Story Time

Graph the Planet 2020

Seth Summersett

Willi Ballenthin

FIREEYE™

Seth Summersett

- Developer
- Reverse Engineer
- Data
- Mgmt

@williballenthin

- Reverse Engineer
- Forensics
- Malware

goals of an intrusion investigation

- determine earliest and most recent **dates of compromise**
- enumerate methods of **access to environment**, including:
  - initial compromise
  - persistent malware
  - methods of lateral movement
- scope the compromise
  - identify **compromised systems**
  - describe **data exposure**
- **attribute** activities to threat groups

phases of intrusion investigation

- there are two aspects of “doing forensics”:
    - artifact identification
    - interpretation

- **artifact identification**: given all collected evidence, which artifacts are related to malicious activity?

- **interpretation**: given all identified artifacts, demonstrate that evidence backs up answers to the *goals of intrusion investigation*.

one task: classify artifacts into buckets

- goal: take a boatload of artifacts and decide if they are relevant to the intrusion investigation.

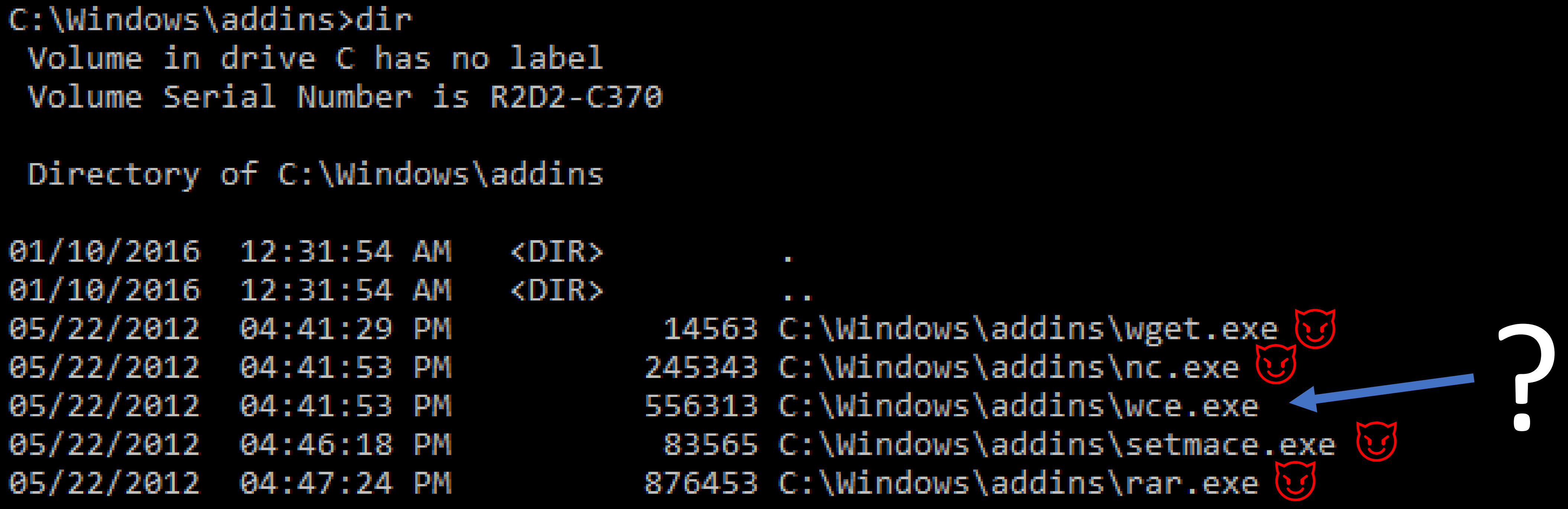
- buckets:
  - **relevant**: attacker actions created or changed the artifact.

    malware payload. persistence key. backdoor file creation timestamp.

  - **not relevant**: legitimate user actions created or changed the artifact.

    os installation date. facebook logon. minesweeper high score.

example

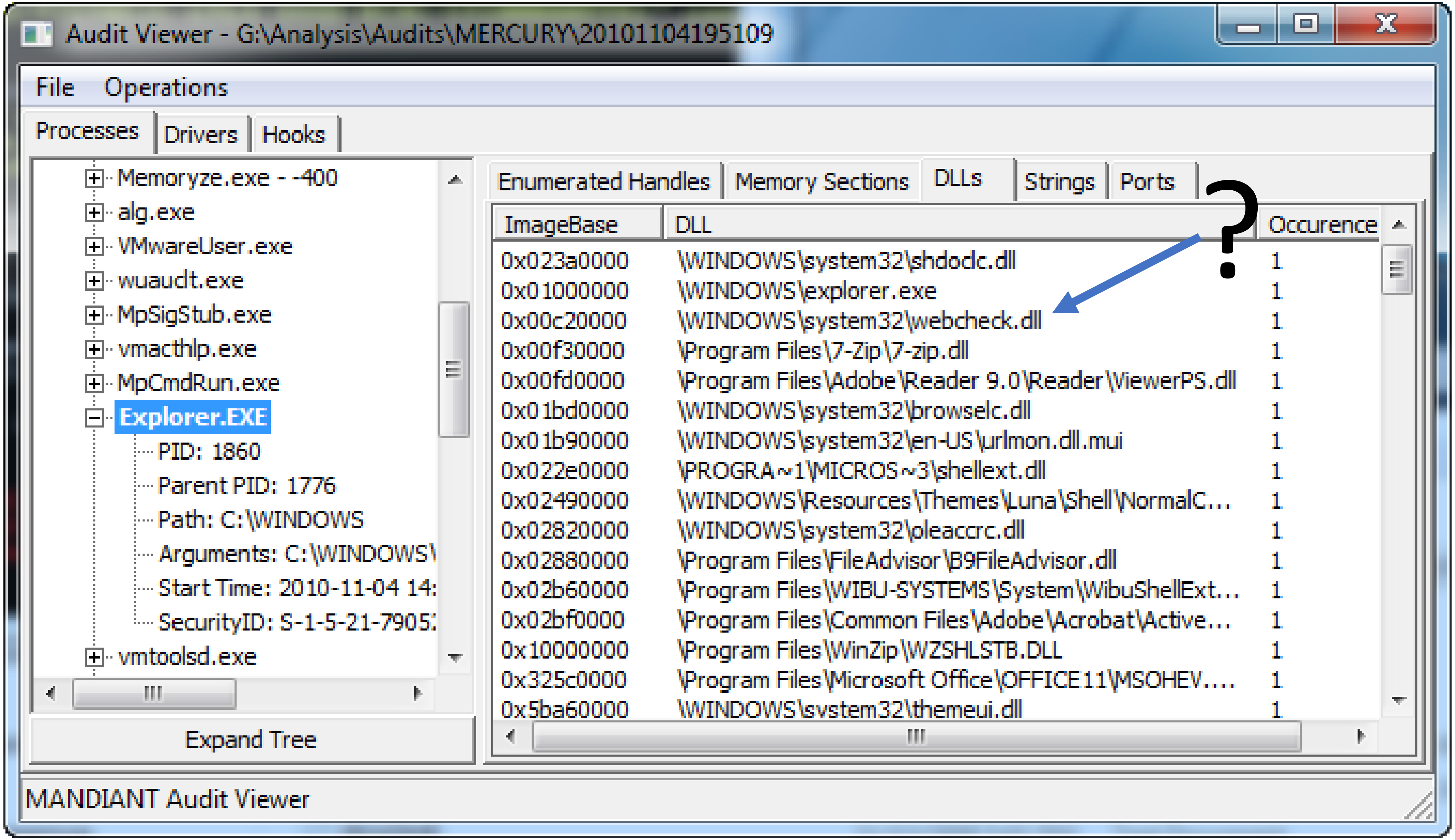


```
C:\Windows\addins>dir
 Volume in drive C has no label
 Volume Serial Number is R2D2-C370

 Directory of C:\Windows\addins

01/10/2016  12:31:54 AM   <DIR>          .
01/10/2016  12:31:54 AM   <DIR>          ..
05/22/2012  04:41:29 PM           14563 C:\Windows\addins\wget.exe 😈
05/22/2012  04:41:53 PM          245343 C:\Windows\addins\nc.exe 😈
05/22/2012  04:41:53 PM          556313 C:\Windows\addins\wce.exe
05/22/2012  04:46:18 PM           83565 C:\Windows\addins\setmace.exe 😈
05/22/2012  04:47:24 PM          876453 C:\Windows\addins\rar.exe 😈
```

example



?

```
⊞ Memoryze.exe - -400
⊞ alg.exe
⊞ VMwareUser.exe
⊞ wuauclt.exe
⊞ MpSigStub.exe
⊞ vmacthlp.exe
⊞ MpCmdRun.exe
⊟ Explorer.EXE
    PID: 1860
    Parent PID: 1776
    Path: C:\WINDOWS
    Arguments: C:\WINDOWS\
    Start Time: 2010-11-04 14:
    SecurityID: S-1-5-21-7905:
⊞ vmtoolsd.exe
```

ImageBase	DLL	Occurence
0x023a0000	\WINDOWS\system32\shdoclc.dll	1
0x01000000	\WINDOWS\explorer.exe	1
0x00c20000	\WINDOWS\system32\webcheck.dll	1
0x00f30000	\Program Files\7-Zip\7-zip.dll	1
0x00fd0000	\Program Files\Adobe\Reader 9.0\Reader\ViewerPS.dll	1
0x01bd0000	\WINDOWS\system32\browselc.dll	1
0x01b90000	\WINDOWS\system32\en-US\urlmon.dll.mui	1
0x022e0000	\PROGRA~1\MICROS~3\shellext.dll	1
0x02490000	\WINDOWS\Resources\Themes\Luna\Shell\NormalC…	1
0x02820000	\WINDOWS\system32\oleaccrc.dll	1
0x02880000	\Program Files\FileAdvisor\B9FileAdvisor.dll	1
0x02b60000	\Program Files\WIBU-SYSTEMS\System\WibuShellExt…	1
0x02bf0000	\Program Files\Common Files\Adobe\Acrobat\Active…	1
0x10000000	\Program Files\WinZip\WZSHLSTB.DLL	1
0x325c0000	\Program Files\Microsoft Office\OFFICE11\MSOHEV.…	1
0x5ba60000	\WINDOWS\system32\themeui.dll	1

is webcheck.dll related to the intrusion?

you might do the following:

- lookup md5 hash of `webcheck.dll` on file system against VirusTotal.
- find other processes that have loaded `webcheck.dll`.
- timeline load of `webcheck.dll` against creation timestamps on file system.
- enumerate registry keys that point to `webcheck.dll`.
- consider files that exist in the same directory as `webcheck.dll`.

is webcheck.dll related to the intrusion?

how would you do the following?

- lookup md5 hash of `webcheck.dll` on file system against VirusTotal.
- find other processes that have loaded `webcheck.dll`.
- timeline load of `webcheck.dll` against creation timestamps on file system.
- enumerate registry keys that point to `webcheck.dll`.
- consider files that exist in the same directory as `webcheck.dll`.

thesis:

our primary investigative tools do not help us easily classify artifacts.

today, we manage alerts **largely in a vacuum** as a single event in time

alert validation is a time-consuming process to **collect and contextualize** metadata

why?

- existing classification tools are typically low dimensional.
    - data is organized into lists or **tables of things**.
    - **one table per artifact** type.
    - links among tables are rare. (lots of development complexity here.)

- meaning:
    - artifacts must be inspected in a vacuum, or manual joining required.
    - they usually cannot provide the context we need to make a decision.

when tools produce independent tables…

to correlate, the analyst must manually do the “join”.

eg. “match the path of the dll in the process listing to the path in the file listing to determine the md5sum”.

pid	process	dll
124	explorer.exe	kernel32.dll
124	explorer.exe	advapi32.dll
124	explorer.exe	webclient.dll

table 1: *volatility loaded modules*

path	created	md5
C:/windows/temp/1.txt	2016-12-10	789abc…
C:/windows/system32/webclient.dll	2017-01-10	d1e2f3…
C:/users/user/Desktop/a.exe	2016-12-11	4a5d6c…

table 2: *sleuthkit file listing*

manual joining is the worst!

- slow
- tedious
- error-prone
- not fun!

manual joining is the worst!

- slow
- tedious
- error-prone
- not fun!

→ this discourages the analyst from asking the questions they mean

  - maybe there is patience for 10 joins, but is that enough?

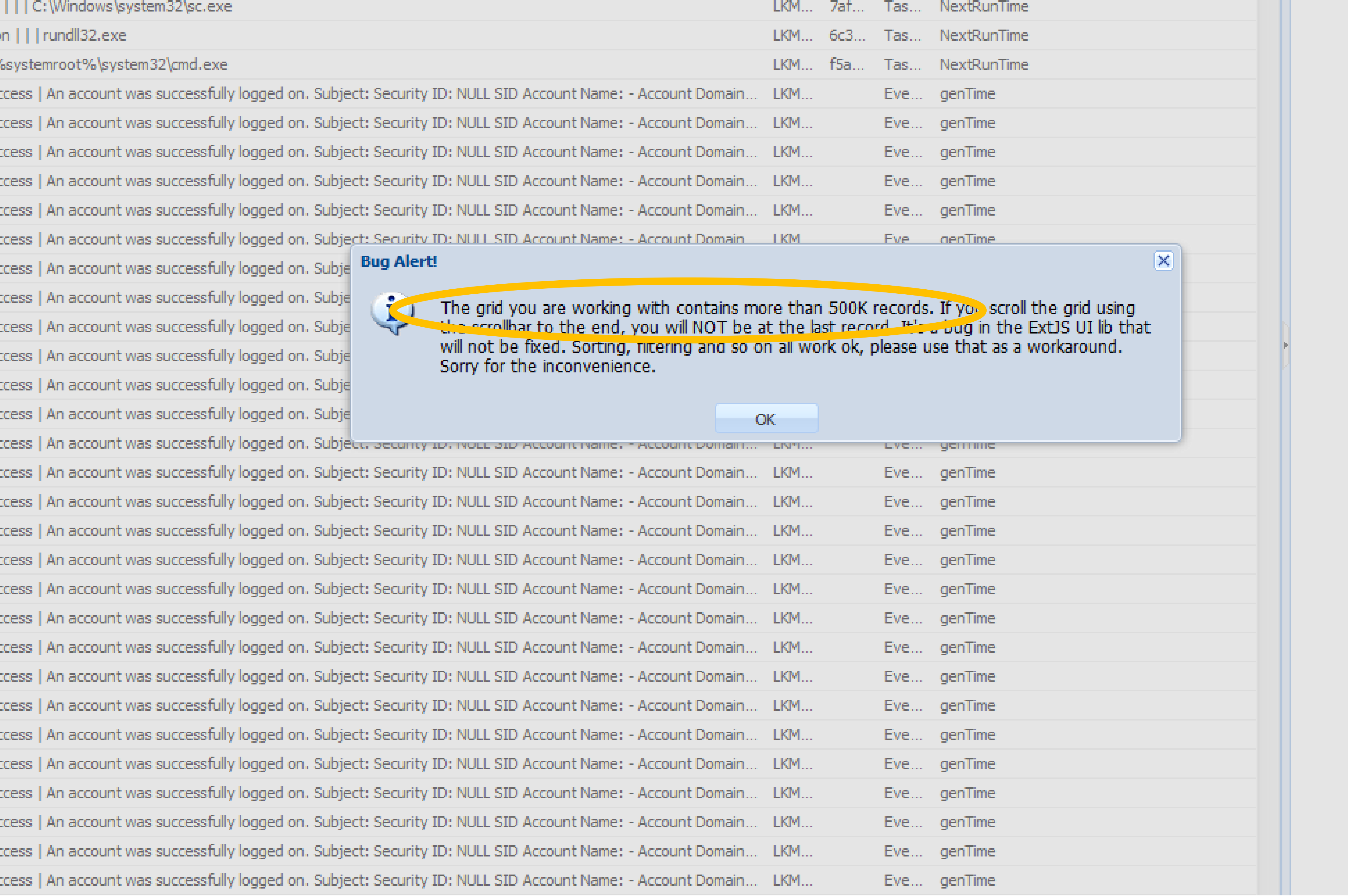
manual joining is the worst!

- slow
- tedious
- error-prone
- not fun!

- this discourages the analyst from asking the questions they mean
  - maybe there is patience for 10 joins, but is that enough?

→ this encourages the analyst to ask questions they **don’t** really mean

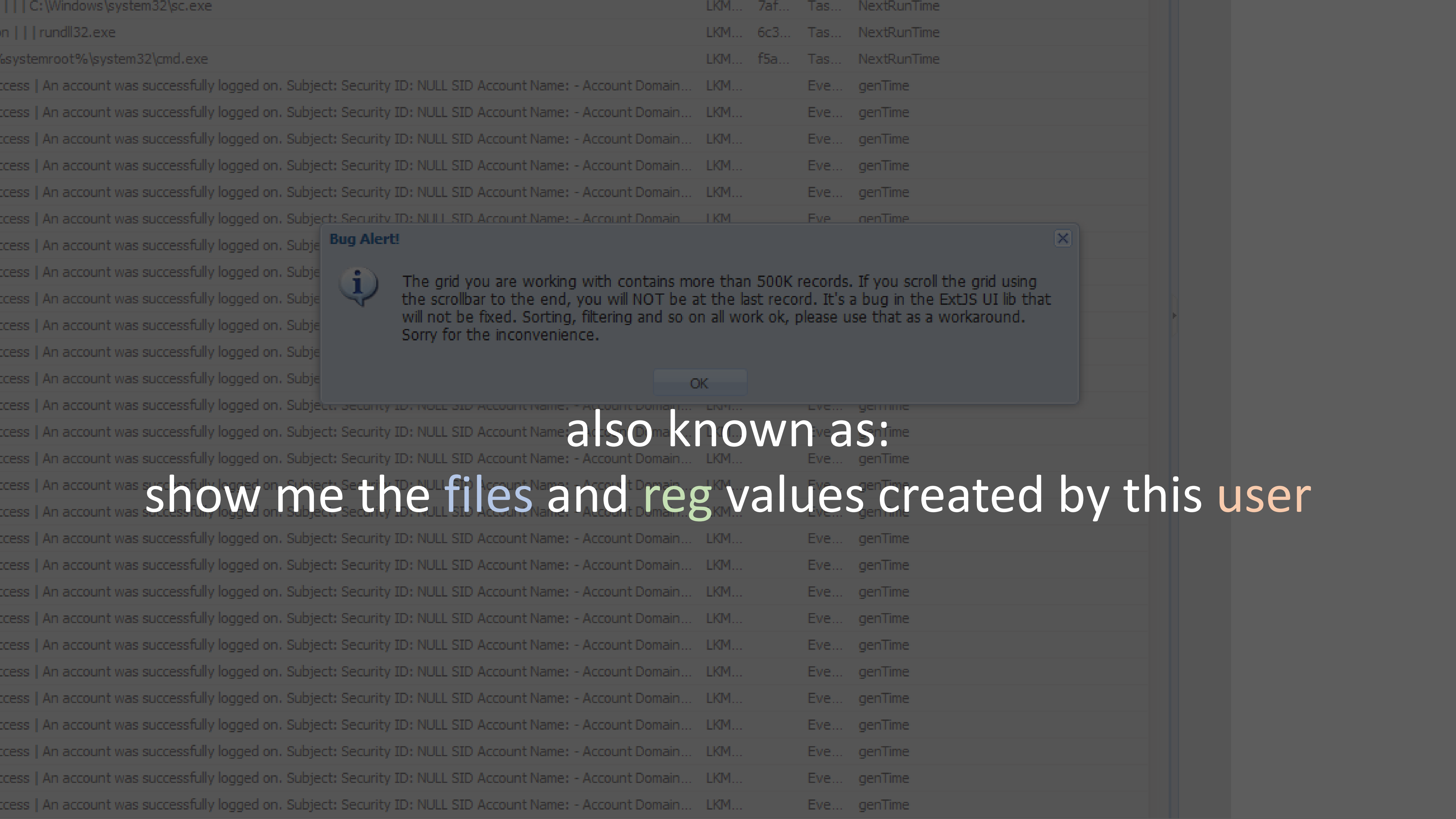
- ask the easy questions that are only moderately helpful



| | | C:\Windows\system32\sc.exe	LKM...	7af...	Tas...	NextRunTime
n | | | rundll32.exe	LKM...	6c3...	Tas...	NextRunTime
%systemroot%\system32\cmd.exe	LKM...	f5a...	Tas...	NextRunTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subje				
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...	LKM...		Eve...	genTime

Bug Alert!

The grid you are working with contains more than 500K records. If you scroll the grid using the scrollbar to the end, you will NOT be at the last record. It's a bug in the ExtJS UI lib that will not be fixed. Sorting, filtering and so on all work ok, please use that as a workaround. Sorry for the inconvenience.



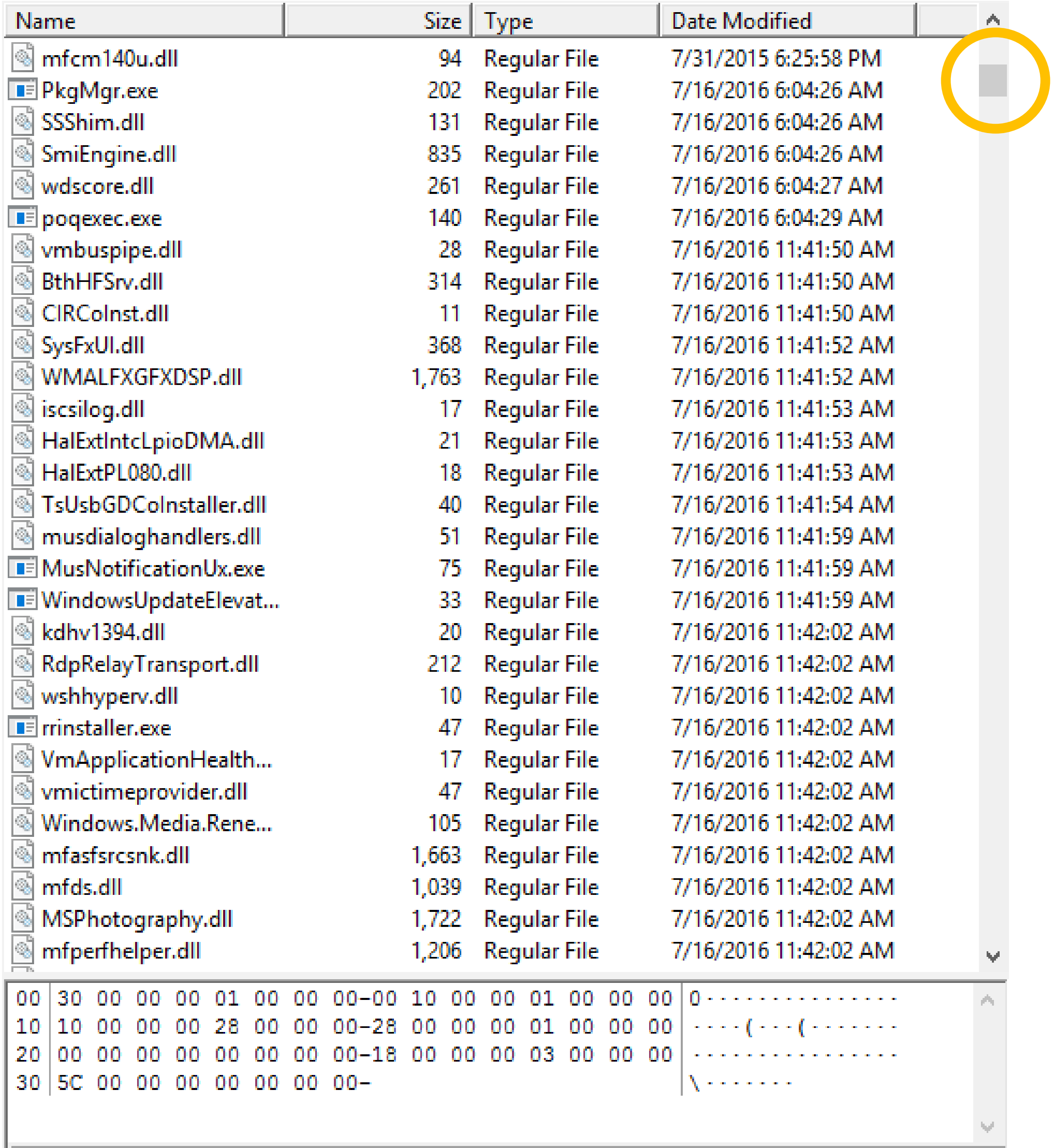
```
| | | C:\Windows\system32\sc.exe                                                                LKM...  7af...  Tas...  NextRunTime
on | | | rundll32.exe                                                                            LKM...  6c3...  Tas...  NextRunTime
%systemroot%\system32\cmd.exe                                                                  LKM...  f5a...  Tas...  NextRunTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain   LKM     Eve     genTime
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subje
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
ccess | An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain...  LKM...  Eve...  genTime
```

Bug Alert!

The grid you are working with contains more than 500K records. If you scroll the grid using the scrollbar to the end, you will NOT be at the last record. It's a bug in the ExtJS UI lib that will not be fixed. Sorting, filtering and so on all work ok, please use that as a workaround. Sorry for the inconvenience.

also known as:

show me the files and reg values created by this user

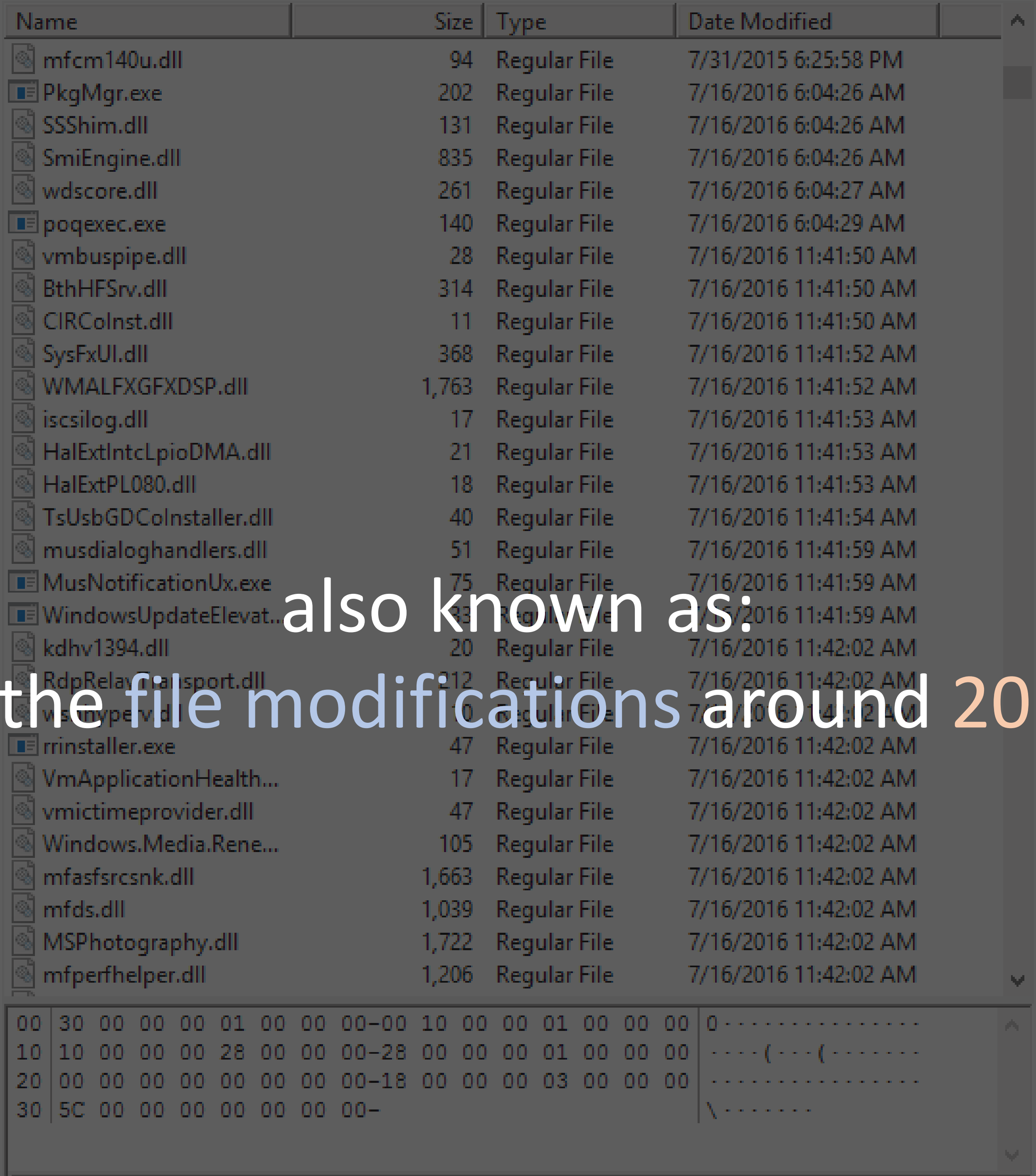


Name	Size	Type	Date Modified
mfcm140u.dll	94	Regular File	7/31/2015 6:25:58 PM
PkgMgr.exe	202	Regular File	7/16/2016 6:04:26 AM
SSShim.dll	131	Regular File	7/16/2016 6:04:26 AM
SmiEngine.dll	835	Regular File	7/16/2016 6:04:26 AM
wdscore.dll	261	Regular File	7/16/2016 6:04:27 AM
poqexec.exe	140	Regular File	7/16/2016 6:04:29 AM
vmbuspipe.dll	28	Regular File	7/16/2016 11:41:50 AM
BthHFSrv.dll	314	Regular File	7/16/2016 11:41:50 AM
CIRCoInst.dll	11	Regular File	7/16/2016 11:41:50 AM
SysFxUI.dll	368	Regular File	7/16/2016 11:41:52 AM
WMALFXGFXDSP.dll	1,763	Regular File	7/16/2016 11:41:52 AM
iscsilog.dll	17	Regular File	7/16/2016 11:41:53 AM
HalExtIntcLpioDMA.dll	21	Regular File	7/16/2016 11:41:53 AM
HalExtPL080.dll	18	Regular File	7/16/2016 11:41:53 AM
TsUsbGDCoInstaller.dll	40	Regular File	7/16/2016 11:41:54 AM
musdialoghandlers.dll	51	Regular File	7/16/2016 11:41:59 AM
MusNotificationUx.exe	75	Regular File	7/16/2016 11:41:59 AM
WindowsUpdateElevat...	33	Regular File	7/16/2016 11:41:59 AM
kdhv1394.dll	20	Regular File	7/16/2016 11:42:02 AM
RdpRelayTransport.dll	212	Regular File	7/16/2016 11:42:02 AM
wshhyperv.dll	10	Regular File	7/16/2016 11:42:02 AM
rrinstaller.exe	47	Regular File	7/16/2016 11:42:02 AM
VmApplicationHealth...	17	Regular File	7/16/2016 11:42:02 AM
vmictimeprovider.dll	47	Regular File	7/16/2016 11:42:02 AM
Windows.Media.Rene...	105	Regular File	7/16/2016 11:42:02 AM
mfasfsrcsnk.dll	1,663	Regular File	7/16/2016 11:42:02 AM
mfds.dll	1,039	Regular File	7/16/2016 11:42:02 AM
MSPhotography.dll	1,722	Regular File	7/16/2016 11:42:02 AM
mfperfhelper.dll	1,206	Regular File	7/16/2016 11:42:02 AM

```
00  30 00 00 00 01 00 00 00-00 10 00 00 01 00 00 00  0...............
10  10 00 00 00 28 00 00 00-28 00 00 00 01 00 00 00  ....(...(.......
20  00 00 00 00 00 00 00 00-18 00 00 00 03 00 00 00  ................
30  5C 00 00 00 00 00 00 00-                         \.......
```

also known as:

timeline the file modifications around 2017-01-10



Name	Size	Type	Date Modified
mfcm140u.dll	94	Regular File	7/31/2015 6:25:58 PM
PkgMgr.exe	202	Regular File	7/16/2016 6:04:26 AM
SSShim.dll	131	Regular File	7/16/2016 6:04:26 AM
SmiEngine.dll	835	Regular File	7/16/2016 6:04:26 AM
wdscore.dll	261	Regular File	7/16/2016 6:04:27 AM
poqexec.exe	140	Regular File	7/16/2016 6:04:29 AM
vmbuspipe.dll	28	Regular File	7/16/2016 11:41:50 AM
BthHFSrv.dll	314	Regular File	7/16/2016 11:41:50 AM
CIRCoInst.dll	11	Regular File	7/16/2016 11:41:50 AM
SysFxUI.dll	368	Regular File	7/16/2016 11:41:52 AM
WMALFXGFXDSP.dll	1,763	Regular File	7/16/2016 11:41:52 AM
iscsilog.dll	17	Regular File	7/16/2016 11:41:53 AM
HalExtIntcLpioDMA.dll	21	Regular File	7/16/2016 11:41:53 AM
HalExtPL080.dll	18	Regular File	7/16/2016 11:41:53 AM
TsUsbGDCoInstaller.dll	40	Regular File	7/16/2016 11:41:54 AM
musdialoghandlers.dll	51	Regular File	7/16/2016 11:41:59 AM
MusNotificationUx.exe	75	Regular File	7/16/2016 11:41:59 AM
WindowsUpdateElevat...	33	Regular File	7/16/2016 11:41:59 AM
kdhv1394.dll	20	Regular File	7/16/2016 11:42:02 AM
RdpRelayTransport.dll	212	Regular File	7/16/2016 11:42:02 AM
wshhyperv.dll	10	Regular File	7/16/2016 11:42:02 AM
rrinstaller.exe	47	Regular File	7/16/2016 11:42:02 AM
VmApplicationHealth...	17	Regular File	7/16/2016 11:42:02 AM
vmictimeprovider.dll	47	Regular File	7/16/2016 11:42:02 AM
Windows.Media.Rene...	105	Regular File	7/16/2016 11:42:02 AM
mfasfsrcsnk.dll	1,663	Regular File	7/16/2016 11:42:02 AM
mfds.dll	1,039	Regular File	7/16/2016 11:42:02 AM
MSPhotography.dll	1,722	Regular File	7/16/2016 11:42:02 AM
mfperfhelper.dll	1,206	Regular File	7/16/2016 11:42:02 AM

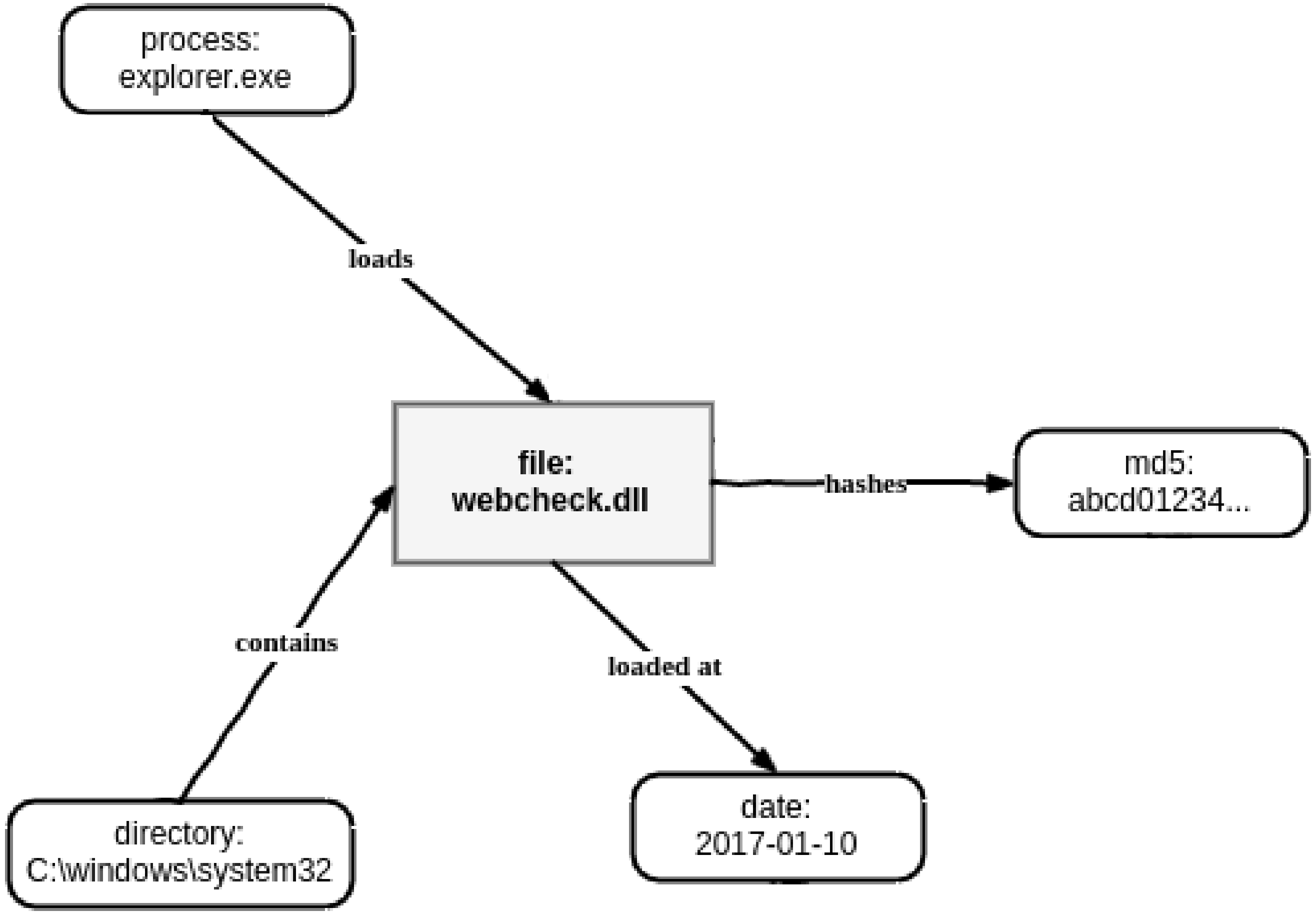
```
00  30 00 00 00 01 00 00 00-00 10 00 00 01 00 00 00  0···············
10  10 00 00 00 28 00 00 00-28 00 00 00 01 00 00 00  ····(···(·······
20  00 00 00 00 00 00 00 00-18 00 00 00 03 00 00 00  ················
30  5C 00 00 00 00 00 00 00-                         \·······
```

proposition:

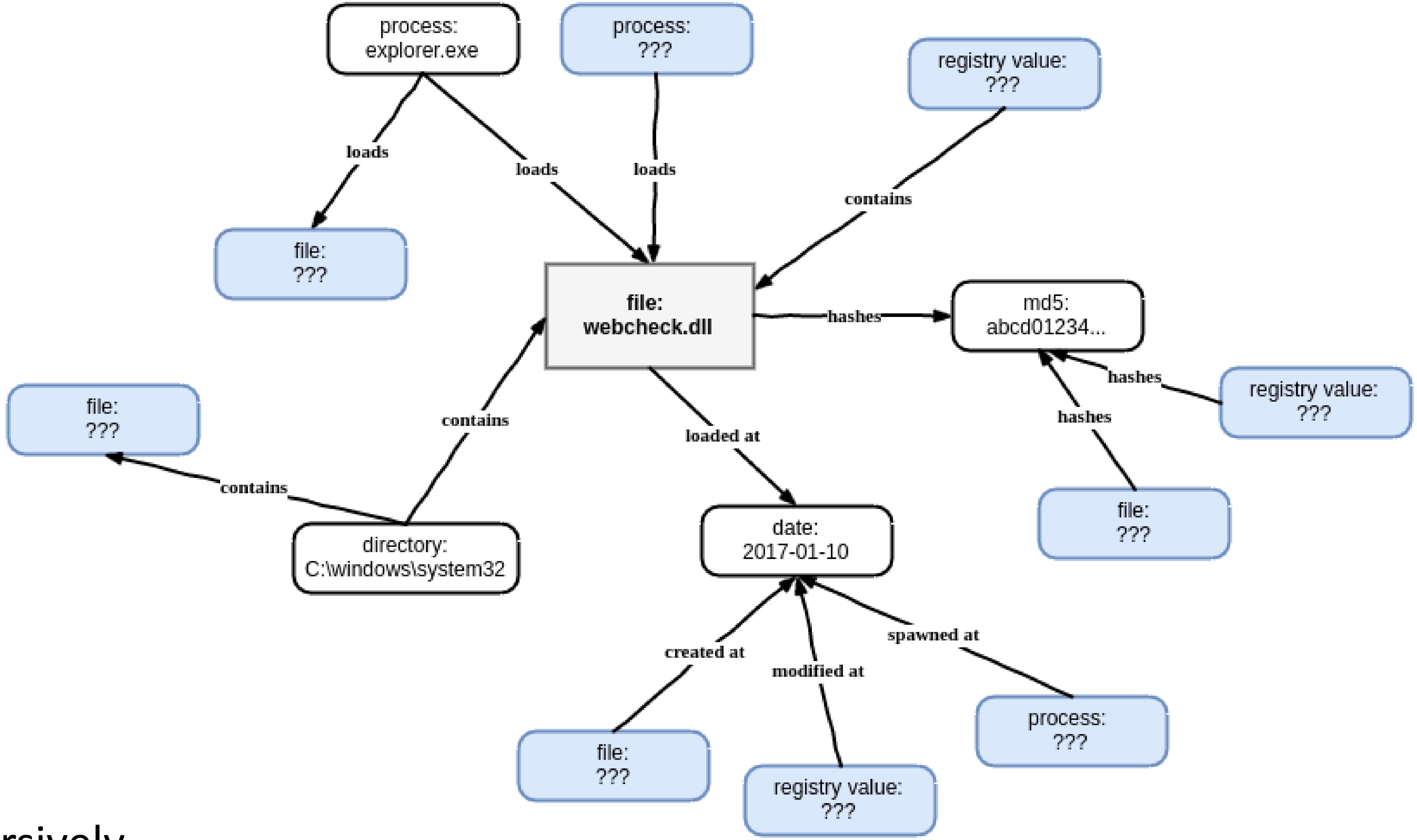
our tools should represent artifacts as a graph

file:
webcheck.dll

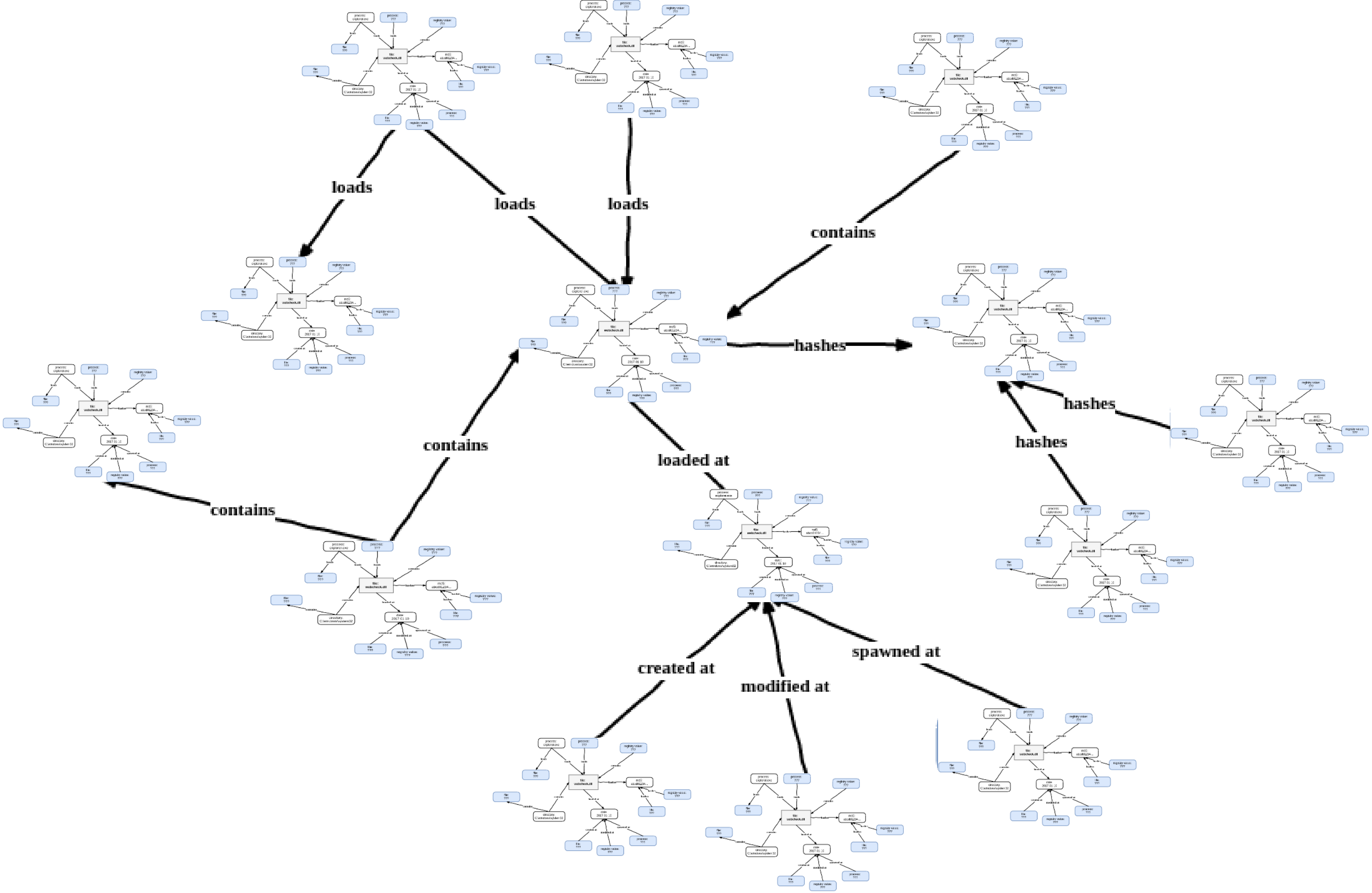
when we have an artifact of interest…

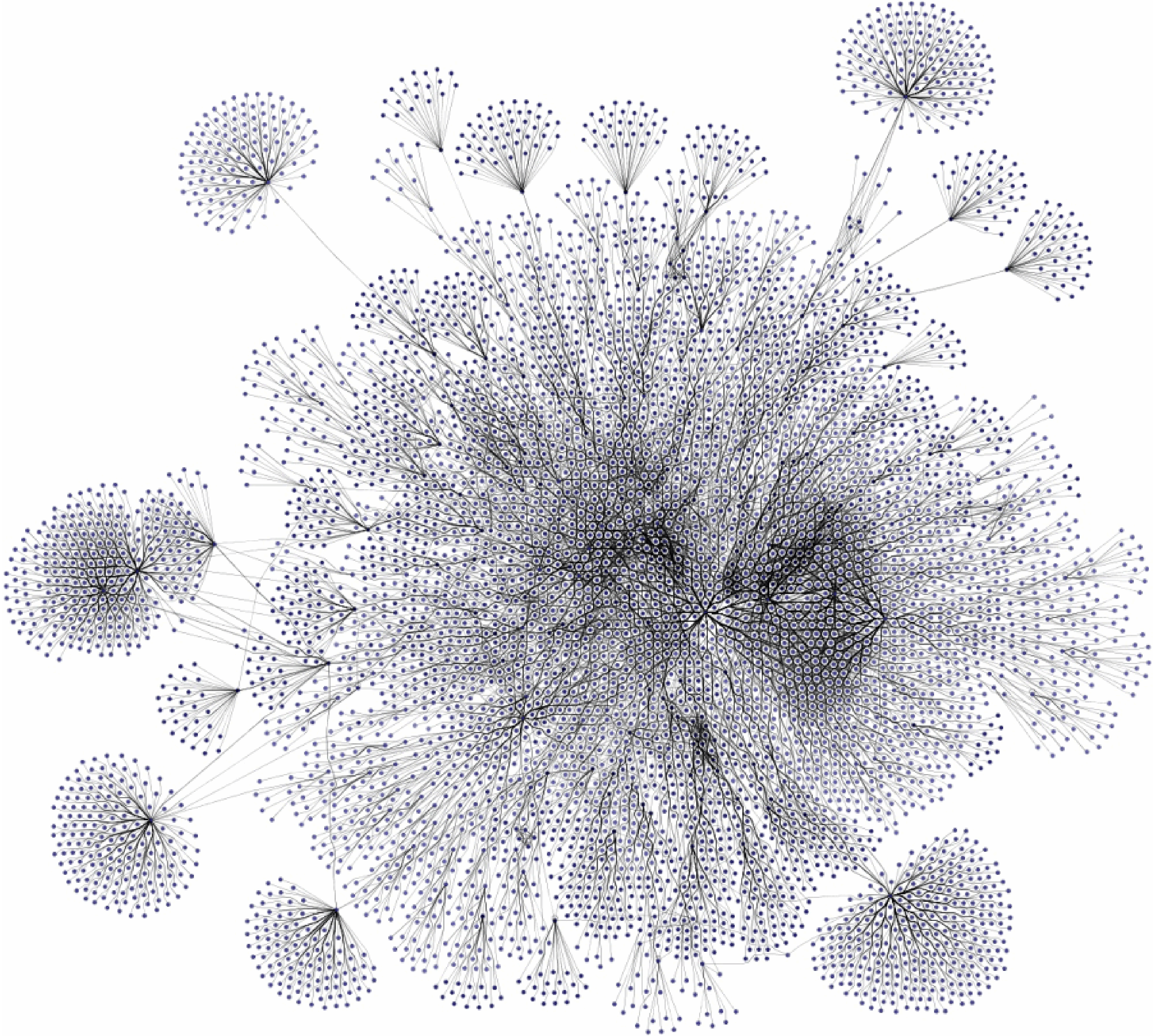


we must be able to ask for every place it’s referenced…



recursively.


loads
loads
loads
contains
hashes
hashes
hashes
contains
contains
loaded at
created at
modified at
spawned at
process: explorer.exe
process: ???
registry value: ???
file: ???
file: webcheck.dll
md5: abcd01234...
registry value: ???
file: ???
directory: C:\windows\system32
date: 2017 01 10
file: ???
registry value: ???
process: ???
loads
contains
hashes
loaded at
created at
modified at
spawned at



StoryTime

StoryTime

represent artifacts in a graph

maintain the graph on each host-based agent

display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes

StoryTime

represent artifacts in a graph

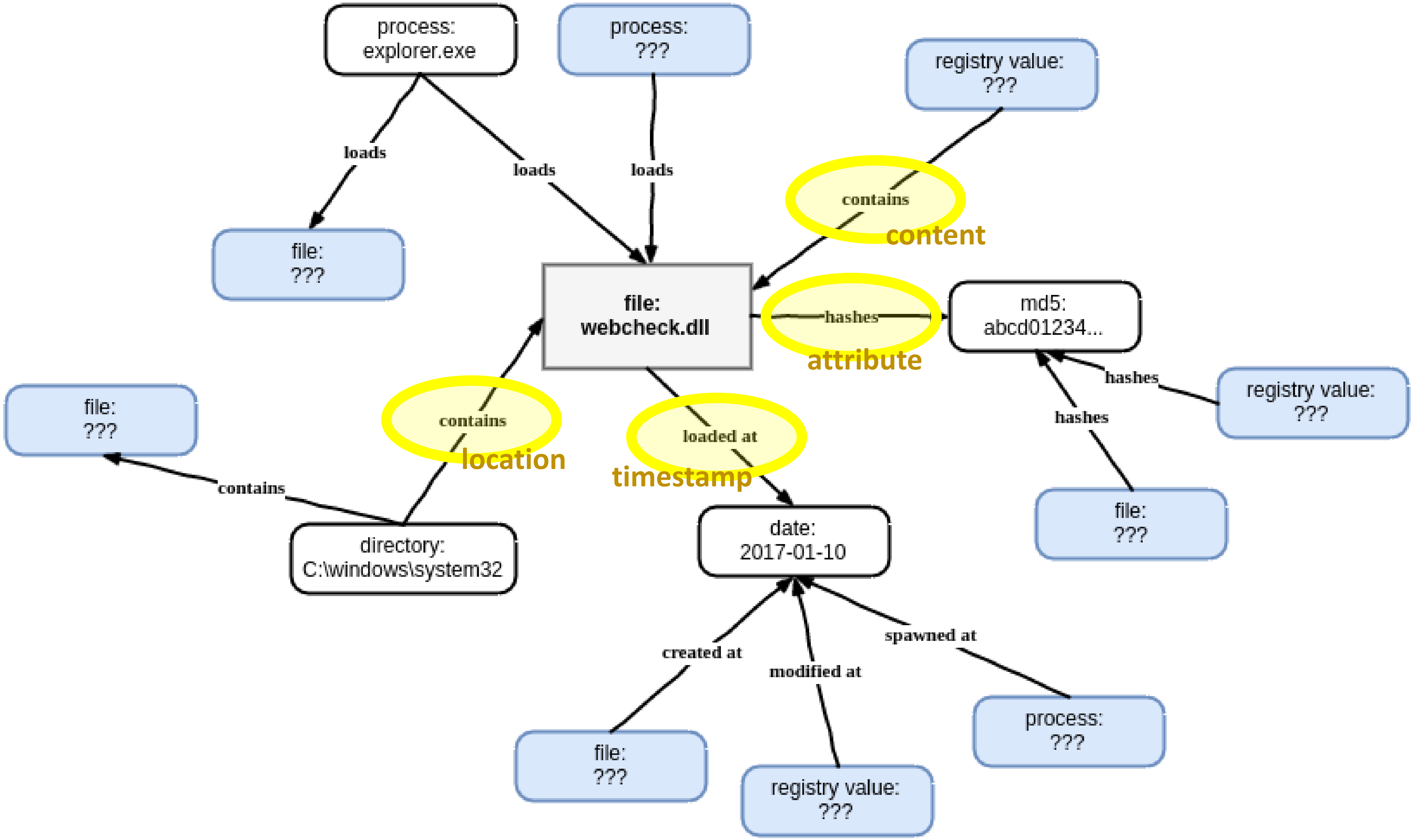
maintain the graph on each host-based agent

display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
content
file:
???
file:
webcheck.dll
hashes
attribute
md5:
abcd01234...
hashes
hashes
registry value:
???
file:
???
contains
location
loaded at
timestamp
contains
file:
???
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???

StoryTime

represent artifacts in a graph

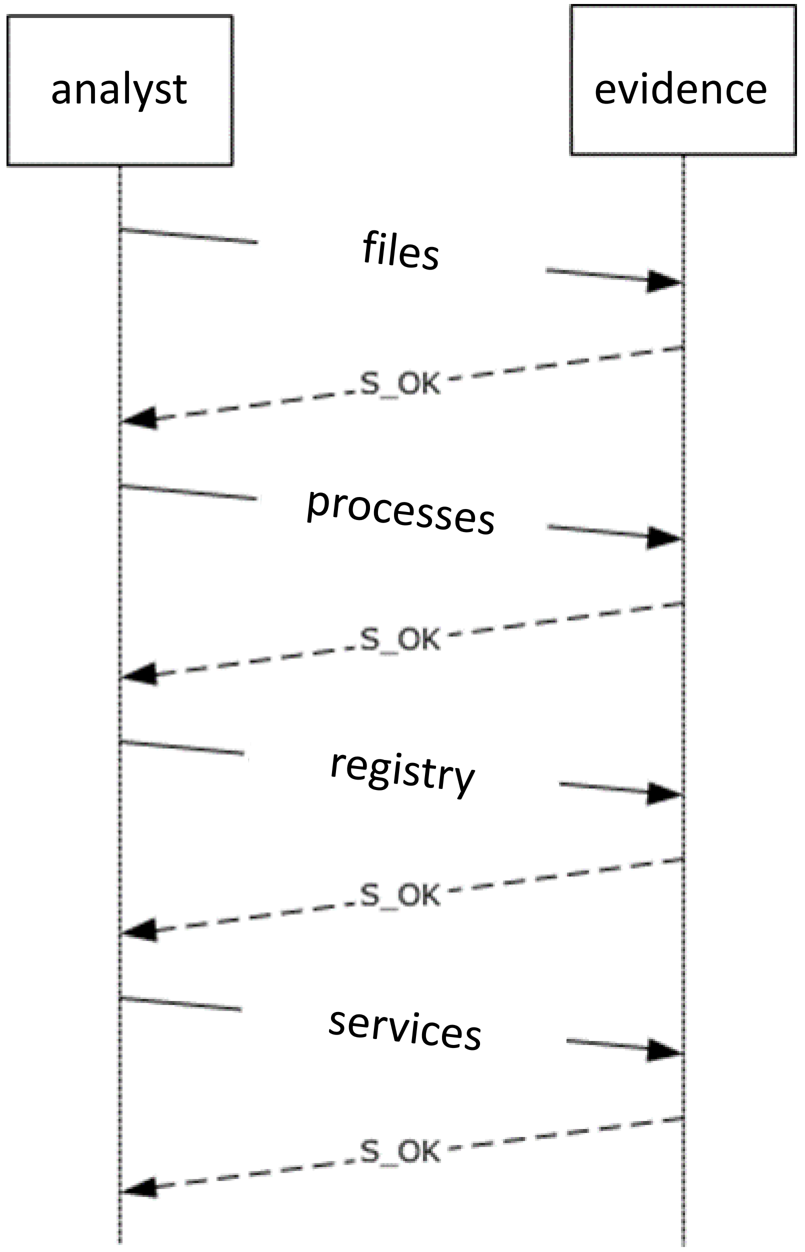
maintain the graph on each host-based agent

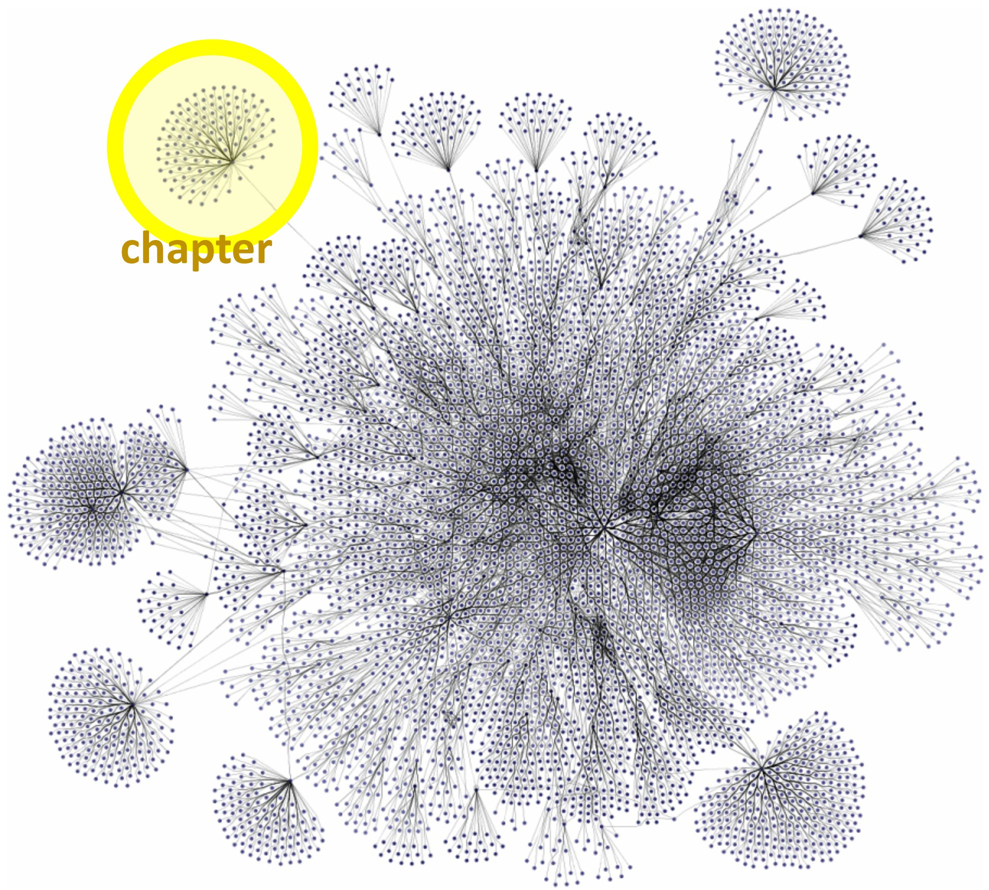
display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes


analyst
evidence
files
S_OK
processes
S_OK
registry
S_OK
services
S_OK


chapter

StoryTime

represent artifacts in a graph

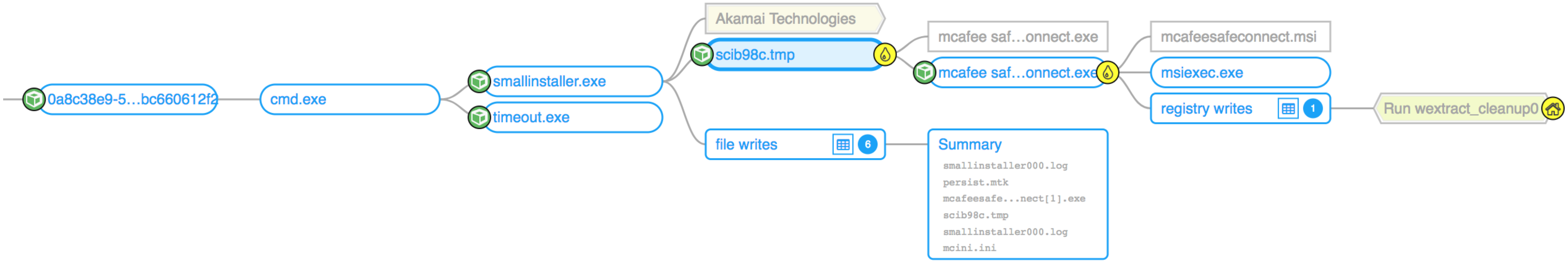
maintain the graph on each host-based agent

display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes


0a8c38e9-5…bc660612f2
cmd.exe
smallinstaller.exe
timeout.exe
Akamai Technologies
scib98c.tmp
file writes
6
mcafee saf…onnect.exe
mcafee saf…onnect.exe
Summary
smallinstaller000.log
persist.mtk
mcafeesafe...nect[1].exe
scib98c.tmp
smallinstaller000.log
mcini.ini
mcafeesafeconnect.msi
msiexec.exe
registry writes
1
Run wextract_cleanup0

StoryTime

represent artifacts in a graph

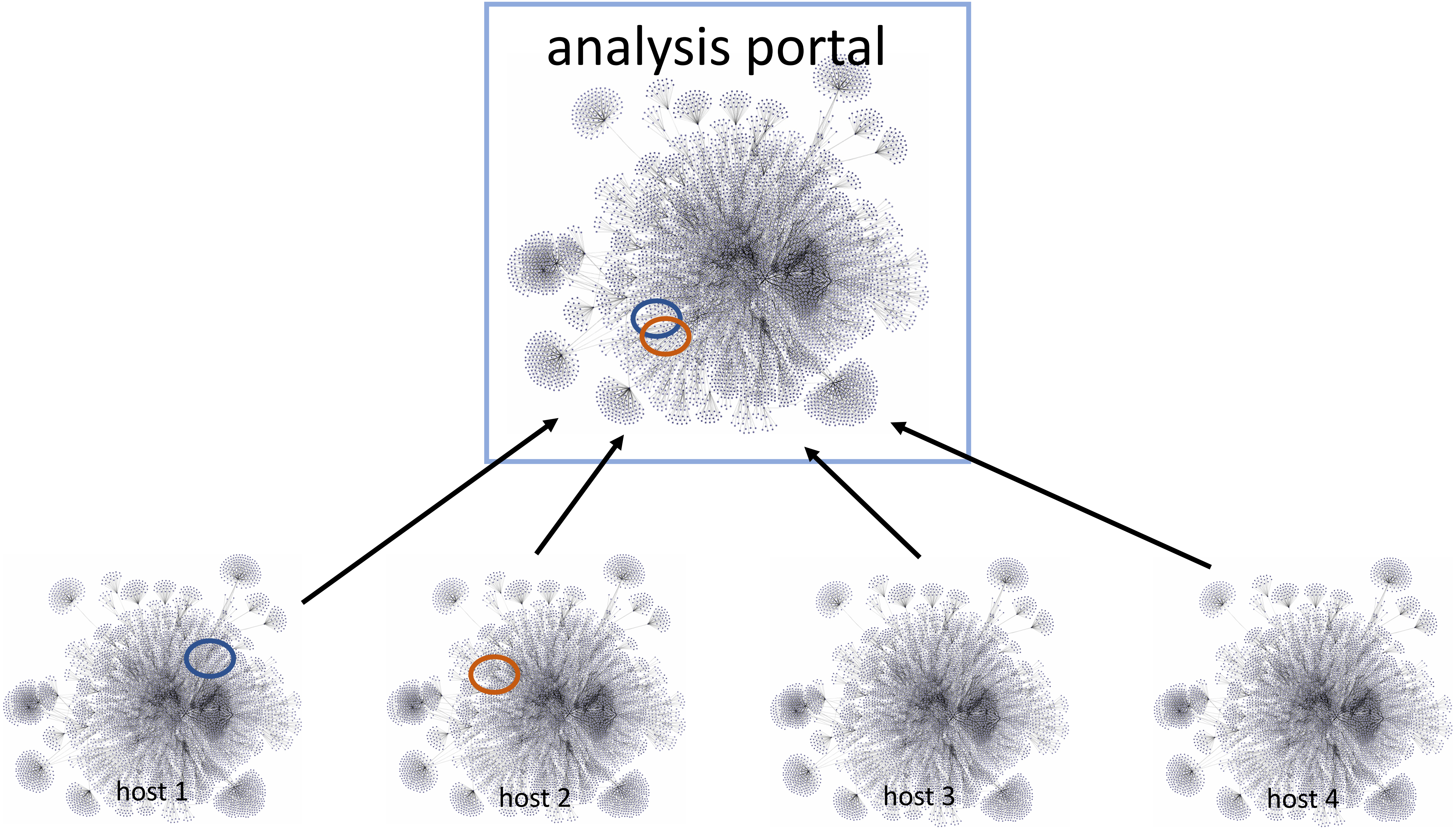
maintain the graph on each host-based agent

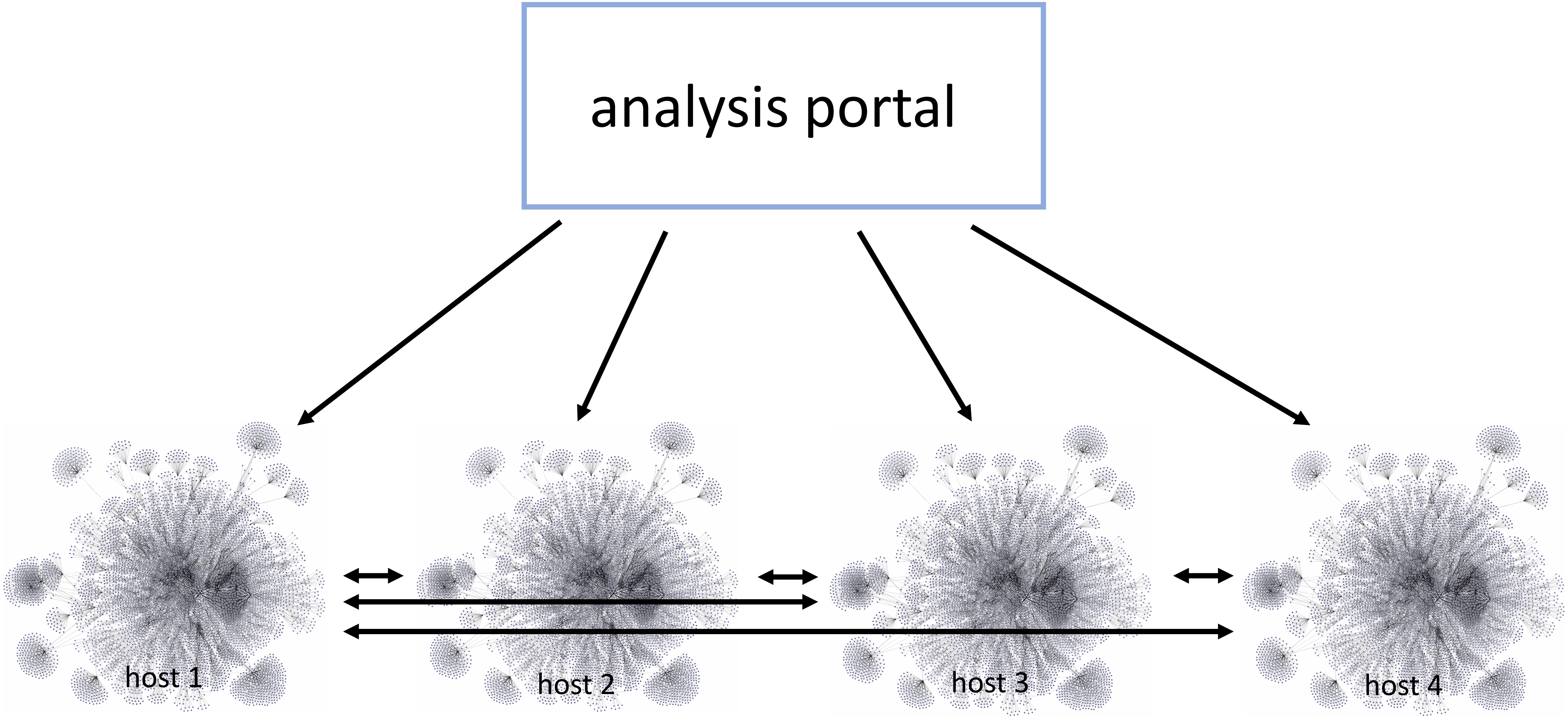
display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes


analysis portal
host 1
host 2
host 3
host 4



analysis portal

host 1

host 2

host 3

host 4

StoryTime

represent artifacts in a graph

maintain the graph on each host-based agent

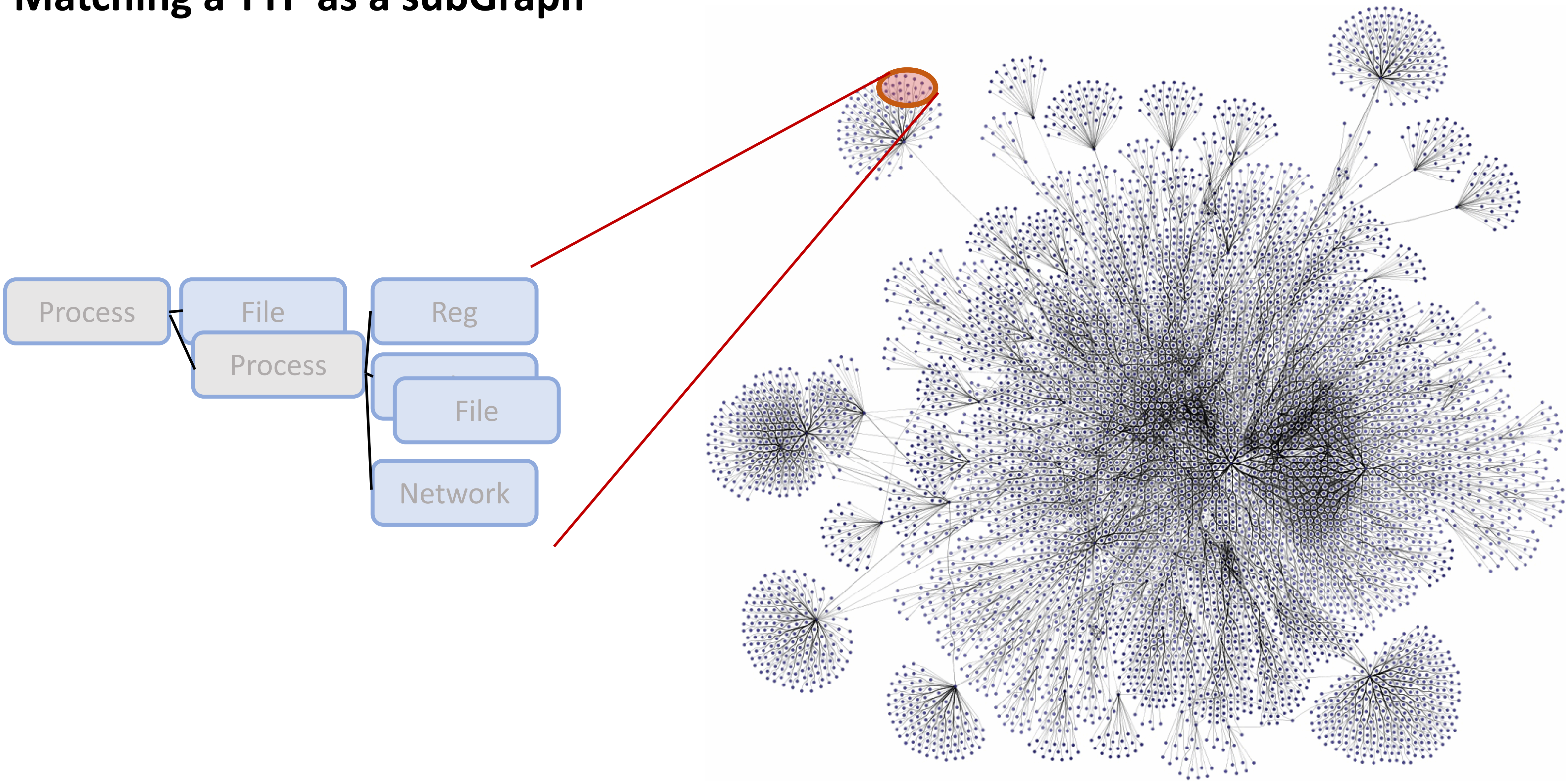
display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes

Matching a TTP as a subGraph



Process

File

Process

Reg

File

Network

StoryTime

represent artifacts in a graph

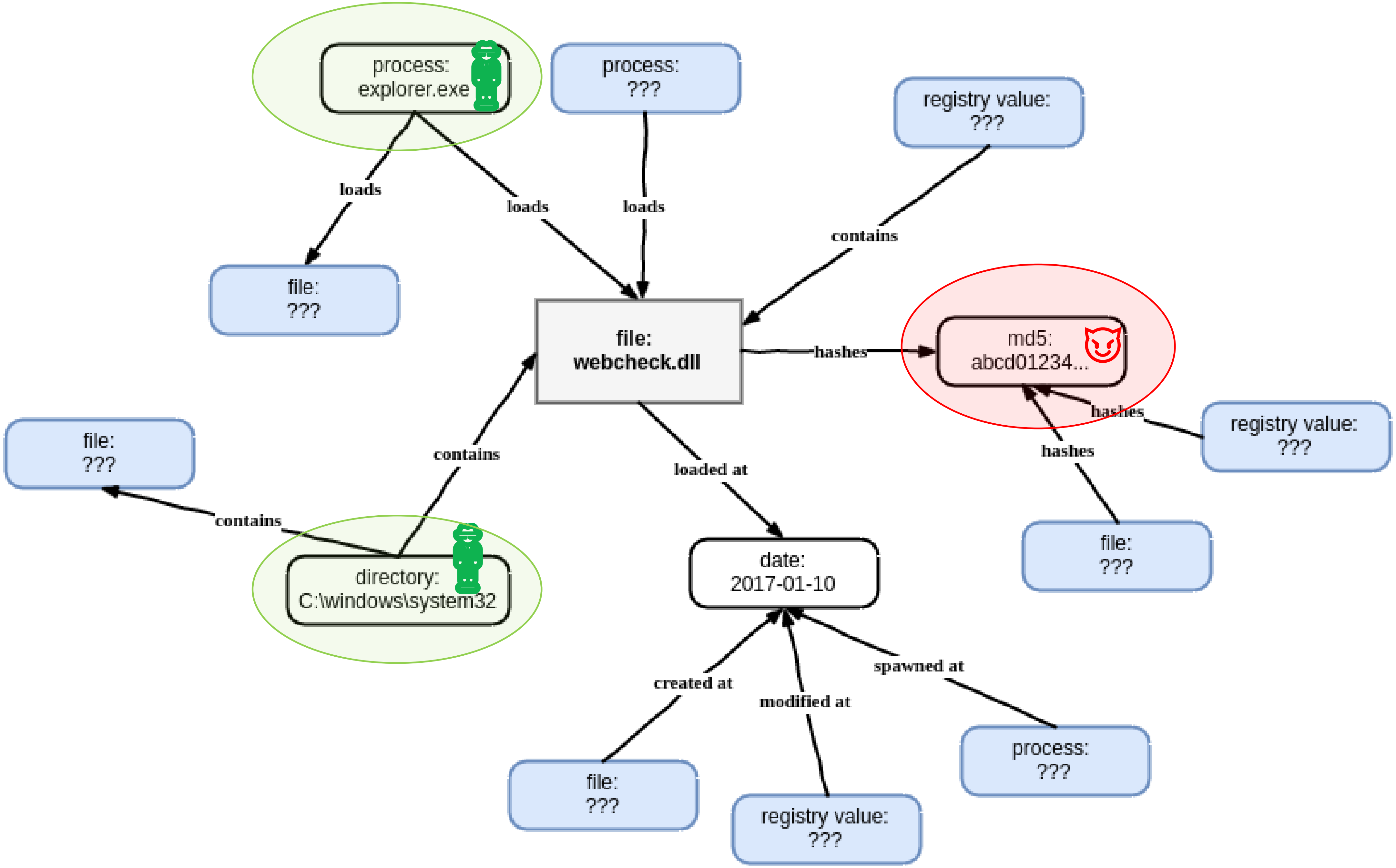
maintain the graph on each host-based agent

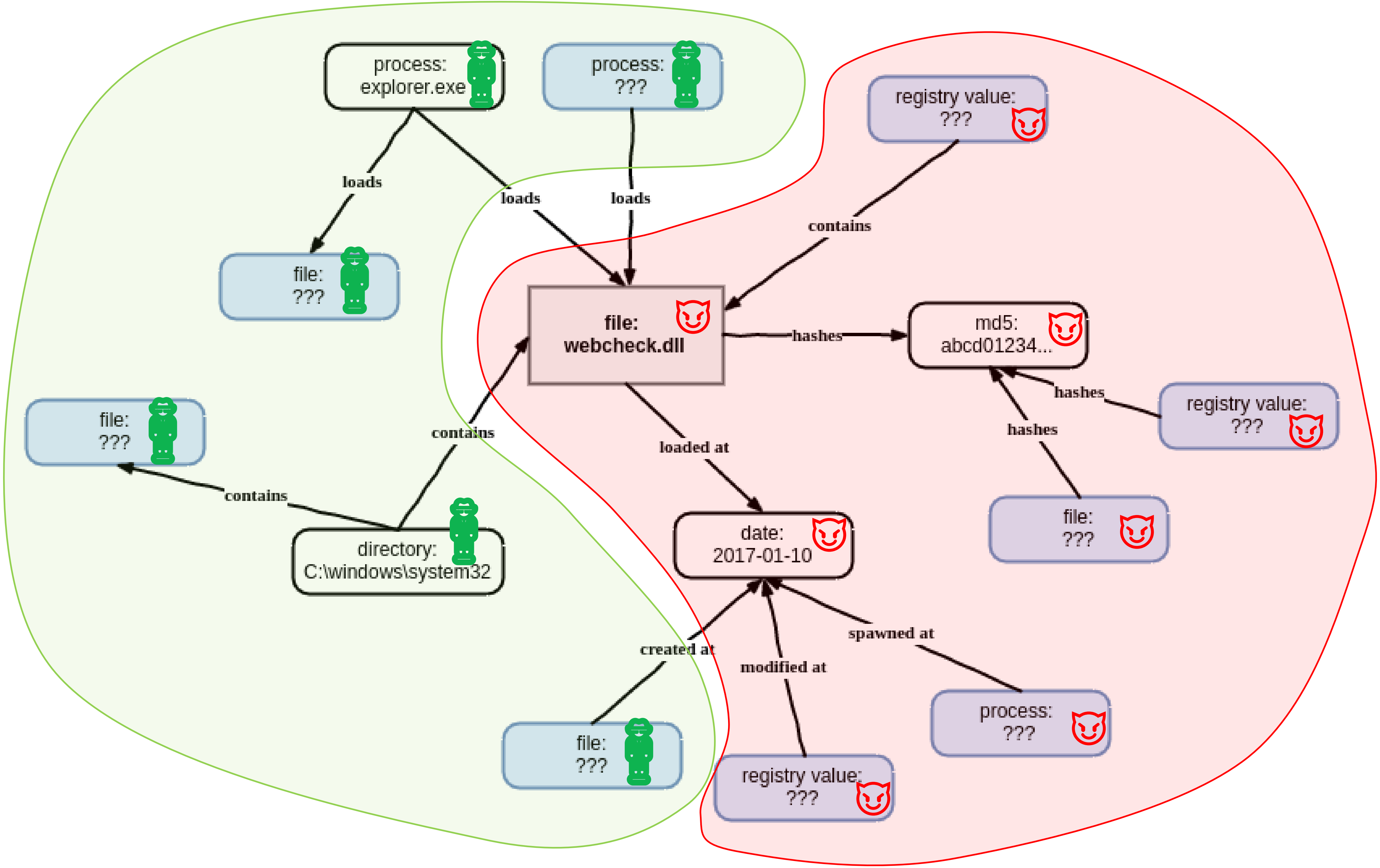
display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

find attacker TTPs as patterns in the graph

partition the graph into relevant sub-graphs and suggest nodes


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
hashes
file:
???
contains
loaded at
file:
???
contains
directory:
C:\windows\system32
date:
2017-01-10
spawned at
created at
modified at
process:
???
file:
???
registry value:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
hashes
file:
???
contains
loaded at
contains
date:
2017-01-10
directory:
C:\windows\system32
created at
modified at
spawned at
process:
???
file:
???
registry value:
???

lessons learned

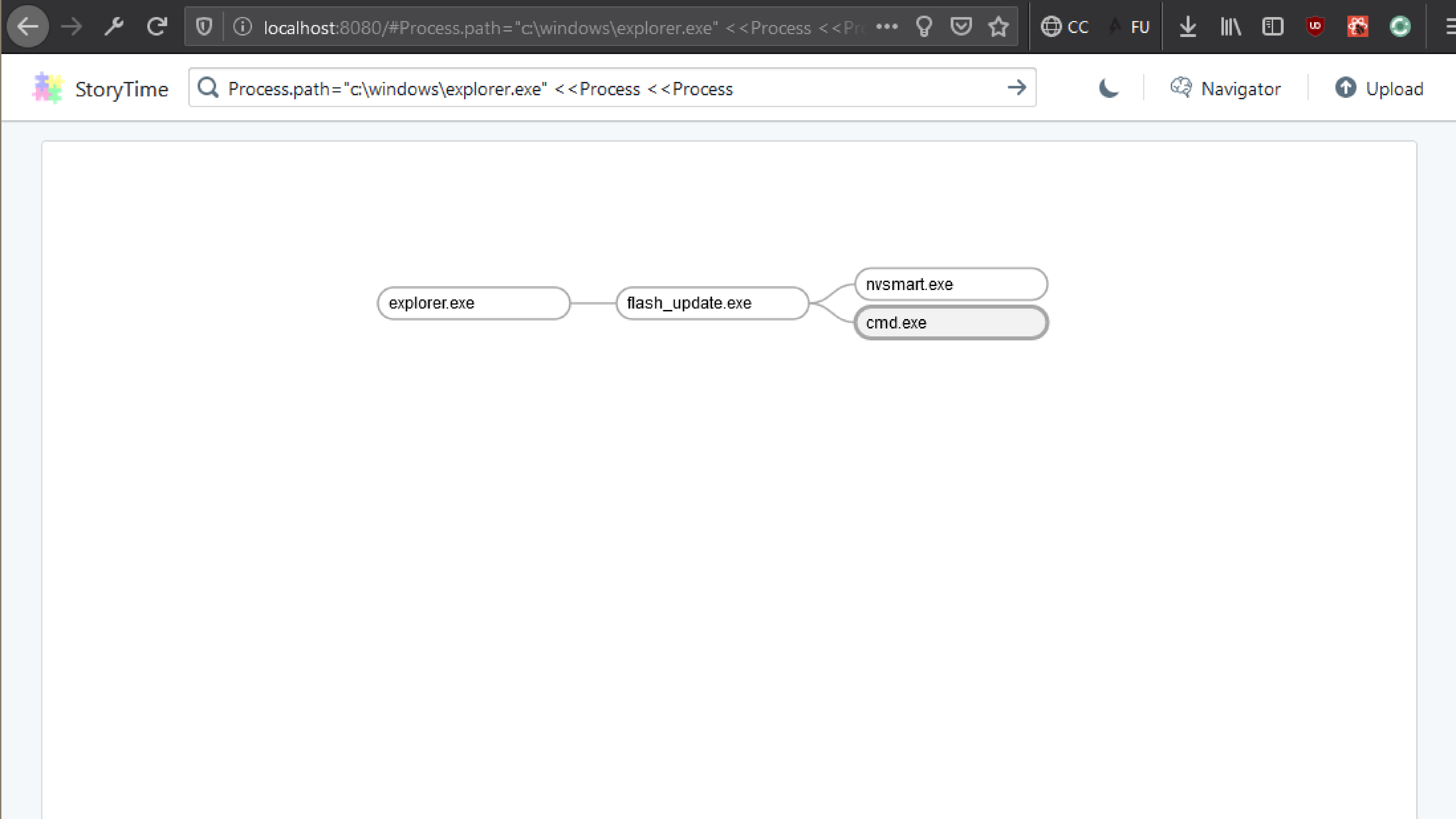
- many advanced analysts still want their grid
    - maybe it’s the data density of a spreadsheet when hunting & consuming data?
    - graph data structure shouldn’t necessarily imply a graph user interface

    - its not a naïve splat of the graph to the screen; tailor graph presentation to guide user
        - in ST, layout order has meaning, and node collapsing implies further context

- its about processing less data, not more

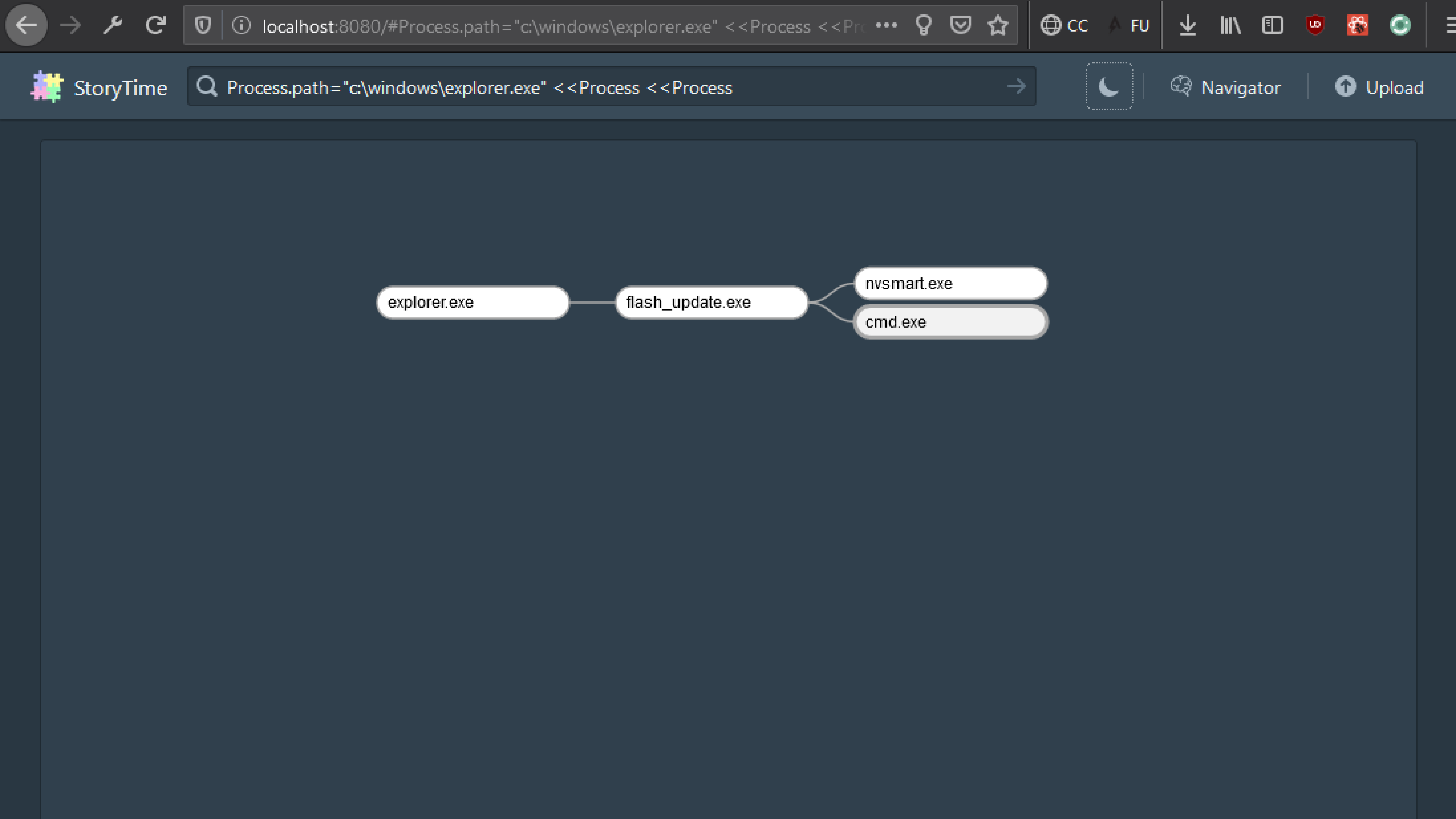
- (like we knew) data model matters: it both limits and enables operations

more detail



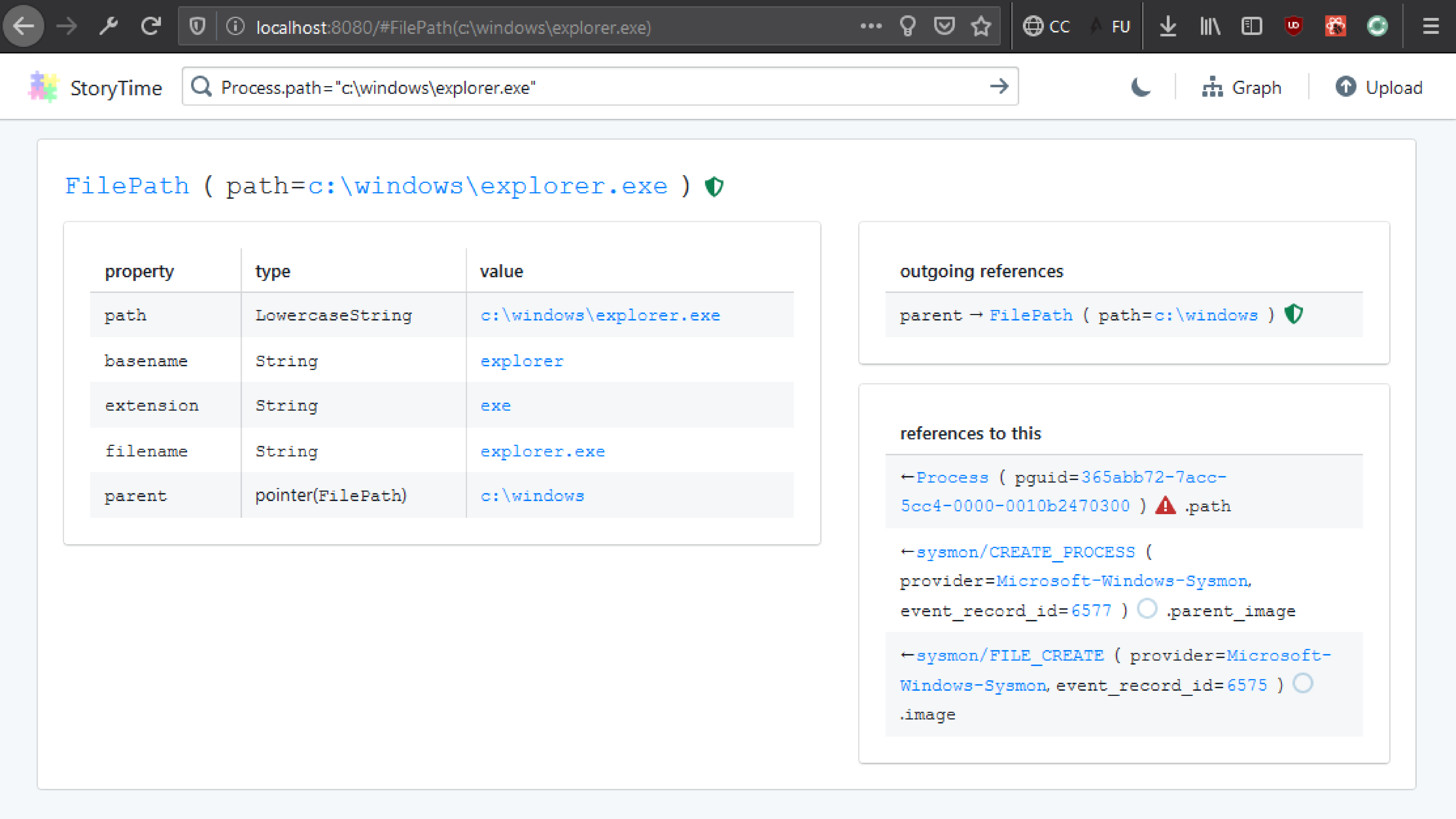
localhost:8080/#Process.path="c:\windows\explorer.exe" <<Process <<Pr

```
Process.path="c:\windows\explorer.exe" <<Process <<Process
```



localhost:8080/#Process.path="c:\windows\explorer.exe" <<Process <<Pr

```
Process.path="c:\windows\explorer.exe" <<Process <<Process
```



localhost:8080/#FilePath(c:\windows\explorer.exe)
CC
FU
StoryTime
Process.path="c:\windows\explorer.exe"
Graph
Upload

FilePath (path=c:\windows\explorer.exe) 🛡

property	type	value
path	LowercaseString	c:\windows\explorer.exe
basename	String	explorer
extension	String	exe
filename	String	explorer.exe
parent	pointer(FilePath)	c:\windows

outgoing references

parent → FilePath (path=c:\windows) 🛡

references to this

←Process (pguid=365abb72-7acc-5cc4-0000-0010b2470300) ⚠ .path

←sysmon/CREATE_PROCESS (provider=Microsoft-Windows-Sysmon, event_record_id=6577) ○ .parent_image

←sysmon/FILE_CREATE (provider=Microsoft-Windows-Sysmon, event_record_id=6575) ○ .image

https://storytime.apps.fireeye.com/index.html

StoryTime

represent artifacts in a graph

represent artifacts in a graph

maintain graph on host-based agents

maintain the graph on each host-based agent

display artifact graph via an intuitive user interface

display the artifact graph via an intuitive user interface

merge host-scoped graphs into global-scoped graph

merge host-scoped graphs into global-scoped graph

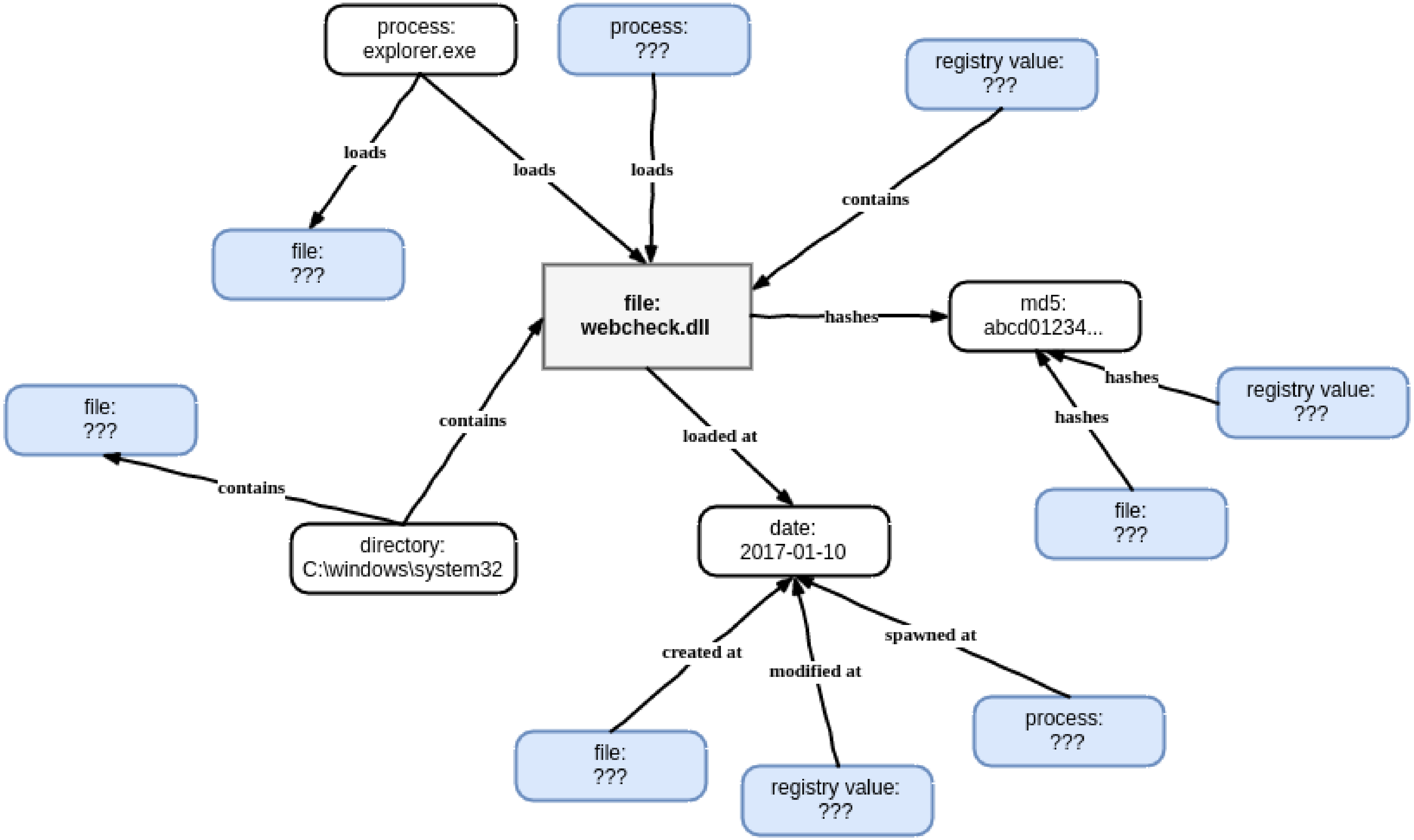
find attacker TTPs as patterns in graph

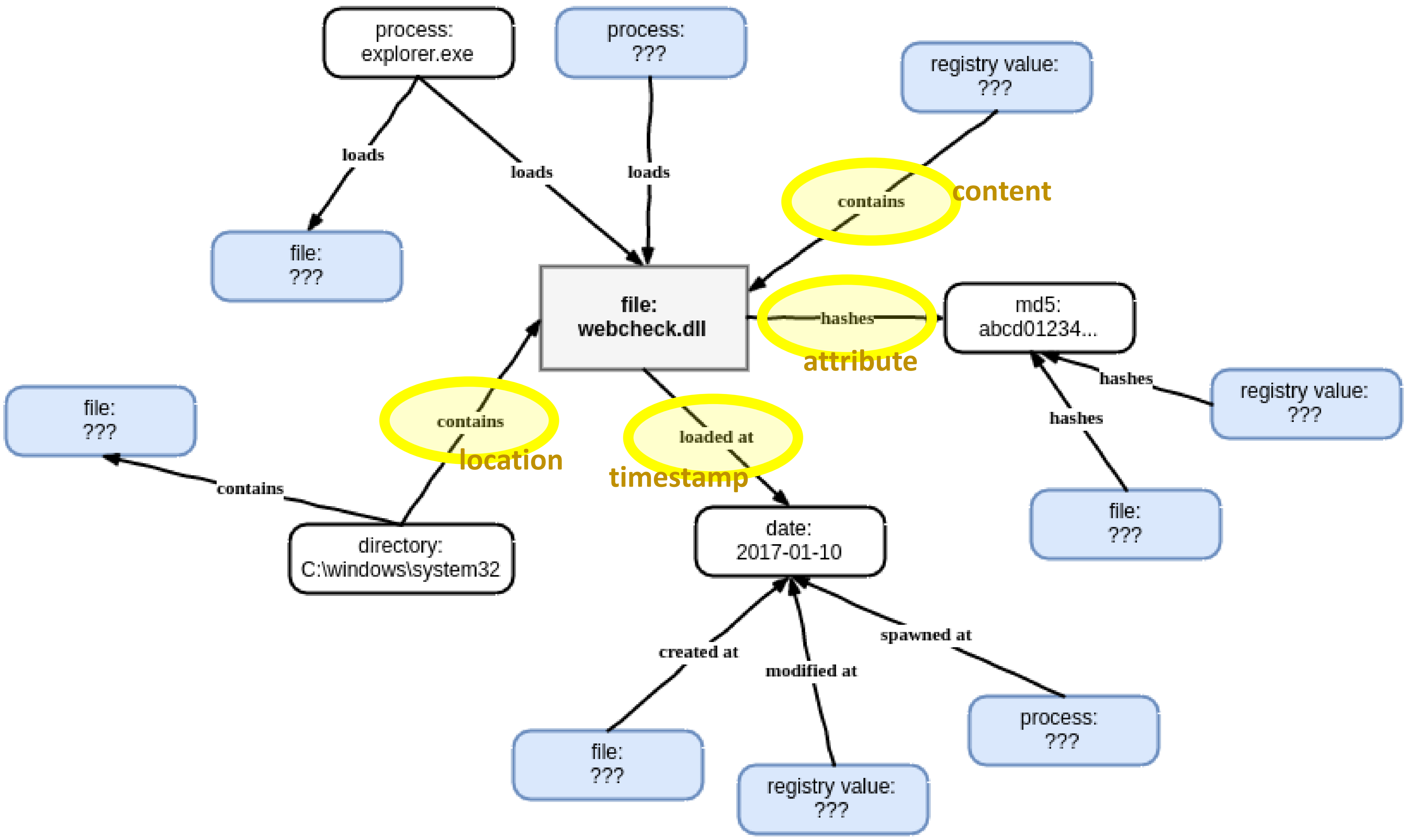
find attacker TTPs as patterns in the graph

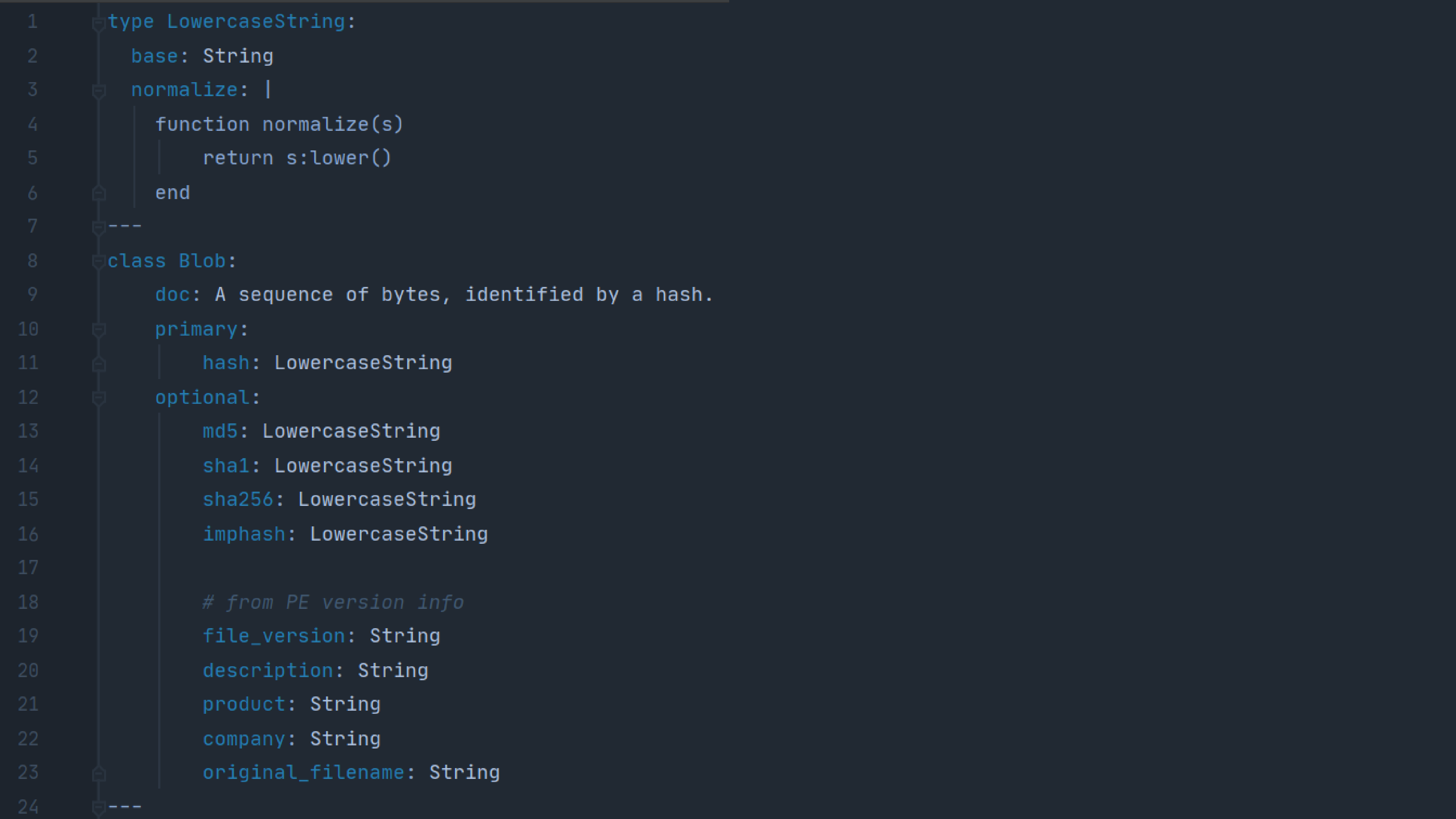
partition graph into sub-graphs and suggest nodes

partition the graph into relevant sub-graphs and suggest nodes

represent artifacts in a graph


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
hashes
file:
???
contains
loaded at
file:
???
contains
directory:
C:\windows\system32
date:
2017-01-10
spawned at
created at
modified at
process:
???
file:
???
registry value:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
content
file:
???
file:
webcheck.dll
hashes
attribute
md5:
abcd01234...
hashes
hashes
registry value:
???
file:
???
contains
location
loaded at
timestamp
contains
file:
???
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???



```
type LowercaseString:
  base: String
  normalize: |
    function normalize(s)
        return s:lower()
    end
---
class Blob:
    doc: A sequence of bytes, identified by a hash.
    primary:
        hash: LowercaseString
    optional:
        md5: LowercaseString
        sha1: LowercaseString
        sha256: LowercaseString
        imphash: LowercaseString

        # from PE version info
        file_version: String
        description: String
        product: String
        company: String
        original_filename: String
---
```

Entity:

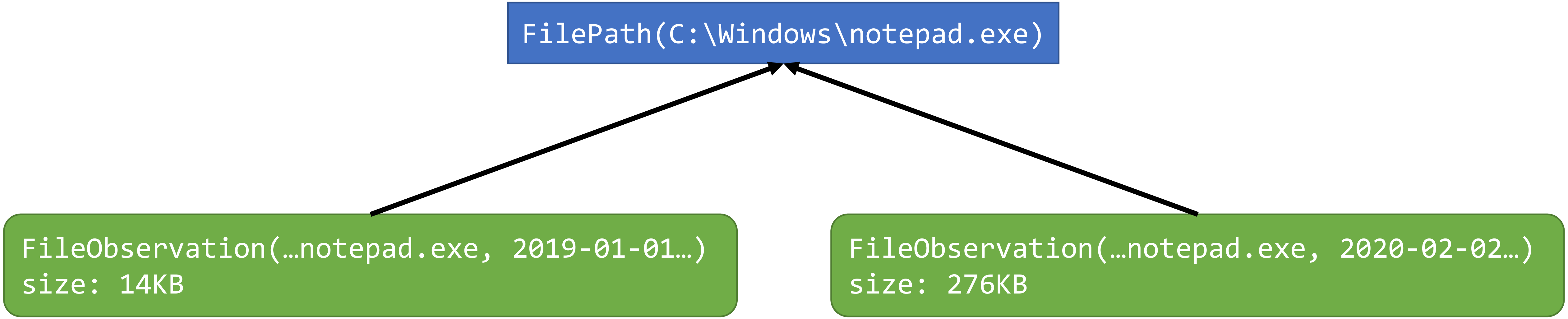
a unique immutable, namable thing/object/term/artifact.

typically quite simple, like a file system path.
instances of it may exist on multiple systems.

Observation:

metadata collected at a point in time.

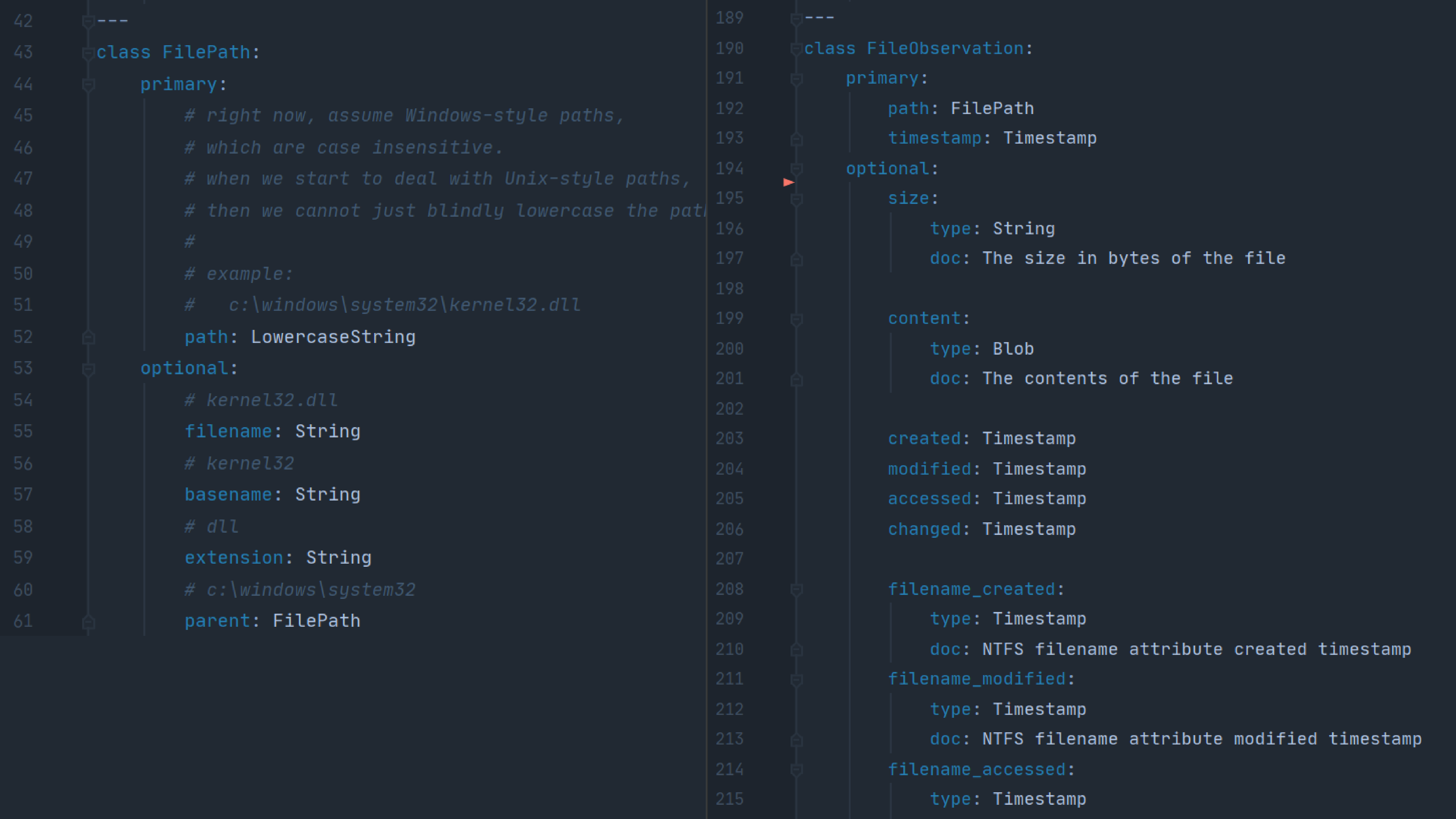
usually has more properties.
often links many entities together.



FilePath(C:\Windows\notepad.exe)

FileObservation(…notepad.exe, 2019-01-01…)
size: 14KB

FileObservation(…notepad.exe, 2020-02-02…)
size: 276KB



```
---
class FilePath:
    primary:
        # right now, assume Windows-style paths,
        # which are case insensitive.
        # when we start to deal with Unix-style paths,
        # then we cannot just blindly lowercase the pat
        #
        # example:
        #   c:\windows\system32\kernel32.dll
        path: LowercaseString
    optional:
        # kernel32.dll
        filename: String
        # kernel32
        basename: String
        # dll
        extension: String
        # c:\windows\system32
        parent: FilePath
```

```
---
class FileObservation:
    primary:
        path: FilePath
        timestamp: Timestamp
    optional:
        size:
            type: String
            doc: The size in bytes of the file

        content:
            type: Blob
            doc: The contents of the file

        created: Timestamp
        modified: Timestamp
        accessed: Timestamp
        changed: Timestamp

        filename_created:
            type: Timestamp
            doc: NTFS filename attribute created timestamp
        filename_modified:
            type: Timestamp
            doc: NTFS filename attribute modified timestamp
        filename_accessed:
            type: Timestamp
```

entities and observations leads to a graph that is bipartite-ish

intel, like “is it malware”, propagates to entities.
this makes sense, because entities are usually global concepts.

but this makes fetching metadata about a thing more complex

- e.g. “As of yesterday, the hash of `C:\windows\notepad.exe` was `XXX`”
- maybe this forces us to be more correct

to merge graphs:

- entities coalesce together
- observations remain unconnected

outstanding issues

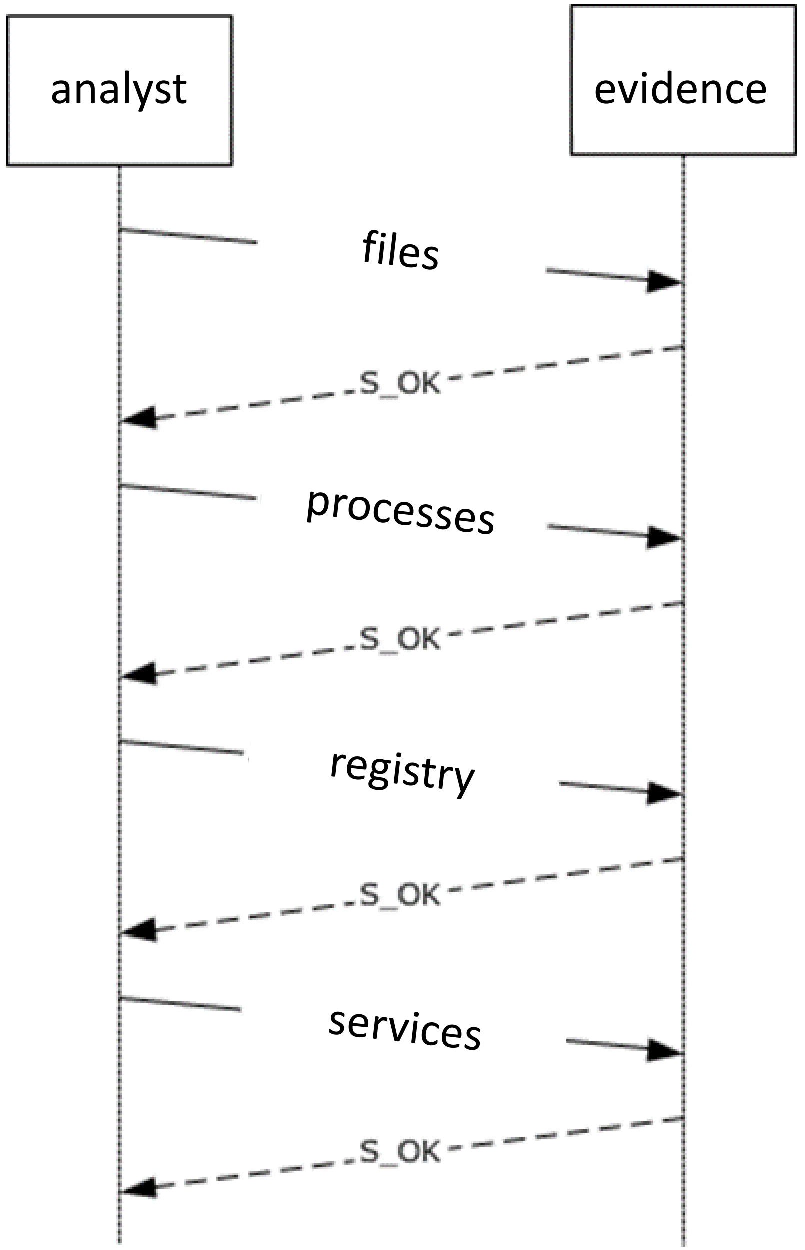
how to represent things with unclear/not-agreed-upon identity?

- e.g. processes (OS recycles PIDs, sysmon has its own GUIDs, etc.)
  - We've seen PIDs reused within a single second on windows systems making time+PID inaccurate when time is seconds granularity

how to find the right level of abstraction?

- want: a level that encourages reasoning
- but: schema dictates (restricts) how data can be accessed

maintain graph on host-based agents



problem:

in typical investigations, there is repeated fetch of artifacts via high-latency process.

“given this alert for `foo.exe`, fetch the file”

“then list processes and find `foo.exe`”

“then see what files `foo.exe` wrote to”

“then collect those”

“then see if any are configured for persistence”

each step might take many minutes to complete

☹

solution:

maintain artifact graph on each endpoint.

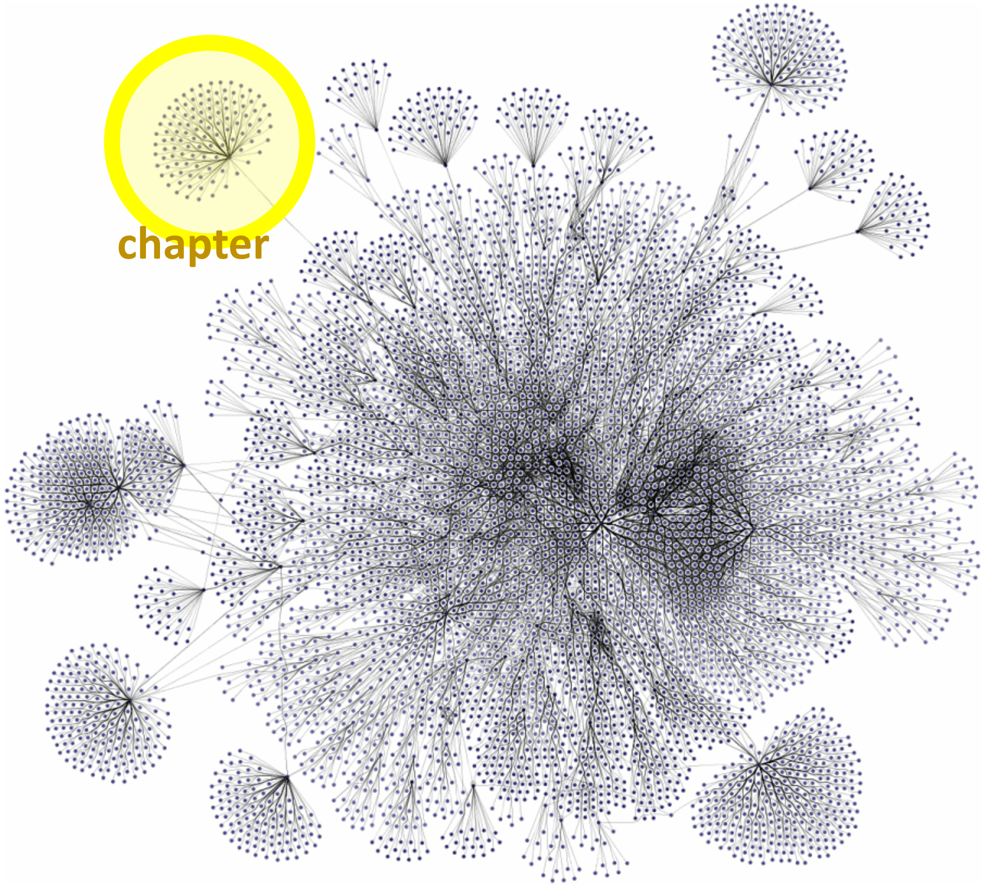
when there is an alert,

  locate associated node in graph,

  collect the subgraph of neighboring nodes,

  return it in *one* roundtrip (or less).

→ system guesses what the analyst will need



this supported real investigations

data sources:

- endpoint agent events, e.g. file writes, process exec, net connection
- play at home: sysmon

nuances:

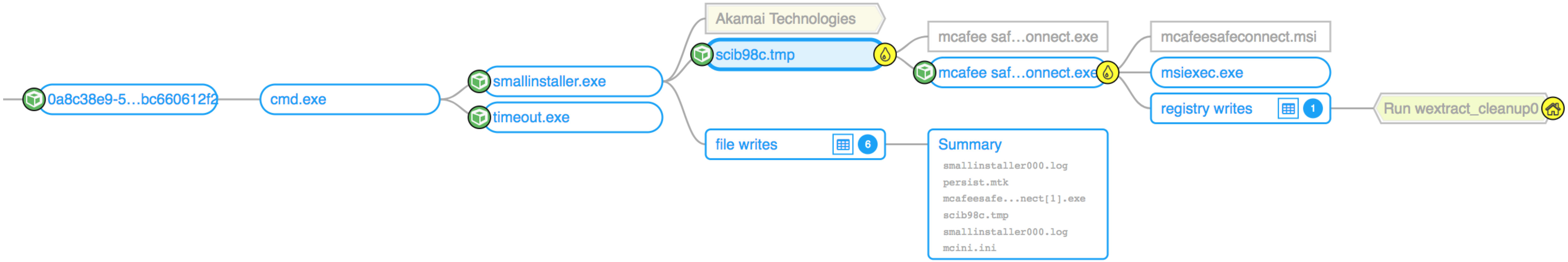
- how big of a graph do you maintain? which nodes to prune?
    - Current system uses type-based aging (process nodes last longer than file or registry nodes...and so on keeping more valuable artifacts for longer)

let’s say you see lateral movement…

- tired: query multiple hosts and stitch a central graph together

- *wired*: host to host graph traversal
  - federate the “global” graph among many endpoints
  - let them query each other, peer-to-peer

display artifact graph via an intuitive user interface



Summary

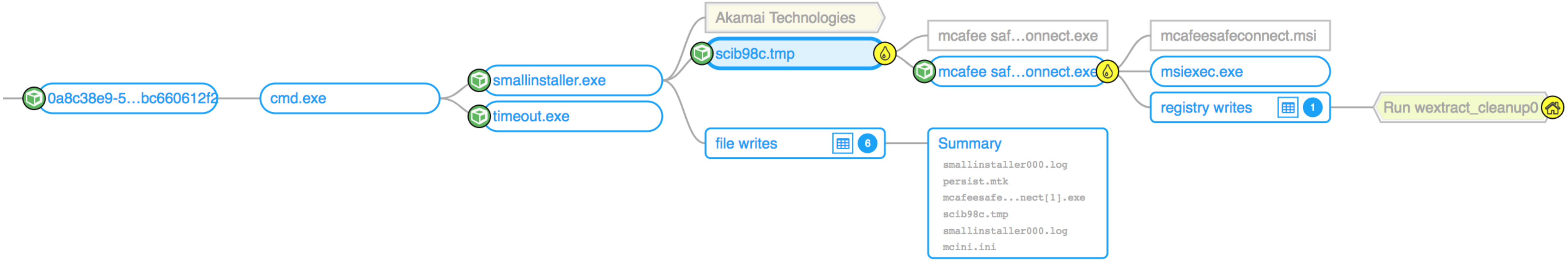
```
smallinstaller000.log
persist.mtk
mcafeesafe...nect[1].exe
scib98c.tmp
smallinstaller000.log
mcini.ini
```

graph relationship visualization

Alert is shown prominently with a shaded blue background -- a process event

Chapter contains context for how this suspect process came about

- lineage: what happened **before** alert
- along with: what happened **after** the alert.



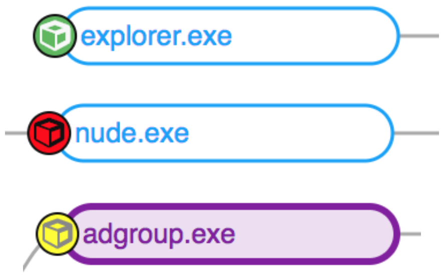
```
smallinstaller000.log
persist.mtk
mcafeesafe...nect[1].exe
scib98c.tmp
smallinstaller000.log
mcini.ini
```

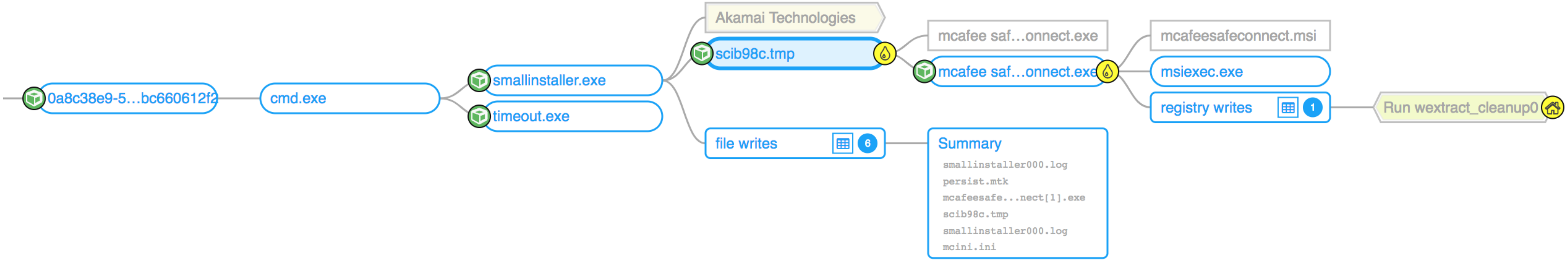
disposition context

Known good

Known bad

Unseen

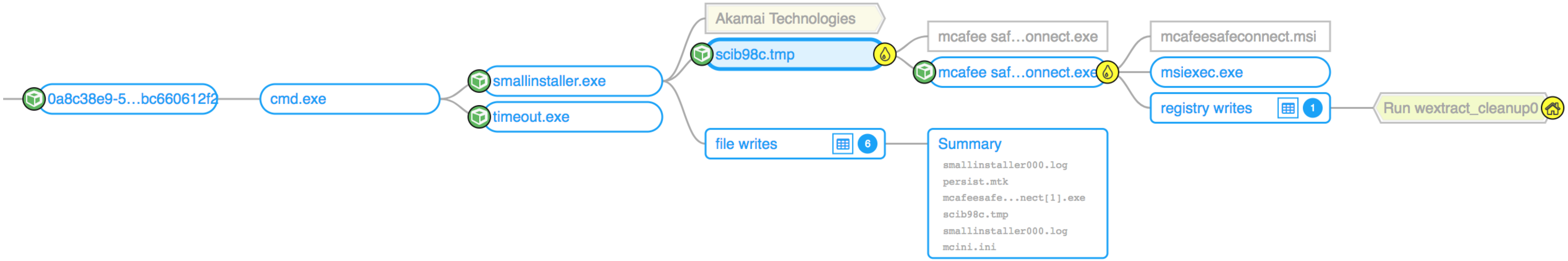




location context

Dropped and Executed

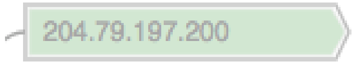
Wrote to persistence Location



network context

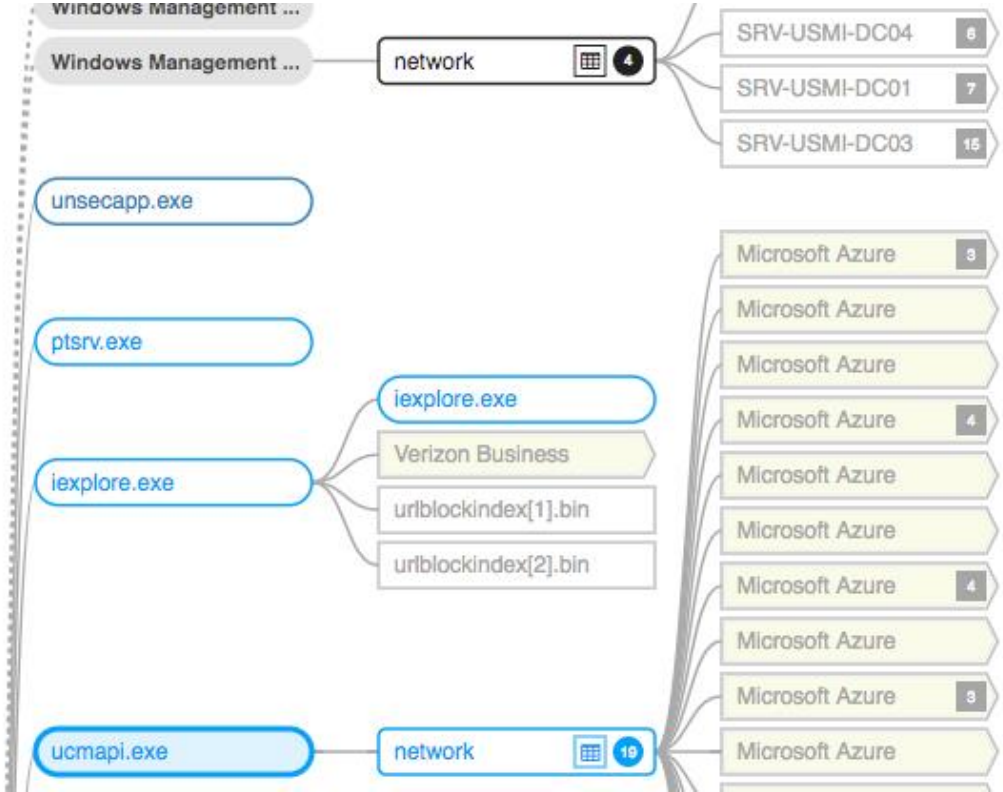
Resolve IP to organization

Context show network is benign



Internal IPs converted to hostnames

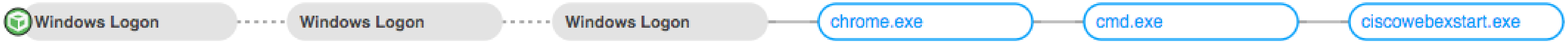
Context to show bad network connection is RED

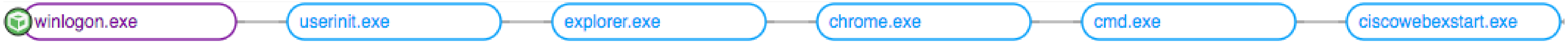


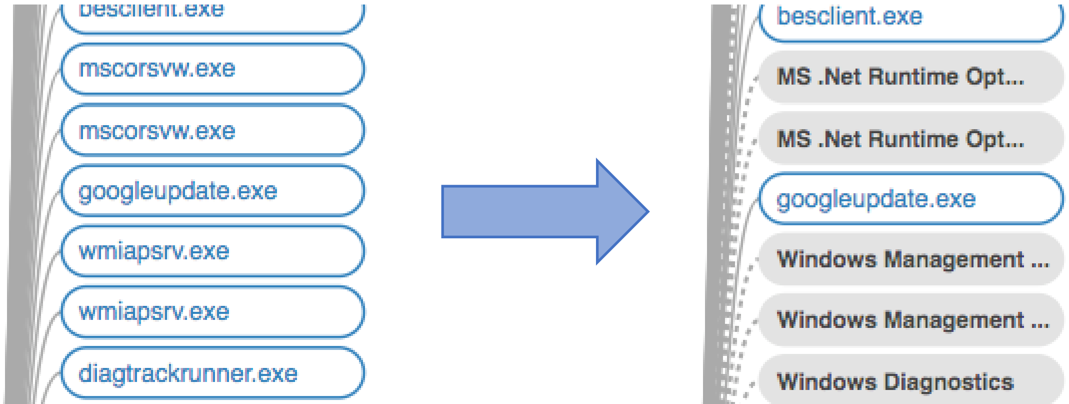
Windows Internals context

Fading nodes into background that are “known”

- Help junior analysts learn common patterns without many years experience
- Filter out unnecessary analysis

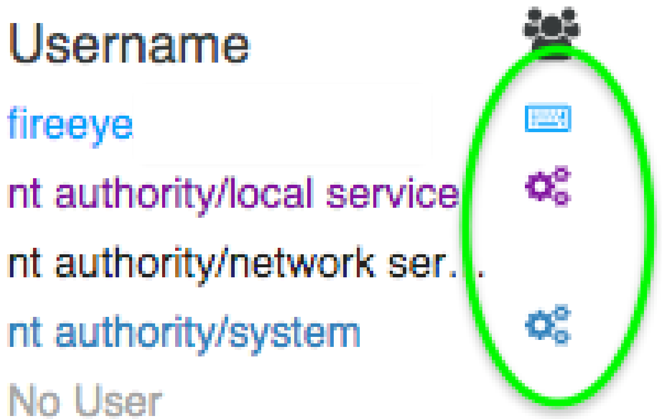






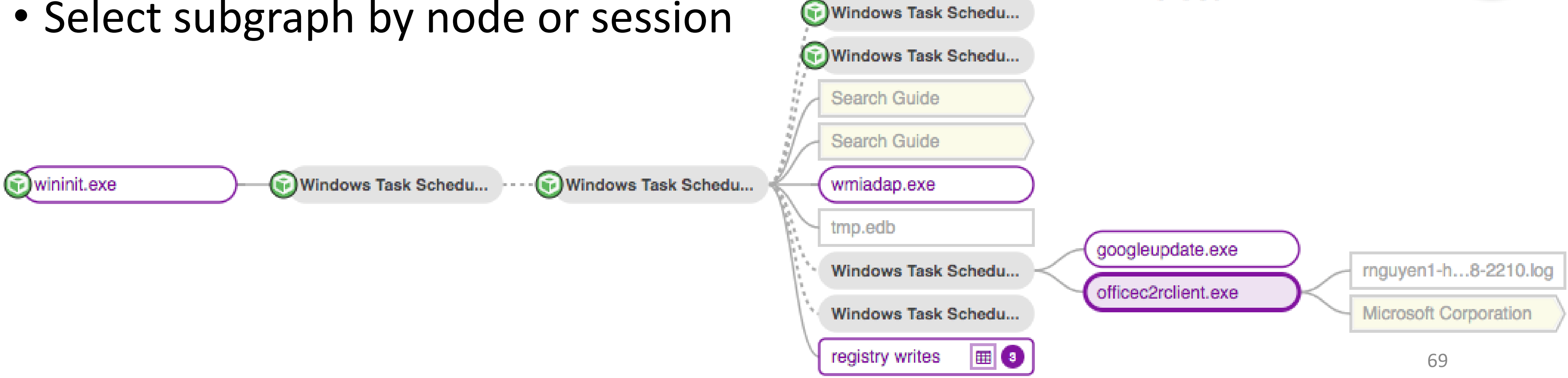
user context

- Color nodes based on user
- Identify session types created by user
  - Interactive (local to machine) ⌨
  - Remote Interactive (remote with UI)
  - Service ⚙
- Select subgraph by node or session



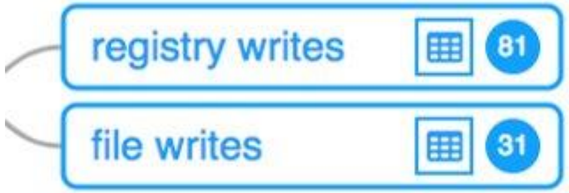
Username

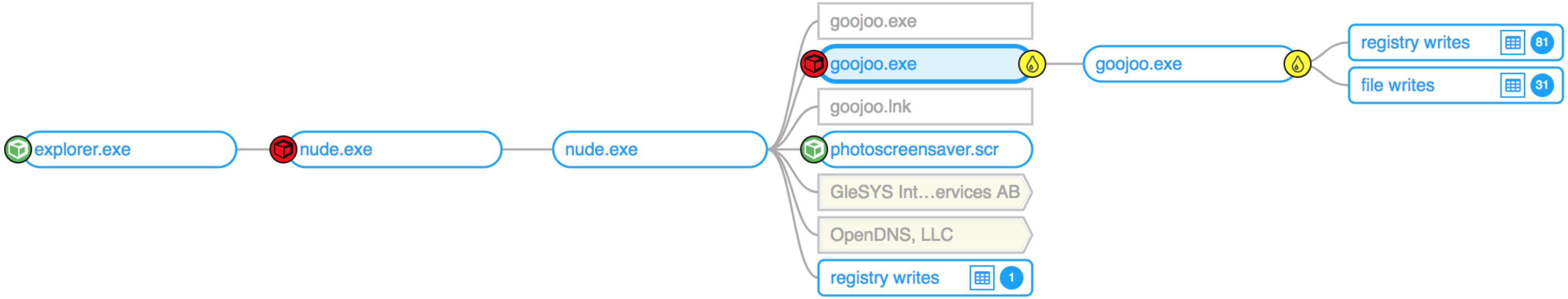
- fireeye ⌨
- nt authority/local service ⚙
- nt authority/network ser...
- nt authority/system ⚙
- No User

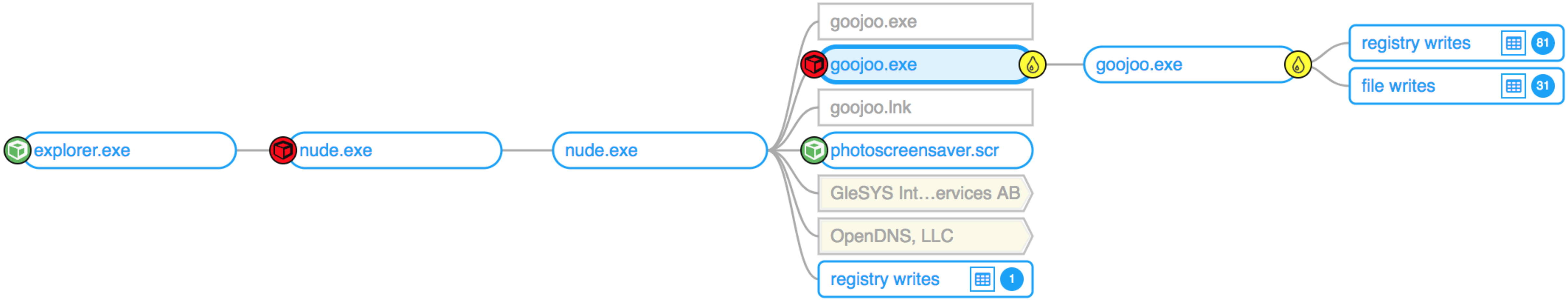


Collapsing Clutter

- Reduce displaying 100s of nodes in chapter down to most important
- Present user with most important information first, to make decision
  - Let use decide when to dig in further
  - Present summaries with limited information
- Can present data in grid view if desired







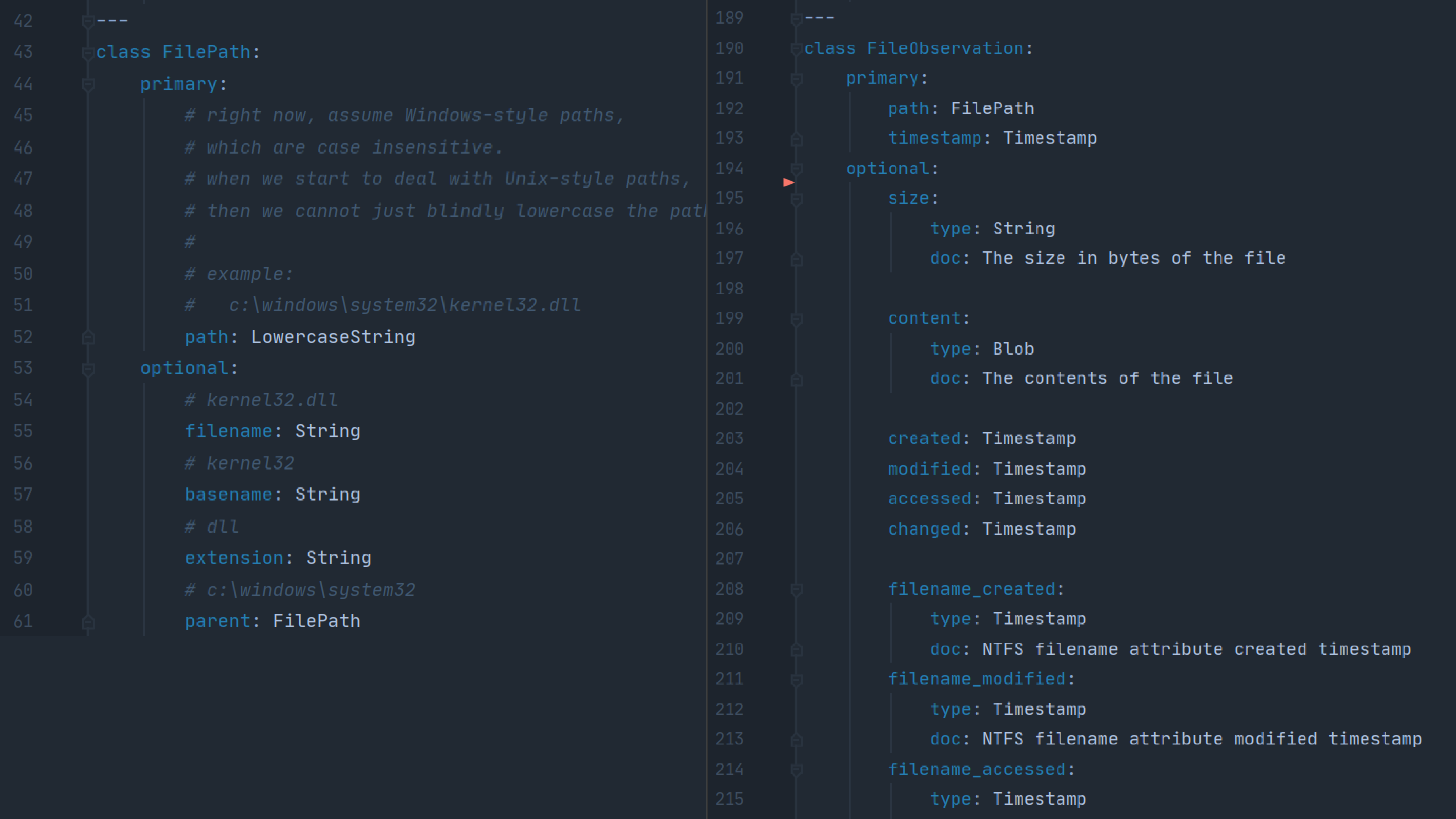
- Two MalwareGaurd detections in single chapter
- Able to see execution started from `explorer.exe`, aka user double-clicked
- Able to see executable was dropped and then executed
- Easy to see that screensaver is part of package
- Simple access to network, registry and file writes associated with chapter

merge host-scoped graphs into global-scoped graph

graph is designed to merge well

- entities coalesce together, across host, investigation, organization
  - every node has a URI derived from its primary properties
    - enables many other things: caching, performance, etc.
  - `FilePath(C:\windows\notepad.exe)` is a global concept

- observations don’t collide
  - Primary properties include key + timestamp (+ maybe host)
  - `FileObservation(C:\windows\notepad.exe, 2020-01-01…, dc-hostname)`



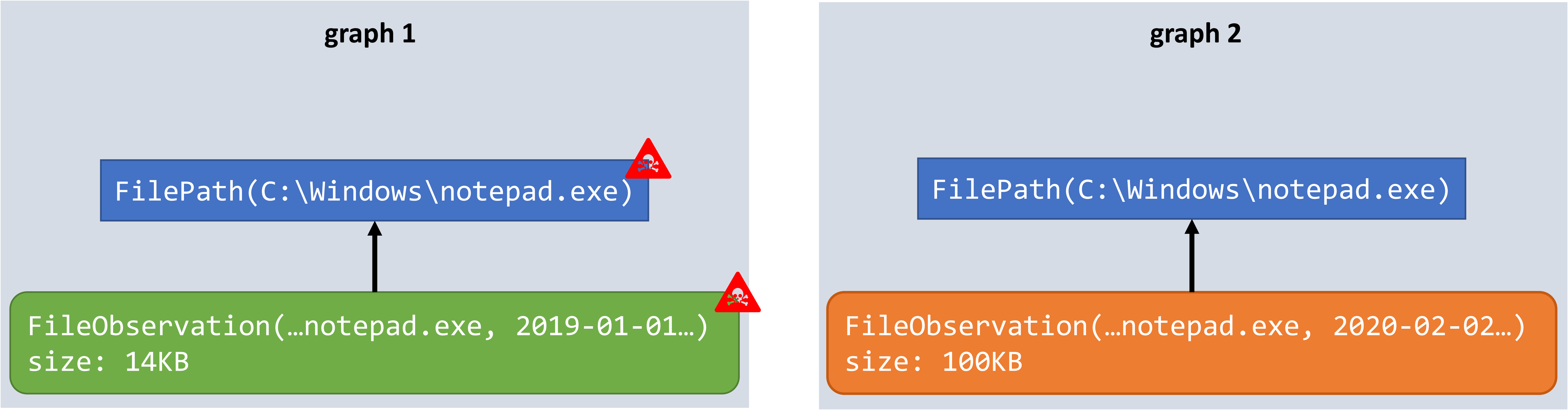
```
---
class FilePath:
    primary:
        # right now, assume Windows-style paths,
        # which are case insensitive.
        # when we start to deal with Unix-style paths,
        # then we cannot just blindly lowercase the pat
        #
        # example:
        #   c:\windows\system32\kernel32.dll
        path: LowercaseString
    optional:
        # kernel32.dll
        filename: String
        # kernel32
        basename: String
        # dll
        extension: String
        # c:\windows\system32
        parent: FilePath
```

```
---
class FileObservation:
    primary:
        path: FilePath
        timestamp: Timestamp
    optional:
        size:
            type: String
            doc: The size in bytes of the file

        content:
            type: Blob
            doc: The contents of the file

        created: Timestamp
        modified: Timestamp
        accessed: Timestamp
        changed: Timestamp

        filename_created:
            type: Timestamp
            doc: NTFS filename attribute created timestamp
        filename_modified:
            type: Timestamp
            doc: NTFS filename attribute modified timestamp
        filename_accessed:
            type: Timestamp
```



graph 1

FilePath(C:\Windows\notepad.exe)

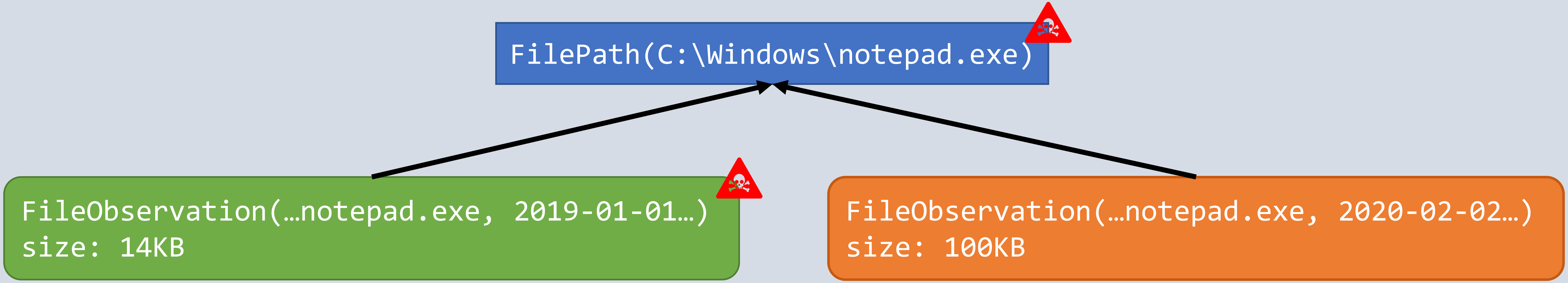
FileObservation(…notepad.exe, 2019-01-01…)
size: 14KB

graph 2

FilePath(C:\Windows\notepad.exe)

FileObservation(…notepad.exe, 2020-02-02…)
size: 100KB

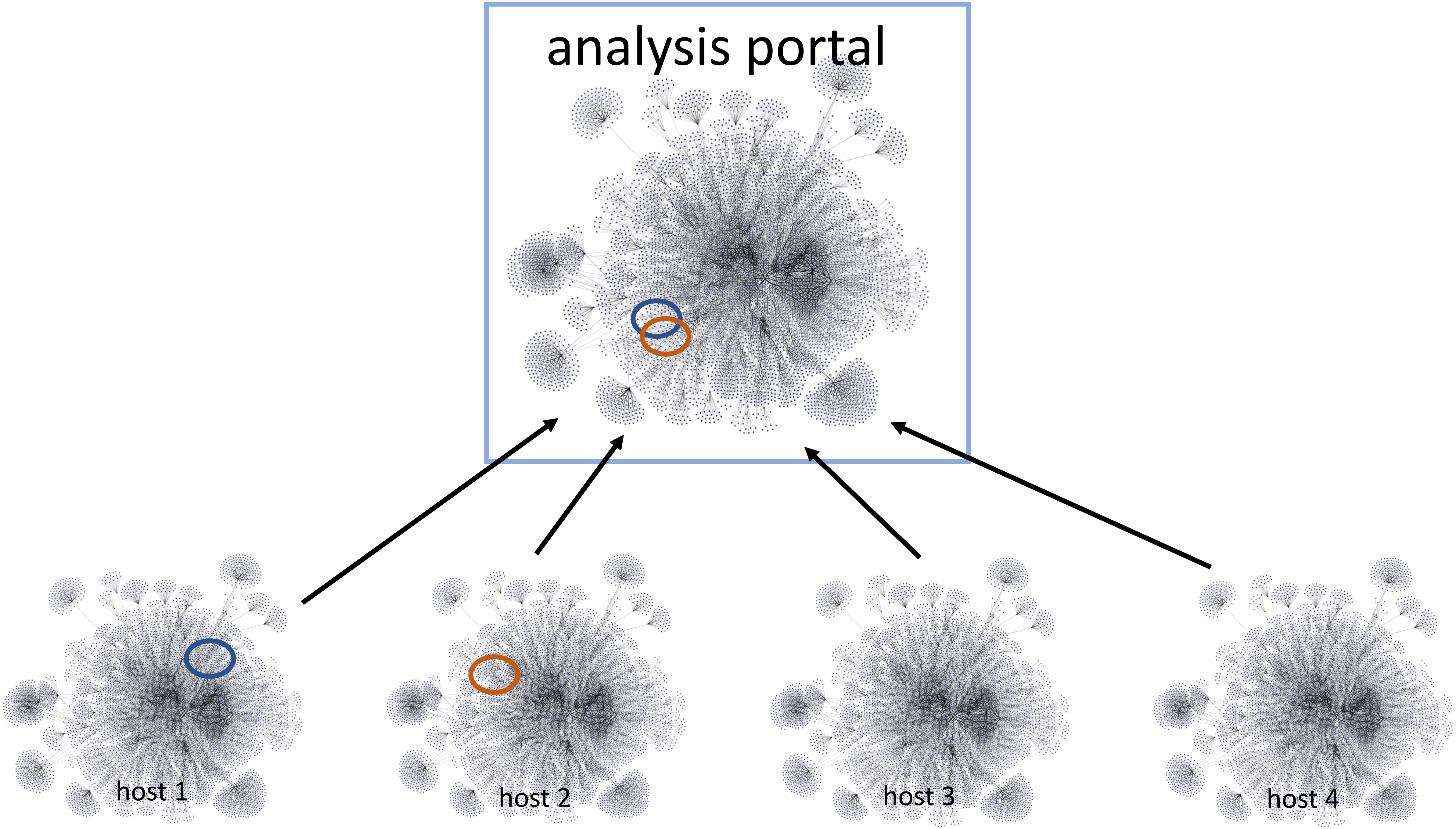
merged graphs



FilePath(C:\Windows\notepad.exe)

FileObservation(…notepad.exe, 2019-01-01…)
size: 14KB

FileObservation(…notepad.exe, 2020-02-02…)
size: 100KB



analysis portal

host 1

host 2

host 3

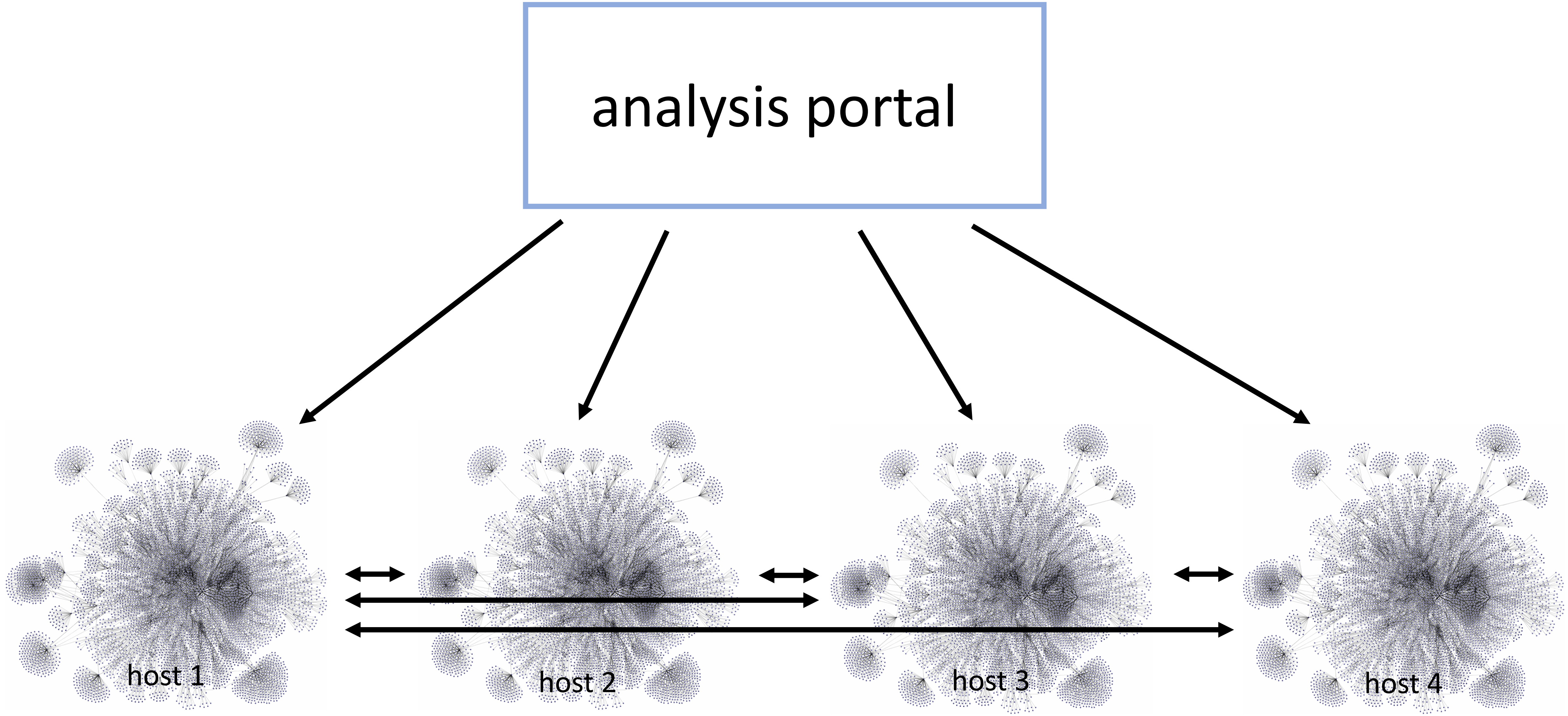
host 4

let’s say you see lateral movement…

- tired: query multiple hosts and stitch a central graph together

- *wired*: host to host graph traversal
  - federate the “global” graph among many endpoints
  - let them query each other, peer-to-peer

→ each endpoint becomes an autonomous agent that investigates the rest of the enterprise



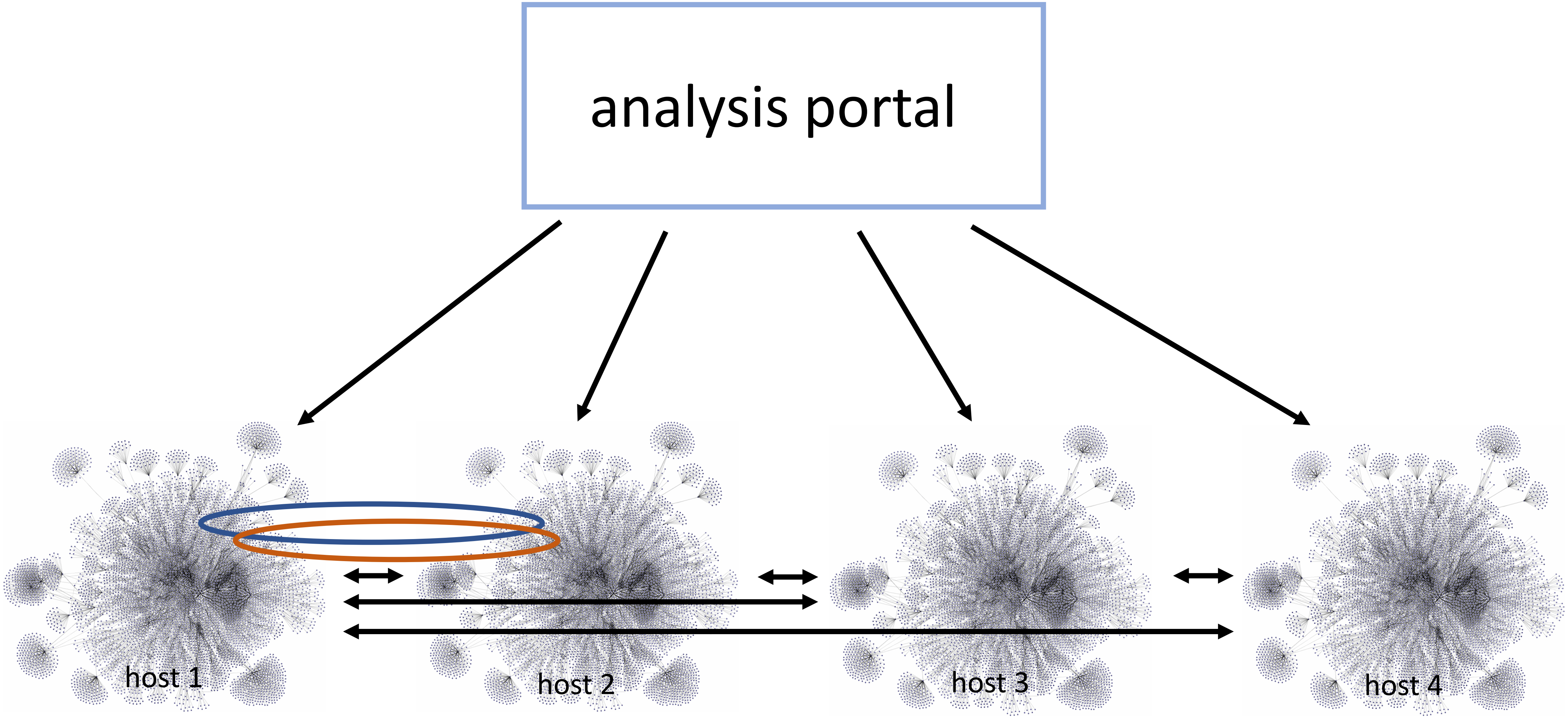
analysis portal

host 1

host 2

host 3

host 4



analysis portal

host 1

host 2

host 3

host 4

find attacker TTPs as patterns in graph

what is a TTP?

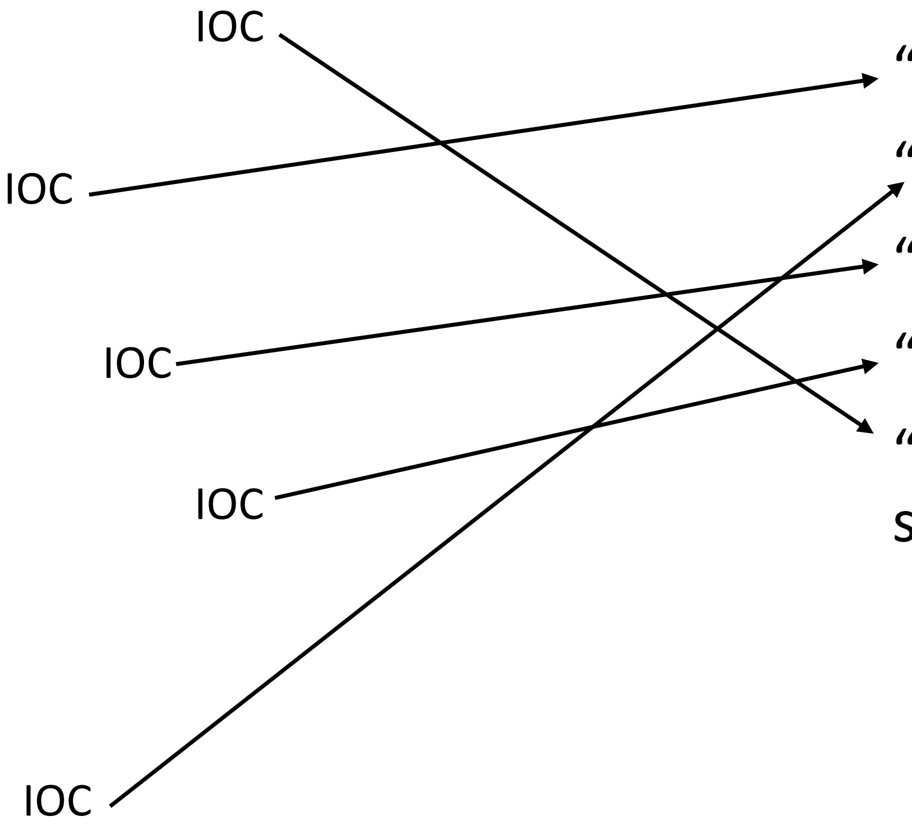
threats follow a certain sequence of events during the attack life-cycle – the attacker fingerprint.

hypothesis:

detect the graph sub-structure created by a TTP rather the individual TTP events

problem:

Example TTPs we create IOCs for today:



IOC

IOC

IOC

IOC

IOC

“attacker send attachment in email with exploit”

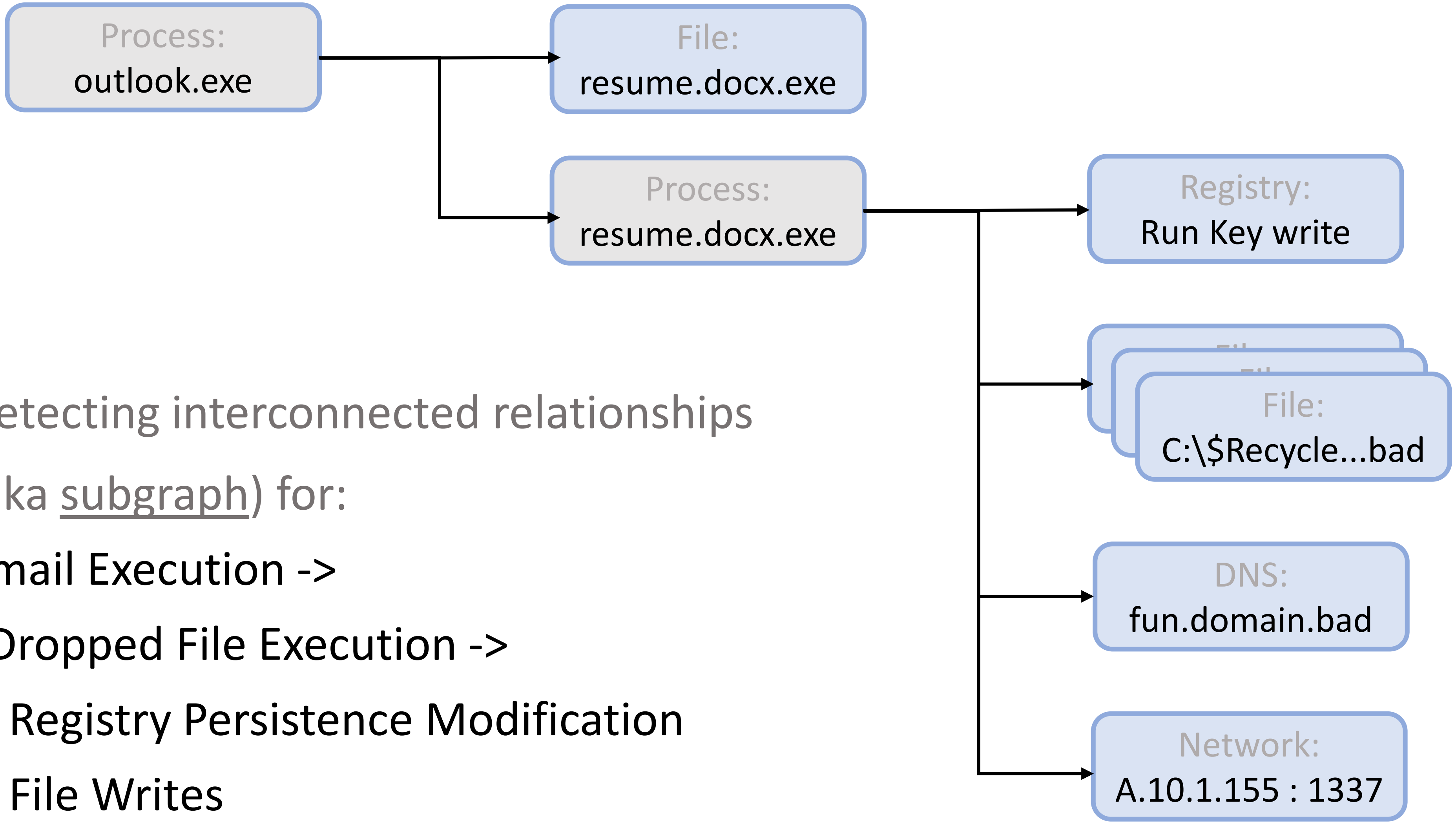
“malware executes”

“creates persistence in the registry Run Key”

“stages exfil files in Recycle Bin”

“exfils data using encrypted C2 over non-standard ports”

solution:



Detecting interconnected relationships

(aka subgraph) for:

Email Execution ->

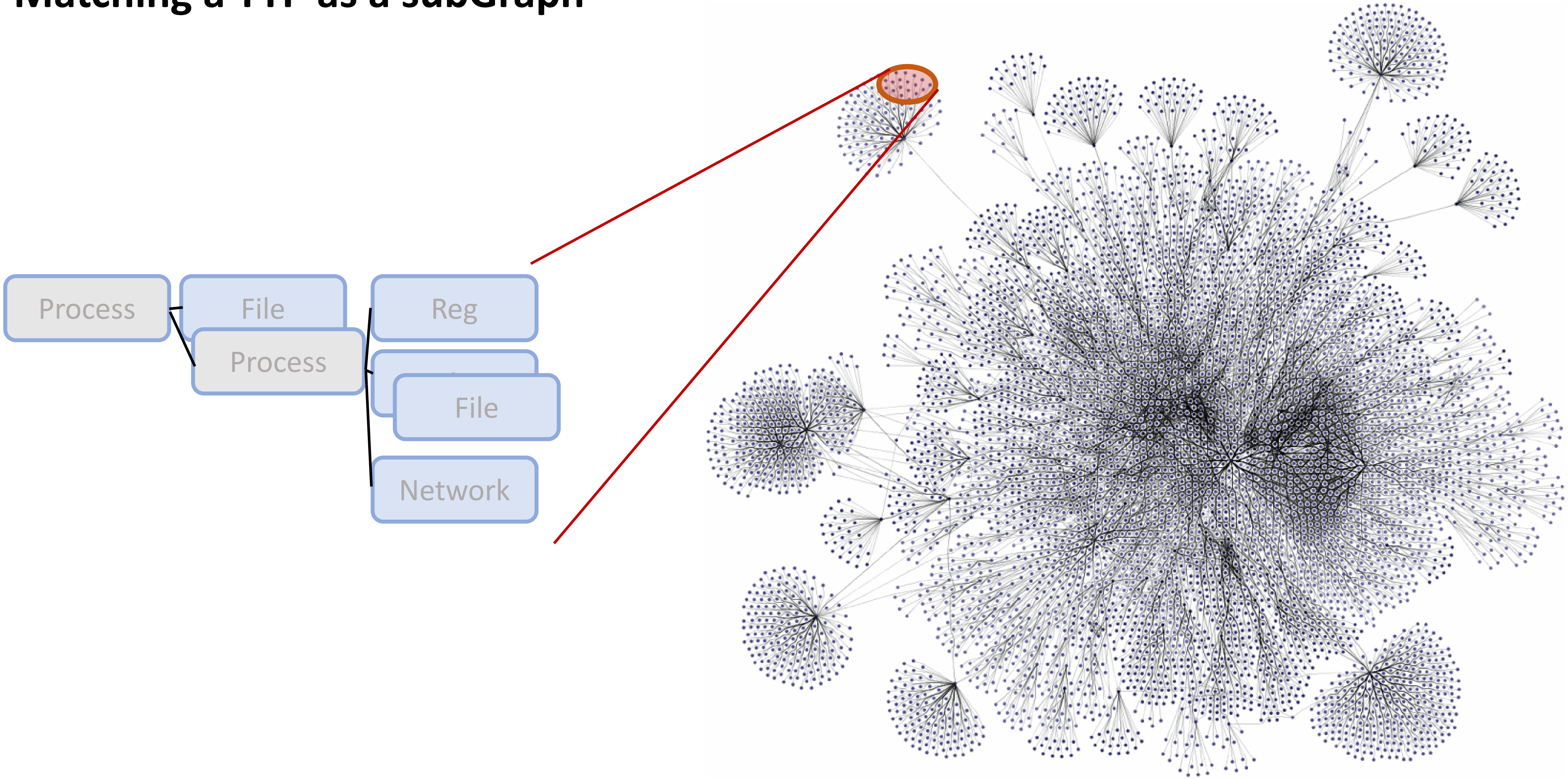
Dropped File Execution ->

Registry Persistence Modification

File Writes

Network Connection

Matching a TTP as a subGraph



Process

File

Process

Reg

File

Network

Converting TTPs to subgraphs for matching!

data sources:

- Intel and IOCs

nuances:

- how big can subgraph be but still generically detect and/or locate new unseen malicious activity?

achilles heel:

- how many FPs???

partition graph into sub-graphs and suggest nodes hypothesis:

related artifacts form a connected subgraph of the entire artifact graph

intuition

- related things happen around the same *time* (temporal locality)
  - generalized: similar values when the type is continuous

    eg. timestamps, file size, entropy
- related things happen around the same *place* (spatial locality)
  - generalized: equal values when the type is discrete

    eg. current directory, user account, md5 hash

- if event *A* is related to event *B*, and event *B* is related to event *C*, then event *A* is related to event *C* (transitive property)

so what?

- the artifact classification phase can be done by graph partitioning
    - goal: find boundaries between the “relevant” and “not relevant” subgraphs

- here’s an effective technique:
    1. start with a known-relevant artifact, and
    2. recursively explore its neighbors,
    3. until only non-relevant artifacts found.

    - this is analogous to what a human does: they follow the thread

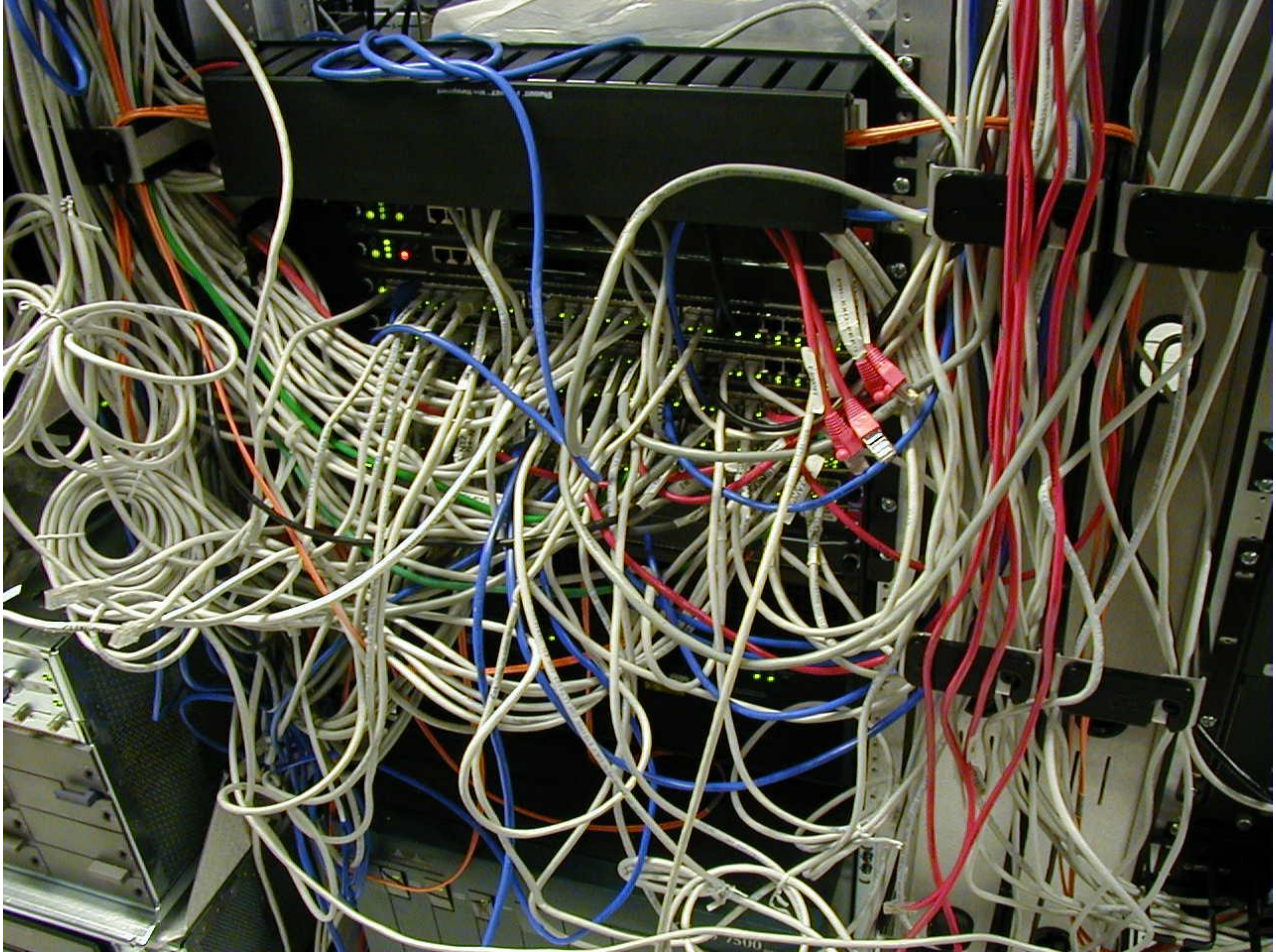
Threat Score Propagation

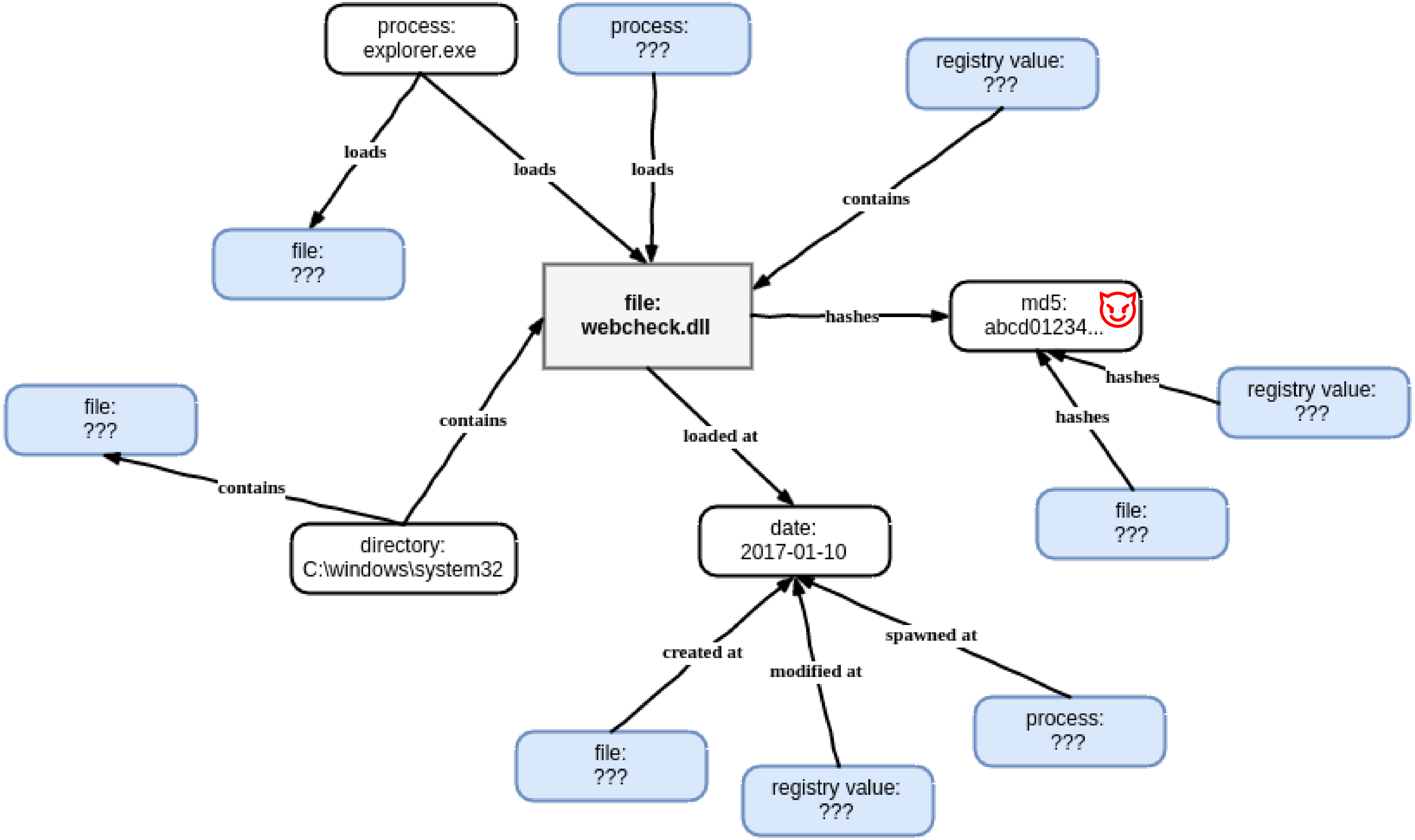
What we implemented first

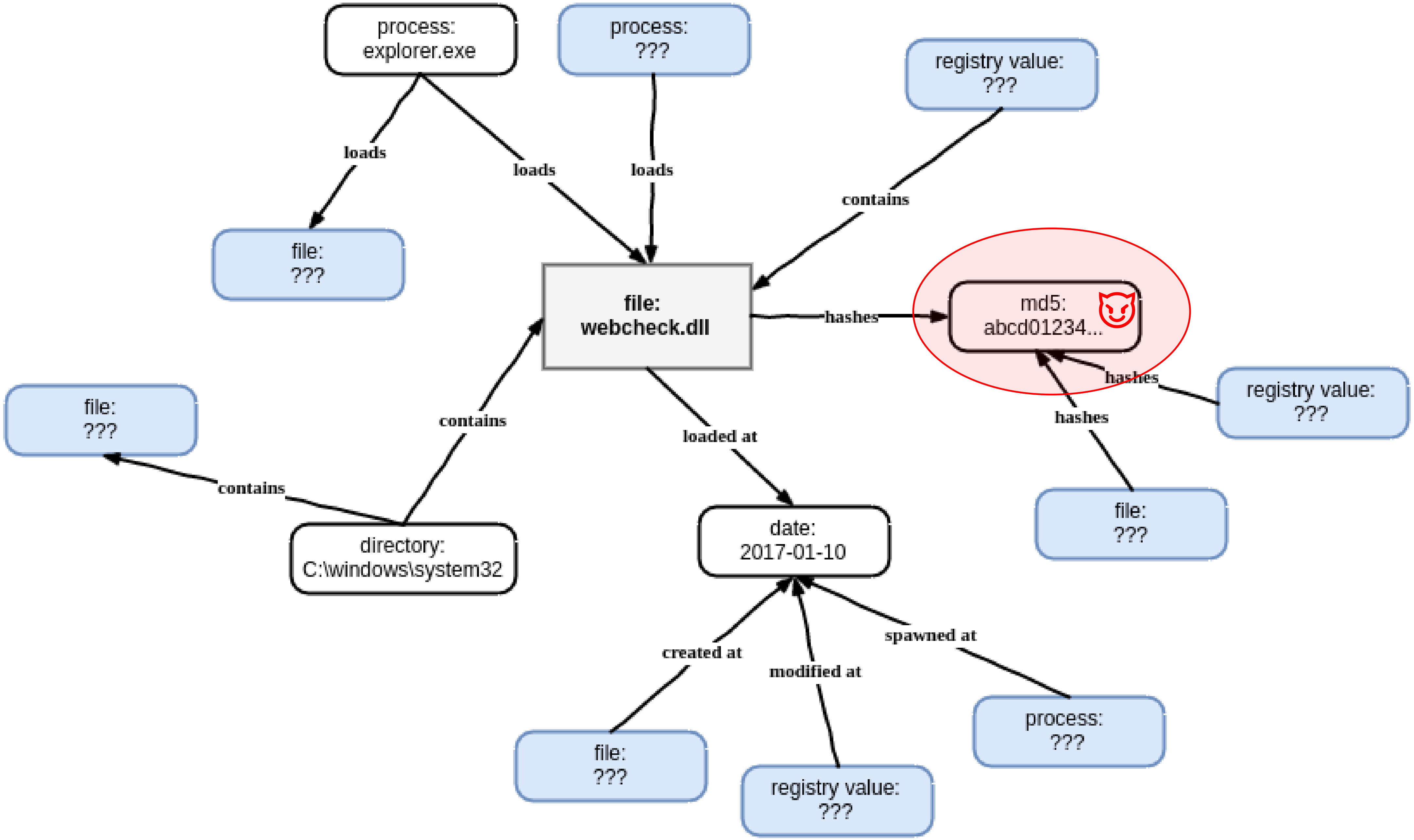
Given suspicious node propagate score from suspicious node to neighbor nodes in the graph

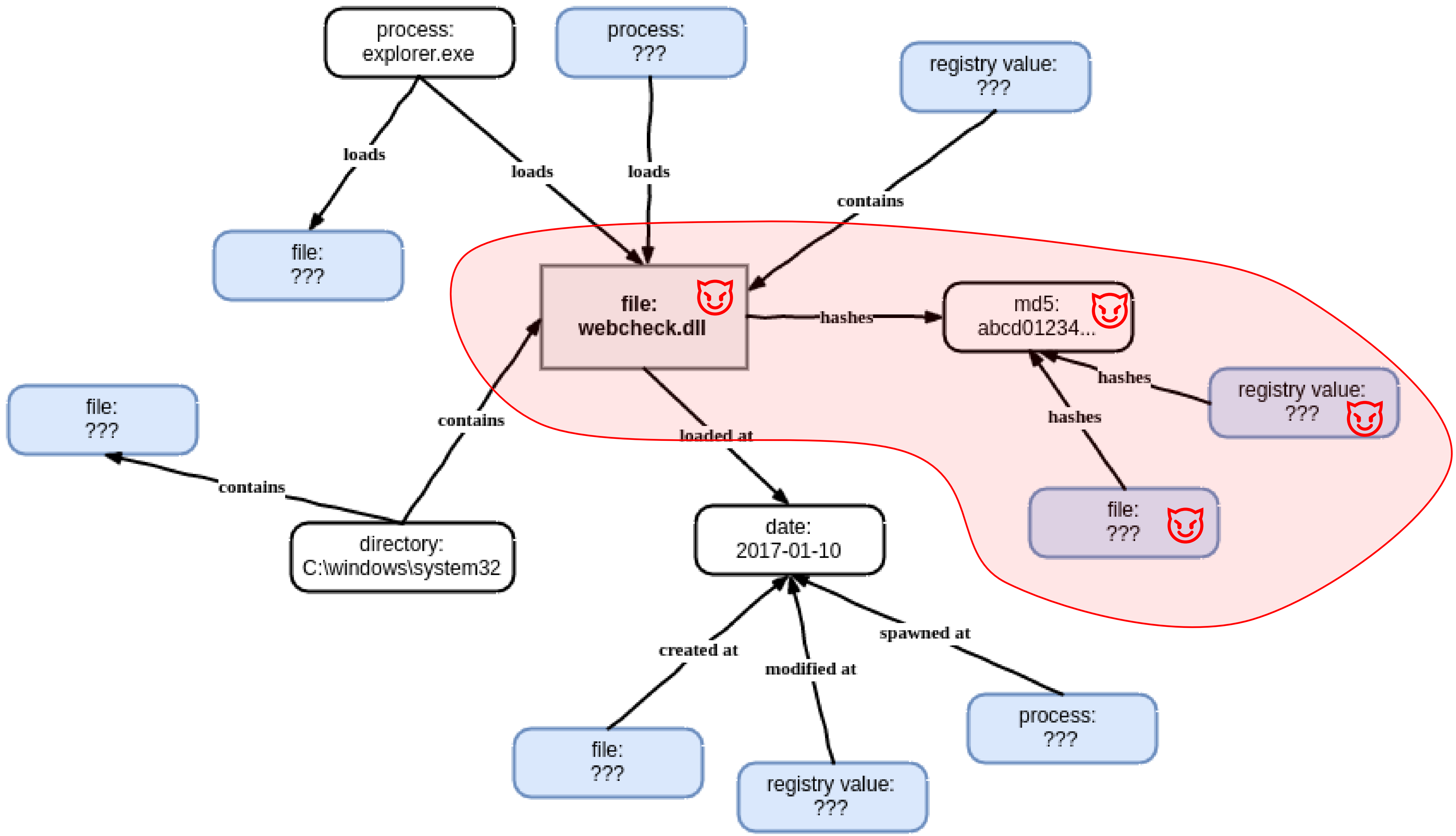
Enables weak signal detection when multiple weak signals within the same neighborhood propagate scores to meet a given threshold for detection

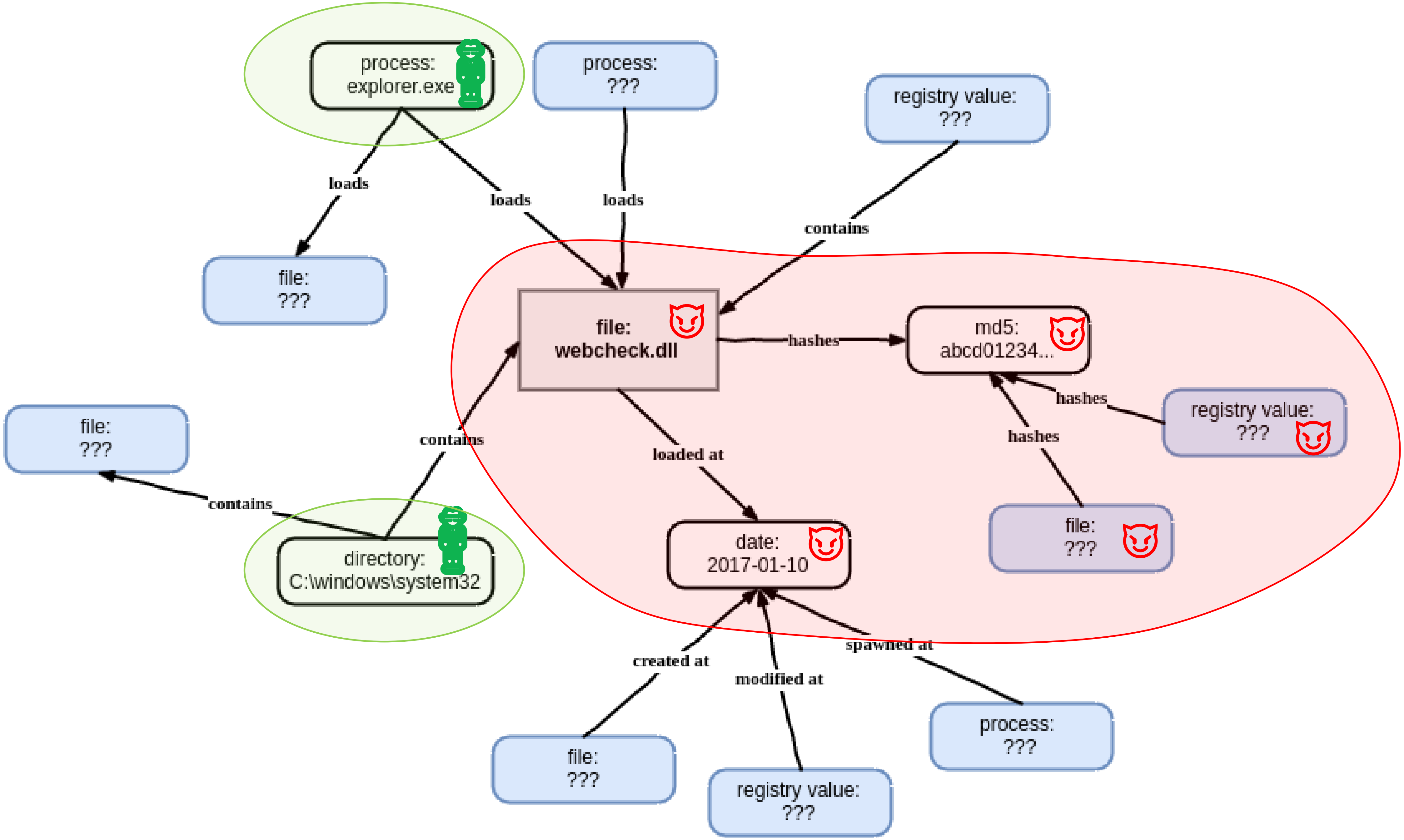
Tested with PageRank and HITS algorithms they follow the thread

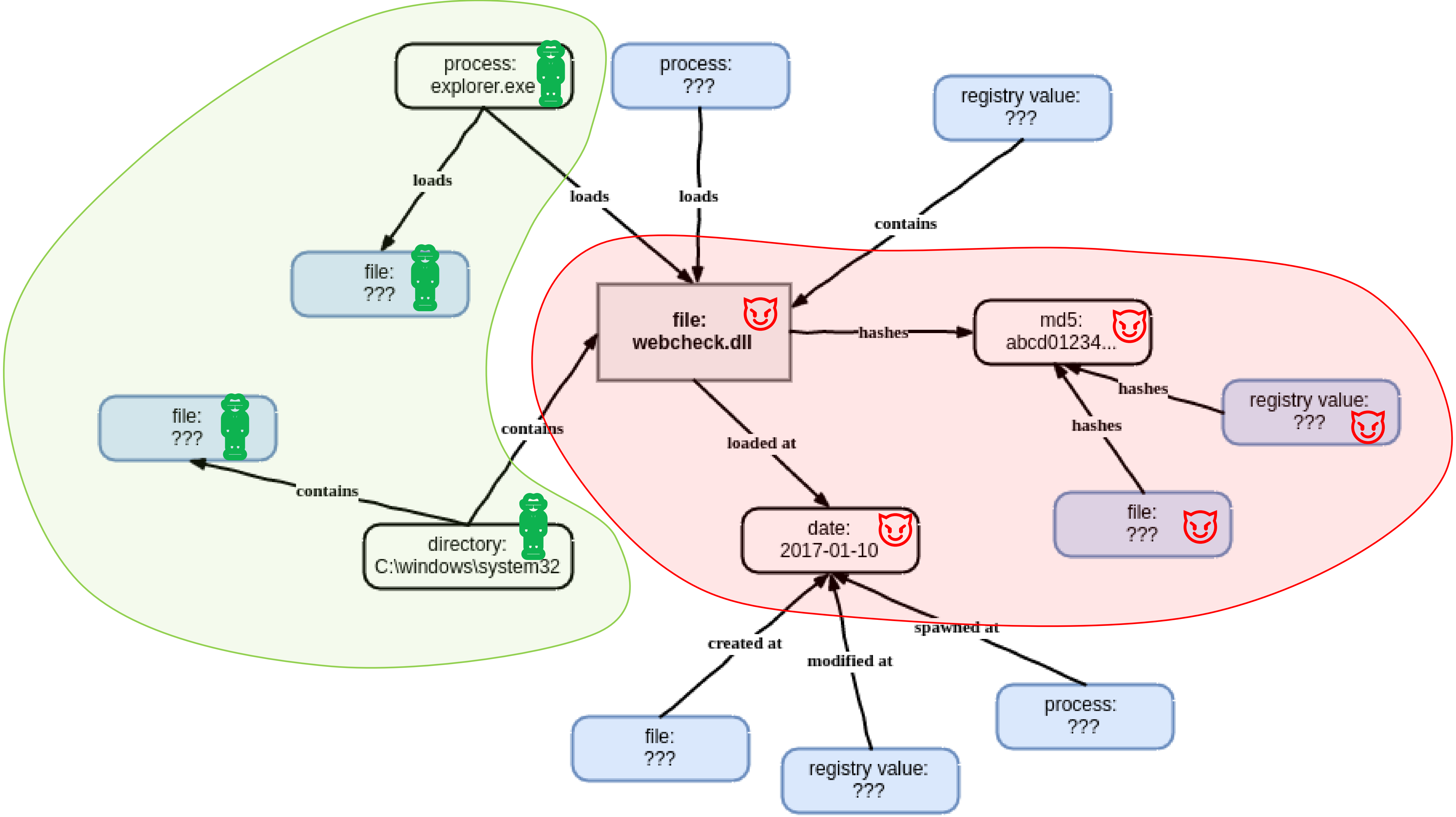


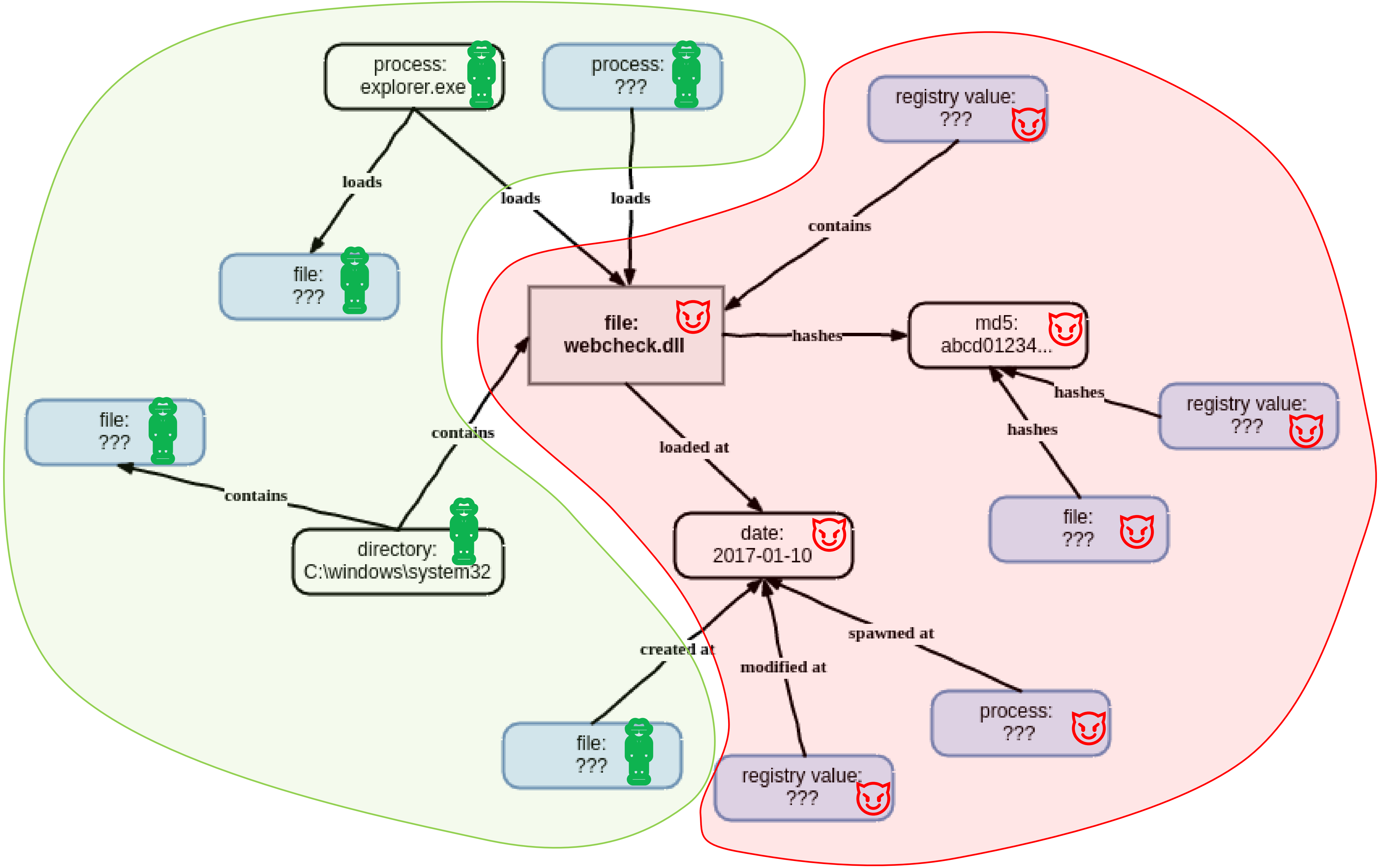

process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
😈
hashes
registry value:
???
hashes
file:
???
contains
loaded at
contains
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
😈
hashes
hashes
registry value:
???
file:
???
contains
contains
loaded at
file:
???
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
hashes
file:
???
contains
loaded at
file:
???
contains
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
hashes
file:
???
contains
loaded at
file:
???
contains
directory:
C:\windows\system32
date:
2017-01-10
created at
modified at
spawned at
file:
???
registry value:
???
process:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
contains
hashes
loaded at
file:
???
contains
file:
???
directory:
C:\windows\system32
date:
2017-01-10
spawned at
created at
modified at
process:
???
file:
???
registry value:
???


process:
explorer.exe
process:
???
registry value:
???
loads
loads
loads
contains
file:
???
file:
webcheck.dll
hashes
md5:
abcd01234...
hashes
registry value:
???
file:
???
contains
hashes
loaded at
contains
file:
???
directory:
C:\windows\system32
date:
2017-01-10
spawned at
created at
modified at
process:
???
file:
???
registry value:
???

algorithmic considerations

supernodes (nodes with many edges)

- e.g. every process loads `kernel32.dll`
- therefore, naïve N-degree traversal quickly explodes
- potential mitigation: weight or threshold nodes by degree
- nicely intuitive: items in a smaller directory are probably more closely related

works: swarm algorithm that randomly walks the neighborhood

- output: the nodes (and their weights) most related to the input set
- interpretation: artifacts that might be relevant to a report

lessons learned

lessons learned

- many advanced analysts still want their grid
    - maybe it’s the data density of a spreadsheet when hunting & consuming data?
    - graph data structure shouldn’t necessarily imply a graph user interface

    - its not a naïve splat of the graph to the screen; tailor graph presentation to guide user
        - in ST, layout order has meaning, and node collapsing implies further context

- its about processing less data, not more

- (like we knew) data model matters: it both limits and enables operations