

# Document parsers "research" as passive income

Jaanus Kääp  
Clarified Security

# Who is this guy

- Jaanus Kääp
- Working at Clarified Security
  - Vulnerability testing, research, trainings, cyber excercises
- Lazy

# Why this topic

- Got #11 in MSRC top-100
  - Suprised but happy
- Then found out who is #12
  - JAMES FORSHAW
- **WTF?**



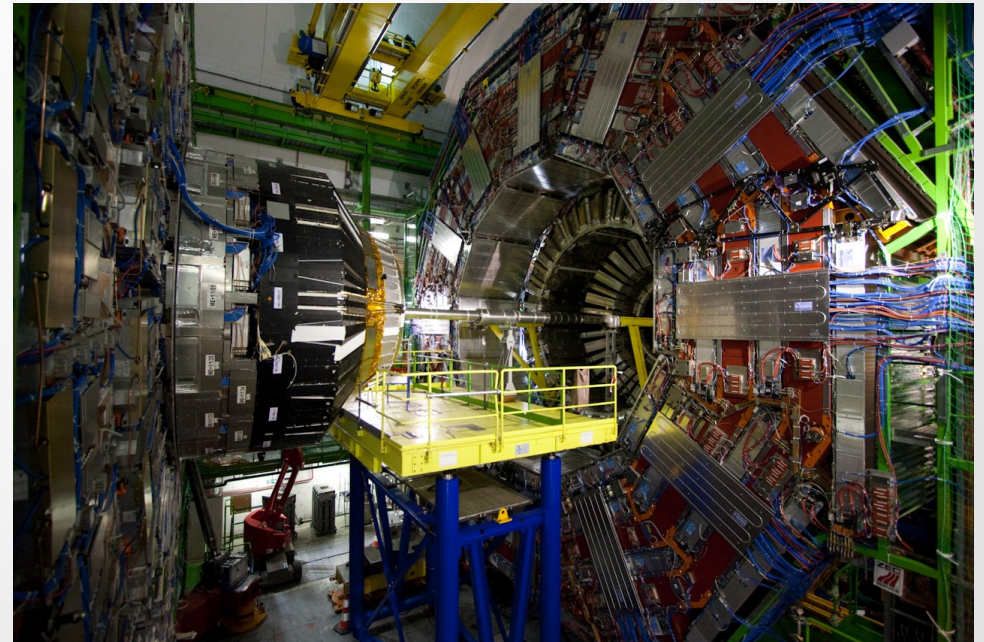
# Why WTF?

- Reported things found by one fuzzing script\*
- Fuzzing with same logic for 3 YEARS!
  - Dumb bit flipping fuzzing, nothing cool!
- My „passive income” through ZDI
- Still placed #11

# „Research” vs Research



VS



# Why WTF?

- Reported things found by one fuzzing script\*
- Fuzzing with same logic for 3 YEARS!
  - Dumb bit flipping fuzzing, nothing cool!
- My „passive income” through ZDI
- Placed #11
- **Pissed me off so here we are**

# Topic itself

- Fuzzing
- My corpus distillation method
- Tools I developed
- Using with other stuff

# Fuzzing

- Simple bit flipping
- 2-6 PC (Zotacs mostly)
  - Electricity cost nothing in Estonia
- Nothing special
- Except fuzzing set



# Fuzzing set

- As much functionality as possible
- Lazy == no protocol implementation
- Very common filetype
- Multiple parsers

# Corpus distillation

- What you need
  - Huge number of initial files
  - Application that can read them
  - Time and/or computing power
- What you do
  - Code coverage with every input
  - Analyse the coverage of all the files
  - Minimize the set

# Code coverage

- Open source – simple (special flags)
- Closed source
  - Trace the code (dead slow)
  - Some tools/libs: Pin, DynamoRIO
  - Intel® PT\*
  - Write coverage tool yourself

# Code coverage

- Basic blocks breakpoints
- First idea:
  - Breakpoint to every basic block
- First implementation
  - Set breakpoints
  - Write down each bp-event
  - Continue execution

# How to get basic blocks

- IDA pro + IDAPython
- Each basic block
  - RVA from base address

# First process

- Prep
  - IDA analysis
  - Basic blocks file generation
- Execution
  - Insert breakpoint
  - Catch 0xCC exceptions
  - If in the basic block list
    - Record location
    - Replace 0xCC with original value
    - $EIP = EIP - 1$

# First run

- Foxit software
  - 611 927 breakpoint
  - Conf: 8 sec wait
  - 180 seconds for setup
  - 30 seconds for execution
  - TOTAL: ~210s/execution == 411 runs per day
- **TOO SLOW**

# How to speed up?

- Most time was spent on setting breakpoints
- What is breakpoint
  - 0xCC
- Why not set them in executable?



# How to get basicblocks

- IDA pro + IDAPython
- Each basic block
  - RVA from base address
  - RVA/Offset in the file
  - Original value

# New process

- Prep
  - IDA analysis
  - Basic blocks file generation
  - Modification of the exe/dll files
- Execution
  - Catch 0xCC exceptions
  - If in the basic block list
    - Record location
    - Replace 0xCC with original value
    - $EIP = EIP - 1$

# Second run

- Foxit software
  - 611 927 breakpoint
  - Conf: 8 sec wait
  - 30 seconds for execution
  - TOTAL: ~30s/execution
- **MUCH BETTER**

# Additional optimization

- Reducing basic blocks count
  - Analyse some(100-1000) files
  - Take some(1-25) files with most coverage
  - Add them to final set
  - Remove basicblocks covered by them

# Third run

- Foxit software (simple example)
  - <600 000 breakpoint
  - Conf: 8 sec wait
  - 10 seconds for execution
  - TOTAL: ~10s/execution
- **25% overhead only for close source software**

DEMO

# How large initial set you need?

What does corpus distillation look like at Google scale? Turns out we have a large index of the web, so we cranked through 20 terabytes of SWF file downloads followed by 1 week of run time on 2,000 CPU cores to calculate the minimal set of about 20,000 files. Finally, those same 2,000 cores plus 3 more weeks of runtime

# Google server room





# My server room

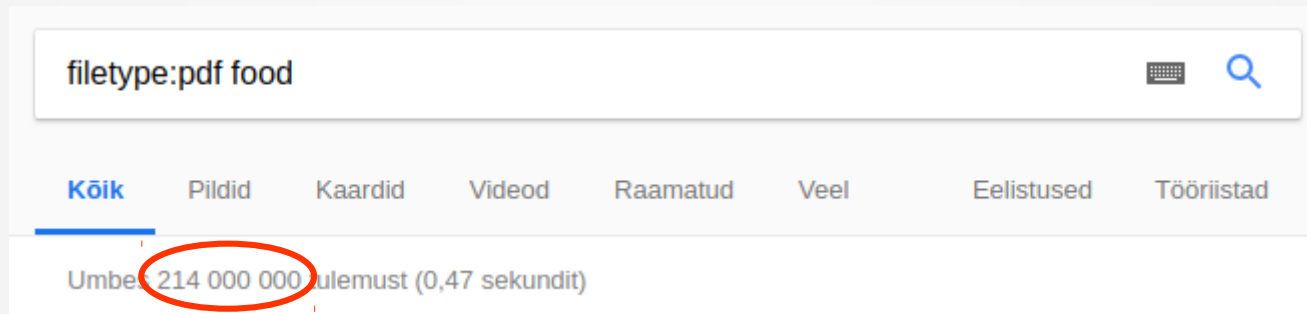


# Final sets

Software	Initial set	Final set
PDF	~1 500 000	2216
DOC	~1 500 000	1309
XLS	~1 500 000	1951
PPT	~1 500 000	1379
SWF	~1 500 000	1495*

# How to get these files?

- Google „filetype:pdf“



# How to get these files?

< Goooooooooooooogle

Eelmine

2 3 4 5 6 7 8 9 10 11

# Additional problems

- Not real pdf files
- DDOS protection

# Solution

- Searches
  - filetype:pdf aa
  - filetype:pdf ab
  - filetype:pdf ac
- Not real pdf files
  - Magic value - %PDF
- DDOS protection
  - It's all about timing
    - **48** seconds wait

# Additional tricks

- Collecting files from multiple IPs
- Anyone here from Google?
  - Please close your eyes and ears for 1-2 minute
  - Possible violation of Terms of Service

## Additional tricks

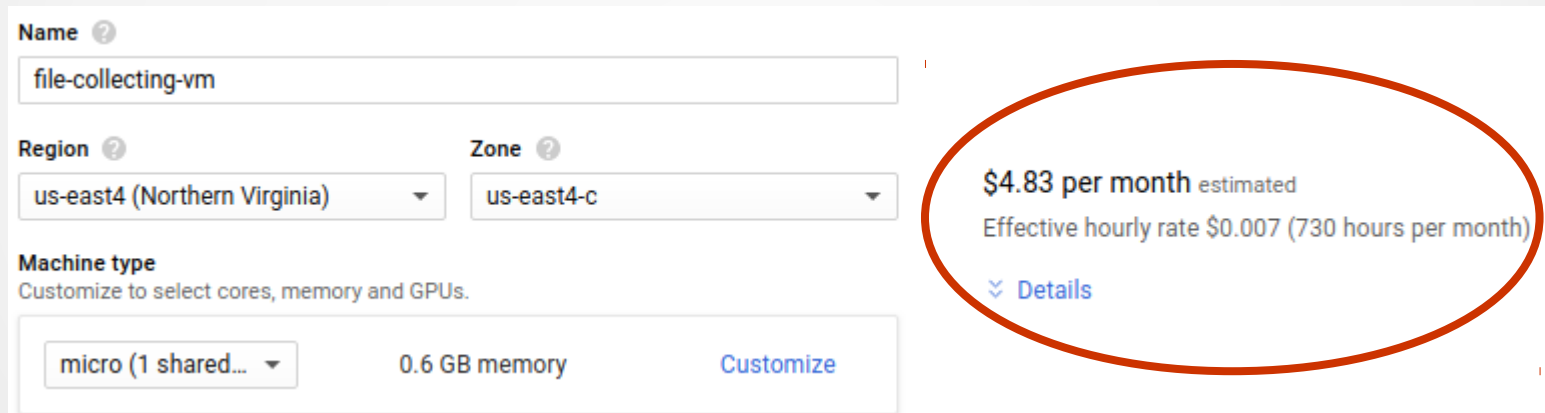


Google Cloud



# Additional tricks

- Every VM gets its own public IP
- Even the smallest and cheapest one



The screenshot shows the configuration interface for a Google Cloud VM. The 'Name' field is 'file-collecting-vm'. The 'Region' is 'us-east4 (Northern Virginia)' and the 'Zone' is 'us-east4-c'. The 'Machine type' is 'micro (1 shared...)' with '0.6 GB memory'. A 'Customize' button is visible. To the right, a pricing summary is circled in red, showing '\$4.83 per month estimated' and 'Effective hourly rate \$0.007 (730 hours per month)'. A 'Details' link is also present.

Field	Value
Name	file-collecting-vm
Region	us-east4 (Northern Virginia)
Zone	us-east4-c
Machine type	micro (1 shared...)
Memory	0.6 GB
Price (estimated)	\$4.83 per month
Effective hourly rate	\$0.007 (730 hours per month)

# Additional tricks

- IT gets just a bit better

Name <sup>?</sup>  
file-collecting-vm

Region <sup>?</sup> us-west1 (Oregon) Zone <sup>?</sup> us-west1-b

Machine type  
Customize to select cores, memory and GPUs.

micro (1 shared... 0.6 GB memory [Customize](#)

**\$4.28 per month** estimated  
Effective hourly rate \$0.006 (720 hours per month)

Your first 744 hours of f1-micro instance usage are free this month. [Learn more](#)

[Details](#)

# Results & CVE-s

Vendor	CVE count
Microsoft	27
Adobe	45
Apple*	2

- Bit over 2 per month
- Actually more findings - lot from Foxit
  - Vendor not giving CVE-s == no CVE (pure laziness)

# How bad against others

- Did use doc fileset to fuzz smaller office software
  - Libreoffice (64bit)
    - 32 bit + full page heap == crash.....
  - WPS office
  - Polaris office
- 5days \* 24hours \* 16 VMs
- Microsoft & Adobe seem lot better after that

# Unique crashes (reverified)

Software	ITERATIONS	NOT NULL	NULL
Libreoffice	~64K(~800/day)	6	2
WPS	~256K(~3200/day)	24	7
Polaris	~120K(~1500/day)	60	47

# Unique crashes (reverified)

Software	DEP	OOBR	OVERFLOW	UAF	UNINITED	???	NULL
Libreoffice		2		2		2	2
WPS	1	14	2	1		6	7
Polaris		32	1	5	6	16	47

# My toolset „Rehepapp”

- Analyzer tool & IDA scripts for BB list
- Tracer software
- Server for gathering data & analysis
- Scripts for file collection & coverage
- Supporting software for data modification

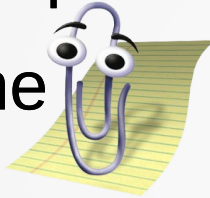
# Along with others

- Input set for AFL
  - Already good coverage
  - WinAFL for closed stuff
- Help for RE by coverage info into IDA
  - Scripts will be included in future
- Possible future work
  - Replace most with Intel PT via WindowsIntelPT

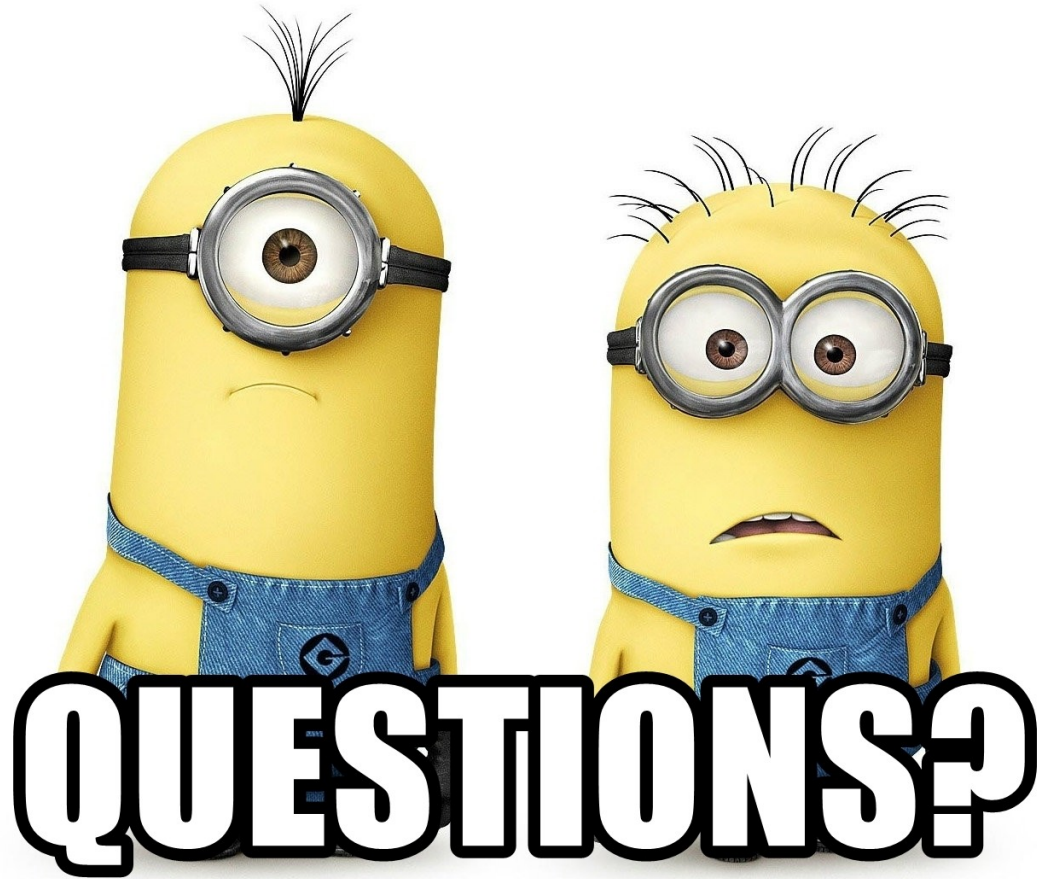


# Tools & sets

- Tool address
  - <https://github.com/FoxHex0ne/Rehepapp>
- For POC participants only: doc fileset
  - Ask from me



# Q & A





# Thank you

@FoxHex0ne

Jaanus.kaap@gmail.com