Pinworm

Man in the middle for metadata

@bigezy_(bigezy)

@itsasstime (saci)

[need to add Introduction section]

My previous related

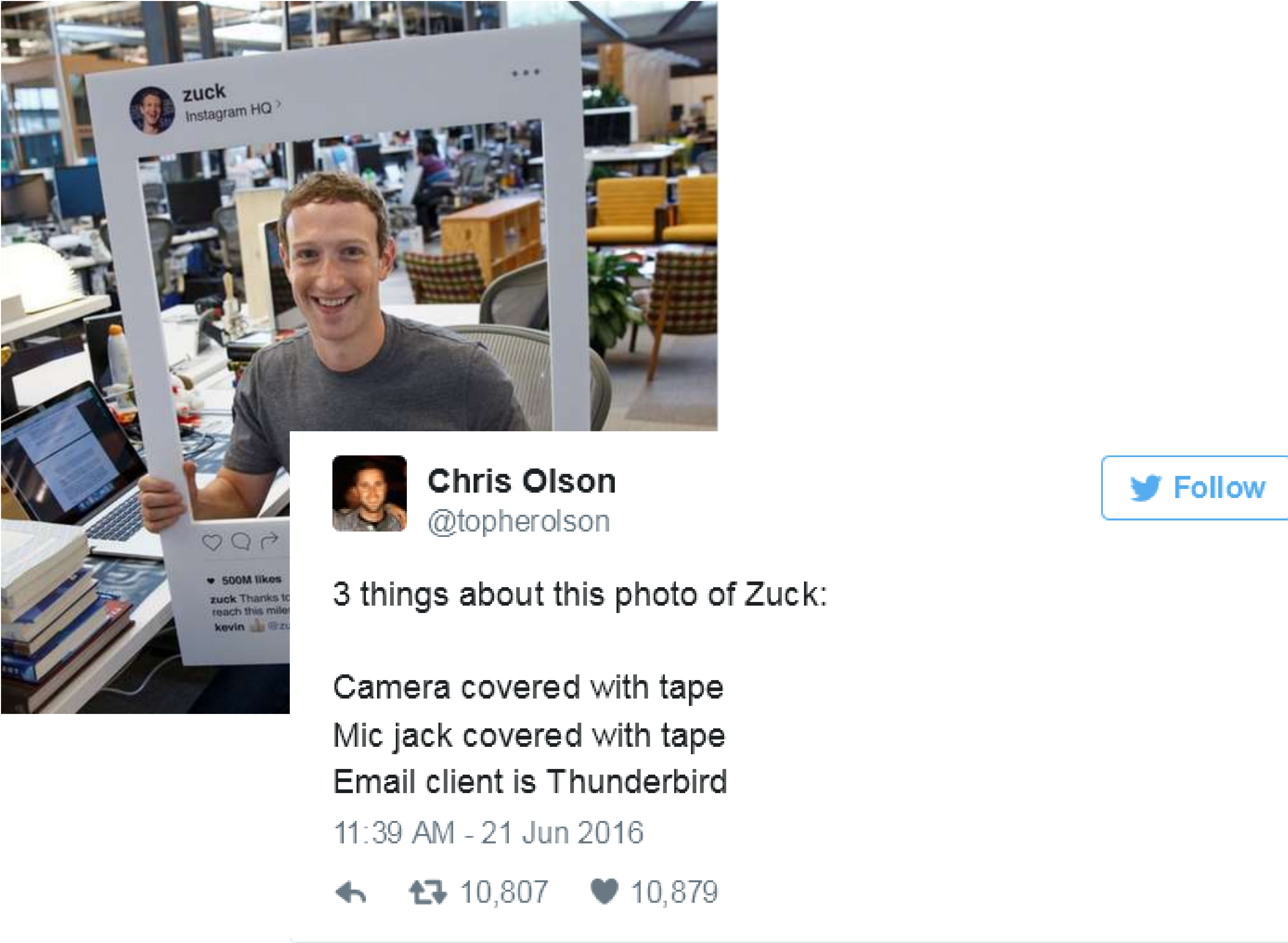
- Badger: Blackhat 2014 https://www.blackhat.com/docs/us-14/materials/us-14-Rogers-Badger-The-Networked-Security-State-Estimation-Toolkit.pdf

- Kobra: BsidesLV 2015 https://youtu.be/gOMVOv6VX50

Previous work from other people

- IRPtracker https://www.osronline.com/article.cfm?article=199

- IRPmon https://github.com/MartinDrab/IRPMon



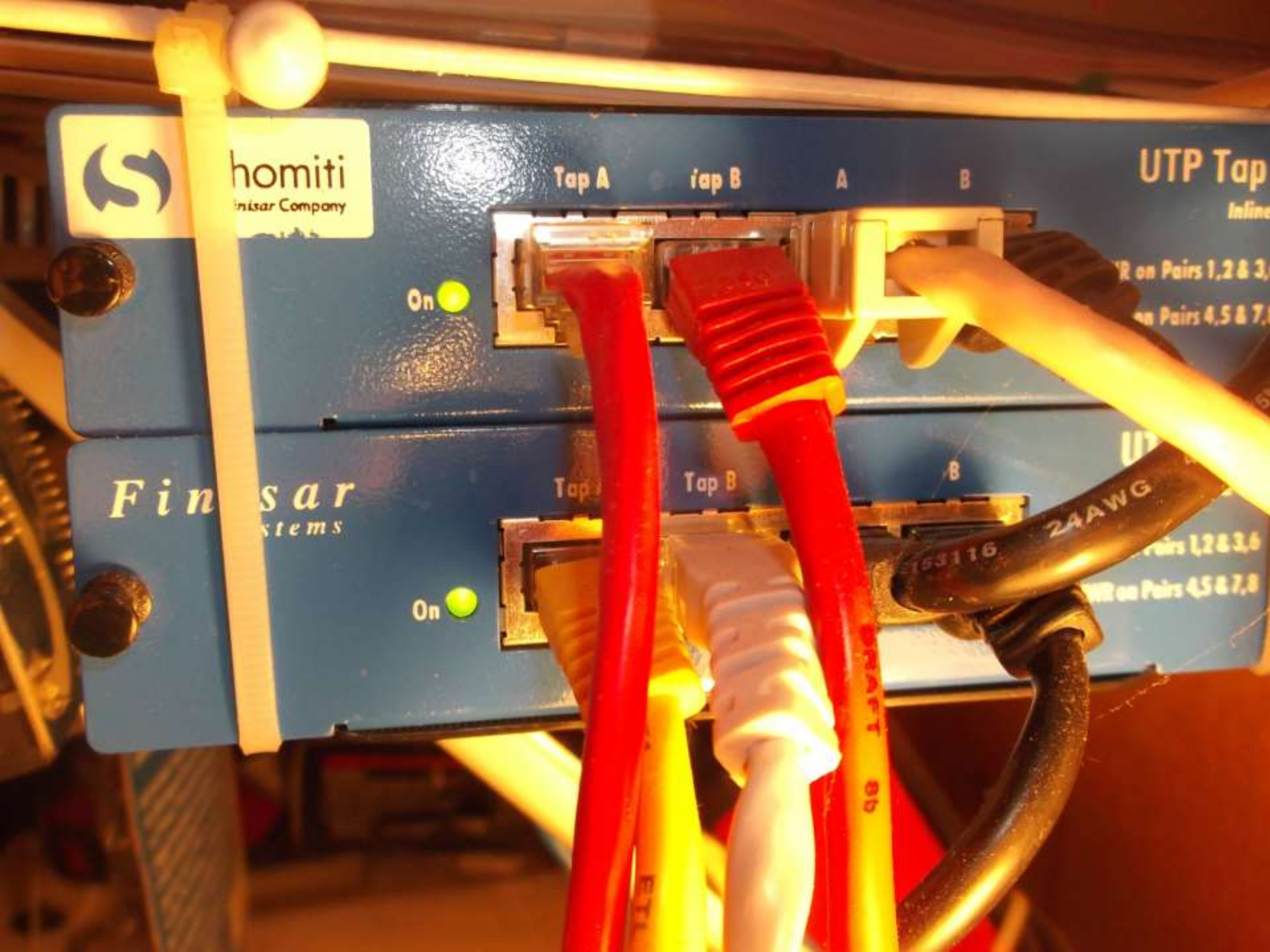
Chris Olson
@topherolson

3 things about this photo of Zuck:

Camera covered with tape
Mic jack covered with tape
Email client is Thunderbird

11:39 AM - 21 Jun 2016

10,807 10,879


homiti
inisar Company
Tap A
ıap B
A
B
UTP Tap
Inline
R on Pairs 1,2 & 3,
on Pairs 4,5 & 7,
On
Fin sar
stems
Tap
Tap B
B
UT
Pairs 1,2 & 3,6
R on Pairs 4,5 & 7,8
On
24AWG
153116
DRAFT
8b
ETL

ACT
10
100
LNK
TX
5

Data observations

- Inside packets 998,802,444
- Outside packets 1,371,775,557
- Observed Destinations Firewalled Side 29,829
  - 29,525 resolved via reverse lookup

[2016 Observations]

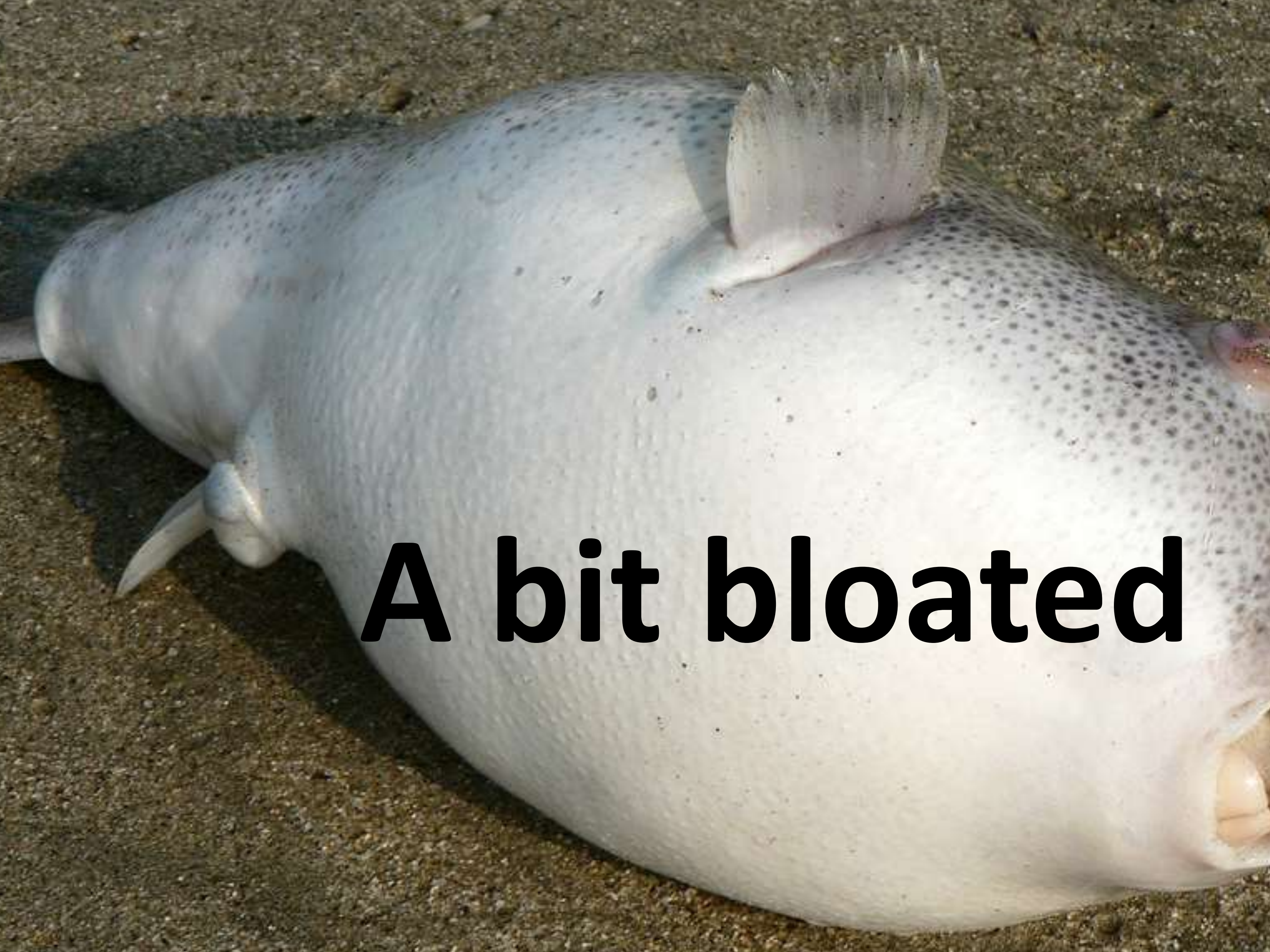
- Destinations per https connection up 400 percent for certain media sites

- SSL data is streaming out even when the pc is idle.

- More details to be added here later

Is it 1984?

- Mouse Movements
  - List of sites that track mouse movements from data (will be released at talk)
- Key Strokes?
- Microphone?
- Video?



A bit bloated

IRP

- Previous projects such as IRPtracker (link in attachment) proved interesting, but, sadly this program is limited to 32 bits.

- But how to track 64 bit IRB process calls

- Luckily there was a great start into this forensic research using an IRP sniffer based on work done by Martin Drab IRPmon

IRPmon downfalls

- IRPmon was a good start and showed the gaps needed to provide for man in the middle

- More precision and information was needed regarding the data flows within the operating system

- Device calls needed an in memory datastore so it would be possible to virtualize flows and pinpoint any duplication of input devices within the OS

IRP Sniffing

- Instrumentation of process access to devices

- Devices provide I/O services that can be used by processes to harvest information
  - Keyboard
  - Mouse
  - Microphone
  - Video

Mouse movement

- What processes are interested in the mouse movements in your browser?
- What network traffic is then generated as a result of these calls?
- Use multiple data sources to get access to those things:
  - 1 The object table can include device objects
  - 2 IRP requests

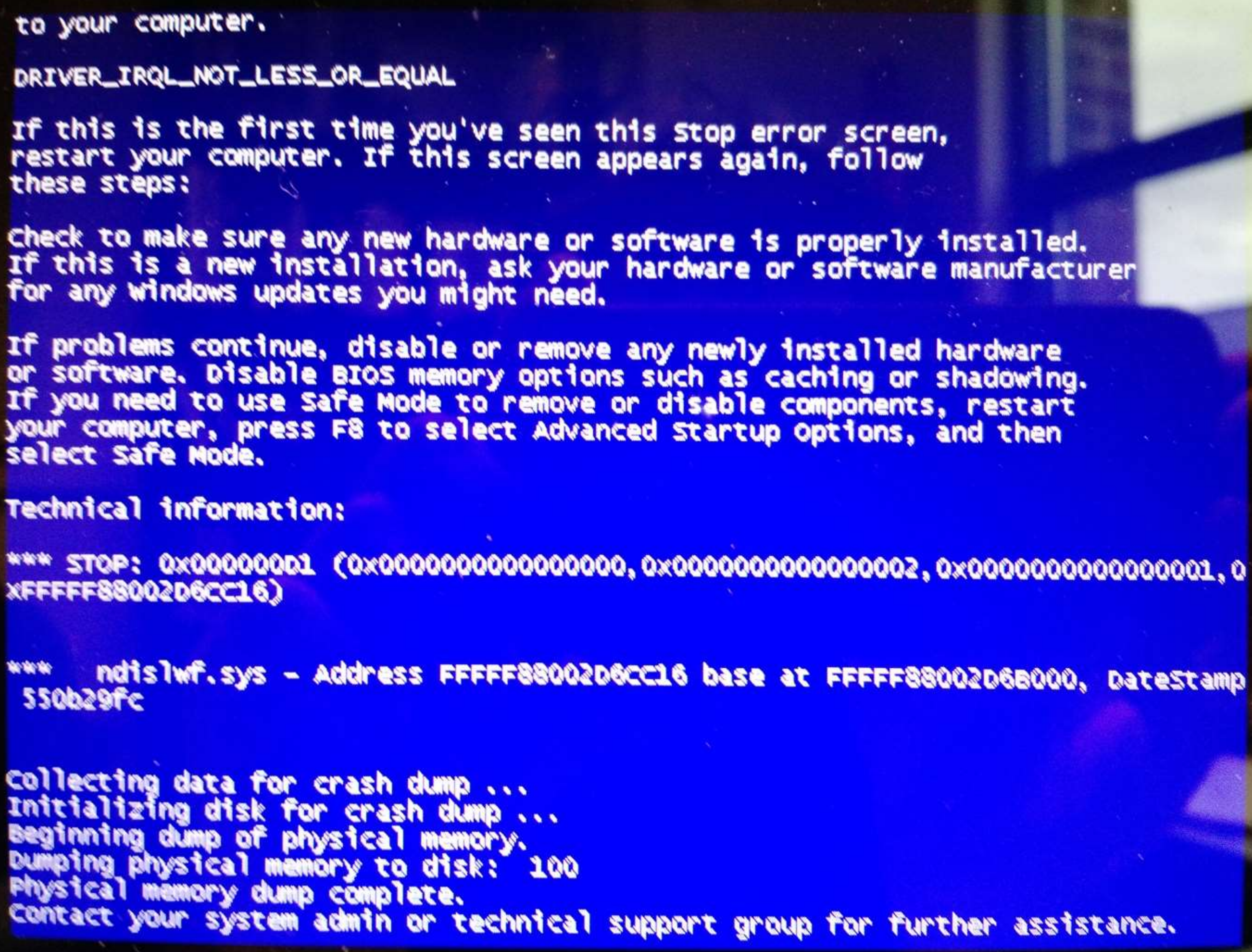
Why Windows 7/8?

- Windows 10 will work with chrome or firefox add on (future work)
- We are building this framework from scratch and are providing it to the community for to ensure integrity of data communications with privacy in mind
- Fuck windows 10 (cause I wanted 3 reasons)

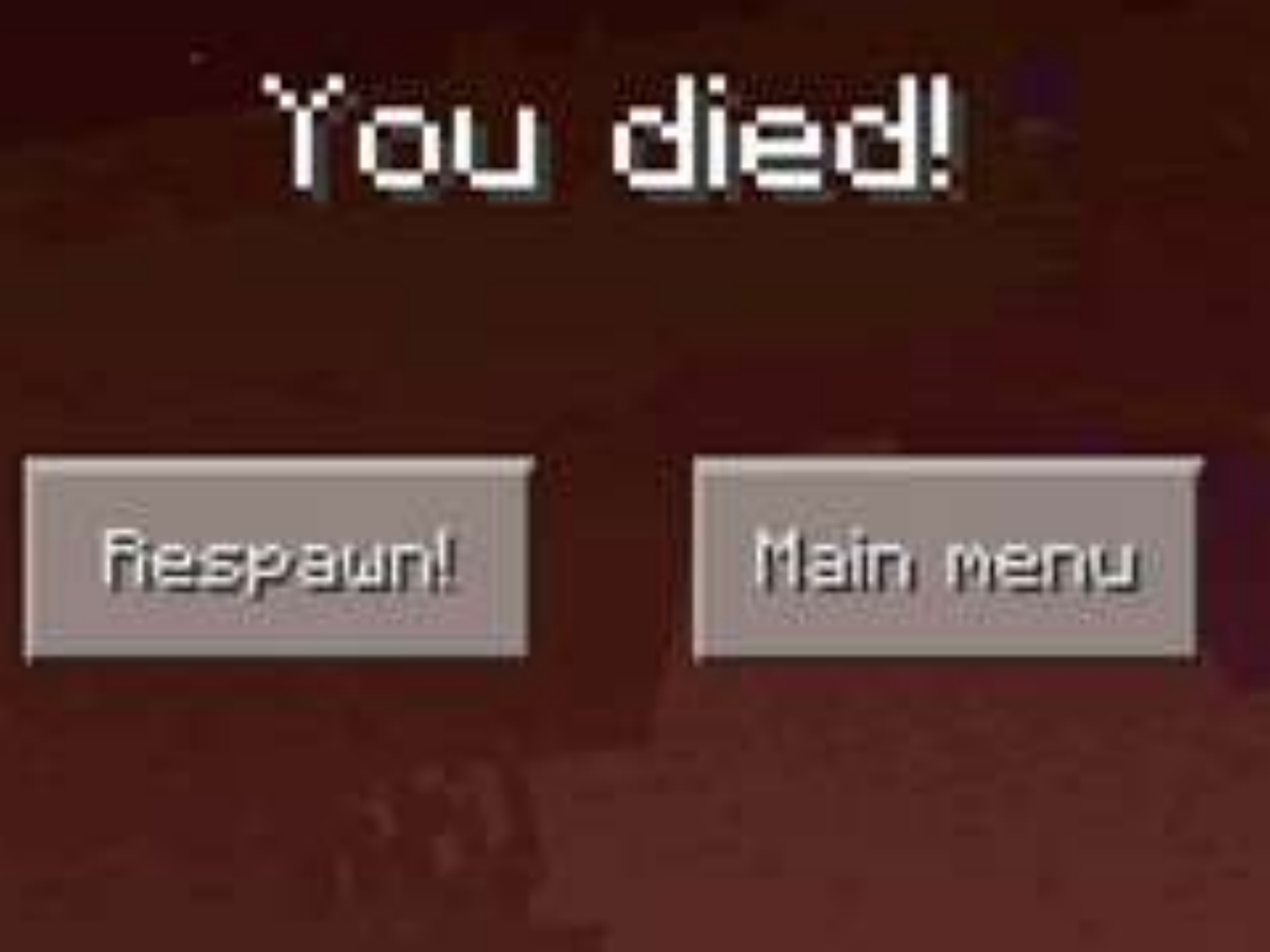
Easy mode

- Meeting your adversary at his own level of abstraction makes finding breaches of privacy easy

- Getting to this level of abstraction however requires repeated failure at accessing the kernel level drivers



```
to your computer.

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000D1 (0x0000000000000000,0x0000000000000002,0x0000000000000001,0
xFFFFF88002D6CC16)


***   ndislwf.sys - Address FFFFF88002D6CC16 base at FFFFF88002D6B000, DateStamp
 550b29fc


Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk:  100
Physical memory dump complete.
Contact your system admin or technical support group for further assistance.
```



You died!

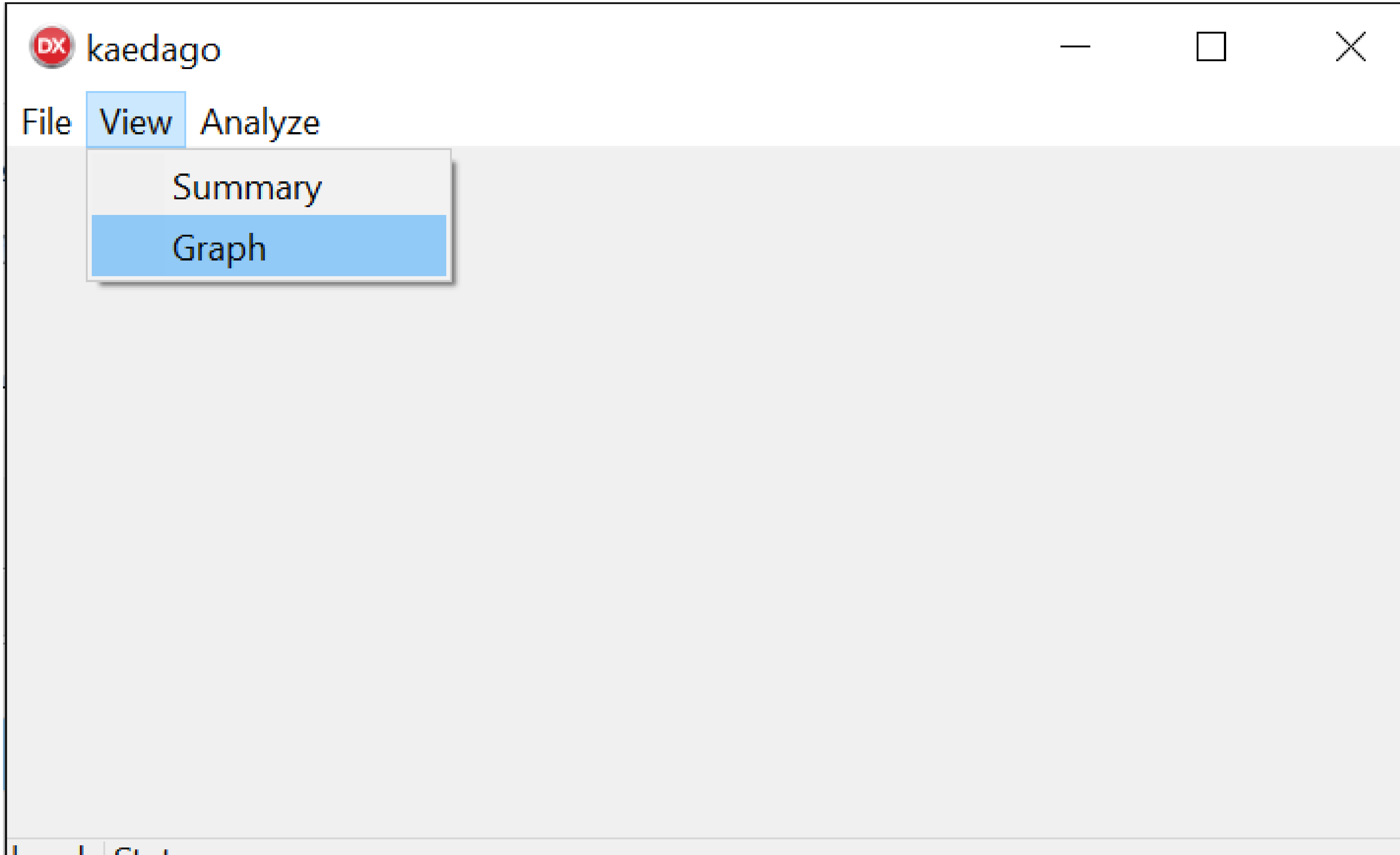
Pinworm Core (Old school code slides)

Pulling the processes

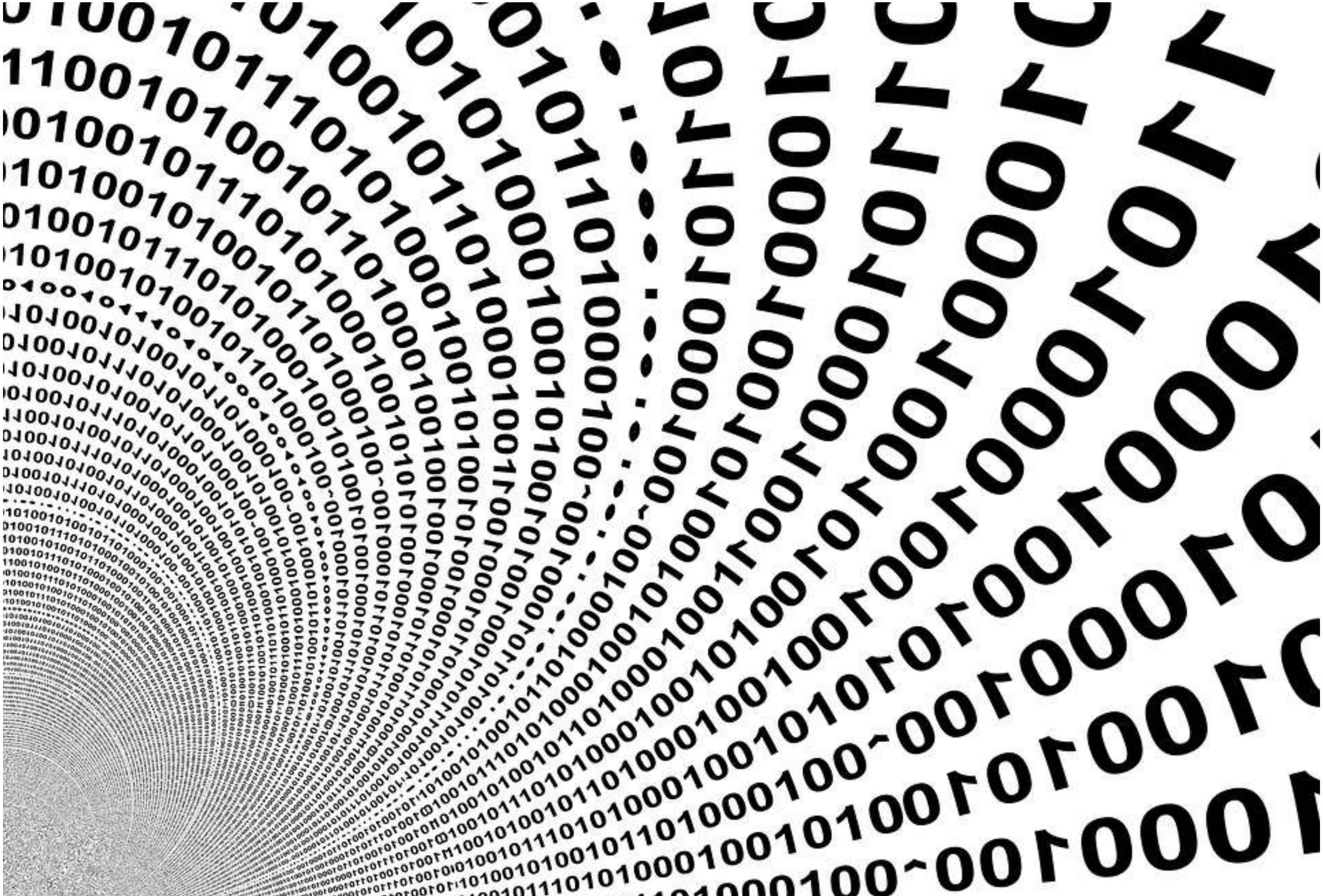
```
void InitializeProcessState()
{
unsigned __int64 eproc = 0x00000000;
int   current_PID = 0;
int   start_PID = 0;
int   i_count = 0;

PLIST_ENTRY plist_active_procs;
InitializeListHead(&ProcessStateList);
// Get the address of the current EPROCESS
eproc = (unsigned __int64)PsGetCurrentProcess();
start_PID = *((int *)(eproc + _kpPIDOFFSET));
current_PID = start_PID;
```

```
while (1)
{


//check end of list
if ((i_count >= 1) && (start_PID == current_PID))
break;


DbgPrint("Test: PID is %d\n", current_PID);
//Insert Item
//allocate
addNewProcess(eproc);


plist_active_procs = (LIST_ENTRY *)(eproc + _kpFLINKOFFSET);
eproc = (unsigned __int64)plist_active_procs->Flink;
eproc = eproc - _kpFLINKOFFSET;
current_PID = *((int *)(eproc + _kpPIDOFFSET));
i_count++;
}
```

Don’t Panic there is also a UI!


kaedago
File
View
Analyze
Summary
Graph

Moar Code first



Client Kung fu:

```
int  main(int argc, char** argv) {
HANDLE hDevice;
BOOL status;
hDevice = CreateFile(L"\\\\.\\KBR",
        GENERIC_READ | GENERIC_WRITE, 0,NULL,
        OPEN_EXISTING, 0, NULL);
```

```
// Variable for the run-time callout identifier
UINT32 CalloutId;
HANDLE engineHandle;
UINT32 gStreamCalloutIdV4, gFlowEstablishedCalloutIdV4;
UINT32 kIPV4Outbound;
PVOID * NotificationHandle;
LIST_ENTRY ProcessStateList;
ProcessProfile * state;
PDEVICE_OBJECT pDeviceObject = NULL;
```

```
// Pool tags used for memory allocations
#define WSKTCP_SOCKET_POOL_TAG ((ULONG)'sksw')
#define WSKTCP_BUFFER_POOL_TAG ((ULONG)'bksw')
#define WSKTCP_GENERIC_POOL_TAG ((ULONG)'xksw')
```

```
// callout functions
VOID NTAPI
ClassifyFn(
IN const FWPS_INCOMING_VALUES0  *inFixedValues,
IN const FWPS_INCOMING_METADATA_VALUES0  *inMetaValues,
IN OUT VOID  *layerData,IN const FWPS_FILTER0  *filter,
IN UINT64  flowContext,
IN OUT FWPS_CLASSIFY_OUT0  *classifyOut)
```

```
// Process the IRP
//Parse and print to debugger with a tag for filtering
NTSTATUS status, pstatus = STATUS_SUCCESS;
PVOID userBuffer, cmdBuffer;
ULONG xferSize;
ULONG       i, thNum;
// The stack location contains the user buffer info
PIO_STACK_LOCATION pIrpStack = IoGetCurrentIrpStackLocation(pIrp);

// Dig out the Device Extension from the Device object
// Determine the length of the request
xferSize = pIrpStack->Parameters.Write.Length;
// Obtain user buffer pointer
userBuffer = pIrp->AssociatedIrp.SystemBuffer;
//allocate buffer and copy for no reason
cmdBuffer = ExAllocatePool(NonPagedPool, xferSize);
if (cmdBuffer == NULL) {
            // buffer didn't allocate???
            status = STATUS_INSUFFICIENT_RESOURCES;
            xferSize = 0; }
else {
            RtlCopyMemory(cmdBuffer, userBuffer, xferSize);   }
```

To add a filter that references a callout (documented in the Windows Driver Kit(WDK))

- Invoke the functions in the following order
  - Call to register the callout with the filter engine.
  - Call FwpmCalloutAdd0 to add the callout to the system
  - Call FwpmFilterAdd0 to add the filter that references the callout to the system.

Case Study 1

- What does a process within a browser do in regards to mouse movement?

- Where does the forked mouse movement data go when its sent to the internet?

- Does this data exfiltration forking occur inside the browser, in kernel land, or in user space?

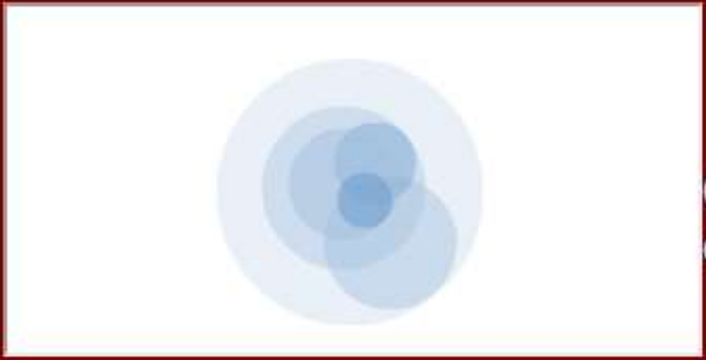
Demo

- www.kaedago.com/saci

- Notice the frames to the left

- We will use these frames to demonstrate the injection of data using pinworm

X: 150, Y: 259



Saci is a one legged demon who wears a magic red cap and
demon is known as a prankster and is often blamed for misf

Saci has a variety of magical powers:

- Has the ability to disappear, often people say only the g
  - He is often depicted disappearing in a dust devil. T spin dance of Saci.
- Can transform himself into Matita Pereira, an evasive b thin air
- Grants wishes to those who capture him and steal his h
  - His hat is known for smelling sulfurous and those w to be haunted by the foul smell.
- Extraordniarily nimble depsite having only one leg. It is cross-legged (quite the extrodinary feat for someone wi

Capturing Saci is a difficult task, but there are many ways th
even capture, Saci:

- The easiest way to evade Saci is to cross a stream. Sa
- Popular ways of slowing Saci down include dropping a

Case Study 2

- We have two sniffer instrumented computers. One computer has been using Windows 7 for the last few years.
- We will use a control computer with a recently installed and patched copy of windows 7.
- What are the differences in network traffic between the two computers?
- What are the results from using different patched browsers on both PC's?

Man in the middle demo

- Show keyboard injection using pinworm

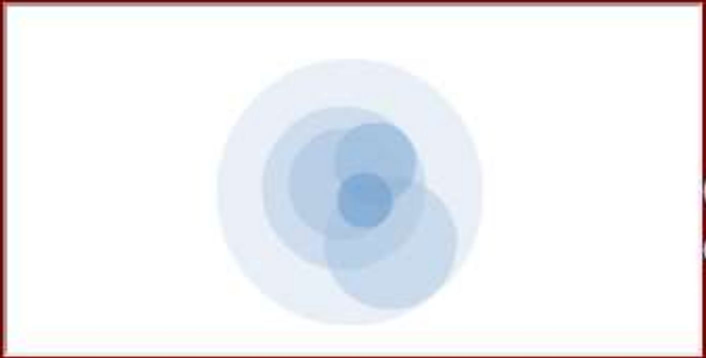
- Show mouse movement

- Show microphone injection

- Show video injection

X: 150, Y: 259

FUCKTARDSPACEH
ALLALT



Saci is a one legged demon who wears a magic red cap and
demon is known as a prankster and is often blamed for misf

Saci has a variety of magical powers:

- Has the ability to disappear, often people say only the g
  - He is often depicted disappearing in a dust devil. T
    spin dance of Saci.
- Can transform himself into Matita Pereira, an evasive b
  thin air
- Grants wishes to those who capture him and steal his h
  - His hat is known for smelling sulfurous and those w
    to be haunted by the foul smell.
- Extraordniarily nimble depsite having only one leg. It is
  cross-legged (quite the extrodinary feat for someone wi

Capturing Saci is a difficult task, but there are many ways th
even capture, Saci:

- The easiest way to evade Saci is to cross a stream. Sa
- Popular ways of slowing Saci down include dropping a

Social Media Demo

- Process instrumentation of social media sites
  - Redacted.com
  - Example.com
  - Whattillthetalk.com

Toolchain framework

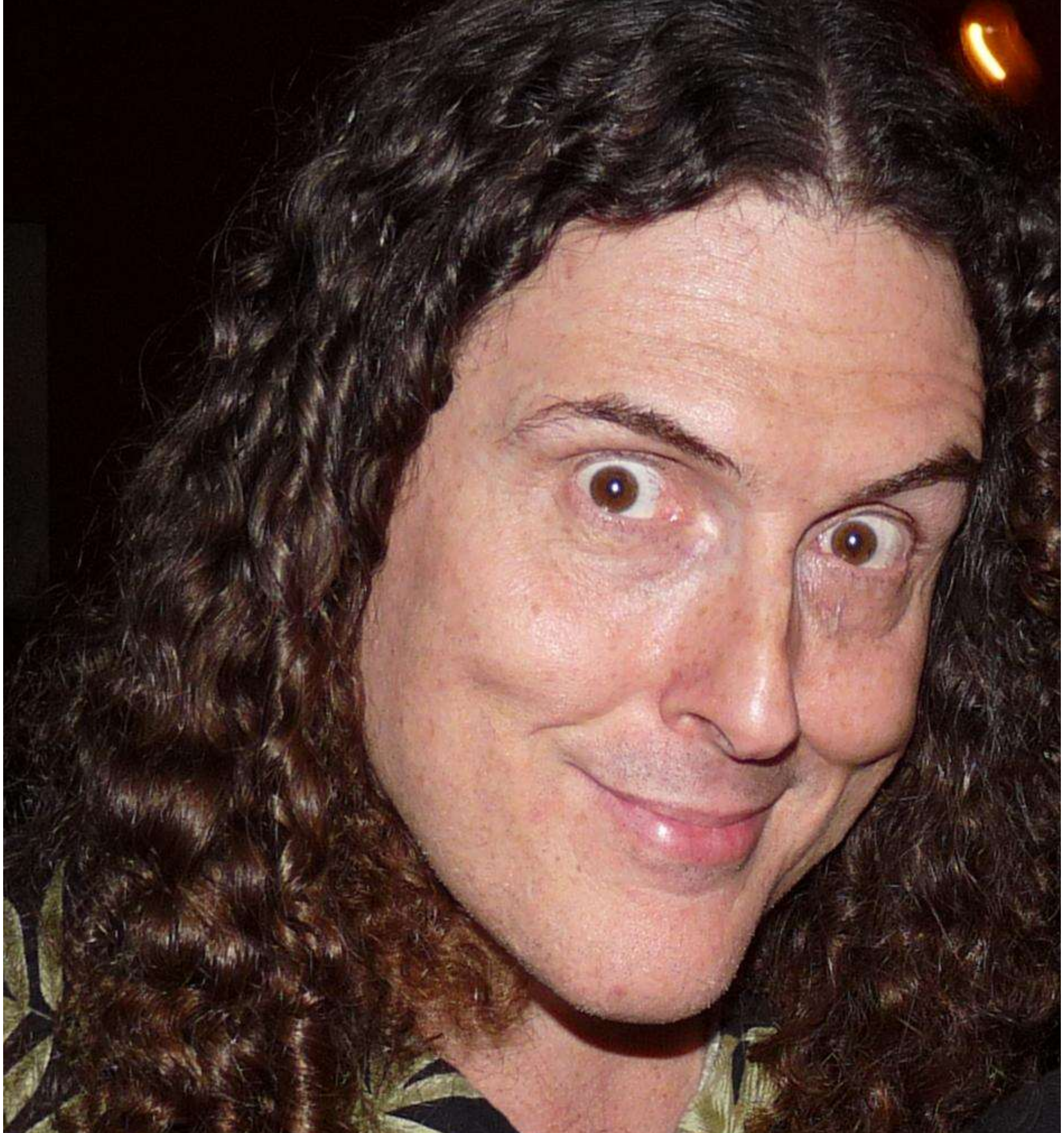
- UI client and cone of silence are still in alpha
- Framework will be released when its ready
- Pinworm and the test site available now

Releasing of Pinworm github.com/bigezy/pinworm including:

1. Sniffer to instrument device driver calls.

2. Http Server code to display metadata collected on users

3. Man in the middle client for interception of device owners private information and white noise generator.



Thanks