Path Of LeAst Resistance

Accelerating the search for vulnerable functions

@aCaltum

https://github.com/ezrac/POLAR

This presentation contains the general insights and opinions of its author, Ezra Caltum. I am speaking on behalf of myself only, and the views and opinions contained in this presentation should not be attributed to my employer.
The information in this presentation is provided for informational and educational purposes only and is not to be relied upon for any other purpose. Use at your own risk!
I makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. I accept no duty to update this presentation based on more current information. I disclaim all liability for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of or reliance on the content of this presentation.
No computer system can be absolutely secure.
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
*Other names and brands may be claimed as the property of others.

```
# cat /etc/passwd | grep ecaltum
ecaltum:X::1:34:Ezra (badg0at) Caltum:/Israel/TelAviv:/bin/zsh

# cat /etc/groups
MexicansInIsrael:x:52:ecaltum…
dc9723:x:9723:iamit, ikotler, ecaltum
bsidestlv:x:201906:....
IntelAttackResearch:x:50:...
IAF:x:111:...
```

This is a personal hobby research

And it’s built on the shoulders of giants.

https://github.com/ezrac/POLAR

I love telling stories

Embedded System

Linux Based

100s of Executables

10s of libraries

Everything dynamically compiled

And like every IoT

CC0 Creative Commons Free for commercial use No attribution required
https://pixabay.com/en/pasta-spaghetti-food-italian-1463928/

How do you exploit it?

Where do you start?

Luck plays a big part


© QuickMeme

The bug was in a library

Who calls the library?

How?

If a tree falls in the forest, and nobody hears it….

So we have a problem

Let’s describe the problem

1. Identify all the executables dynamically compiled with the library.

Let’s describe the problem

1. Identify all the executables dynamically compiled with the library.
2. Identify all the function usages among all the executables.

Let’s describe the problem

1. Identify all the executables dynamically compiled with the library.
2. Identify all the function usages among all the executables.
3. Identify a place where no sanitization was performed

Let’s describe the problem

1. Identify all the executables dynamically compiled with the library.
2. Identify all the function usages among all the executables.
3. Identify a place where no sanitization was performed
4. ...

Let’s describe the problem

1. Identify all the executables dynamically compiled with the library.
2. Identify all the function usages among all the executables.
3. Identify a place where no sanitization was performed
4. ...
5. Profit

Identify executables linked to the library.

That’s simple:

- nm -DA

Identify executables linked to the library.

That’s simple:

- nm -DA
- Simple to do

Identify executables linked to the library.

That’s simple:

- nm -DA
- Simple to do
- Simple to search

Identify executables linked to the library.

That’s simple:

- nm -DA
- Simple to do
- Simple to search
- grep-able

```
bin/umount:         U shmdt
bin/umount:         U shmget
bin/umount:         U sigaction
bin/umount:         U sigaddset
bin/umount:         U sigemptyset
bin/umount:         U sigfillset
bin/umount:         U signal
bin/umount:         U sigprocmask
bin/umount:         U sigsuspend
bin/umount:         U sin
bin/umount:         U sleep
bin/umount:         U snprintf
bin/umount:         U socket
bin/umount:         U sprintf
bin/umount:         U sqrt
bin/umount:         U srand
bin/umount:         U sscanf
bin/umount:0000c490 T _start
bin/umount:         U stat64
bin/umount:         U statfs64
bin/umount:00068e54 B stderr
bin/umount:00068e64 B stdin
bin/umount:00068e50 B stdout
bin/umount:         U stime
```

Thank you for coming to my talk.

Not really

It’s not scalable.

Not really

It’s not scalable.

I can’t query it.

Not really

It’s not scalable.

I can’t query it.

It’s not “nice”

Problem

- Model the relationships on a well known format.

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)
- …

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)
- …
- And the most important thing. Managers love shiny colors and visualizations.

Problem

- **Model the relationships on a well known format.**
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)
- …
- And the most important thing. Managers love shiny colors and visualizations.

Let’s model the problem

We have a file.

The file can:

Let’s model the problem

We have a file.

The file can:

- Import a symbol from another file (library)

Let’s model the problem

We have a file.

The file can:

- Import a symbol from another file (library)
- Export a symbol for another file

Let’s model the problem

We have a file.

The file can:

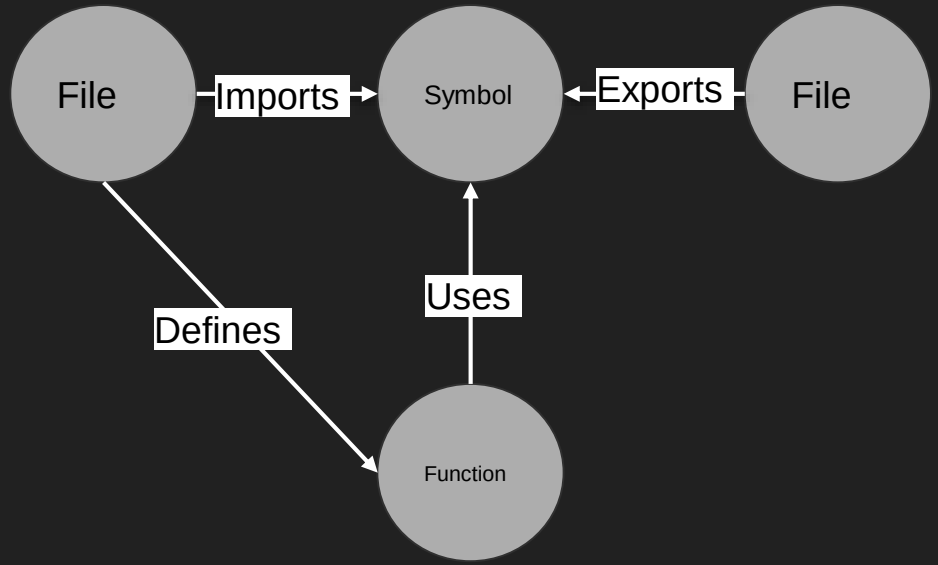
- Import a symbol from another file (library)
- Export a symbol for another file
- Use the symbol in it’s functions

Let’s model the problem

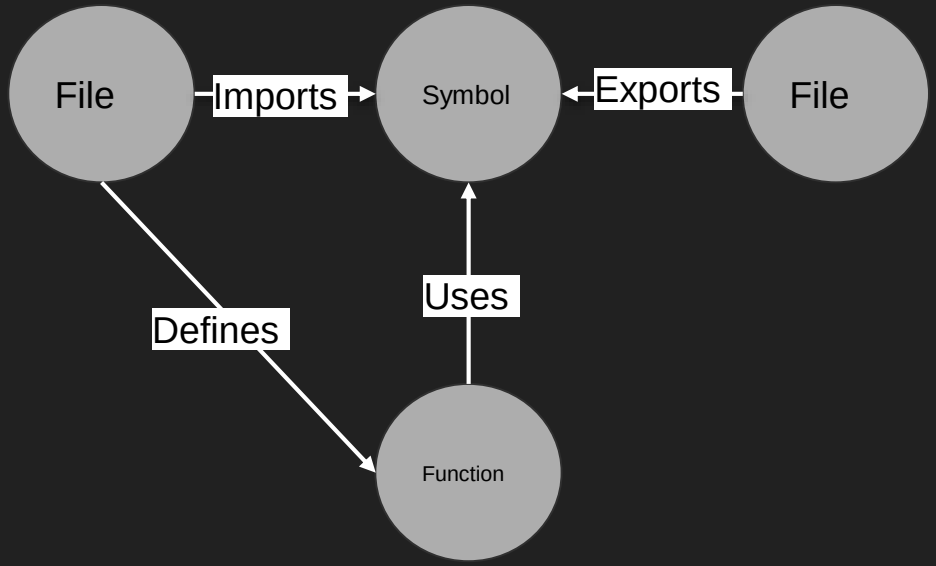
We have a file.

The file can:

- Import a symbol from another file (library)
- Export a symbol for another file
- Use the symbol in it’s functions



YAY! It looks like a graph



Graphs

- Graph
  - Node
  - Edges

Graphs

They are amazing.

We use them all the time:

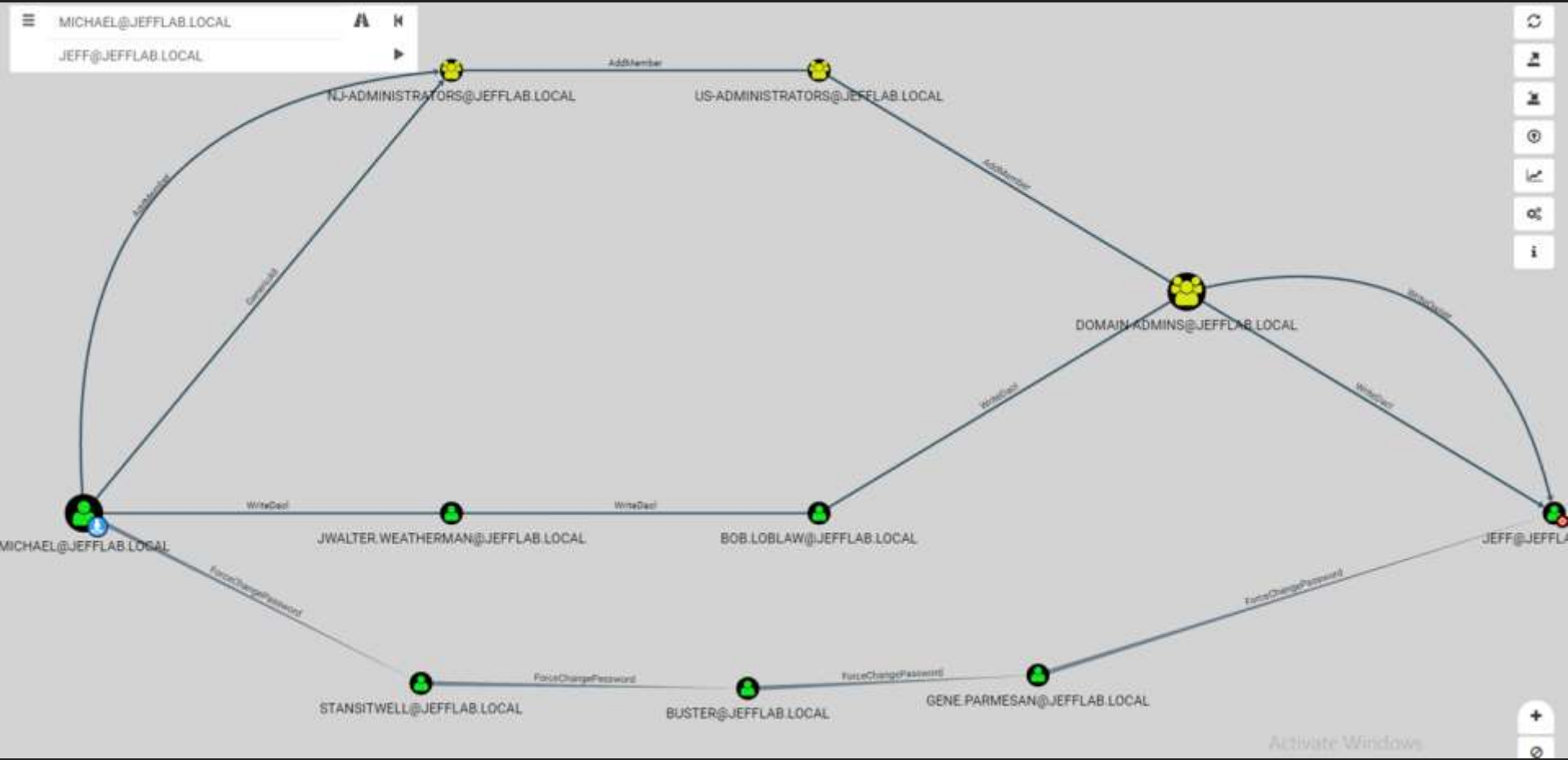
- Facebook

Graphs

They are amazing.

We use them all the time:

- Facebook
- Bloodhound

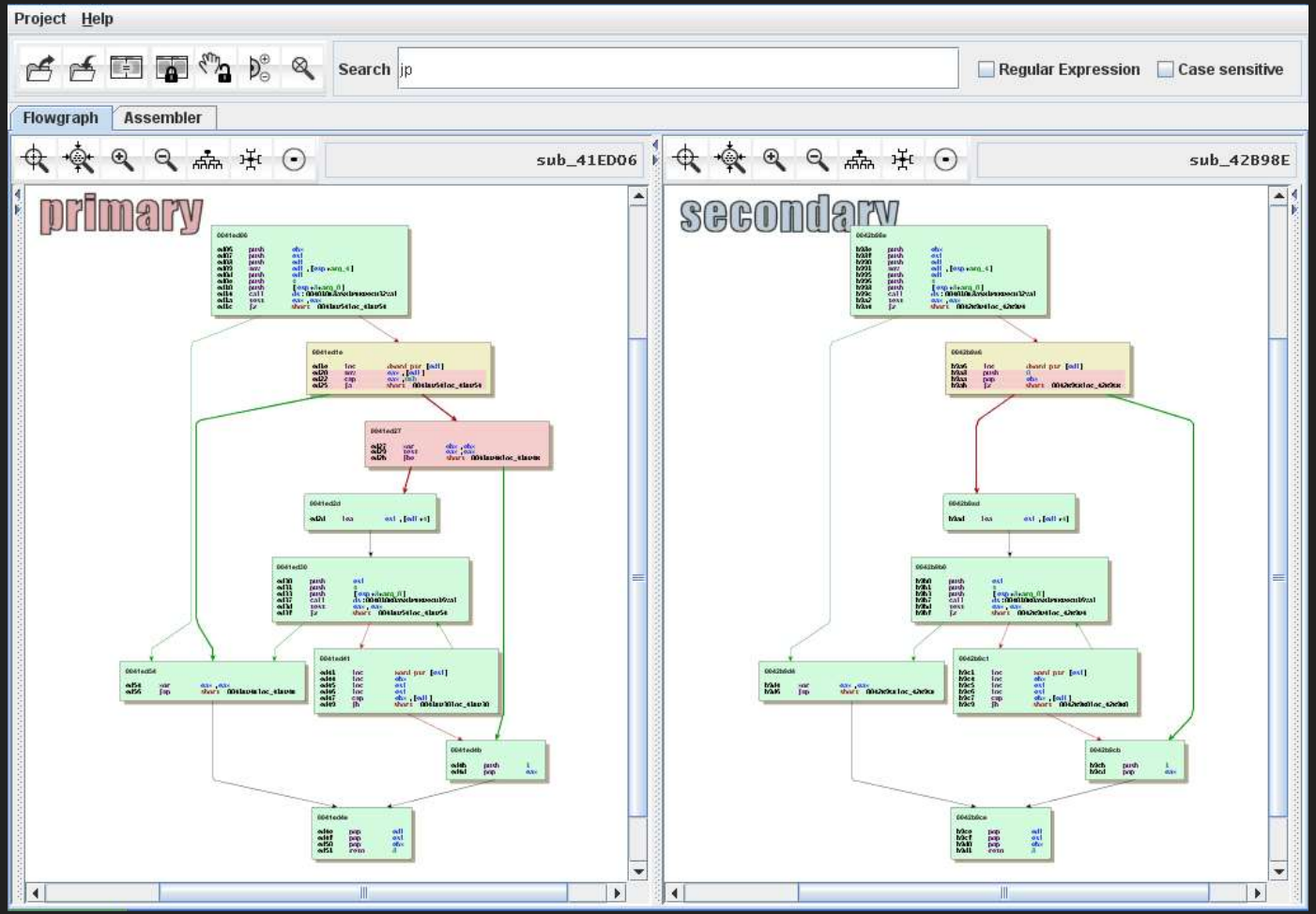

MICHAEL@JEFFLAB.LOCAL
JEFF@JEFFLAB.LOCAL
NJ-ADMINISTRATORS@JEFFLAB.LOCAL
US-ADMINISTRATORS@JEFFLAB.LOCAL
AddMember
AddMember
AddMember
GenericAll
DOMAIN ADMINS@JEFFLAB.LOCAL
WriteDacl
WriteDacl
WriteDacl
WriteDacl
WriteDacl
MICHAEL@JEFFLAB.LOCAL
JWALTER.WEATHERMAN@JEFFLAB.LOCAL
BOB.LOBLAW@JEFFLAB.LOCAL
JEFF@JEFFLA
ForceChangePassword
ForceChangePassword
ForceChangePassword
ForceChangePassword
STANSITWELL@JEFFLAB.LOCAL
BUSTER@JEFFLAB.LOCAL
GENE.PARMESAN@JEFFLAB.LOCAL
Activate Windows

Graphs

They are amazing.

We use them all the time:

- Facebook
- Bloodhound
- BinNavi/BinDiff



@aCaltum

https://github.com/ezrac/POLAR

“We treat the executable as a graph of graphs”

Thomas Dullen (halvar flake) et.al.

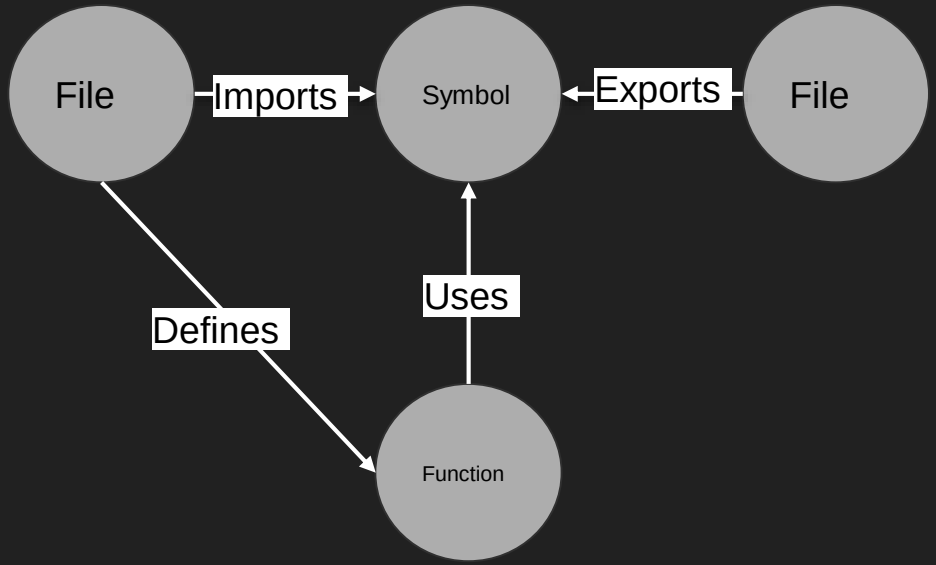
https://static.googleusercontent.com/media/www.zynamics.com/en//downloads/bindiffsstic05-1.pdf

Graphs

They are amazing.

They had been used by Law Enforcement Officials for a long time. (ie. ibm i2, xanalys)

(we will return to that in a few)


File
Imports
Symbol
Exports
File
Defines
Uses
Function

Problem

- Model the relationships on a well known format.
- **Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).**
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)
- …
- And the most important thing. Managers love shiny colors and visualizations.

How are we going to build those graphs?

- Radare2
  - For the dissasembly - the IDA version is also available.
- Neo4j
  - Why write a DB from scratch when we can use something available?
- https://github.com/ezrac/polar
  - A sh!tty parser

Let’s take a look at the code

https://github.com/ezrac/POLAR/blob/master/polar/__init__.py

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- **Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)**
- …
- And the most important thing. Managers love shiny colors and visualizations.

Let’s run it

polar-parse -db "bolt://neo4j:ezra@localhost:7687" -d squash-fs-root/*

Let’s run it

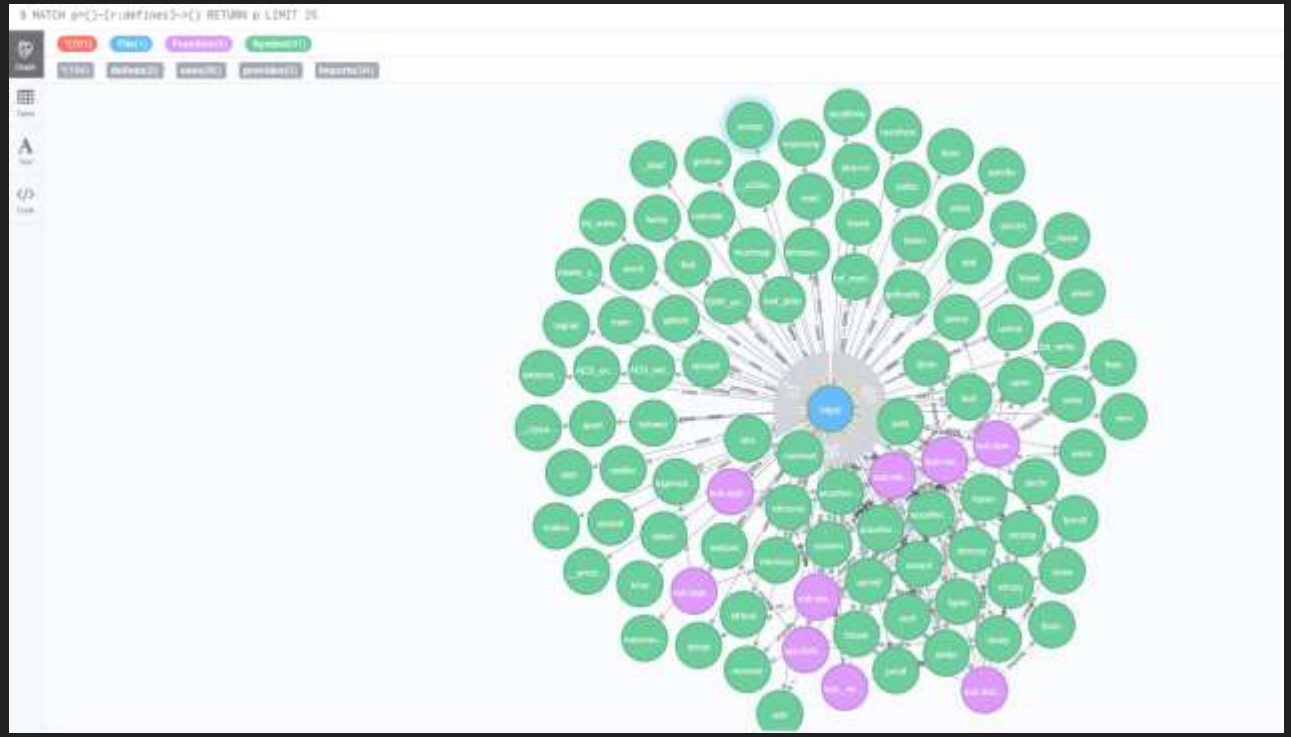
polar-parse -db "bolt://neo4j:ezra@localhost:7687" -d usr/lib

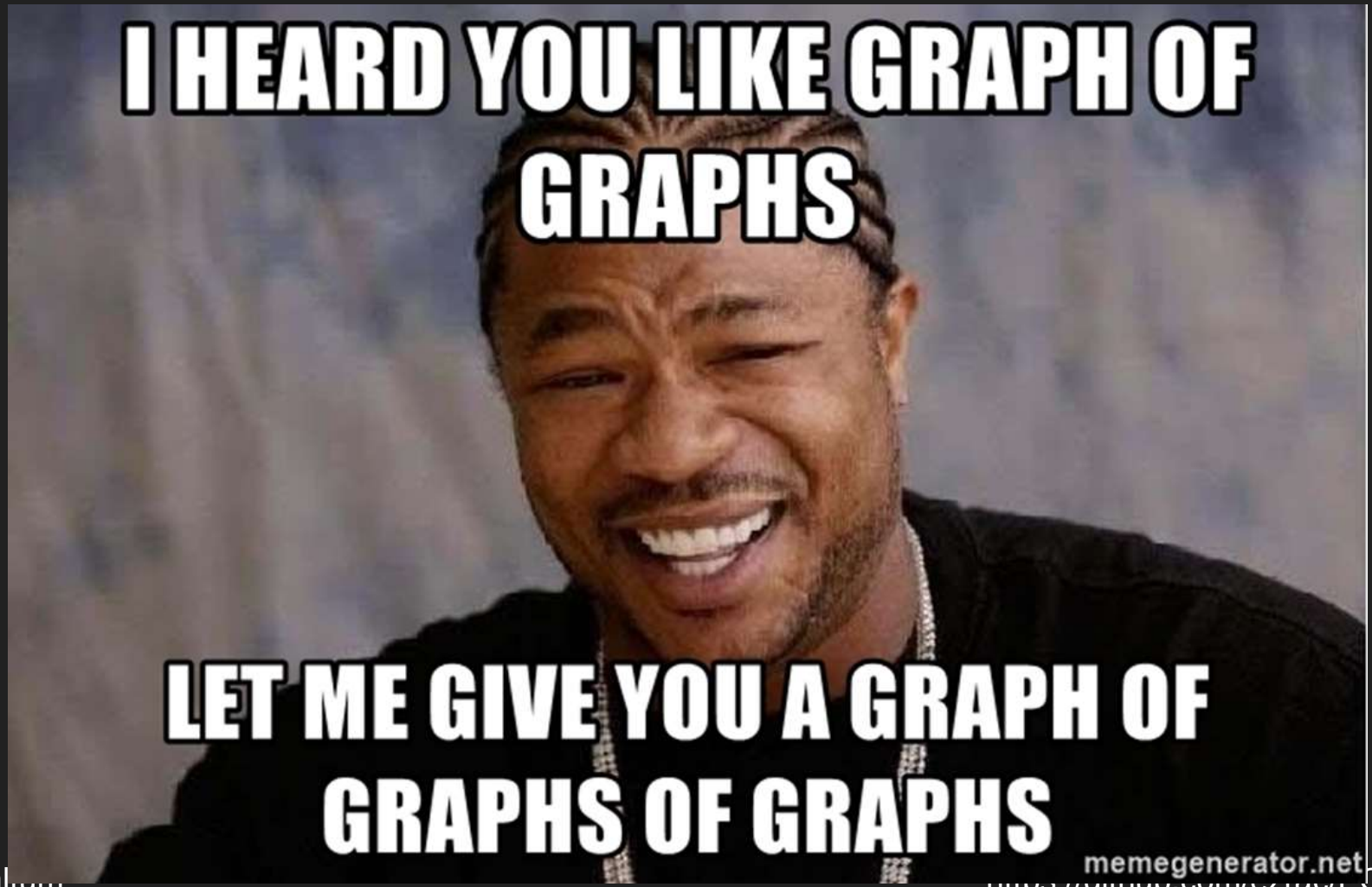
It actually takes some time, so trust me on this one

Problem

- Model the relationships on a well known format.
- Represent the relationships in a format that I can query (instead of writing a IDA script every time I need it).
- Had you ever tried adding multiple files to a single IDA project (as multiple segments for example?)
- …
- **And the most important thing. Managers love shiny colors and visualizations.**

POLAR Screenshot




I HEARD YOU LIKE GRAPH OF GRAPHS
LET ME GIVE YOU A GRAPH OF GRAPHS OF GRAPHS
memegenerator.net

Syntax time

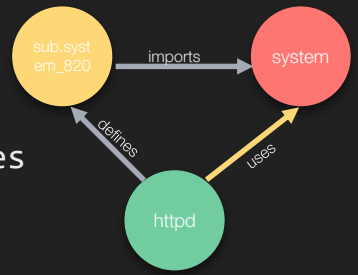
Cypher 101

Is “ASCII Art”

Cypher 101

(c:Function{name:'sub.system_820'})

(b:Symbol{name:'system'})



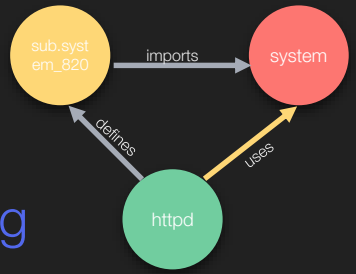
<-[:defines]-

-[:uses]->

(a:File{name:'httpd'})

```
(c:Function{name:'sub.system_820'})<-[:defines]-
(a:File)-[:uses]->(b:Symbol{name:'system'})
```

Cypher 101



Variable we're returning

Json Key-Value

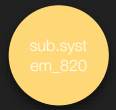
```
(a:File{name:'ht
    tpd'})
```

Node Type

Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})

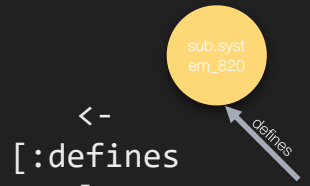
```
(c:Function{name:'sub.system_
            820'})
```



Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})<-[:defines]-

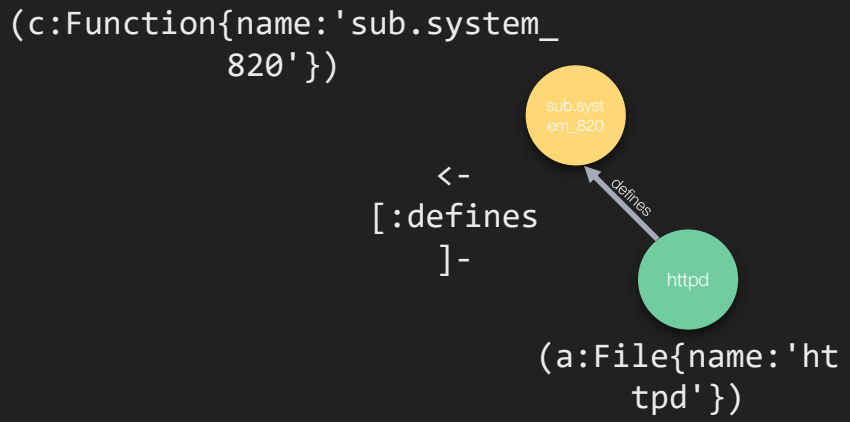
```
(c:Function{name:'sub.system_
           820'})
```




```
    <-
[:defines
    ]-
```

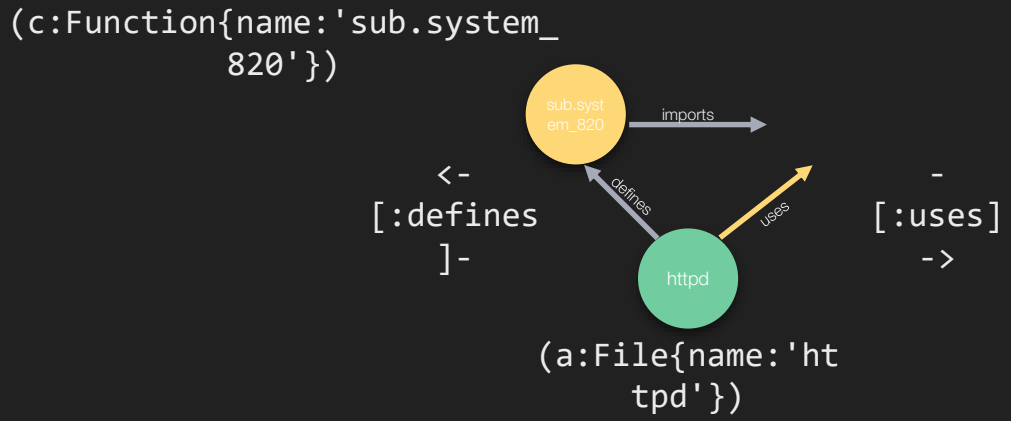
Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})<-[:defines]-(a:File)



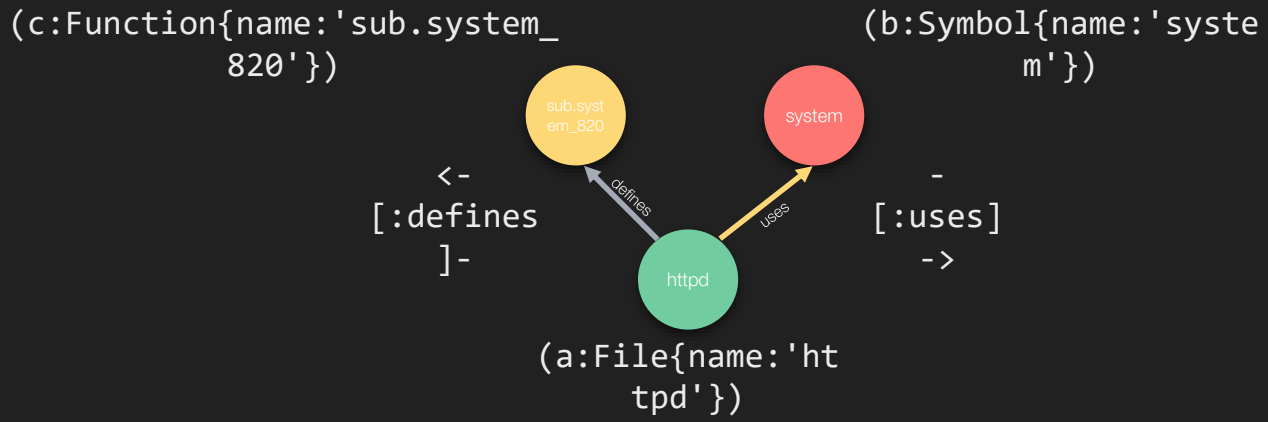
Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})<-[:defines]-(a:File)-[:uses]->



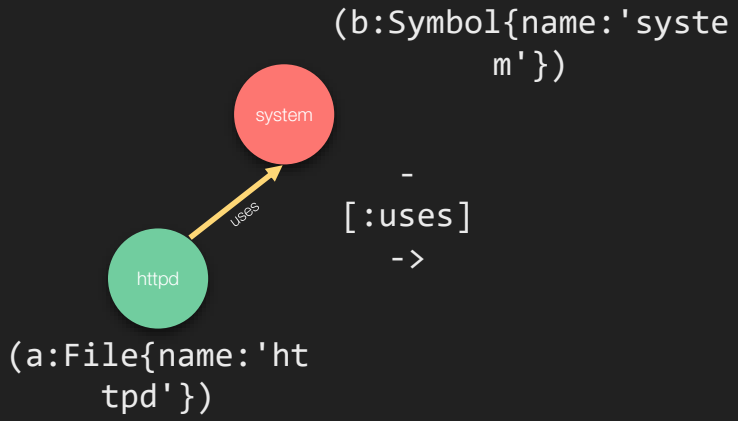
Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})<-[:defines]-(a:File)-[:uses]->(b:Symbol{name:'system'})



Cypher 101: Return

MATCH (c:Function{name:'sub.system_820'})<-[:defines]-(a:File)-[:uses]->(b:Symbol{name:'system'}) RETURN a,b



Now that we had set-up the background lets perform the vulnerability “investigation” using queries

Detective work - 101

We have a suspect?

Who this suspect talks with?

Is this suspect usually seen with other suspects?

Where are all the suspects involved?

How often does this subject talks?

Demo Time!

IoT Case

We have a suspect?

In IOT stuff, our favorite suspects are popen|system|execve

```
MATCH (a:File)-->(b:Symbol) WHERE b.name in ['system', 'execve', 'popen'        ]
RETURN a,b
```

With whom the suspect is talking?

polar-disassemble -db "bolt://neo4j:ezra@localhost:7687" -f "usr/sbin/httpd:system"

MATCH (c:Function)-->(b:Symbol) WHERE b.name in ['system', 'execve', 'popen'] return b,c

Is this suspect usually seen with other suspects?

```
MATCH (c:Function)<-[:defines]-(a:File)-->(b:Symbol), (c)-->(b2:Symbol) WHERE b.name =~ '.*system.*' and b2.name =~ '.*print.*' RETURN b,b2,c
```

Is this suspect usually seen with other suspects, doing shady stuff?

```
MATCH (c:Function)<-[:defines]-(a:File)-->(b:Symbol), (c)-->(b2:Symbol) WHERE b.name =~ '.*system.*' and b2.name =~ '.*print.*'  and c.decompilation CONTAINS '__s' RETURN b,b2,c
```

Where are all the suspects involved?

```
MATCH (a:File)-->(b:Symbol) WHERE b.name in
['strcpy','strcpyA','strcpyW','wcscpy','_tcscpy','_mbscpy','StrCpy','StrCpyA','StrCpyW','lstrcpy','lstrcpyA','lstrcpyW','_tccpy','_mbccpy','_ftcscpy','strncpy'
,'wcsncpy','_tcsncpy','_mbsncpy','_mbsnbcpy','StrCpyN','StrCpyNA','StrCpyNW','StrNCpy','strcpynA','StrNCpyA','StrNCpyW','lstrcpyn','lstrcpynA','lstrc
pynW','strcat','strcatA','strcatW','wcscat','_tcscat','_mbscat','StrCat','StrCatA','StrCatW','lstrcat','lstrcatA','lstrcatW','StrCatBuff','StrCatBuffA','StrCatBuff
W','StrCatChainW','_tccat','_mbccat','_ftcscat','strncat','wcsncat','_tcsncat','_mbsncat','_mbsnbcat','StrCatN','StrCatNA','StrCatNW','StrNCat','StrNCat
A','StrNCatW','lstrncat','lstrcatnA','lstrcatnW','lstrcatn','sprintfW','sprintfA','wsprintf','wsprintfW','wsprintfA','sprintf','swprintf','_stprintf','wvsprintf','wvsprint
fA','wvsprintfW','vsprintf','_vstprintf','vswprintf','_snwprintf','_snprintf','_sntprintf','nsprintf','wvsprintf','wvsprintfA','wvsprintfW','vsprintf','_vstprintf','vswpri
ntf','strncpy','wcsncpy','_tcsncpy','_mbsncpy','_mbsnbcpy','StrCpyN','StrCpyNA','StrCpyNW','StrNCpy','strcpynA','StrNCpyA','StrNCpyW','lstrcpyn','lstr
cpynA','lstrcpynW','_fstrncpy','strncat','wcsncat','_tcsncat','_mbsncat','_mbsnbcat','StrCatN','StrCatNA','StrCatNW','StrNCat','StrNCatA','StrNCatW','lstr
ncat','lstrcatnA','lstrcatnW','lstrcatn','_fstrncat','alloca','_alloca','strlen','wcslen','_mbslen','_mbstrlen','StrLen','lstrlen','memcpy','RtlCopyMemory','Copy
Memory','wmemcpy']  RETURN a,b limit 1000
```

How often does this subject talks?

MATCH (a:File)-[r:uses]->(b:Symbol) RETURN b, count(r) as usages order by usages desc

Network aware stuff?

MATCH (a:File)--(b:Symbol) WHERE b.name in ['accept', 'bind', 'listen', 'recv'] return a,b

What are you leaking from me?

```
MATCH (a:File)--(c:Function)-->(b:Symbol) WHERE b.name in ['sendto', 'send',
'sendmsg'            ] and c.decompilation contains 'Authorization'  return a, c
```

```
MATCH (a:File)--(c:Function)-->(b:Symbol) WHERE b.name in ['sendto', 'send',
'sendmsg'            ] and c.decompilation contains 'HTTP'  return a, c
```

Little Experiment (just wrote it this morning)

- MATCH (c:Function)<-[:defines]-(a:File{name:'httpd'})-->(b:Symbol), (c)-->(b2:Symbol) WHERE b.name =~ '.*memcpy.*' and not b2.name in ['strlen', 'strcpy'] RETURN b,b2,c

Main points

This is not automatic

It still needs the work of the vulnerability researcher

It helped me. I hope it can help you

This is an open source project

If it helps you, or you have a cool query, share it with me

TODO

I’ll be releasing the playbook over the weekend.

An IDA version of the script is on the works.

I need your help:

- Do you have any ideas of stuff I can find?
- How to best manage the “playbook” ? Git? Wiki? Google Docs ?

The Real Story Behind this talk

I was not going to release this

I felt that there was no need

“Haters gonna hate” – When asking for advice somebody told me that this is irrelevant

I was unable to explain the usage at the beginning – thus people thought this is a tool to do graphical representations

My friends told me that the worst thing that could happen, is that nobody would use it

But I had got some very good feedback

Acknowledgements

@iiamit

@inbarraz

@shiftreduce

@realgam3

Resources

https://github.com/joxeankoret/diaphora by @matalaz

https://www.zynamics.com/software.html by @halvarflake

https://static.googleusercontent.com/media/www.zynamics.com/en//downloads/bindiffsstic05-1.pdf by @halvarflake

https://github.com/ShiftLeftSecurity/codepropertygraph by @fabsx00

http://mlsec.org/joern/ by @fabsx00

Questions?