries
  - Self-compiled binaries
- Difficult to find **all** exploitable addresses

# Lessons learnt

- Staring at objdump's for weeks will give you headache



# Lessons learnt

- Staring at objdump's for weeks will give you headache
- Automate everything right from the start

# Cache Template Attacks

- Automatically find any secret-dependent cache access
- Can be used for attacks and to improve software

# Cache Template Attacks

- Automatically find any secret-dependent cache access
- Can be used for attacks and to improve software
- Examples:
  - Cache-based keylogger
  - Automatic attacks on crypto algorithms

# Cache Template Attacks

## Profiling Phase

- Preprocessing step to find exploitable addresses automatically
  - w.r.t. “events” (keystrokes, encryptions, ...)
  - called “Cache Template”

# Cache Template Attacks

## Profiling Phase

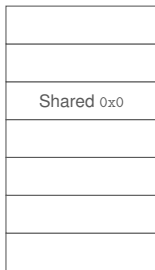
- Preprocessing step to find exploitable addresses automatically
  - w.r.t. “events” (keystrokes, encryptions, ...)
  - called “Cache Template”

## Exploitation Phase

- Monitor exploitable addresses

# Profiling Phase

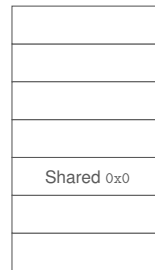
Attacker address space



Cache

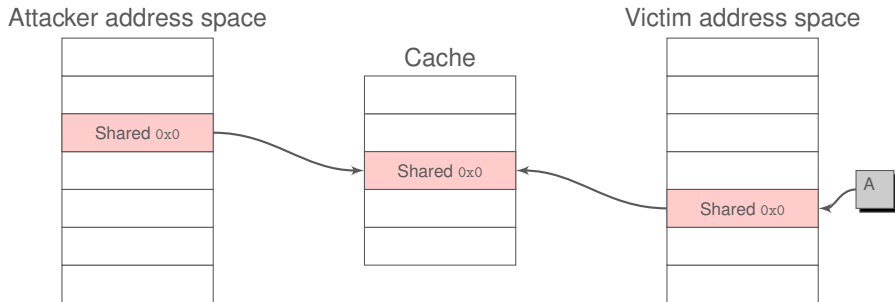


Victim address space



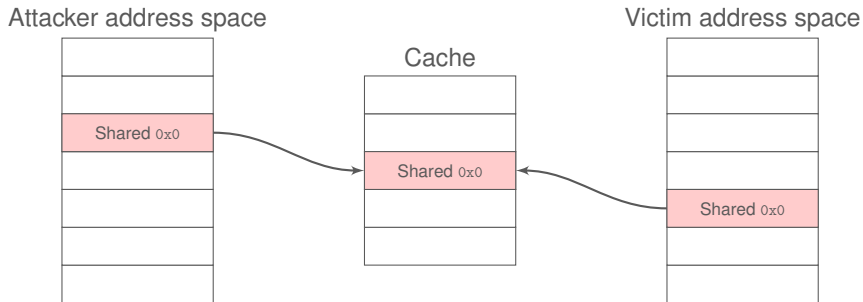
Cache is empty

# Profiling Phase



Attacker triggers an event

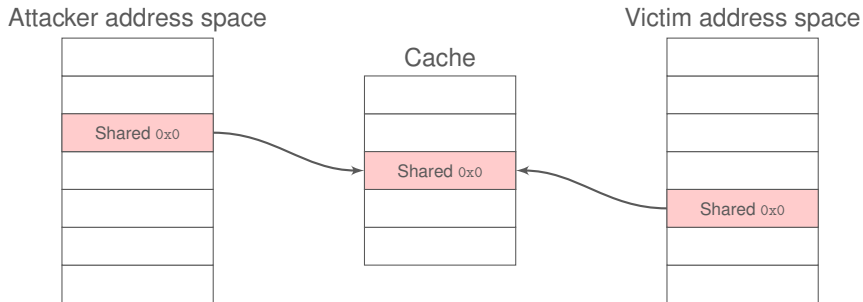
# Profiling Phase



Attacker checks one address for cache hits (“Reload”)

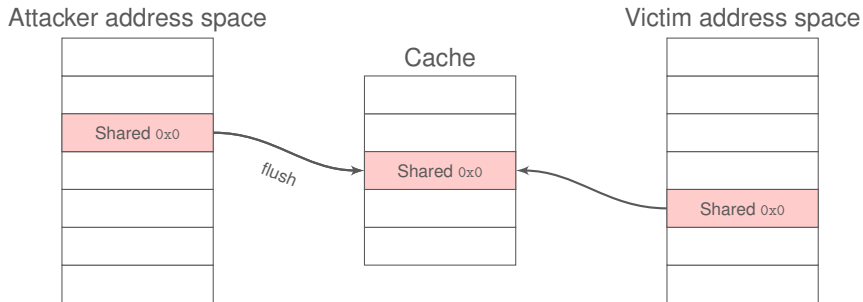


# Profiling Phase



Update cache hit ratio (per event and address)

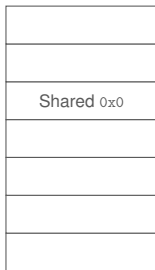
# Profiling Phase



Attacker flushes shared memory

# Profiling Phase

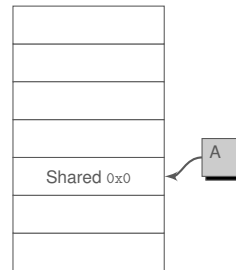
Attacker address space



Cache



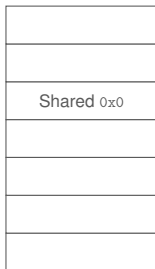
Victim address space



Repeat for higher accuracy

# Profiling Phase

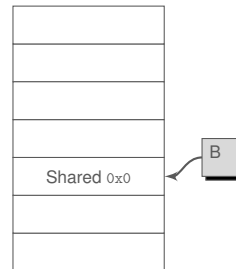
Attacker address space



Cache



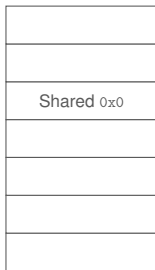
Victim address space



Repeat for all events

# Profiling Phase

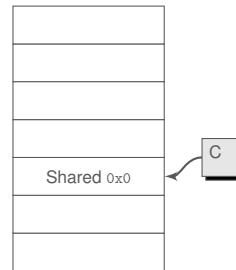
Attacker address space



Cache



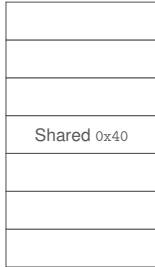
Victim address space



Repeat for all events

# Profiling Phase

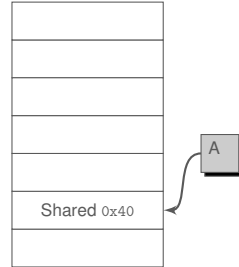
Attacker address space



Cache



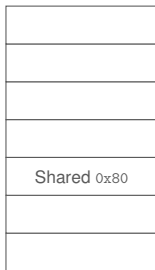
Victim address space



Continue with next address

# Profiling Phase

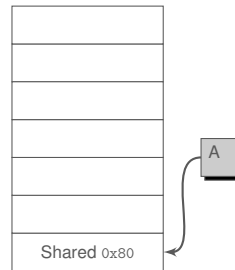
Attacker address space



Cache

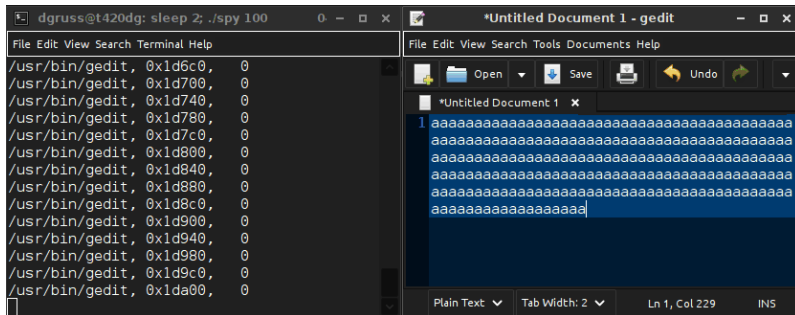


Victim address space



Continue with next address

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 229 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 229' and 'INS'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d6c0, 0
```

```
/usr/bin/gedit, 0x1d700, 0
```

```
/usr/bin/gedit, 0x1d740, 0
```

```
/usr/bin/gedit, 0x1d780, 0
```

```
/usr/bin/gedit, 0x1d7c0, 0
```

```
/usr/bin/gedit, 0x1d800, 0
```

```
/usr/bin/gedit, 0x1d840, 0
```

```
/usr/bin/gedit, 0x1d880, 0
```

```
/usr/bin/gedit, 0x1d8c0, 0
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

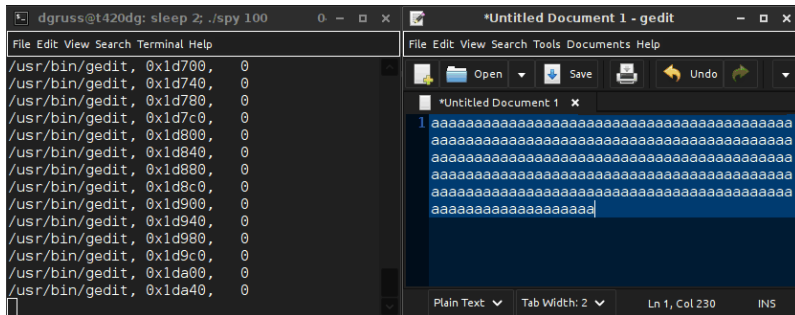
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 229 INS
```



# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 230 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 230'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d700, 0
```

```
/usr/bin/gedit, 0x1d740, 0
```

```
/usr/bin/gedit, 0x1d780, 0
```

```
/usr/bin/gedit, 0x1d7c0, 0
```

```
/usr/bin/gedit, 0x1d800, 0
```

```
/usr/bin/gedit, 0x1d840, 0
```

```
/usr/bin/gedit, 0x1d880, 0
```

```
/usr/bin/gedit, 0x1d8c0, 0
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

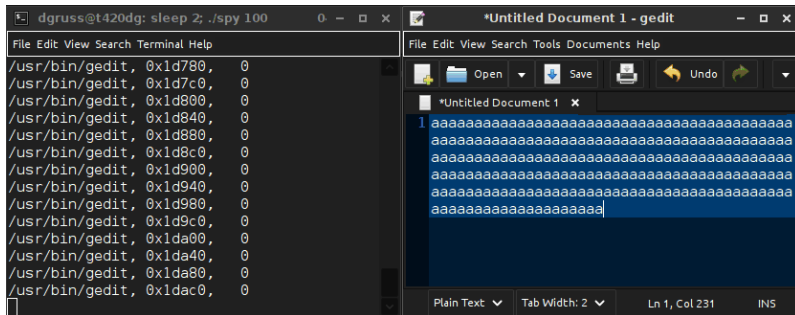
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 230 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 231 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 231' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0xd780, 0
```

```
/usr/bin/gedit, 0xd7c0, 0
```

```
/usr/bin/gedit, 0xd800, 0
```

```
/usr/bin/gedit, 0xd840, 0
```

```
/usr/bin/gedit, 0xd880, 0
```

```
/usr/bin/gedit, 0xd8c0, 0
```

```
/usr/bin/gedit, 0xd900, 0
```

```
/usr/bin/gedit, 0xd940, 0
```

```
/usr/bin/gedit, 0xd980, 0
```

```
/usr/bin/gedit, 0xd9c0, 0
```

```
/usr/bin/gedit, 0xda00, 0
```

```
/usr/bin/gedit, 0xda40, 0
```

```
/usr/bin/gedit, 0xda80, 0
```

```
/usr/bin/gedit, 0xdac0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

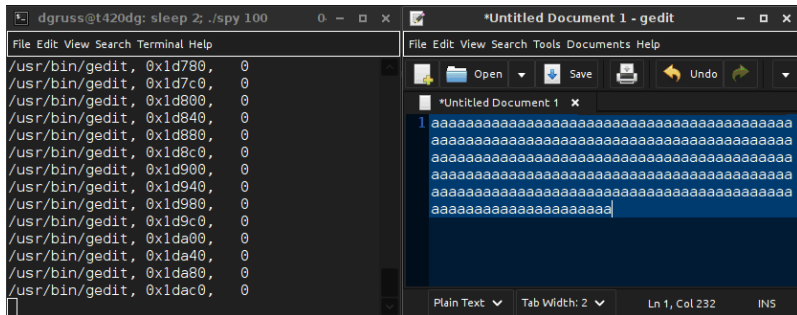
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 231 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 232 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 232'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0xd780, 0
```

```
/usr/bin/gedit, 0xd7c0, 0
```

```
/usr/bin/gedit, 0xd800, 0
```

```
/usr/bin/gedit, 0xd840, 0
```

```
/usr/bin/gedit, 0xd880, 0
```

```
/usr/bin/gedit, 0xd8c0, 0
```

```
/usr/bin/gedit, 0xd900, 0
```

```
/usr/bin/gedit, 0xd940, 0
```

```
/usr/bin/gedit, 0xd980, 0
```

```
/usr/bin/gedit, 0xd9c0, 0
```

```
/usr/bin/gedit, 0xda00, 0
```

```
/usr/bin/gedit, 0xda40, 0
```

```
/usr/bin/gedit, 0xda80, 0
```

```
/usr/bin/gedit, 0xdac0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

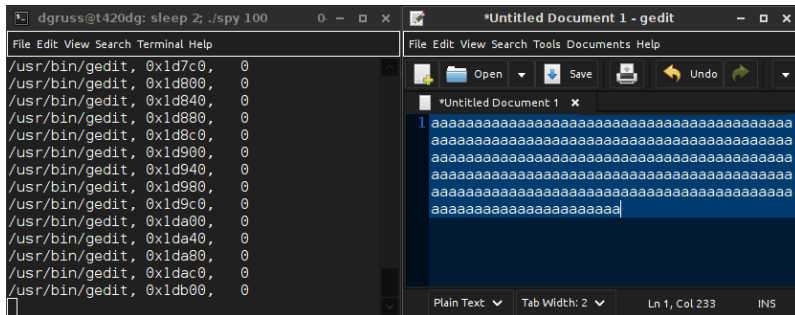
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 232 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 233 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 233'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d7c0, 0
```

```
/usr/bin/gedit, 0x1d800, 0
```

```
/usr/bin/gedit, 0x1d840, 0
```

```
/usr/bin/gedit, 0x1d880, 0
```

```
/usr/bin/gedit, 0x1d8c0, 0
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

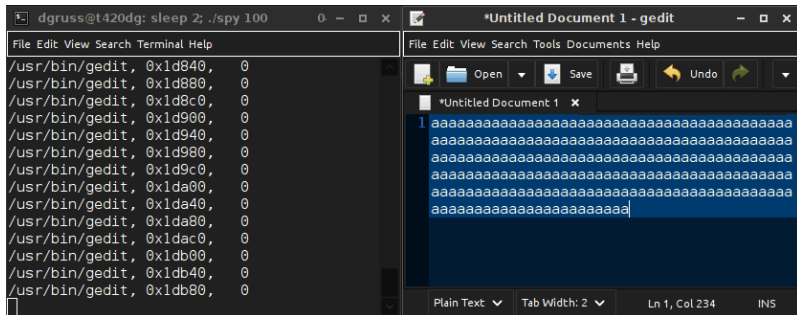
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 233 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 234 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 234'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d840, 0
```

```
/usr/bin/gedit, 0x1d880, 0
```

```
/usr/bin/gedit, 0x1d8c0, 0
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

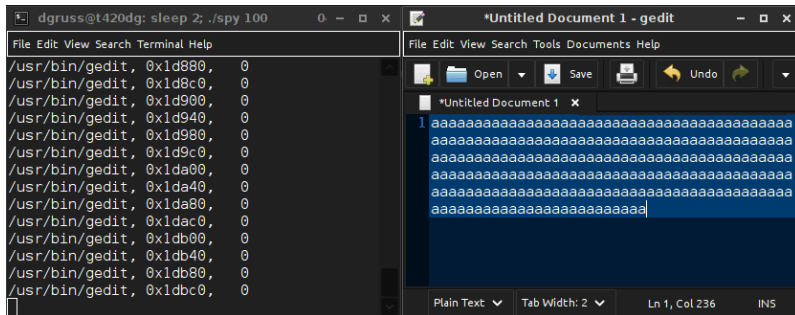
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 234 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 236 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 236' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d880, 0
```

```
/usr/bin/gedit, 0x1d8c0, 0
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

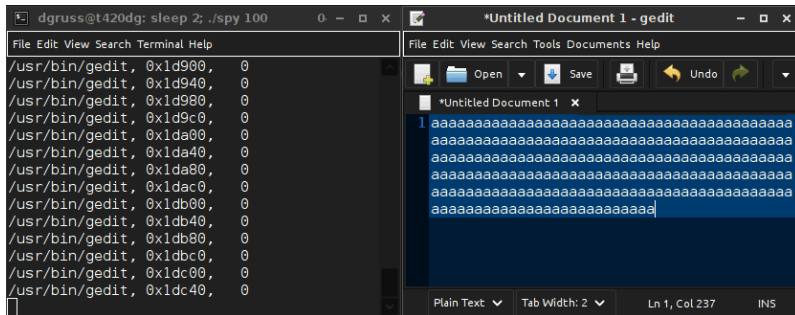
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 236 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 237 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 237' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d900, 0
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

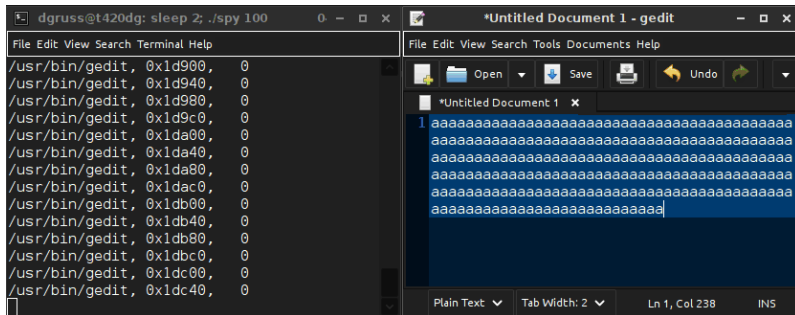
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 237 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 238 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 238' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d900, 0
/usr/bin/gedit, 0x1d940, 0
/usr/bin/gedit, 0x1d980, 0
/usr/bin/gedit, 0x1d9c0, 0
/usr/bin/gedit, 0x1da00, 0
/usr/bin/gedit, 0x1da40, 0
/usr/bin/gedit, 0x1da80, 0
/usr/bin/gedit, 0x1dac0, 0
/usr/bin/gedit, 0x1db00, 0
/usr/bin/gedit, 0x1db40, 0
/usr/bin/gedit, 0x1db80, 0
/usr/bin/gedit, 0x1dbc0, 0
/usr/bin/gedit, 0x1dc00, 0
/usr/bin/gedit, 0x1dc40, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

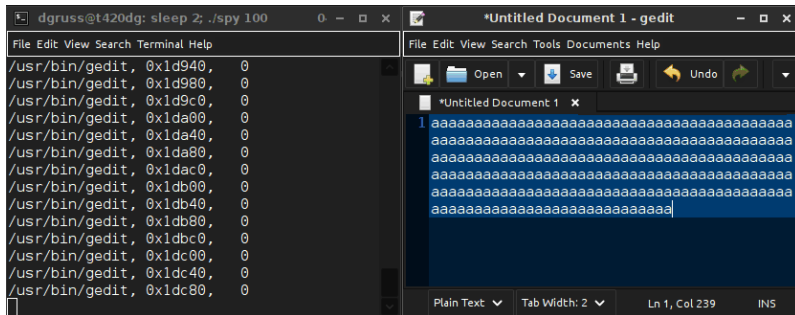
```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|
```

```
Plain Text Tab Width: 2 Ln 1, Col 238 INS
```



# Profiling a Single Event



The image shows a terminal window and a gedit editor window side-by-side. The terminal window, titled 'dgruss@t420dg: sleep 2; ./spy 100', displays a list of memory addresses and their corresponding values, all of which are 0. The gedit editor window, titled '\*Untitled Document 1 - gedit', shows a document with a single line of text consisting of 239 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 239' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d940, 0
```

```
/usr/bin/gedit, 0x1d980, 0
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

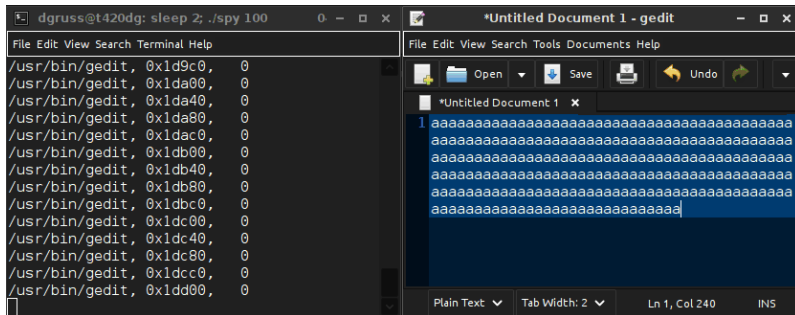
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 239 INS
```

# Profiling a Single Event



The image shows two side-by-side windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are '0'. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a single line of text consisting of 240 'a' characters, with the first character highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 240'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1d9c0, 0
```

```
/usr/bin/gedit, 0x1da00, 0
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

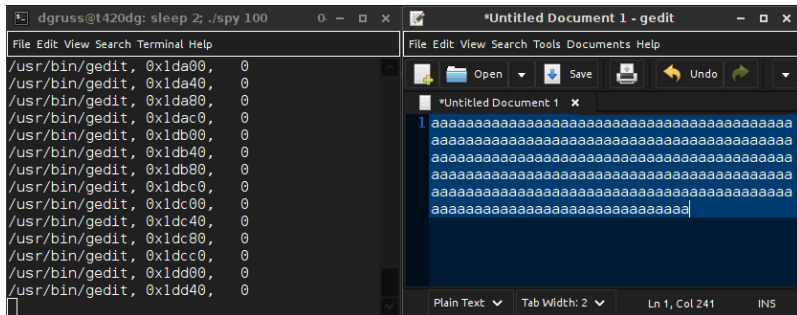
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 240 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 241 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 241' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1da00, 0
/usr/bin/gedit, 0x1da40, 0
/usr/bin/gedit, 0x1da80, 0
/usr/bin/gedit, 0x1dac0, 0
/usr/bin/gedit, 0x1db00, 0
/usr/bin/gedit, 0x1db40, 0
/usr/bin/gedit, 0x1db80, 0
/usr/bin/gedit, 0x1dbc0, 0
/usr/bin/gedit, 0x1dc00, 0
/usr/bin/gedit, 0x1dc40, 0
/usr/bin/gedit, 0x1dc80, 0
/usr/bin/gedit, 0x1dcc0, 0
/usr/bin/gedit, 0x1dd00, 0
/usr/bin/gedit, 0x1dd40, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

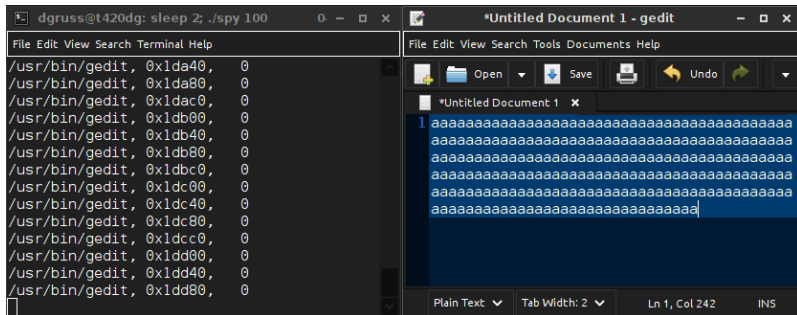
```
Open Save Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 241 INS
```

# Profiling a Single Event



The image shows two side-by-side windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of a long string of 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 242' and 'INS'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1da40, 0
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

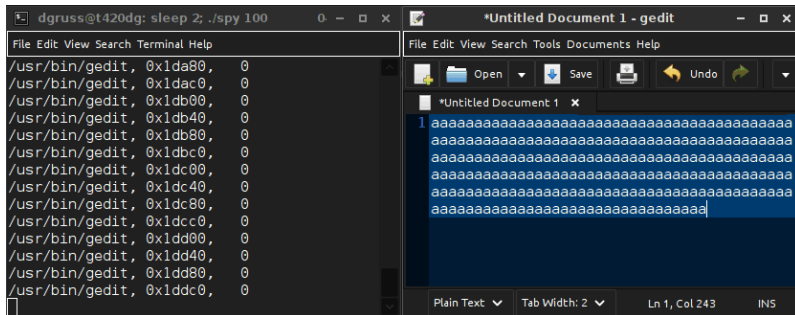
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 242 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 243 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 243' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1da80, 0
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

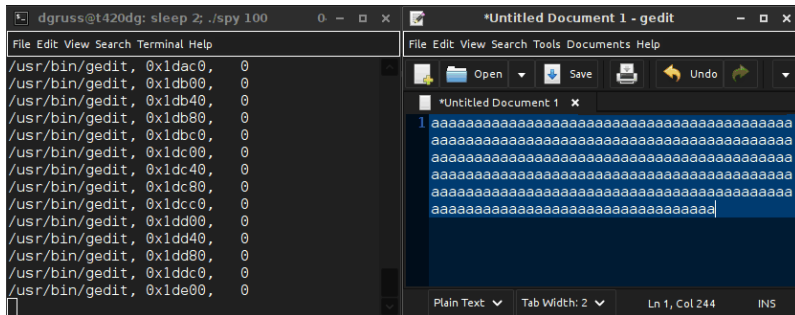
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 243 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 244 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 244'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dac0, 0
```

```
/usr/bin/gedit, 0x1db00, 0
```

```
/usr/bin/gedit, 0x1db40, 0
```

```
/usr/bin/gedit, 0x1db80, 0
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

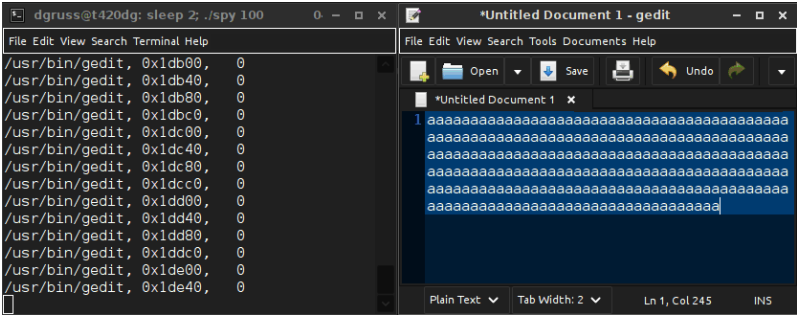
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

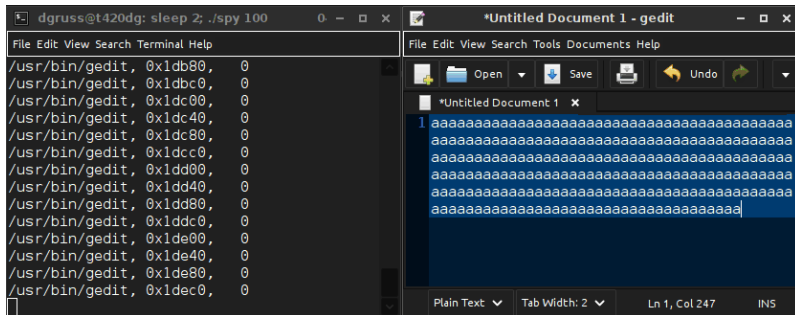
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 244 INS
```

# Profiling a Single Event



# Profiling a Single Event



The image shows two side-by-side windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 247 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 247' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0xldb80, 0
```

```
/usr/bin/gedit, 0xldb0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

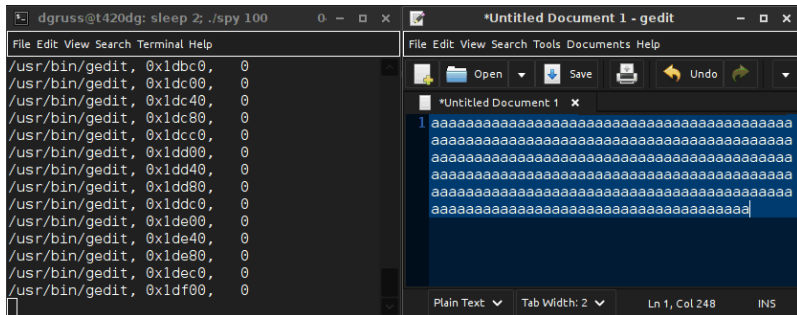
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 247 INS
```



# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 248 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 248'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

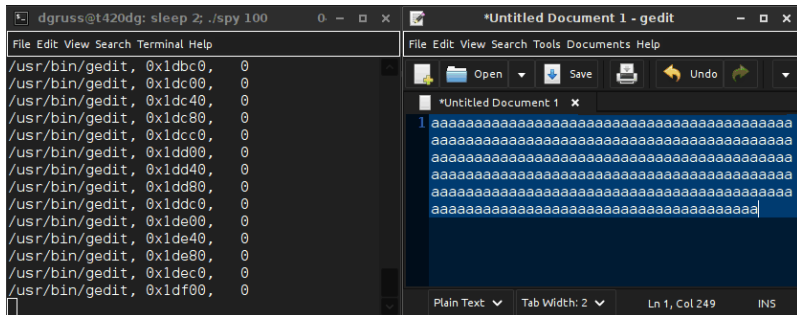
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 248 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 249 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 249' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dbc0, 0
```

```
/usr/bin/gedit, 0x1dc00, 0
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

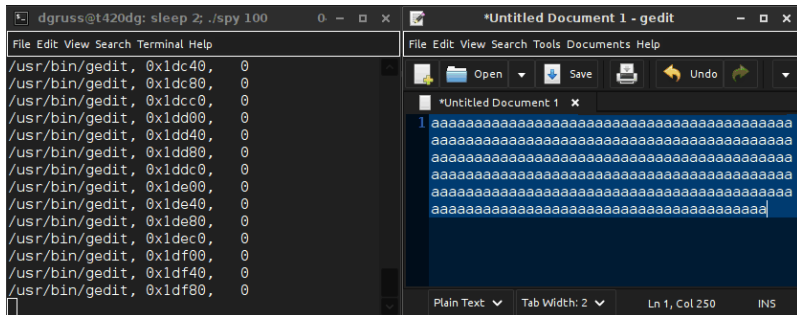
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 249 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 250 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 250'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dc40, 0
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

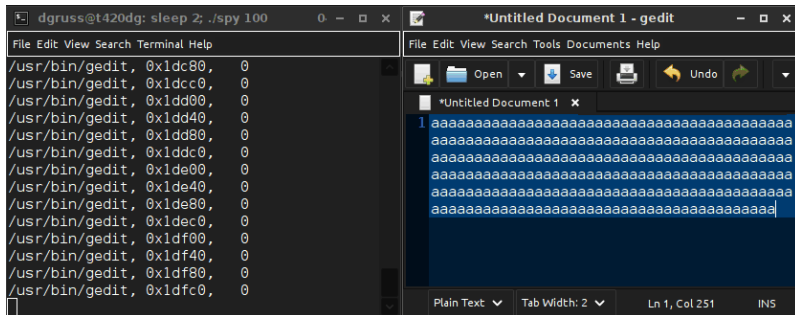
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 250 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 251 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 251'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dc80, 0
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

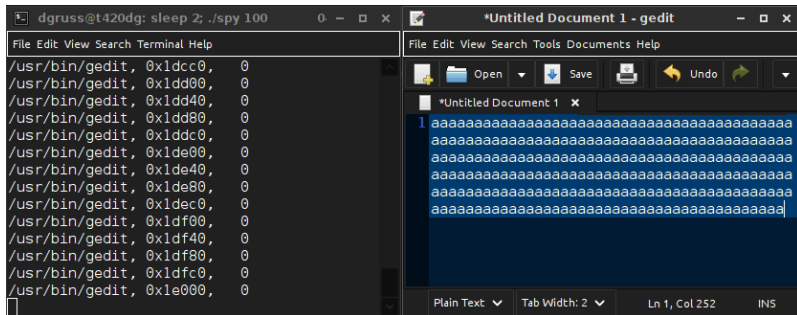
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 251 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 252 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 252'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dcc0, 0
```

```
/usr/bin/gedit, 0x1dd00, 0
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

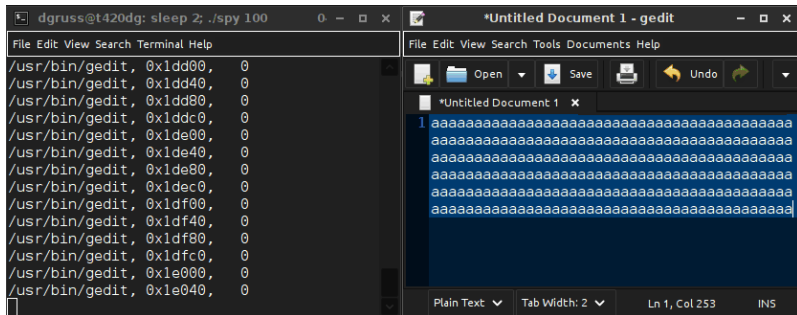
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 252 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 253 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 253'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dd00, 0
/usr/bin/gedit, 0x1dd40, 0
/usr/bin/gedit, 0x1dd80, 0
/usr/bin/gedit, 0x1ddc0, 0
/usr/bin/gedit, 0x1de00, 0
/usr/bin/gedit, 0x1de40, 0
/usr/bin/gedit, 0x1de80, 0
/usr/bin/gedit, 0x1dec0, 0
/usr/bin/gedit, 0x1df00, 0
/usr/bin/gedit, 0x1df40, 0
/usr/bin/gedit, 0x1df80, 0
/usr/bin/gedit, 0x1dfc0, 0
/usr/bin/gedit, 0x1e000, 0
/usr/bin/gedit, 0x1e040, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

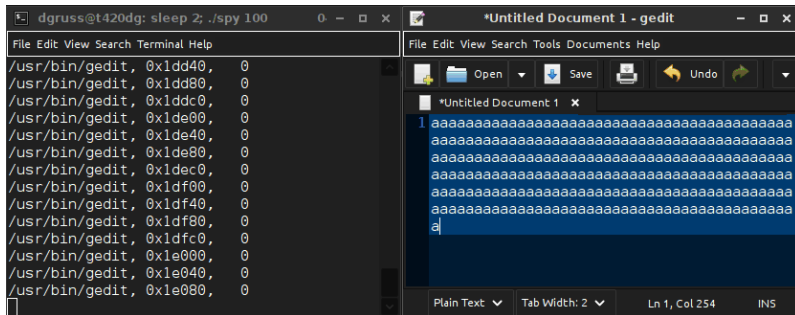
```
Open Save Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 253 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 254 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 254' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dd40, 0
```

```
/usr/bin/gedit, 0x1dd80, 0
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

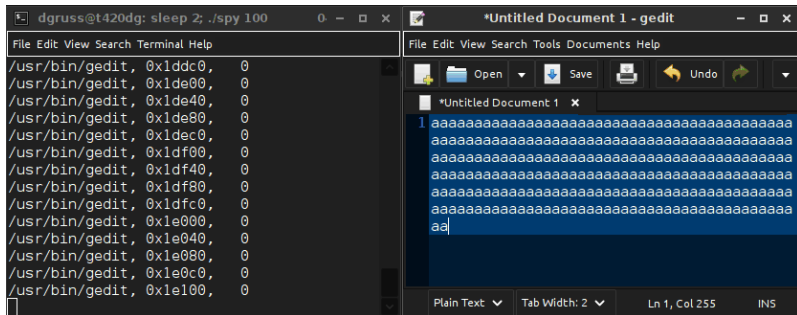
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
a
```

```
Plain Text Tab Width: 2 Ln 1, Col 254 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 255 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 255'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1ddc0, 0
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

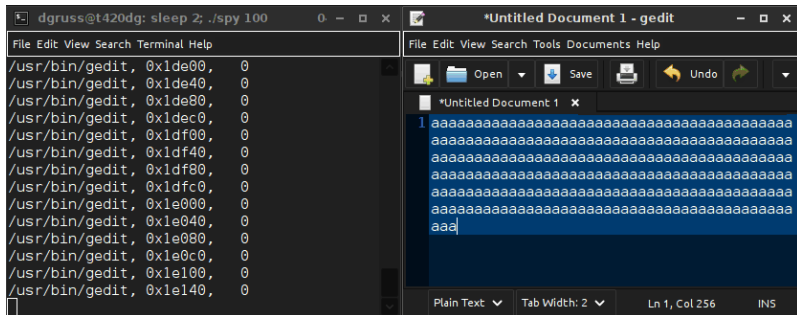
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aa
```

```
Plain Text Tab Width: 2 Ln 1, Col 255 INS
```



# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 256 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 256'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1de00, 0
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
/usr/bin/gedit, 0x1e140, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

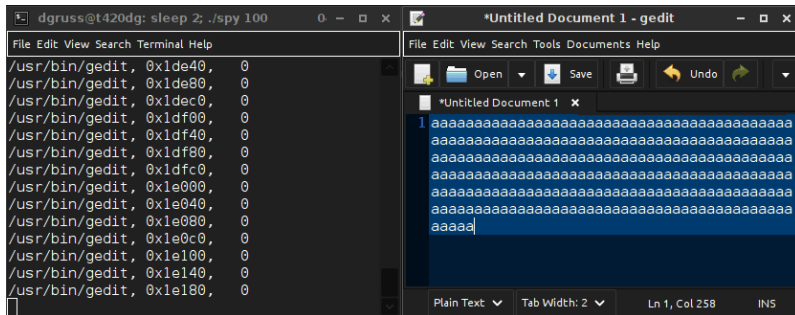
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 256 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 258 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 258'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1de40, 0
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
/usr/bin/gedit, 0x1e140, 0
```

```
/usr/bin/gedit, 0x1e180, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

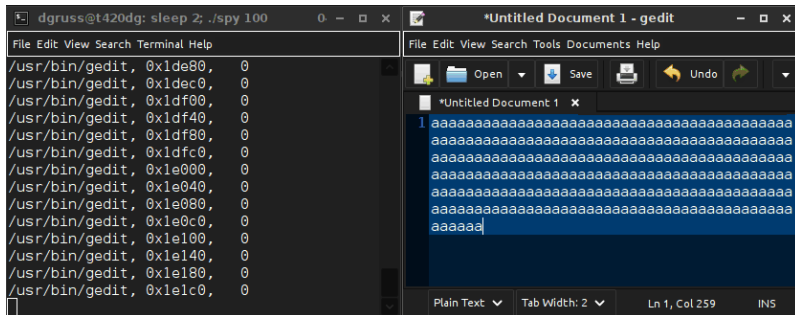
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 258 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 259 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 259' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1de80, 0
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
/usr/bin/gedit, 0x1e140, 0
```

```
/usr/bin/gedit, 0x1e180, 0
```

```
/usr/bin/gedit, 0x1e1c0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

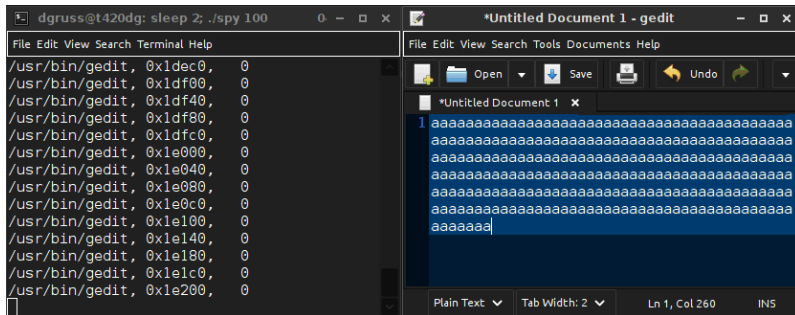
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 259 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 260 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 260' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1dec0, 0
```

```
/usr/bin/gedit, 0x1df00, 0
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
/usr/bin/gedit, 0x1e140, 0
```

```
/usr/bin/gedit, 0x1e180, 0
```

```
/usr/bin/gedit, 0x1e1c0, 0
```

```
/usr/bin/gedit, 0x1e200, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

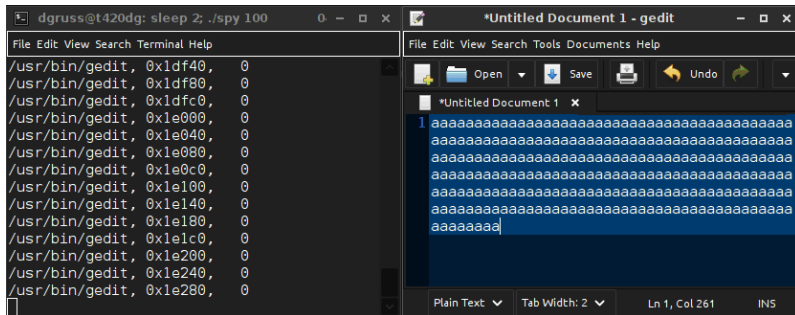
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 260 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 261 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 261' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1df40, 0
```

```
/usr/bin/gedit, 0x1df80, 0
```

```
/usr/bin/gedit, 0x1dfc0, 0
```

```
/usr/bin/gedit, 0x1e000, 0
```

```
/usr/bin/gedit, 0x1e040, 0
```

```
/usr/bin/gedit, 0x1e080, 0
```

```
/usr/bin/gedit, 0x1e0c0, 0
```

```
/usr/bin/gedit, 0x1e100, 0
```

```
/usr/bin/gedit, 0x1e140, 0
```

```
/usr/bin/gedit, 0x1e180, 0
```

```
/usr/bin/gedit, 0x1e1c0, 0
```

```
/usr/bin/gedit, 0x1e200, 0
```

```
/usr/bin/gedit, 0x1e240, 0
```

```
/usr/bin/gedit, 0x1e280, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

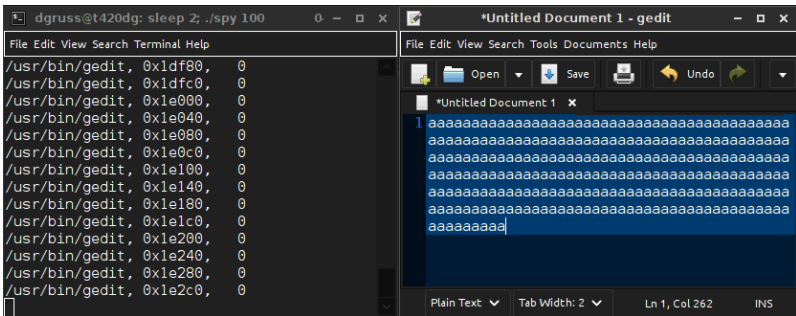
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

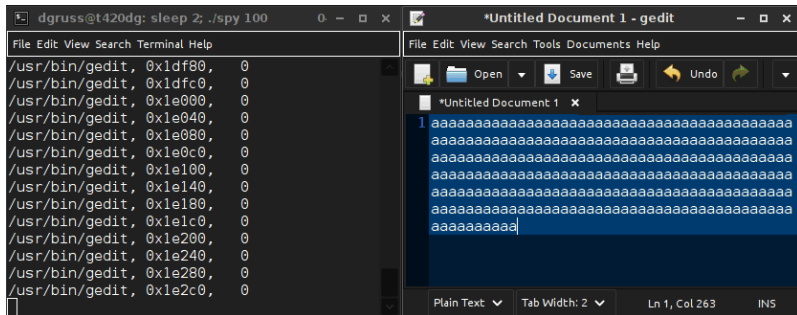
```
aaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 261 INS
```

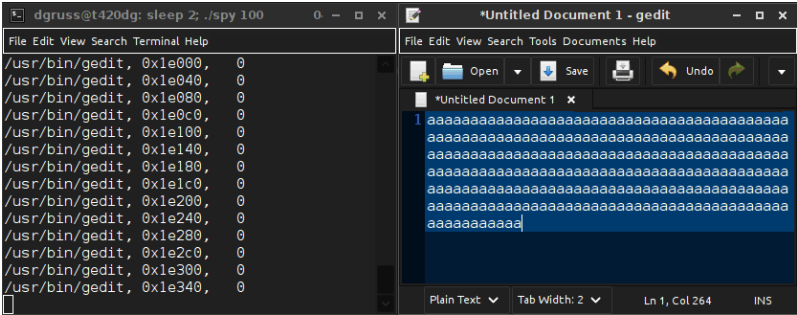
# Profiling a Single Event



## Profiling a Single Event

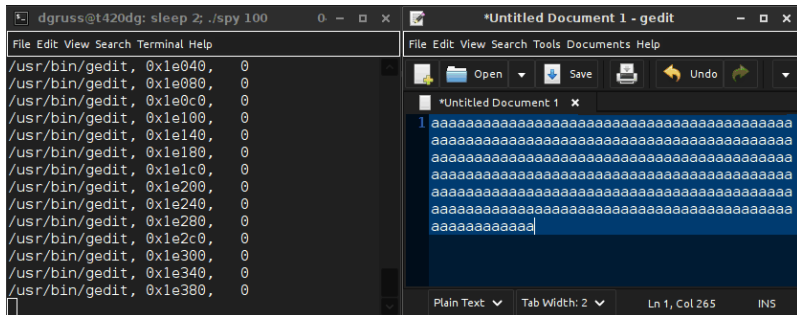


# Profiling a Single Event

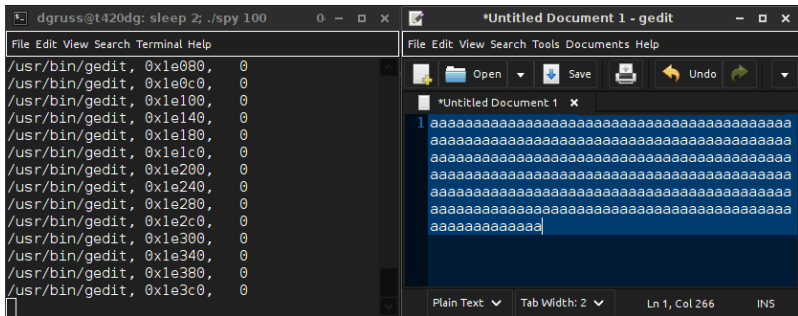




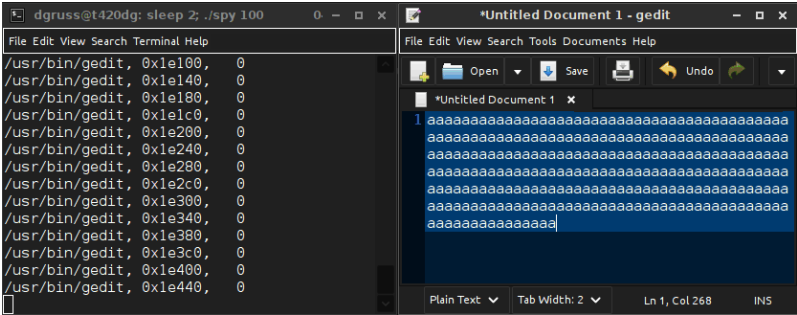
## Profiling a Single Event



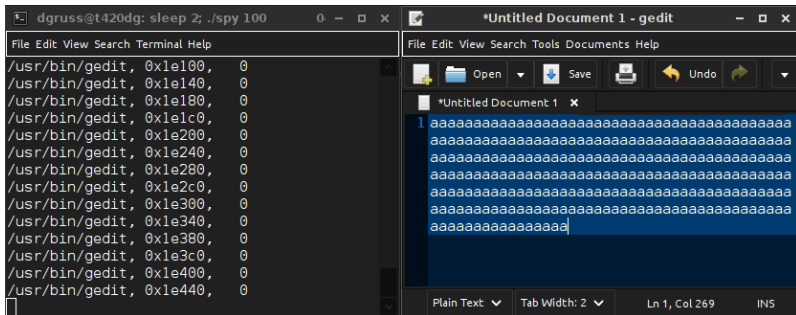
## Profiling a Single Event



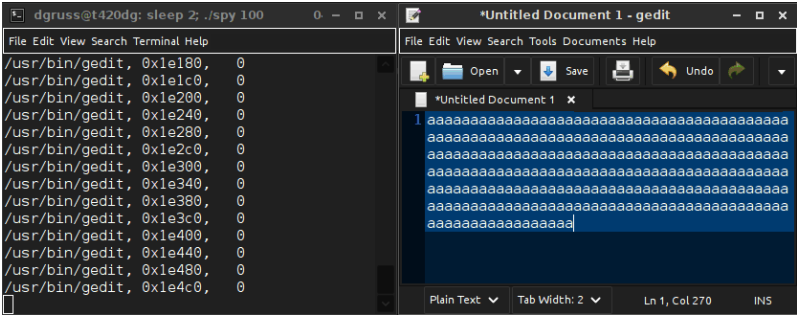
# Profiling a Single Event



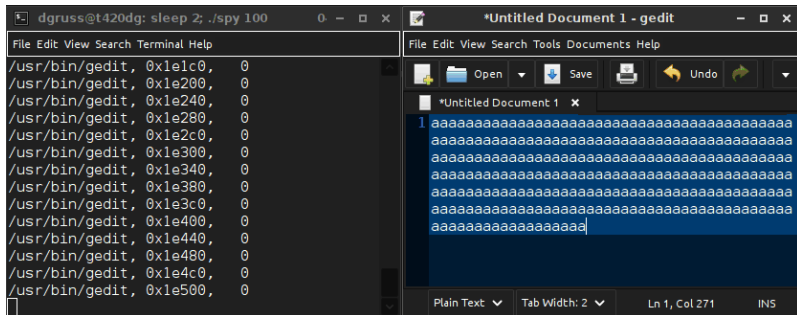
## Profiling a Single Event



# Profiling a Single Event



# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a single line of text consisting of 271 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 271'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e1c0, 0
```

```
/usr/bin/gedit, 0x1e200, 0
```

```
/usr/bin/gedit, 0x1e240, 0
```

```
/usr/bin/gedit, 0x1e280, 0
```

```
/usr/bin/gedit, 0x1e2c0, 0
```

```
/usr/bin/gedit, 0x1e300, 0
```

```
/usr/bin/gedit, 0x1e340, 0
```

```
/usr/bin/gedit, 0x1e380, 0
```

```
/usr/bin/gedit, 0x1e3c0, 0
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

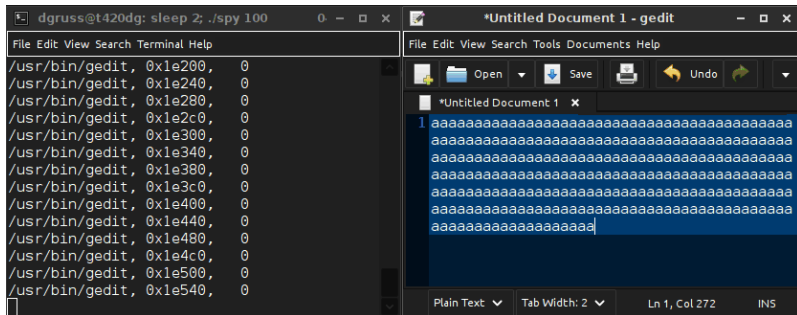
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 271 INS
```

# Profiling a Single Event



The image shows two side-by-side windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a single line of text consisting of 272 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 272'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e200, 0
```

```
/usr/bin/gedit, 0x1e240, 0
```

```
/usr/bin/gedit, 0x1e280, 0
```

```
/usr/bin/gedit, 0x1e2c0, 0
```

```
/usr/bin/gedit, 0x1e300, 0
```

```
/usr/bin/gedit, 0x1e340, 0
```

```
/usr/bin/gedit, 0x1e380, 0
```

```
/usr/bin/gedit, 0x1e3c0, 0
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
/usr/bin/gedit, 0x1e540, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

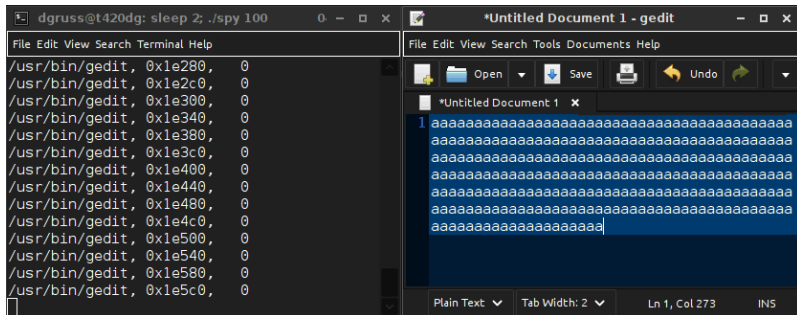
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 272 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 273 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 273' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0xe280, 0
```

```
/usr/bin/gedit, 0xe2c0, 0
```

```
/usr/bin/gedit, 0xe300, 0
```

```
/usr/bin/gedit, 0xe340, 0
```

```
/usr/bin/gedit, 0xe380, 0
```

```
/usr/bin/gedit, 0xe3c0, 0
```

```
/usr/bin/gedit, 0xe400, 0
```

```
/usr/bin/gedit, 0xe440, 0
```

```
/usr/bin/gedit, 0xe480, 0
```

```
/usr/bin/gedit, 0xe4c0, 0
```

```
/usr/bin/gedit, 0xe500, 0
```

```
/usr/bin/gedit, 0xe540, 0
```

```
/usr/bin/gedit, 0xe580, 0
```

```
/usr/bin/gedit, 0xe5c0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

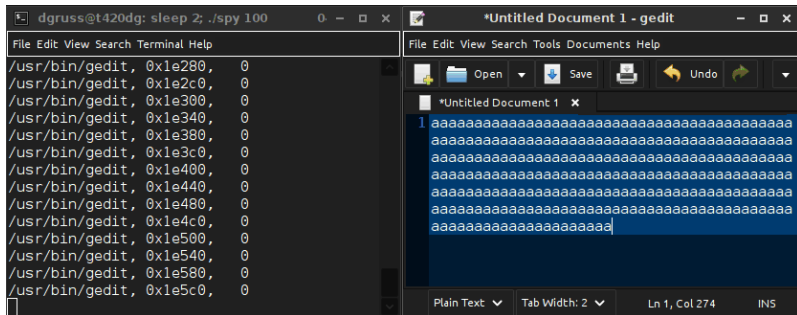
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 273 INS
```



# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 274 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 274'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0xe280, 0
```

```
/usr/bin/gedit, 0xe2c0, 0
```

```
/usr/bin/gedit, 0xe300, 0
```

```
/usr/bin/gedit, 0xe340, 0
```

```
/usr/bin/gedit, 0xe380, 0
```

```
/usr/bin/gedit, 0xe3c0, 0
```

```
/usr/bin/gedit, 0xe400, 0
```

```
/usr/bin/gedit, 0xe440, 0
```

```
/usr/bin/gedit, 0xe480, 0
```

```
/usr/bin/gedit, 0xe4c0, 0
```

```
/usr/bin/gedit, 0xe500, 0
```

```
/usr/bin/gedit, 0xe540, 0
```

```
/usr/bin/gedit, 0xe580, 0
```

```
/usr/bin/gedit, 0xe5c0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

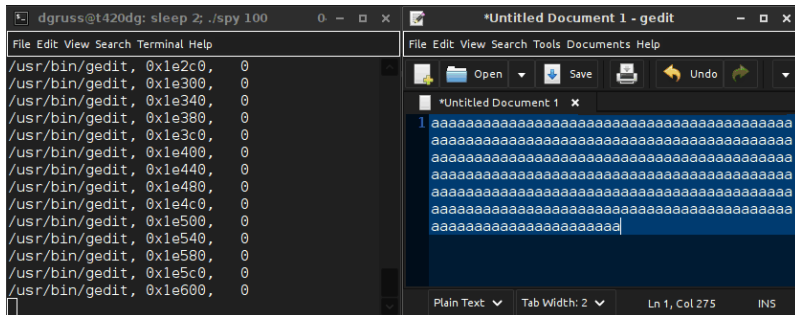
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 274 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all of which are 0. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of 275 'a' characters, which is highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 275'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e2c0, 0
```

```
/usr/bin/gedit, 0x1e300, 0
```

```
/usr/bin/gedit, 0x1e340, 0
```

```
/usr/bin/gedit, 0x1e380, 0
```

```
/usr/bin/gedit, 0x1e3c0, 0
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
/usr/bin/gedit, 0x1e540, 0
```

```
/usr/bin/gedit, 0x1e580, 0
```

```
/usr/bin/gedit, 0x1e5c0, 0
```

```
/usr/bin/gedit, 0x1e600, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

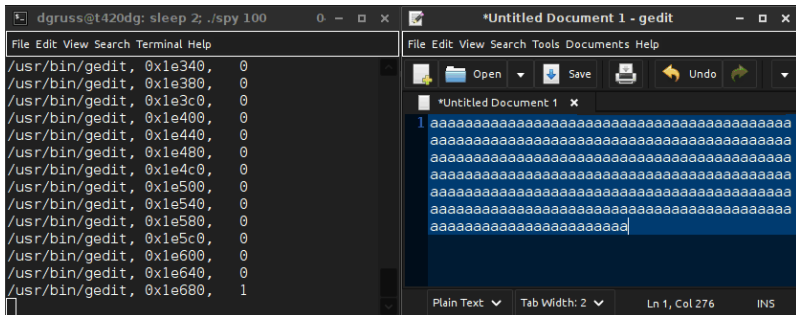
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 275 INS
```

# Profiling a Single Event



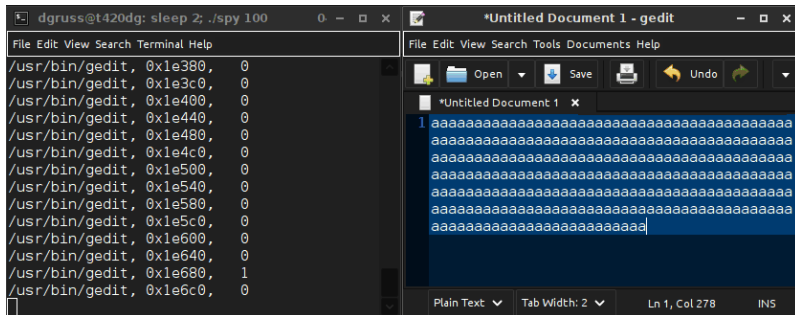
The screenshot displays two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It shows a list of memory addresses and their corresponding values, representing a memory dump or profiling data. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of a long string of 'a' characters, which is highlighted in blue. The terminal output is as follows:

```
dgruss@t420dg: sleep 2; ./spy 100
/usr/bin/gedit, 0x1e340, 0
/usr/bin/gedit, 0x1e380, 0
/usr/bin/gedit, 0x1e3c0, 0
/usr/bin/gedit, 0x1e400, 0
/usr/bin/gedit, 0x1e440, 0
/usr/bin/gedit, 0x1e480, 0
/usr/bin/gedit, 0x1e4c0, 0
/usr/bin/gedit, 0x1e500, 0
/usr/bin/gedit, 0x1e540, 0
/usr/bin/gedit, 0x1e580, 0
/usr/bin/gedit, 0x1e5c0, 0
/usr/bin/gedit, 0x1e600, 0
/usr/bin/gedit, 0x1e640, 0
/usr/bin/gedit, 0x1e680, 1
```

The gedit editor shows the following text:

```
*Untitled Document 1
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all set to 0, except for the last two which are 1. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 278 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 278'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e380, 0
```

```
/usr/bin/gedit, 0x1e3c0, 0
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
/usr/bin/gedit, 0x1e540, 0
```

```
/usr/bin/gedit, 0x1e580, 0
```

```
/usr/bin/gedit, 0x1e5c0, 0
```

```
/usr/bin/gedit, 0x1e600, 0
```

```
/usr/bin/gedit, 0x1e640, 0
```

```
/usr/bin/gedit, 0x1e680, 1
```

```
/usr/bin/gedit, 0x1e6c0, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

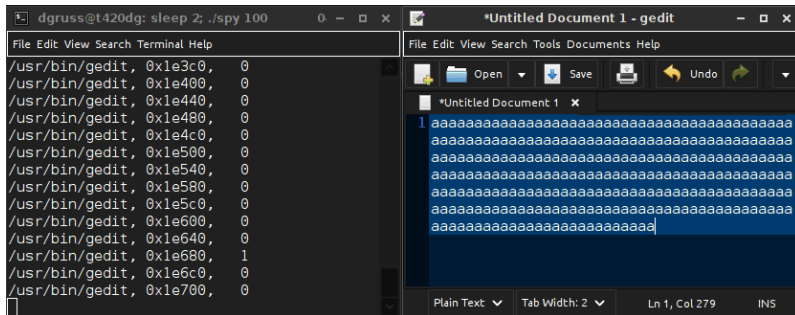
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 278 INS
```

# Profiling a Single Event



The image shows two side-by-side windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, all pointing to '/usr/bin/gedit'. The values are mostly 0, except for one which is 1. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document filled with the character 'a'. The status bar at the bottom indicates 'Ln 1, Col 279' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e3c0, 0
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
/usr/bin/gedit, 0x1e540, 0
```

```
/usr/bin/gedit, 0x1e580, 0
```

```
/usr/bin/gedit, 0x1e5c0, 0
```

```
/usr/bin/gedit, 0x1e600, 0
```

```
/usr/bin/gedit, 0x1e640, 0
```

```
/usr/bin/gedit, 0x1e680, 1
```

```
/usr/bin/gedit, 0x1e6c0, 0
```

```
/usr/bin/gedit, 0x1e700, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

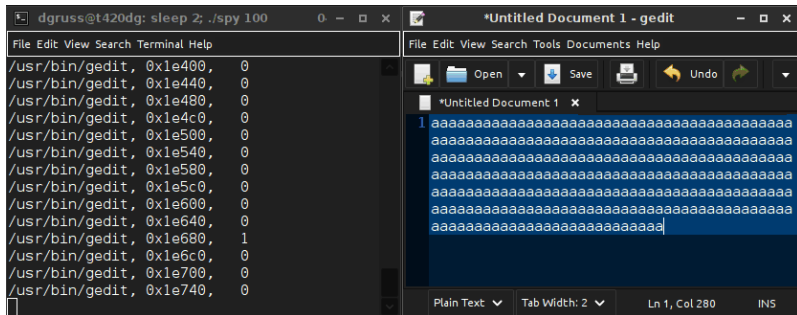
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 279 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, mostly 0, with a '1' at address 0x1e680. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of 280 'a' characters, which is highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 280'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e400, 0
```

```
/usr/bin/gedit, 0x1e440, 0
```

```
/usr/bin/gedit, 0x1e480, 0
```

```
/usr/bin/gedit, 0x1e4c0, 0
```

```
/usr/bin/gedit, 0x1e500, 0
```

```
/usr/bin/gedit, 0x1e540, 0
```

```
/usr/bin/gedit, 0x1e580, 0
```

```
/usr/bin/gedit, 0x1e5c0, 0
```

```
/usr/bin/gedit, 0x1e600, 0
```

```
/usr/bin/gedit, 0x1e640, 0
```

```
/usr/bin/gedit, 0x1e680, 1
```

```
/usr/bin/gedit, 0x1e6c0, 0
```

```
/usr/bin/gedit, 0x1e700, 0
```

```
/usr/bin/gedit, 0x1e740, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Print Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

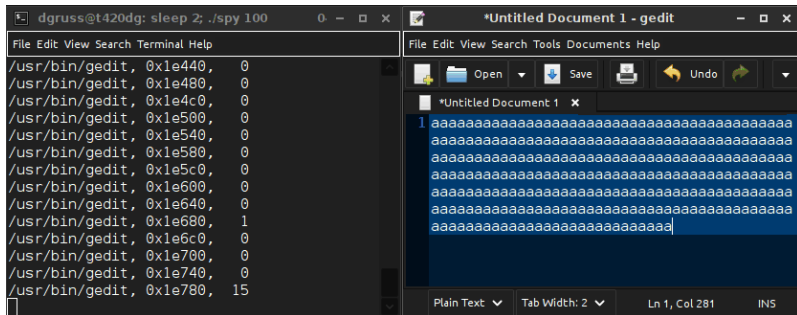
```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 280 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, with the last entry being '/usr/bin/gedit, 0x1e780, 15'. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 281 'a' characters. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 281' and 'INS' mode.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e440, 0
/usr/bin/gedit, 0x1e480, 0
/usr/bin/gedit, 0x1e4c0, 0
/usr/bin/gedit, 0x1e500, 0
/usr/bin/gedit, 0x1e540, 0
/usr/bin/gedit, 0x1e580, 0
/usr/bin/gedit, 0x1e5c0, 0
/usr/bin/gedit, 0x1e600, 0
/usr/bin/gedit, 0x1e640, 0
/usr/bin/gedit, 0x1e680, 1
/usr/bin/gedit, 0x1e6c0, 0
/usr/bin/gedit, 0x1e700, 0
/usr/bin/gedit, 0x1e740, 0
/usr/bin/gedit, 0x1e780, 15
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

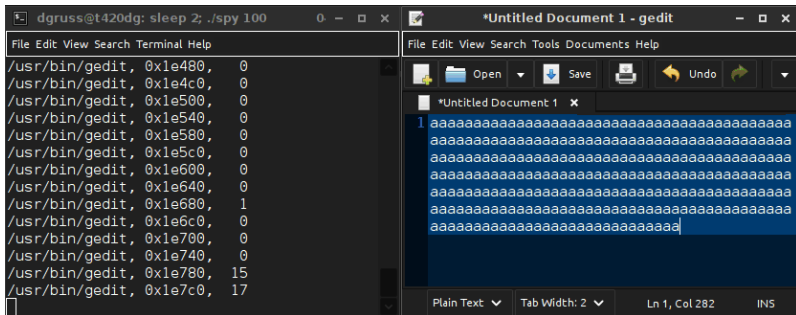
```
Open Save Undo
```

```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 281 INS
```

# Profiling a Single Event



The screenshot shows two windows side-by-side. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, likely from a memory dump or profiler output. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of a long string of 'a' characters, which is highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 282' and 'INS'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e480, 0
/usr/bin/gedit, 0x1e4c0, 0
/usr/bin/gedit, 0x1e500, 0
/usr/bin/gedit, 0x1e540, 0
/usr/bin/gedit, 0x1e580, 0
/usr/bin/gedit, 0x1e5c0, 0
/usr/bin/gedit, 0x1e600, 0
/usr/bin/gedit, 0x1e640, 0
/usr/bin/gedit, 0x1e680, 1
/usr/bin/gedit, 0x1e6c0, 0
/usr/bin/gedit, 0x1e700, 0
/usr/bin/gedit, 0x1e740, 0
/usr/bin/gedit, 0x1e780, 15
/usr/bin/gedit, 0x1e7c0, 17
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

```
Open Save Undo
```

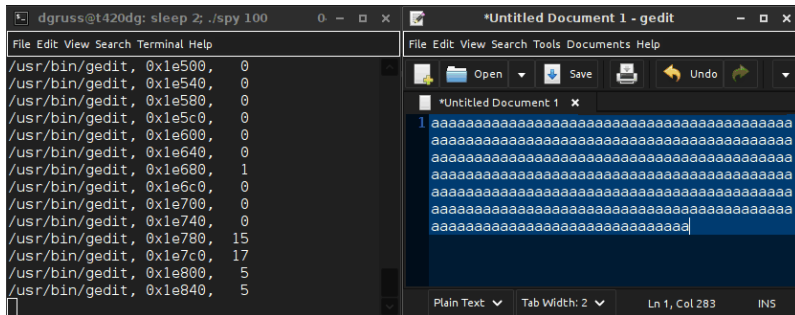
```
*Untitled Document 1 x
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 282 INS
```



# Profiling a Single Event



The screenshot shows two windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, likely from a memory profiler. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 283 'a' characters, which is highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 283'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e500, 0
/usr/bin/gedit, 0x1e540, 0
/usr/bin/gedit, 0x1e580, 0
/usr/bin/gedit, 0x1e5c0, 0
/usr/bin/gedit, 0x1e600, 0
/usr/bin/gedit, 0x1e640, 0
/usr/bin/gedit, 0x1e680, 1
/usr/bin/gedit, 0x1e6c0, 0
/usr/bin/gedit, 0x1e700, 0
/usr/bin/gedit, 0x1e740, 0
/usr/bin/gedit, 0x1e780, 15
/usr/bin/gedit, 0x1e7c0, 17
/usr/bin/gedit, 0x1e800, 5
/usr/bin/gedit, 0x1e840, 5
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

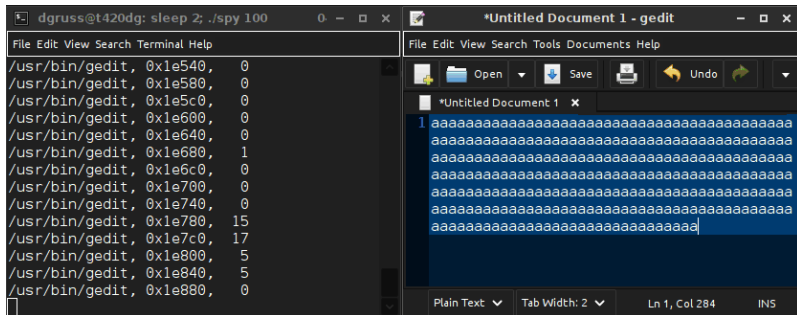
```
Open Save Undo
```

```
*Untitled Document 1
```

```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 283 INS
```

# Profiling a Single Event



The screenshot shows two windows. The left window is a terminal titled 'dgruss@t420dg: sleep 2; ./spy 100'. It displays a list of memory addresses and their corresponding values, representing a profiling event. The right window is a gedit editor titled '\*Untitled Document 1 - gedit'. It shows a document with a single line of text consisting of 284 'a' characters, which is highlighted in blue. The status bar at the bottom of the gedit window indicates 'Ln 1, Col 284' and 'INS'.

```
dgruss@t420dg: sleep 2; ./spy 100
```

```
File Edit View Search Terminal Help
```

```
/usr/bin/gedit, 0x1e540, 0
/usr/bin/gedit, 0x1e580, 0
/usr/bin/gedit, 0x1e5c0, 0
/usr/bin/gedit, 0x1e600, 0
/usr/bin/gedit, 0x1e640, 0
/usr/bin/gedit, 0x1e680, 1
/usr/bin/gedit, 0x1e6c0, 0
/usr/bin/gedit, 0x1e700, 0
/usr/bin/gedit, 0x1e740, 0
/usr/bin/gedit, 0x1e780, 15
/usr/bin/gedit, 0x1e7c0, 17
/usr/bin/gedit, 0x1e800, 5
/usr/bin/gedit, 0x1e840, 5
/usr/bin/gedit, 0x1e880, 0
```

```
*Untitled Document 1 - gedit
```

```
File Edit View Search Tools Documents Help
```

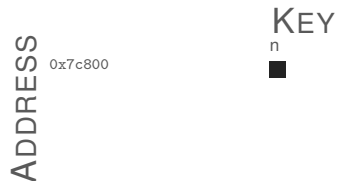
```
Open Save Undo
```

```
*Untitled Document 1 x
```

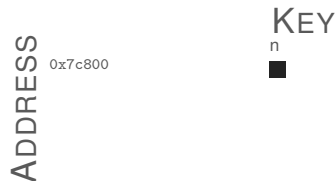
```
1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

```
Plain Text Tab Width: 2 Ln 1, Col 284 INS
```

# Profiling Phase: 1 Event, 1 Address



# Profiling Phase: 1 Event, 1 Address



Example: Cache Hit Ratio for  $(0x7c800, n)$ : 200 / 200

# Profiling Phase: All Events, 1 Address



# Profiling Phase: All Events, 1 Address



Example: Cache Hit Ratio for  $(0x7c800, u)$ : 13 / 200

# Profiling Phase: All Events, 1 Address



Distinguish  $n$  from other keys by monitoring 0x7c800

# Profiling Phase: All Events, All Addresses





# Exploitation Phase

- Monitor addresses from Cache Template

# Exploitation Phase

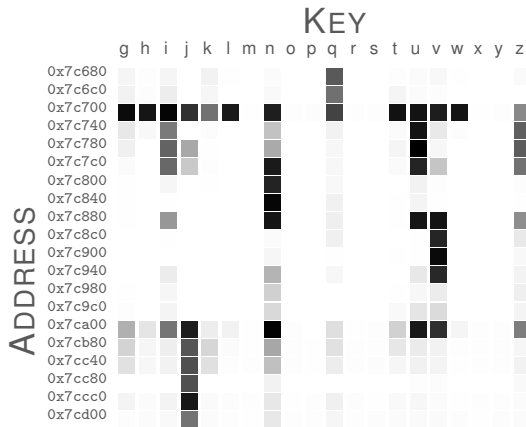
- Monitor addresses from Cache Template
- Report to log file / attacker

# Exploitation Phase

- Monitor addresses from Cache Template
- Report to log file / attacker
- Manual analysis of log file
  - Find password in keypress log, etc.

# Example Attack: Keylogging

- Linux with GTK: monitor keystrokes of specific keys
- Detect groups of keys
- Some keys distinct



# Example Attack: AES T-Tables

## AES T-Table implementation from OpenSSL 1.0.2

- Known-plaintext attack
- Events: encryption with only one fixed key byte

# Example Attack: AES T-Tables

## AES T-Table implementation from OpenSSL 1.0.2

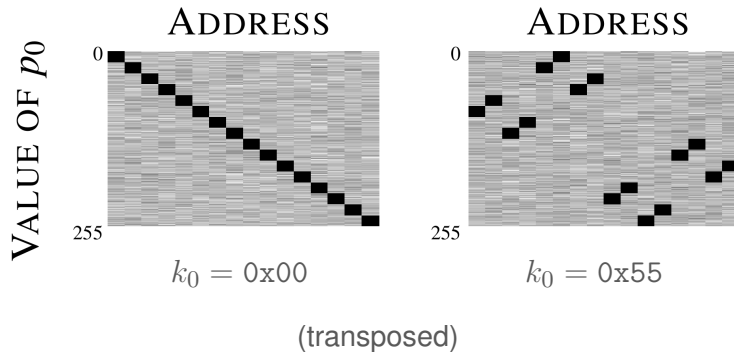
- Known-plaintext attack
- Events: encryption with only one fixed key byte
- Profile each event

# Example Attack: AES T-Tables

## AES T-Table implementation from OpenSSL 1.0.2

- Known-plaintext attack
- Events: encryption with only one fixed key byte
- Profile each event
- Exploitation phase: an auto-generated first-round attack

# Example Attack: AES T-Tables Template





# Cache Template Attacks: How hard is it?

- We let students attack different programs
- Leakage typically found within 30 minutes
- Wide range of programs

# Cache Template Attacks: Example Applications

- Keystrokes in nano, quasselclient, xterm, Pidgin, gksu, Firefox, GNU Emacs (on OSX), mousepad, user32.dll on Windows
- Keystrokes specifically in the password field in KeePassX
- Font rendering in libharfbuzz (used by many Linux applications)
- Action buttons in VLC (play, mute, ...)
- Receiving characters in netcat
- Distinguishing keygroups in gksu
- Monitoring checkbox clicking in gksu

# Cache Template Attacks: Conclusion

- Technique to find any cache side-channel leakage

# Cache Template Attacks: Conclusion

- Technique to find any cache side-channel leakage
- Works on Intel, ARM, and some AMD
- Works even with unknown binaries

# Cache Template Attacks: Conclusion

- Technique to find any cache side-channel leakage
- Works on Intel, ARM, and some AMD
- Works even with unknown binaries
- Large scale automated cache attacks possible!

# Cache Template Attacks: Conclusion

- Technique to find any cache side-channel leakage
- Works on Intel, ARM, and some AMD
- Works even with unknown binaries
- Large scale automated cache attacks possible!
- But: still requires shared memory

# What about Prime+Probe?

- Good: no shared memory

# What about Prime+Probe?

- Good: no shared memory
- Bad: much more difficult due to noise



# What about Prime+Probe?

- Good: no shared memory
- Bad: much more difficult due to noise

→ Let's do something more fun with Prime+Probe!

1. Caches

2. Cache attacks

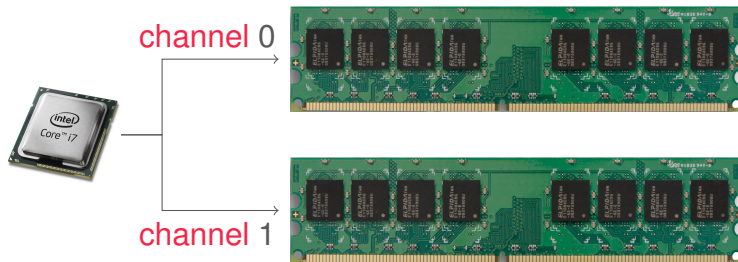
3. Cache Template Attacks

4. Rowhammer

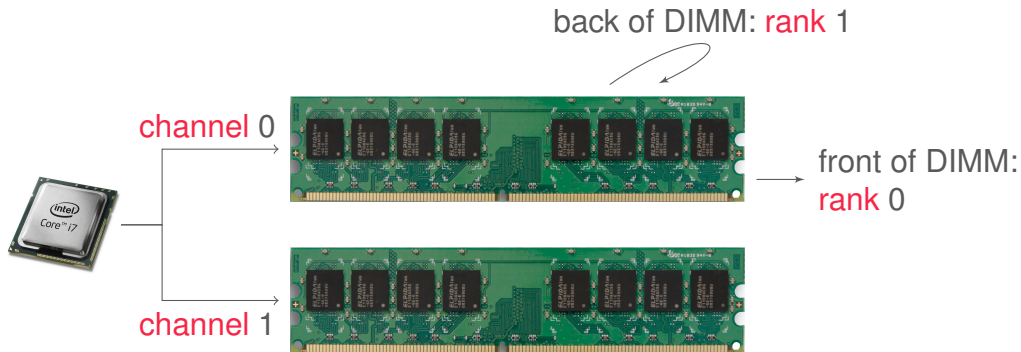
# DRAM organization example



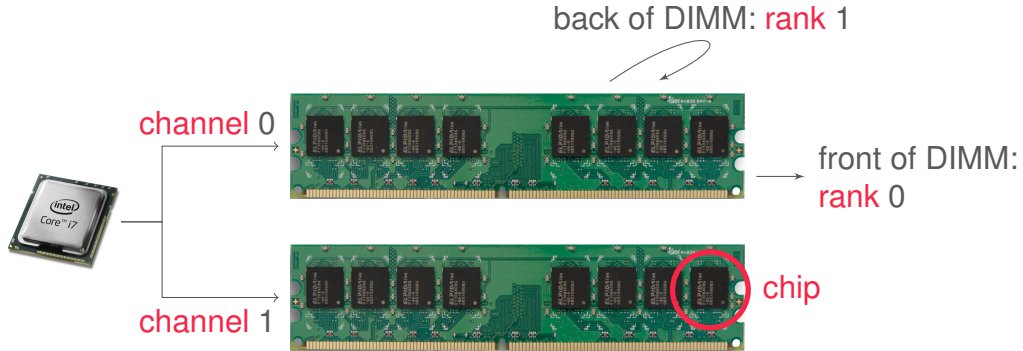
# DRAM organization example



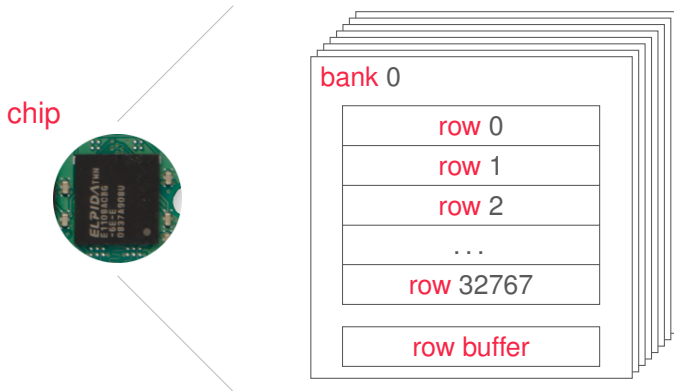
# DRAM organization example



# DRAM organization example



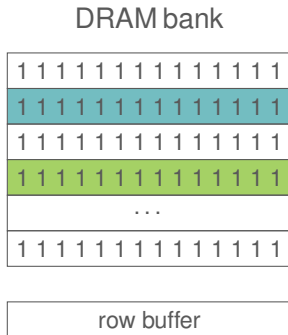
# DRAM organization example



- bits in cells in rows
- access: **activate** row, copy to row buffer
- cells leak → refresh necessary
- cells leak faster upon proximate accesses

# Rowhammer

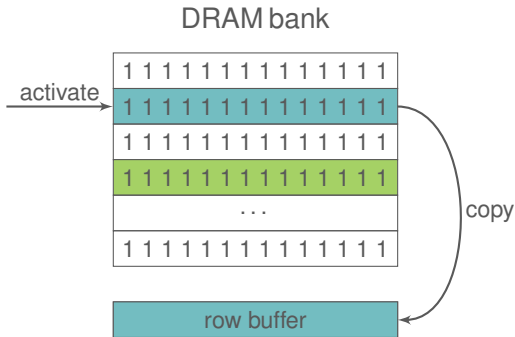
*“It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after” – Motherboard Vice*





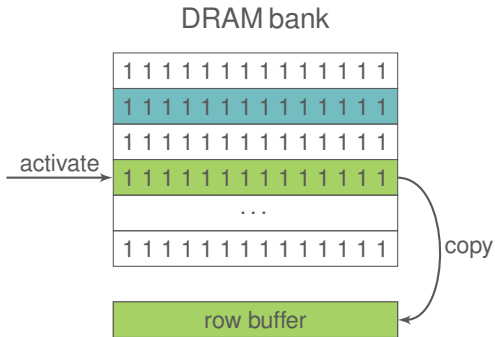
# Rowhammer

*“It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after” – Motherboard Vice*



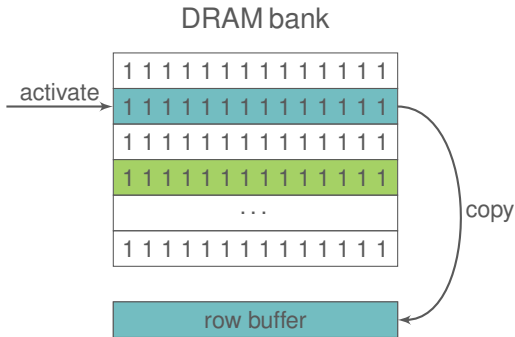
# Rowhammer

*“It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after” – Motherboard Vice*



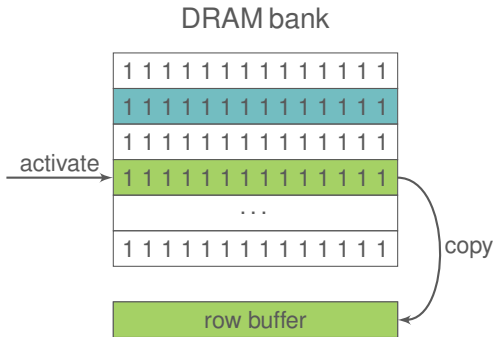
# Rowhammer

*“It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after” – Motherboard Vice*



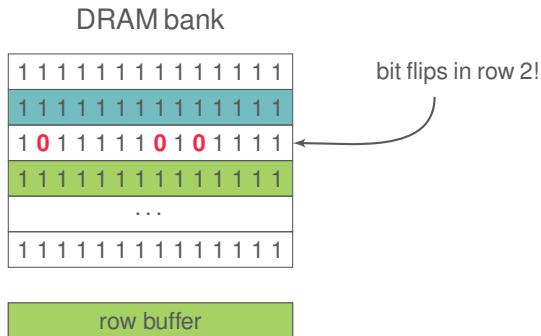
# Rowhammer

*“It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after” – Motherboard Vice*

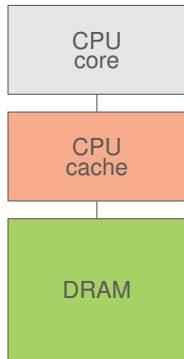


# Rowhammer

*"It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after"* – Motherboard Vice



# Impact of the CPU cache



- only **non-cached accesses** reach DRAM
  - original attacks use `clflush` instruction
- flush line from cache
- next access will be served from DRAM

# Rowhammer

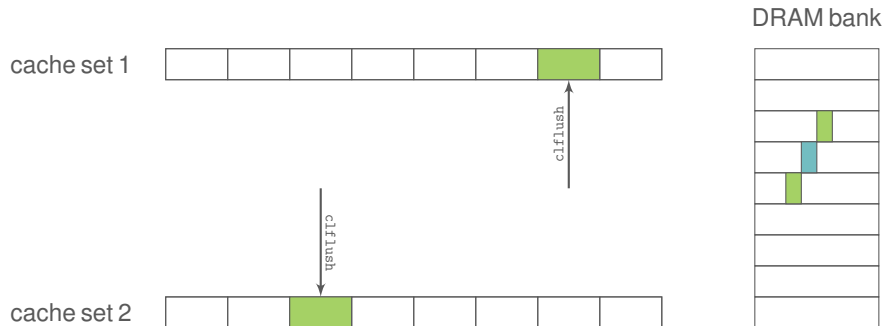
- Rowhammer: DRAM bug that causes bit flips [Kim et al., 2014]
- Bug used in security exploits [Seaborn, 2015]
- Need **non-cached accesses** to reach DRAM
- Very similar to Flush+Reload

# Rowhammer (with clflush)

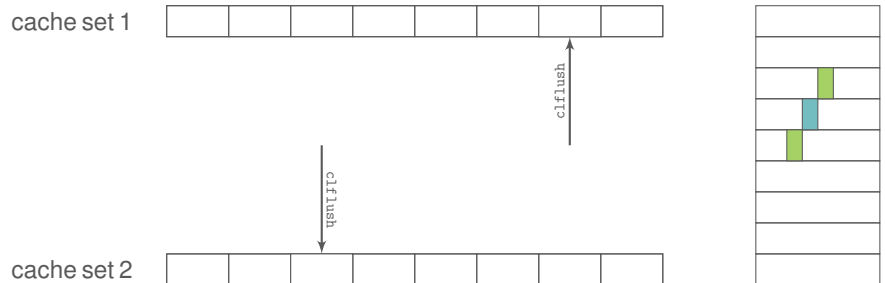




# Rowhammer (with clflush)



# Rowhammer (with clflush)



# Rowhammer (with clflush)

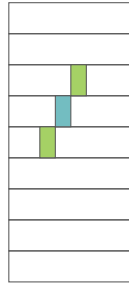
cache set 1



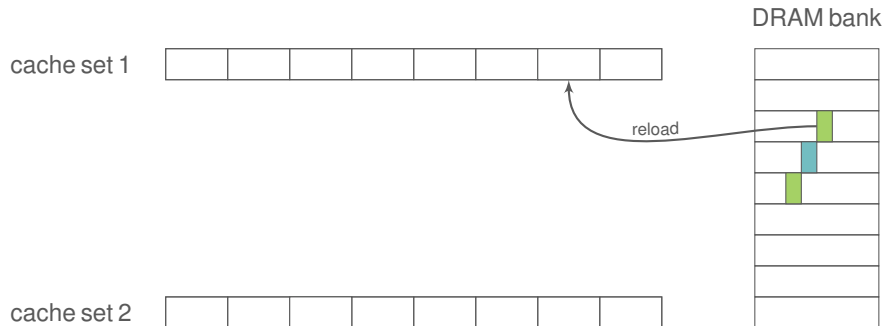
cache set 2



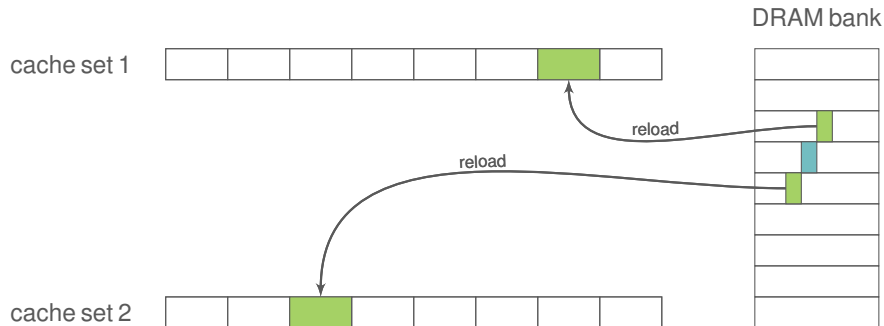
DRAM bank



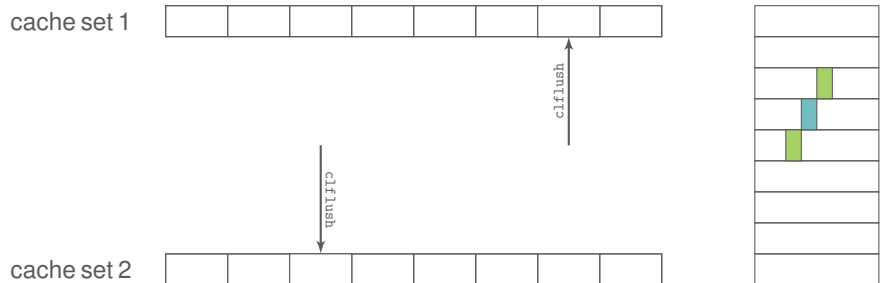
# Rowhammer (with clflush)



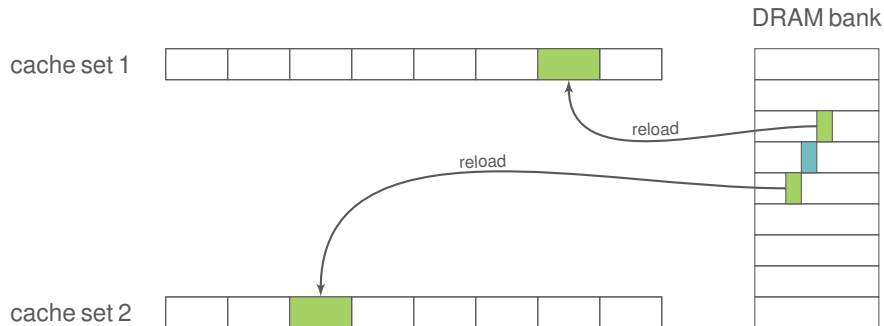
# Rowhammer (with clflush)



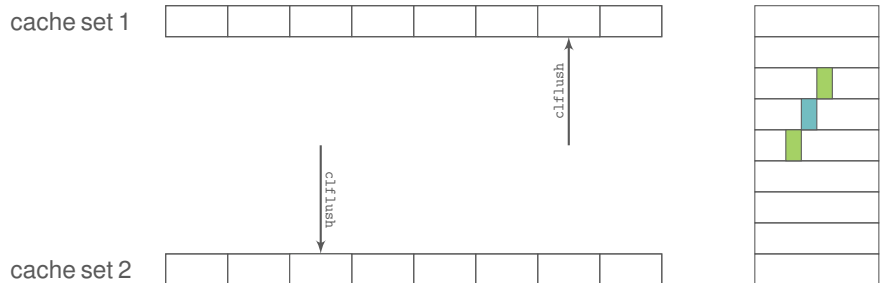
# Rowhammer (with clflush)



# Rowhammer (with clflush)

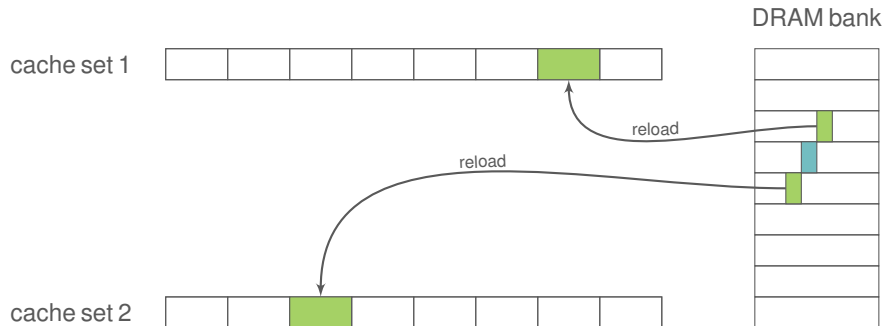


# Rowhammer (with clflush)

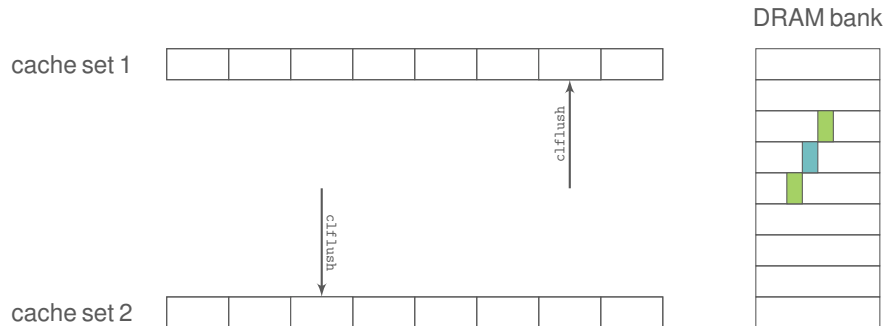




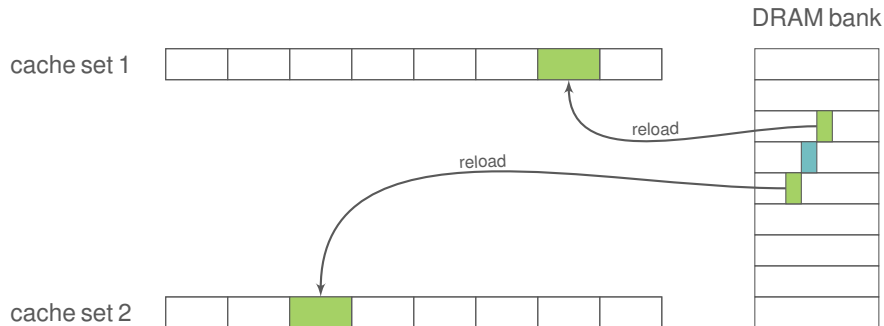
# Rowhammer (with clflush)



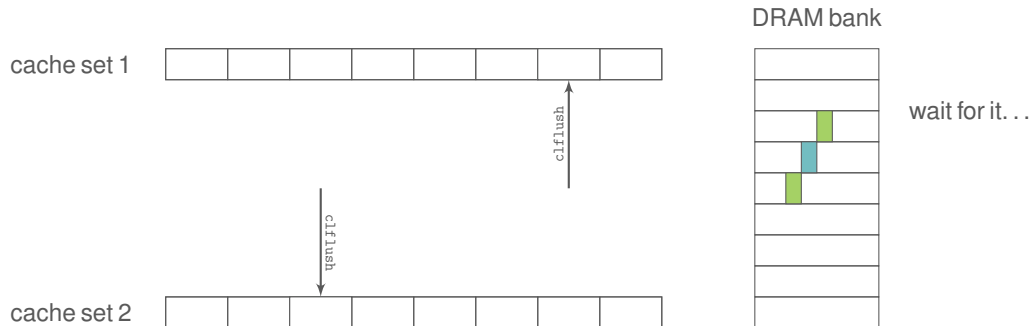
# Rowhammer (with clflush)



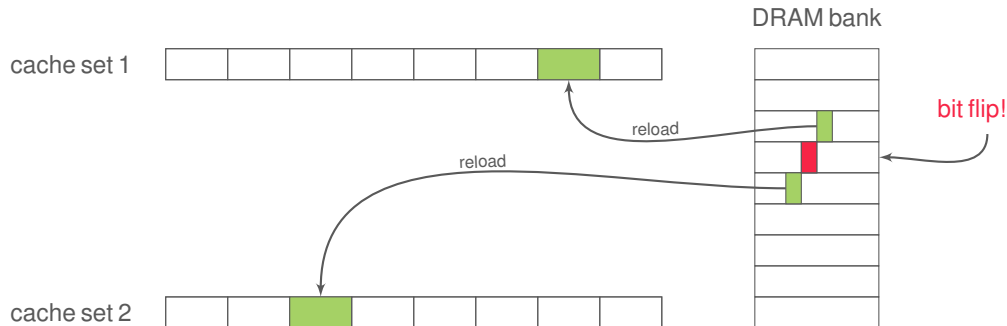
# Rowhammer (with clflush)



# Rowhammer (with clflush)



# Rowhammer (with clflush)



# Rowhammer without clflush?

Idea: use Prime+Probe for Rowhammer instead of Flush+Reload

# Rowhammer without clflush

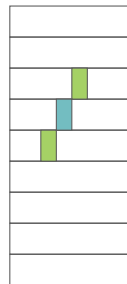
cache set 1



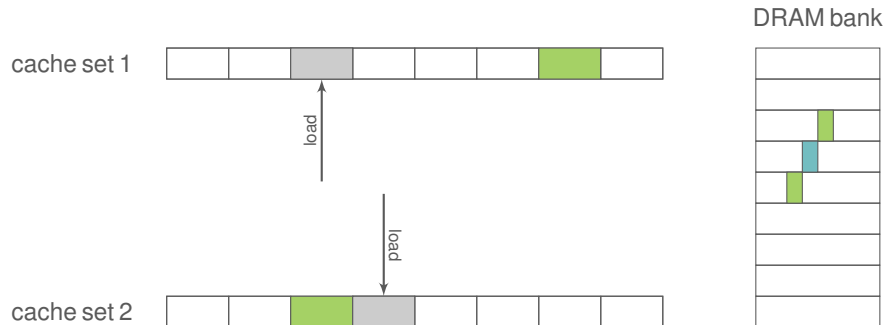
cache set 2



DRAM bank



# Rowhammer without clflush

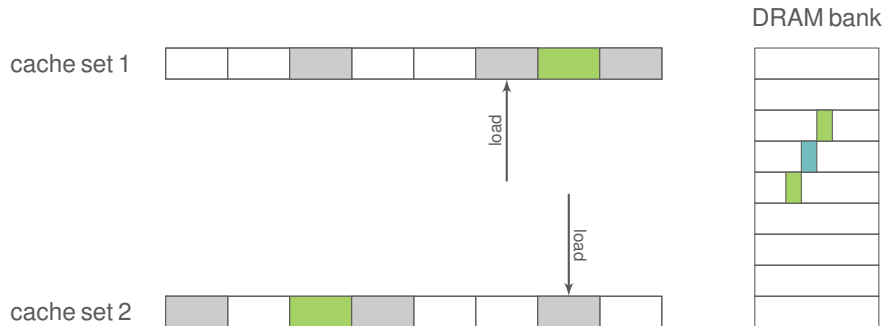




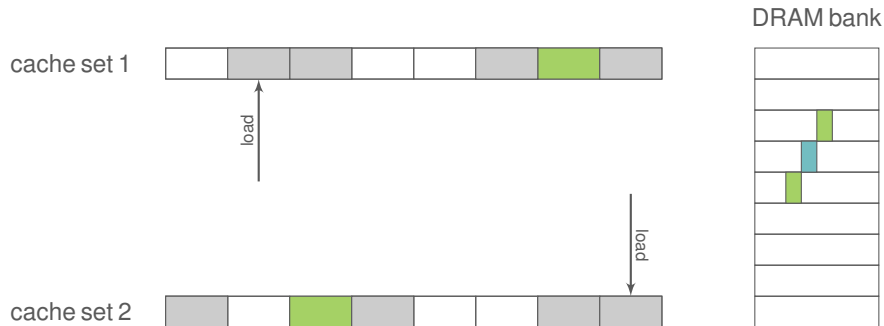
# Rowhammer without clflush



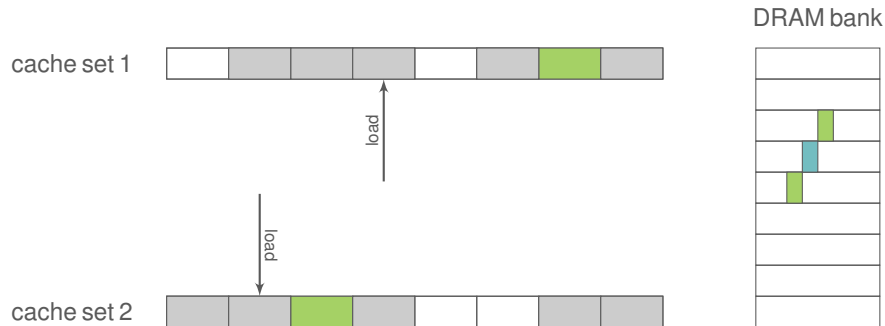
# Rowhammer without clflush



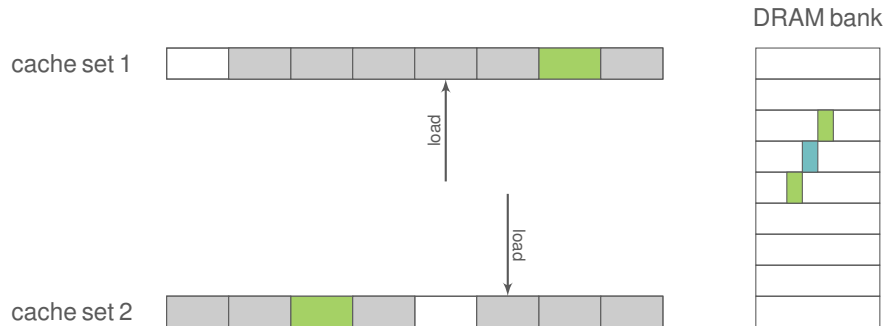
# Rowhammer without clflush



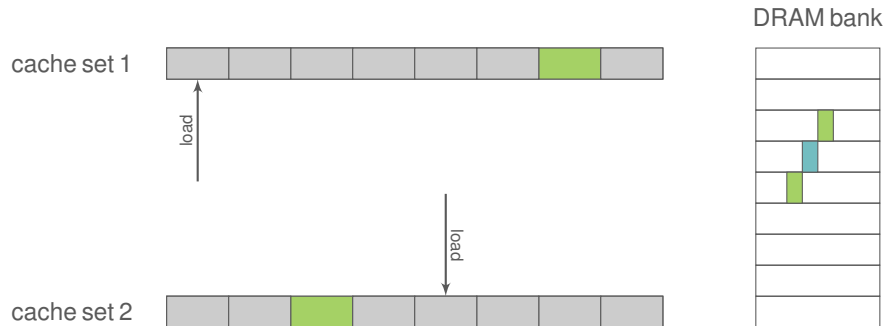
# Rowhammer without clflush



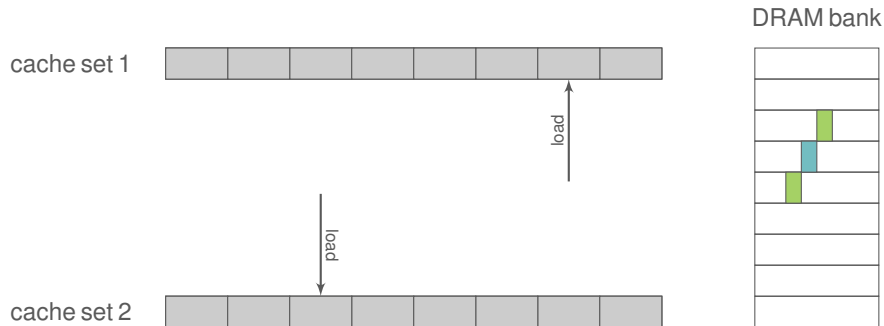
# Rowhammer without clflush



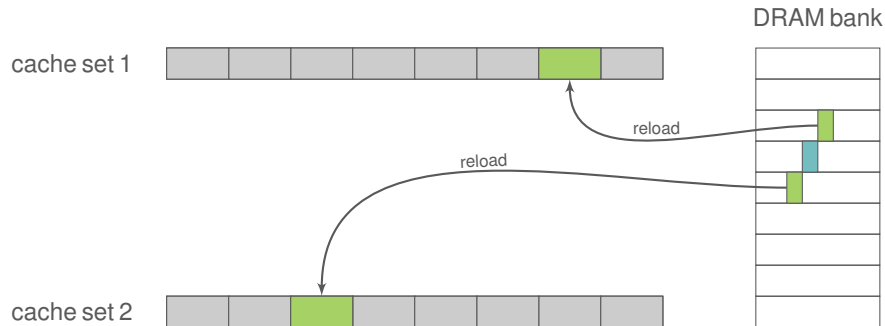
# Rowhammer without clflush



# Rowhammer without clflush



# Rowhammer without clflush





# Rowhammer without clflush

cache set 1

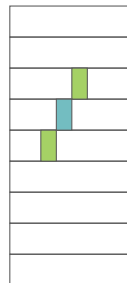


repeat!

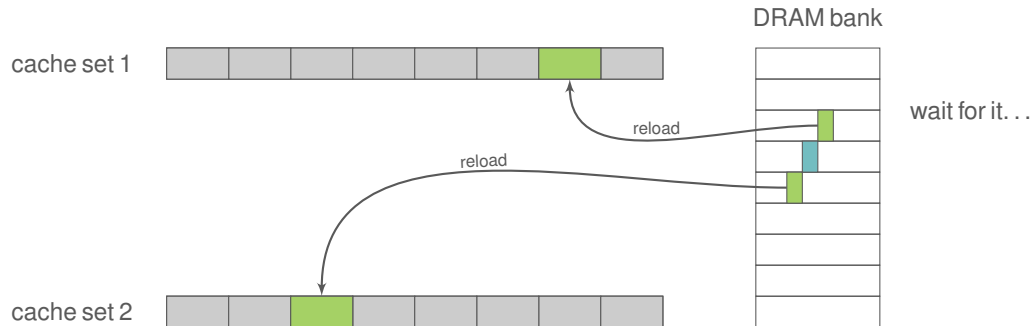
cache set 2



DRAM bank



# Rowhammer without clflush



# Rowhammer without clflush

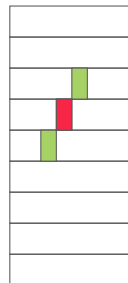
cache set 1



cache set 2



DRAM bank



bit flip!



# Rowhammer without clflush

## Challenges:

1. How to get accurate timing (in JS)?
2. How to get physical addresses (in JS)?
3. Which physical addresses to access?
4. In which order to access them?

# Rowhammer without clflush

## Challenges:

1. How to get accurate timing (in JS)? → easy
2. How to get physical addresses (in JS)? → easy
3. Which physical addresses to access? → already solved
4. In which order to access them? → details today!

# Accurate timing in JS (easy)

- native code: `rdtsc`
- JavaScript: `window.performance.now()`

# Physical addresses in JS (easy)

- fixed map between physical addresses and DRAM
- we reverse-engineered it [Pessl et al., 2015]
- OS optimization: use 2MB pages
- last 21 bits equal for physical address, virtual address, JS array index [Gruss et al., 2015a]
- several DRAM rows per 2MB page

# Which addresses to access? (already solved)

- use eviction  $\rightarrow$  congruent addresses
- slice mapping is known [Maurice et al., 2015a]



## Which addresses to access? (already solved)

- use eviction → congruent addresses
- slice mapping is known [Maurice et al., 2015a]

Remaining challenge: “in which order to access them?”

# Replacement policy on older CPUs

“LRU eviction” memory accesses

cache set



# Replacement policy on older CPUs

“LRU eviction” memory accesses

cache set



- LRU replacement policy: oldest entry first

# Replacement policy on older CPUs

“LRU eviction” memory accesses

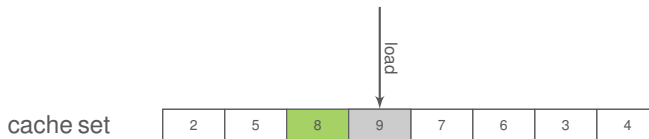
cache set

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 8 | 1 | 7 | 6 | 3 | 4 |
|---|---|---|---|---|---|---|---|

- LRU replacement policy: oldest entry first
- timestamps for every cache line

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp



# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

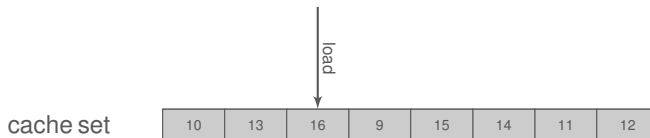
“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on older CPUs

“LRU eviction” memory accesses



- LRU replacement policy: oldest entry first
- timestamps for every cache line
- access updates timestamp

# Replacement policy on recent CPUs

“LRU eviction” memory accesses

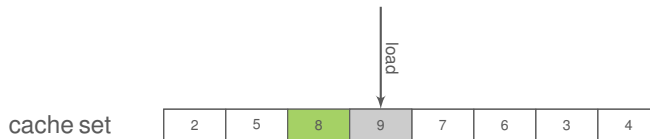
cache set

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 2 | 5 | 8 | 1 | 7 | 6 | 3 | 4 |
|---|---|---|---|---|---|---|---|

- no LRU replacement

# Replacement policy on recent CPUs

“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

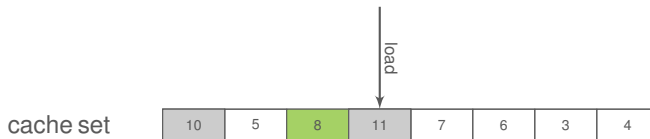
“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

“LRU eviction” memory accesses



- no LRU replacement



# Replacement policy on recent CPUs

“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

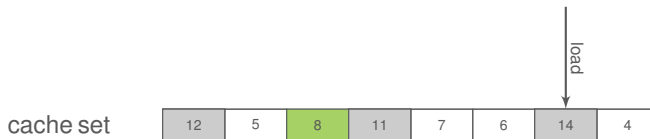
“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

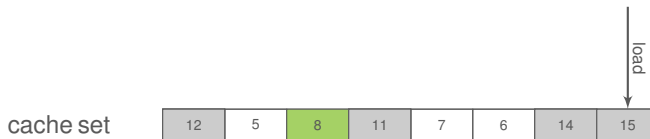
“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

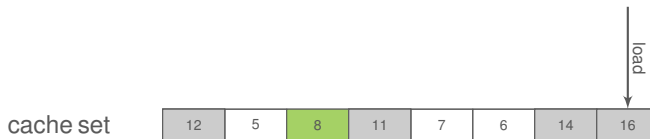
“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

“LRU eviction” memory accesses



- no LRU replacement

# Replacement policy on recent CPUs

“LRU eviction” memory accesses

cache set

|    |   |   |    |   |   |    |    |
|----|---|---|----|---|---|----|----|
| 12 | 5 | 8 | 11 | 7 | 6 | 14 | 16 |
|----|---|---|----|---|---|----|----|

- no LRU replacement
- only 75% success rate on Haswell

# Replacement policy on recent CPUs

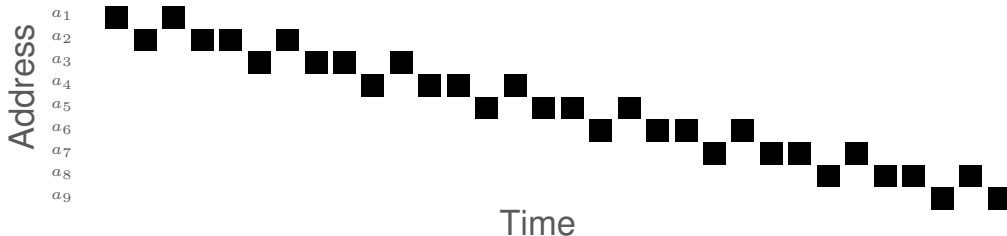
“LRU eviction” memory accesses

cache set

|    |   |   |    |   |   |    |    |
|----|---|---|----|---|---|----|----|
| 12 | 5 | 8 | 11 | 7 | 6 | 14 | 16 |
|----|---|---|----|---|---|----|----|

- no LRU replacement
- only 75% success rate on Haswell
- more accesses → higher success rate, but **too slow**

# Cache eviction strategy (the beginning)





# Cache eviction strategy (new representation)

Represent accesses as a sequence of numbers:

- 1, 2, 1, 2, 2, 3, 2, 3, 3, 4, 3, 4, ...
- can be a long sequence

# Cache eviction strategy properties

- all addresses are congruent
- indistinguishable in terms of the eviction strategy

# Cache eviction strategy properties

- all addresses are congruent

→ indistinguishable in terms of the eviction strategy

- 2, 5, 4, 6, 3, 1, 2, 5, 6, 4, 1, 3, ... all equivalent!

→ 1, 2, 3, 4, 5, 6

# Cache eviction strategy properties

- all addresses are congruent
  - indistinguishable in terms of the eviction strategy
    - 2, 5, 4, 6, 3, 1, 2, 5, 6, 4, 1, 3, ... all equivalent!
      - 1, 2, 3, 4, 5, 6
- adding more **different** addresses can increase eviction rate
- **multiple** accesses to one address can increase the eviction rate

## Cache eviction strategy properties (2)

- multiple accesses, low number of different addresses
- sequences often have repeating sub-sequences

## Cache eviction strategy properties (2)

- multiple accesses, low number of different addresses
- sequences often have repeating sub-sequences
- idea: repetitive sequences can be compressed to loops over sub-sequences

# Cache eviction strategy (compression)

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c <= C; c += 1)
    for (d = 0; d <= D; d += 1)
      *a[s+d];
```

# Cache eviction strategy (compression)

```
for (s = 0; s <= S-D; s += L)
    for (c = 0; c <= C; c += 1)
        for (d = 0; d <= D; d += 1)
            *a[s+d];
```

$D$ : different addresses per inner access loop

$C$ : number of repetitions of the inner access loop

$S$ : total number of different addresses (=set size)

$L$ : overlap parameter for the inner access loops (=step size)



# Cache eviction strategy (compression)

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c <= C; c += 1)
    for (d = 0; d <= D; d += 1)
      *a[s+d];
```

$D$ : different addresses per inner access loop

$C$ : number of repetitions of the inner access loop

$S$ : total number of different addresses (=set size)

$L$ : overlap parameter for the inner access loops (=step size)

Write eviction strategies as:  $\mathcal{P}\text{-}C\text{-}D\text{-}L\text{-}S$

# Cache eviction strategy example

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c < C; c += 1)
    for (d = 0; d < D; d += 1)
      *a[s+d];
```

# Cache eviction strategy example

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c < C; c += 1)
    for (d = 0; d < D; d += 1)
      *a[s+d];
```

- $\mathcal{P}$ -1-1-1-4
- $\mathcal{P}$ -2-2-1-4

# Cache eviction strategy example

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c < C; c += 1)
    for (d = 0; d < D; d += 1)
      *a[s+d];
```

- $\mathcal{P}\text{-}1\text{-}1\text{-}1\text{-}4 \rightarrow 1, 2, 3, 4$
- $\mathcal{P}\text{-}2\text{-}2\text{-}1\text{-}4 \rightarrow 1, 2, 1, 2, 2, 3, 2, 3, 3, 4, 3, 4$

# Cache eviction strategy example

```
for (s = 0; s <= S-D; s += L)
  for (c = 0; c < C; c += 1)
    for (d = 0; d < D; d += 1)
      *a[s+d];
```

- $\mathcal{P}\text{-}1\text{-}1\text{-}1\text{-}4 \rightarrow 1, 2, 3, 4$
- $\mathcal{P}\text{-}2\text{-}2\text{-}1\text{-}4 \rightarrow 1, 2, 1, 2, 2, 3, 2, 3, 3, 4, 3, 4$

We evaluated more than 10000 strategies...

# Cache eviction strategies (simple example)

Which one has the better eviction rate?

- $\mathcal{P}$ -1-1-1-17 (as used in previous work)

- $\mathcal{P}$ -1-1-1-20

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (simple example)

Which one has the better eviction rate?

- $\mathcal{P}$ -1-1-1-17 (as used in previous work)
  - Average eviction rate: 74.46%
  - Not high enough for Rowhammer!
- $\mathcal{P}$ -1-1-1-20
  - Average eviction rate: 99.82%
  - High enough for Rowhammer!

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (simple example)

Which one is faster?

- $\mathcal{P}$ -1-1-1-17 (as used in previous work)
- $\mathcal{P}$ -1-1-1-20 (=20 accesses)

(executed in a loop, on Haswell, 16-way cache)



# Cache eviction strategies (simple example)

Which one is faster?

- $\mathcal{P}$ -1-1-1-17 (as used in previous work)
  - Average execution time: 307 ns
  - Fast enough for Rowhammer!
- $\mathcal{P}$ -1-1-1-20 (=20 accesses)
  - Average execution time: 934 ns
  - Not fast enough for Rowhammer!

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one has the better eviction rate?

- $\mathcal{P}$ -1-1-1-17 (=17 accesses)  
→ Average eviction rate: 74.46%
- $\mathcal{P}$ -2-1-1-17 (=34 accesses)

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one has the better eviction rate?

- $\mathcal{P}$ -1-1-1-17 (=17 accesses)

→ Average eviction rate: 74.46%

- $\mathcal{P}$ -2-1-1-17 (=34 accesses)

→ Average eviction rate: 99.86%

→ High enough for Rowhammer!

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one is faster?

- $\mathcal{P}$ -1-1-1-17 (=17 accesses)  
→ Average execution time: 307 ns
- $\mathcal{P}$ -2-1-1-17 (=34 accesses)

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one is faster?

- $\mathcal{P}$ -1-1-1-17 (=17 accesses)
  - Average execution time: 307 ns
- $\mathcal{P}$ -2-1-1-17 (=34 accesses)
  - Average execution time: 191 ns
  - Fast enough for Rowhammer!

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one has the better eviction rate?

- $\mathcal{P}$ -2-1-1-17 (=34 accesses)
  - Average eviction rate: 99.86%
  - Average execution time: 191 ns
- $\mathcal{P}$ -2-2-1-17 (=64 accesses)

(executed in a loop, on Haswell, 16-way cache)

# Cache eviction strategies (another example)

Which one has the better eviction rate?

- $\mathcal{P}$ -2-1-1-17 (=34 accesses)
  - Average eviction rate: 99.86%
  - Average execution time: 191 ns
- $\mathcal{P}$ -2-2-1-17 (=64 accesses)
  - Average eviction rate: 99.98%
  - Average execution time: 180 ns

(executed in a loop, on Haswell, 16-way cache)

# Rowhammer without clflush

## Challenges:

1. How to get accurate timing (in JS)? → easy
2. How to get physical addresses (in JS)? → easy
3. Which physical addresses to access? → solved
4. In which order to access them?



# Rowhammer without clflush

## Challenges:

1. How to get accurate timing (in JS)? → easy
2. How to get physical addresses (in JS)? → easy
3. Which physical addresses to access? → solved
4. In which order to access them? → solved

# Evaluation on Haswell

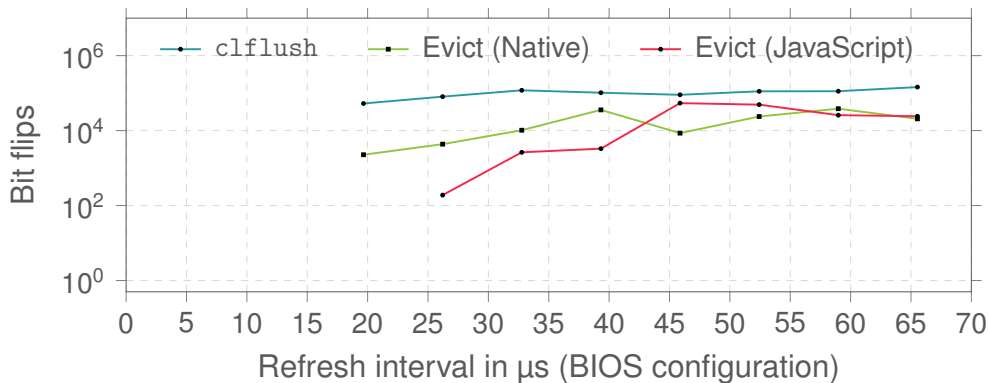


Figure: Number of bit flips within 15 minutes. [Gruss et al., 2016]

# Conclusions

- cache eviction good enough to replace `clflush`
  - independent of programming language and available instructions
  - hardware-fault attack induced in JavaScript
- remote attacks through websites are possible

At least on this laptop (default settings)



ROOT privileges for web apps!

# Cache Side-Channel Attacks and the case of Rowhammer

**Daniel Gruss**  
**IAIK, Graz University of Technology**

April 28, 2016

[Gruss et al., 2015a] Gruss, D., Bidner, D., and Mangard, S. (2015a).

Practical Memory Deduplication Attacks in Sandboxed JavaScript.

In *Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS'15)*.

[Gruss et al., 2016] Gruss, D., Maurice, C., and Mangard, S. (2016).

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript.

In *DIMVA'16*.

[Gruss et al., 2015b] Gruss, D., Spreitzer, R., and Mangard, S. (2015b).

Cache Template Attacks: Automating Attacks on Inclusive Last-Level Caches.

In *USENIX Security Symposium (USENIX Security'15)*.

[Gullasch et al., 2011] Gullasch, D., Bangerter, E., and Krenn, S. (2011).

Cache Games – Bringing Access-Based Cache Attacks on AES to Practice.

In *IEEE Symposium on Security and Privacy (S&P'11)*.

[Hund et al., 2013] Hund, R., Willems, C., and Holz, T. (2013).

Practical Timing Side Channel Attacks against Kernel Space ASLR.

In *IEEE Symposium on Security and Privacy – S&P*, pages 191–205. IEEE.

[Inci et al., 2015] Inci, M. S., Gulmezoglu, B., Irazoqui, G., Eisenbarth, T., and Sunar, B. (2015).

Seriously, get off my cloud! Cross-VM RSA Key Recovery in a Public Cloud.

*Cryptology ePrint Archive, Report 2015/898*.

[Kim et al., 2014] Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J. H., Lee, D., Wilkerson, C., Lai, K., and Mutlu, O. (2014).

Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors.

In *ACM/IEEE International Symposium on Computer Architecture (ISCA'14)*.

[Liu et al., 2015] Liu, F., Yarom, Y., Ge, Q., Heiser, G., and Lee, R. B. (2015).

Last-Level Cache Side-Channel Attacks are Practical.

In *IEEE Symposium on Security and Privacy (S&P'15)*.

[Maurice et al., 2015a] Maurice, C., Le Scouarnec, N., Neumann, C., Heen, O., and Francillon, A. (2015a).

Reverse Engineering Intel Last-Level Cache Complex Addressing Using Performance Counters.

In *International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*.

[Maurice et al., 2015b] Maurice, C., Neumann, C., Heen, O., and Francillon, A. (2015b).

C5: Cross-Cores Cache Covert Channel.

In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'15)*.

[Percival, 2005] Percival, C. (2005).

Cache Missing for Fun and Profit.

URL: <http://daemonology.net/hyperthreading-considered-harmful/>.

[Pessl et al., 2015] Pessl, P., Gruss, D., Maurice, C., Schwarz, M., and Mangard, S. (2015).

Reverse Engineering Intel DRAM Addressing and Exploitation.

*arXiv:1511.08756*.



[Seaborn, 2015] Seaborn, M. (2015).

Exploiting the DRAM rowhammer bug to gain kernel privileges.

<http://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html>.

[Yarom and Falkner, 2014] Yarom, Y. and Falkner, K. (2014).

FLUSH+RELOAD: A High Resolution, Low Noise, L3 Cache Side-Channel Attack.

In *USENIX Security Symposium (USENIX Security'14)*.

[Yarom et al., 2015] Yarom, Y., Ge, Q., Liu, F., Lee, R. B., and Heiser, G. (2015).

Mapping the Intel Last-Level Cache.

*Cryptology ePrint Archive, Report 2015/905*.