On the Forensic Validity of Approximated Audit Logs

Noor Michael, ***Jaron Mink***, Jason Liu,
Sneha Gaur, Wajih Ul Hassan, and Adam Bates

University of Illinois at Urbana-Champaign

Audit Logs are Invaluable

Audit Logs are Invaluable

- Records history of executed events
  - Kernel-level frameworks track application syscalls

Audit Logs are Invaluable

- Records history of executed events
  - Kernel-level frameworks track application syscalls

- 75% of analysts [1] believe logs are the most important resource when investigating threats

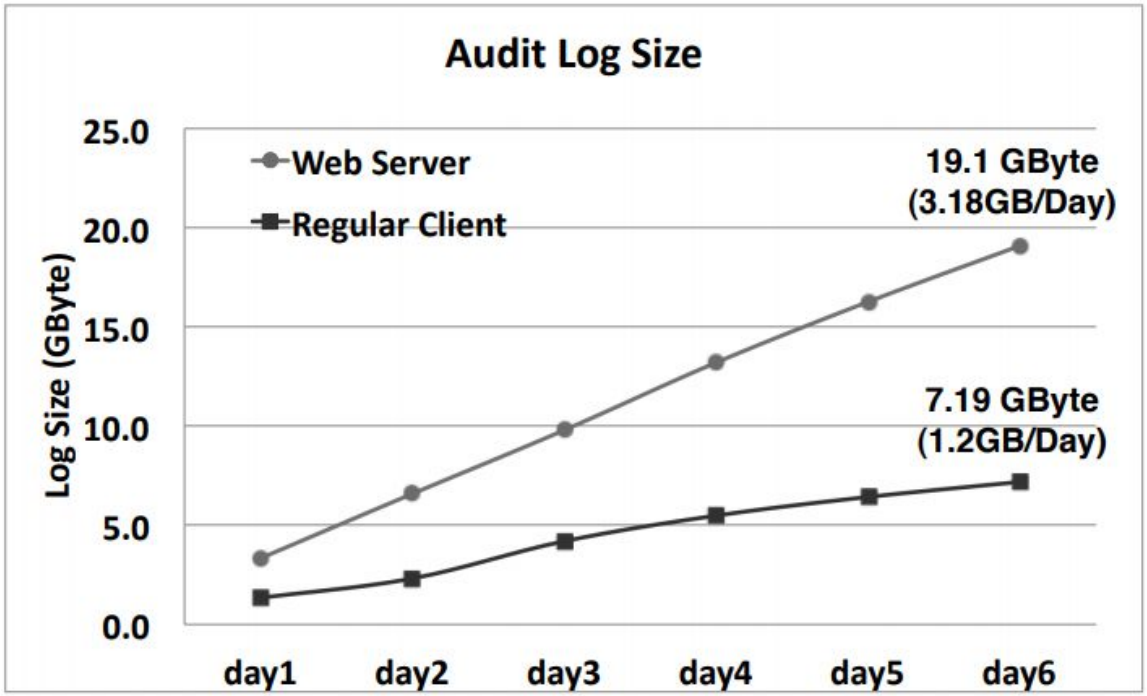
[1] Carbon Black. 2018. Global Incident Response Threat Report. https://www. carbonblack.com/global-incident-response-threat-report/november-2018/

Audit Logs are Invaluable
… but Burdensome

- Records history of executed events
  - Kernel-level frameworks track application syscalls

- 75% of analysts [1] believe logs are the most important resource when investigating threats

[2]



[1] Carbon Black. 2018. Global Incident Response Threat Report. https://www. carbonblack.com/global-incident-response-threat-report/november-2018/

[2] Lee et. al. LogGC: Garbage Collecting Audit Log. In Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security (CCS '13)

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine where process A sent data

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine where process A sent data

```
  1: <Proc A, t_001, send, server.com>
                  ...
 99: <Proc A, t_099, send, server.com>
```

Original Log

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine where process A sent data

```
  1: <Proc A, t_001, send, server.com>
                 ...
 99: <Proc A, t_099, send, server.com>
```

Original Log

```
  1: <Proc A, t_0XX send, server.com>
```

Approximated Log

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine where process A sent data

```
 1: <Proc A, t_001, send, server.com>
                ...
99: <Proc A, t_099, send, server.com>
```

→

```
1: <Proc A, t_0XX send, server.com>
```

Original Log

Approximated Log

The same conclusion is reached with either log

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

```
 1: <Proc A, t_001, send, server.com>
                ...
99: <Proc A, t_099, send, server.com>
```

Original Log

```
1: <Proc A, t_0XX send, server.com>
```

Approximated Log

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine whether Proc A was using a covert timing channel[1]

```
 1: <Proc A, t_001, send, server.com>
                ...
99: <Proc A, t_099, send, server.com>
```

Original Log

```
1: <Proc A, t_0XX send, server.com>
```

Approximated Log

[1] Cabuk, S., Brodley, C. E., & Shields, C. (2004, October). IP covert timing channels: design and detection. In *Proceedings of the 11th ACM conference on Computer and communications security*

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine whether Proc A was using a covert timing channel[1]

```
 1: <Proc A, t_001, send, server.com>
                ...
99: <Proc A, t_099, send, server.com>
```

Original Log

```
1: <Proc A, t_0XX send, server.com>
```

Approximated Log

Conclusions may differ!

[1] Cabuk, S., Brodley, C. E., & Shields, C. (2004, October). IP covert timing channels: design and detection. In *Proceedings of the 11th ACM conference on Computer and communications security*

Audit Log Reduction Techniques

Insight: The entire audit log is not often required

Information may be:

- not needed for investigation goal
- redundant
- reasonably approximated

Investigation Goal: Determine whether Proc A was using a covert timing channel[1]

```
 1: <Proc A, t_001, send, server.com>
                ...
99: <Proc A, t_099, send, server.com>
```

→

```
 1: <Proc A, t_0XX send, server.com>
```

Original Log **Approximated Log**

Conclusions may differ!

How much information is kept for arbitrary goals under different threat models?

[1] Cabuk, S., Brodley, C. E., & Shields, C. (2004, October). IP covert timing channels: design and detection. In *Proceedings of the 11th ACM conference on Computer and communications security*

Formalizing Forensic Metrics

Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

Edges: Causal Events

Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

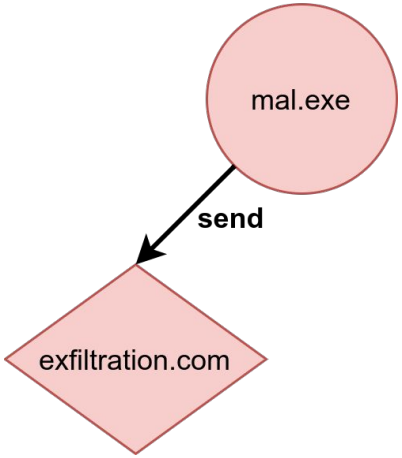
Edges: Causal Events

Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

Edges: Causal Events

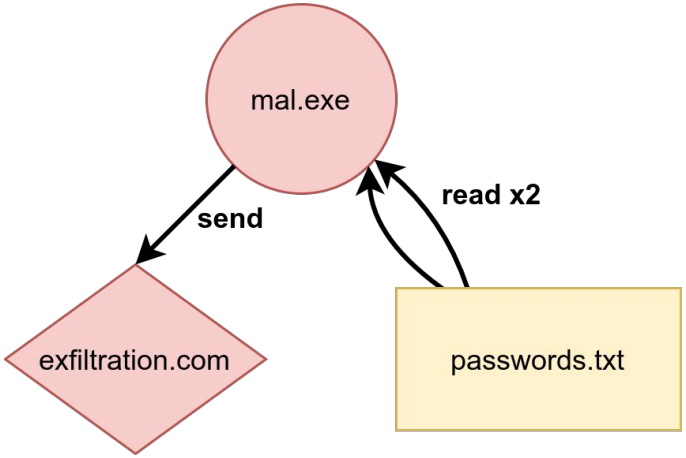


Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

Edges: Causal Events

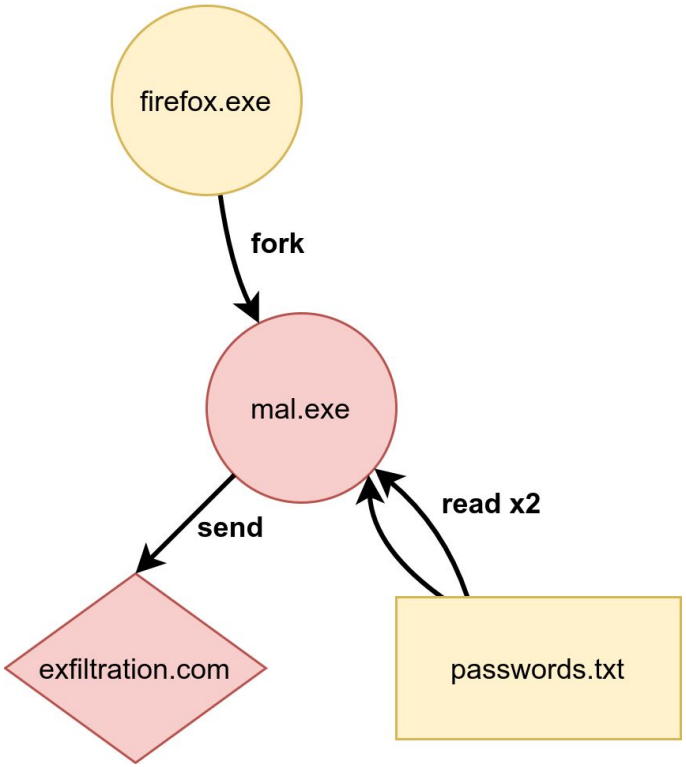


Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

Edges: Causal Events

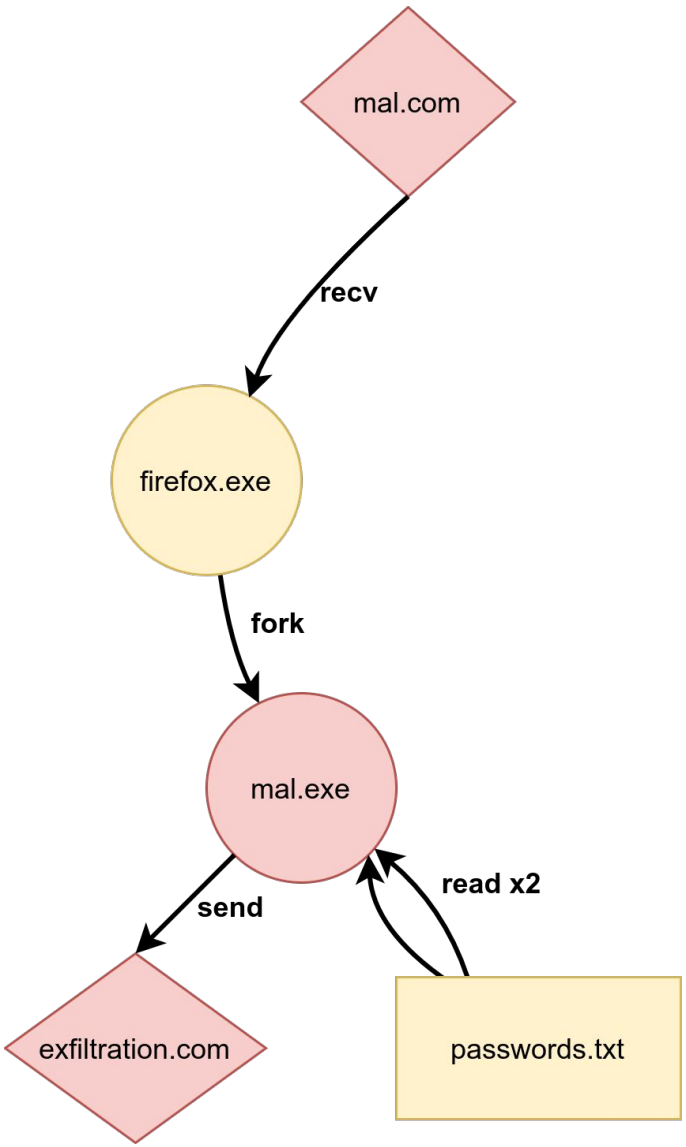


Formalizing Forensic Metrics

Provenance Graph

Nodes: System Objects

Edges: Causal Events

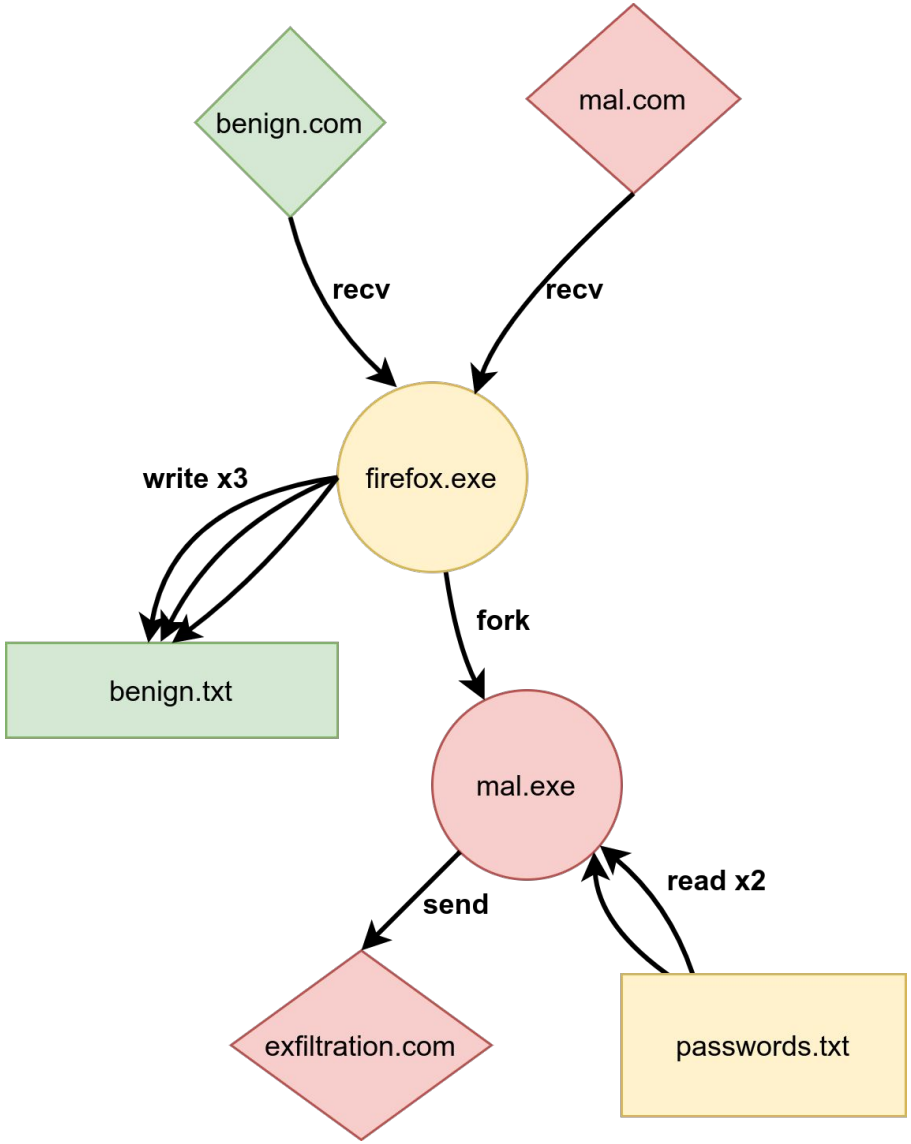


Formalizing Forensic Metrics

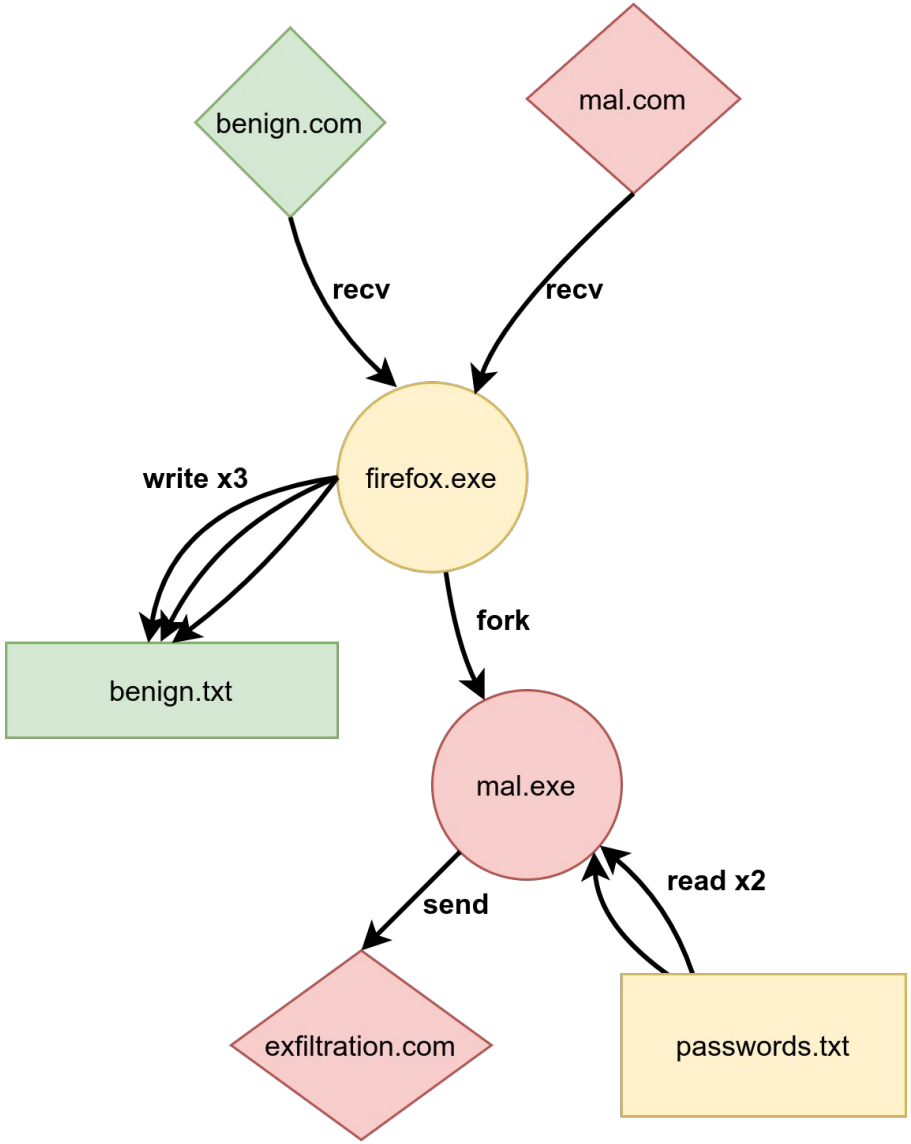
Provenance Graph

Nodes: System Objects

Edges: Causal Events



Formalizing Forensic Metrics

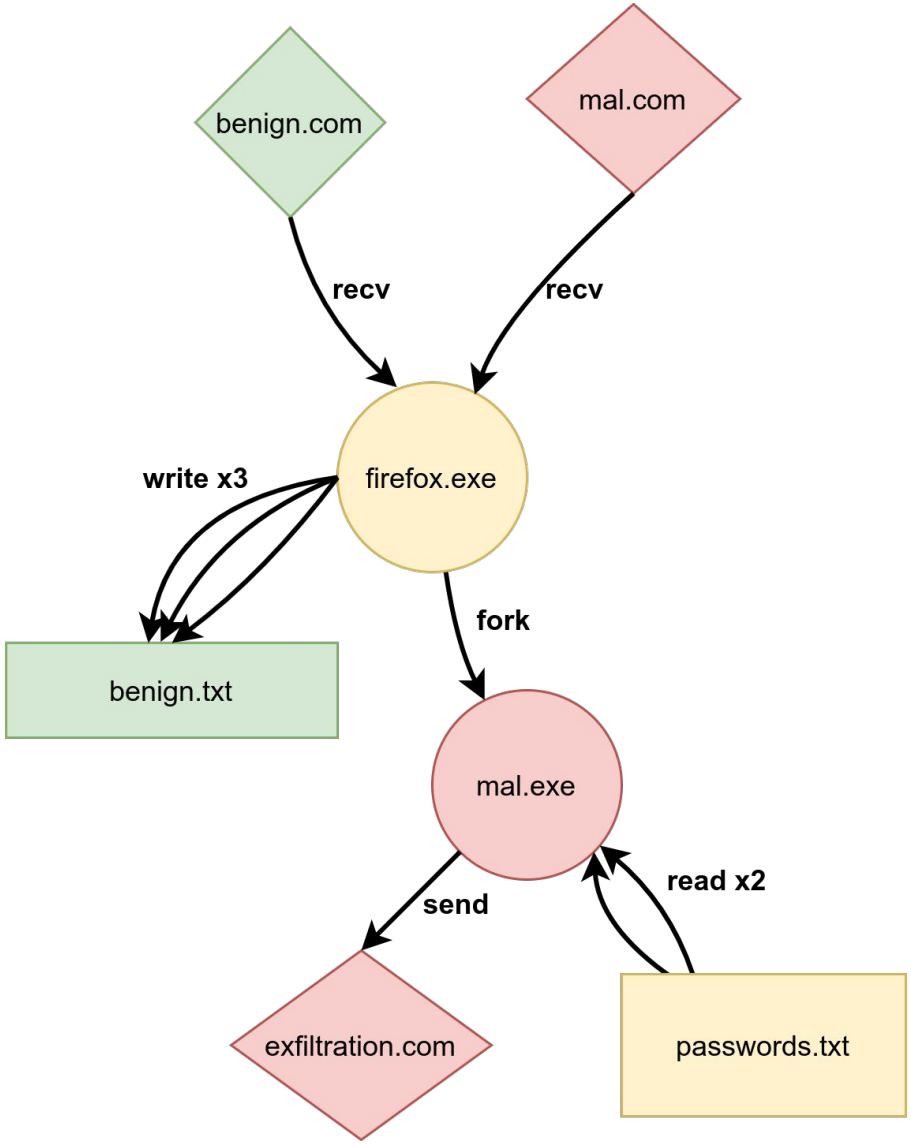


Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information



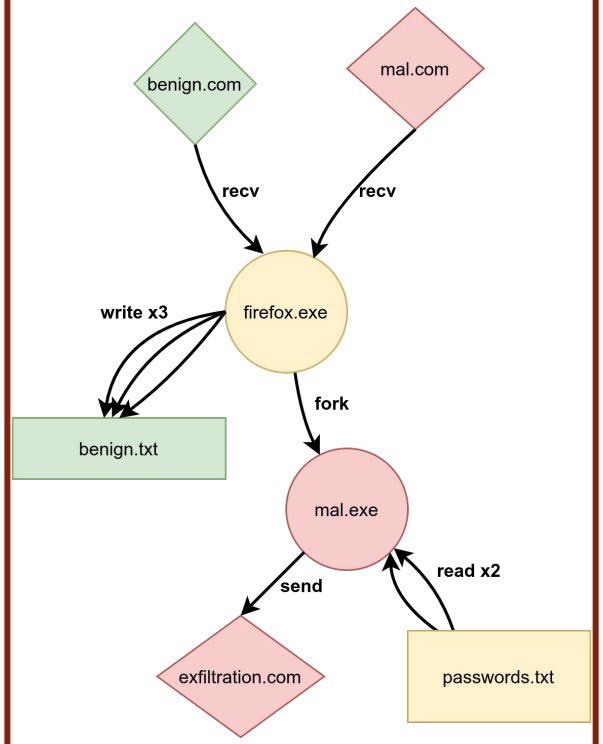
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

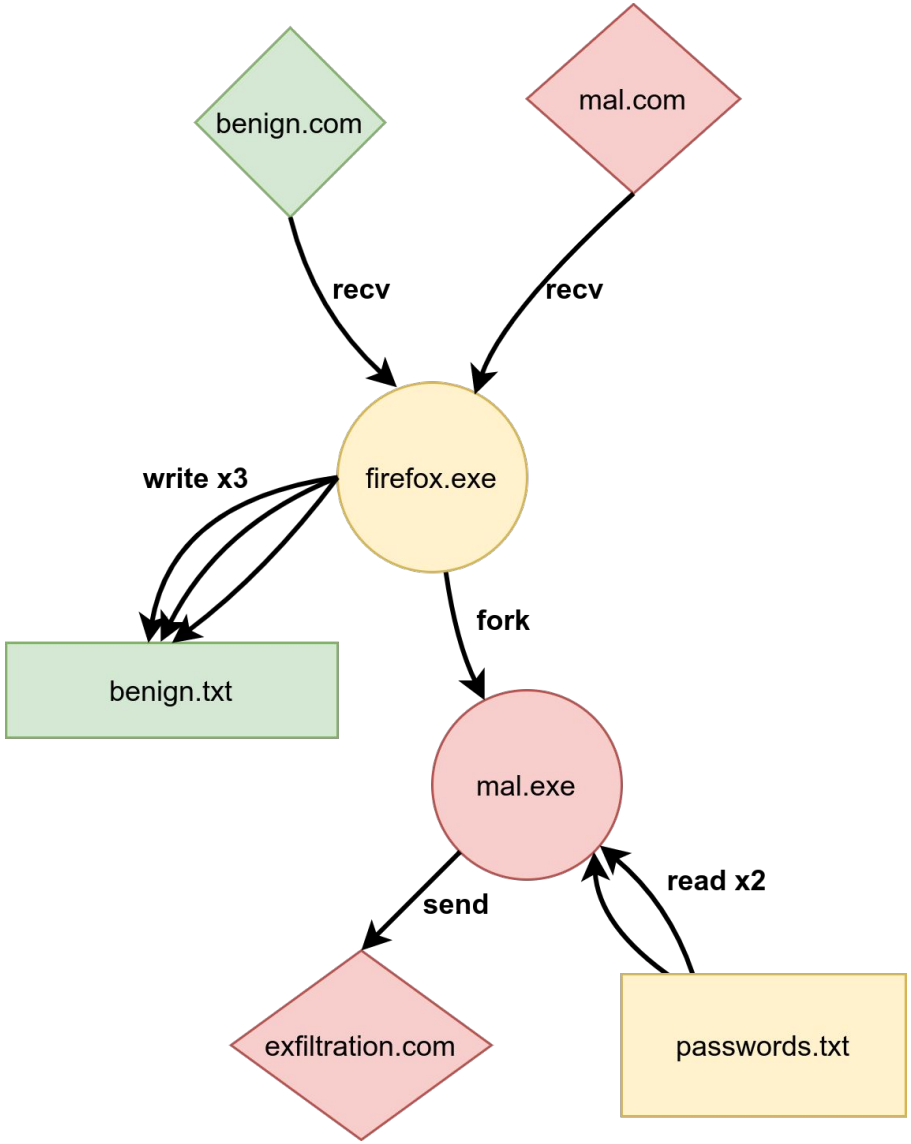
Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information





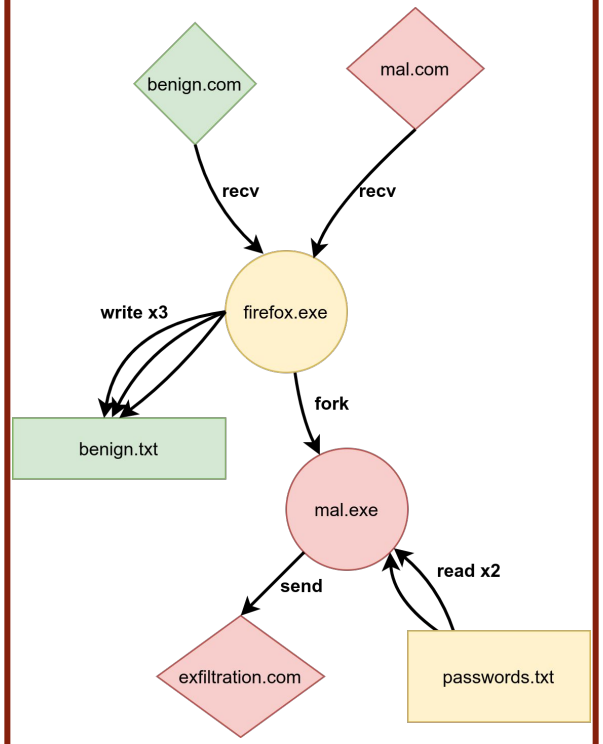
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information



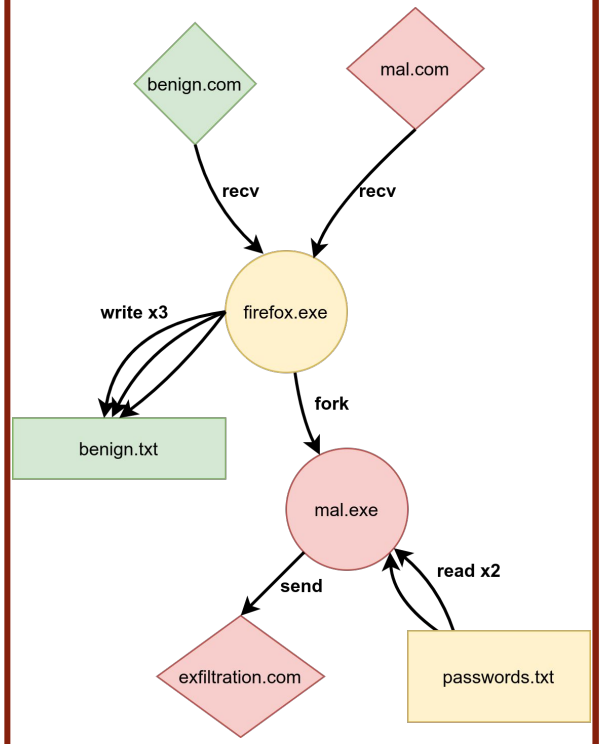
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information



Causality-Preserving

(based on Xu et. al.[1])

Threat Model:
Abides by system level abstractions

Preserves:
Information flow

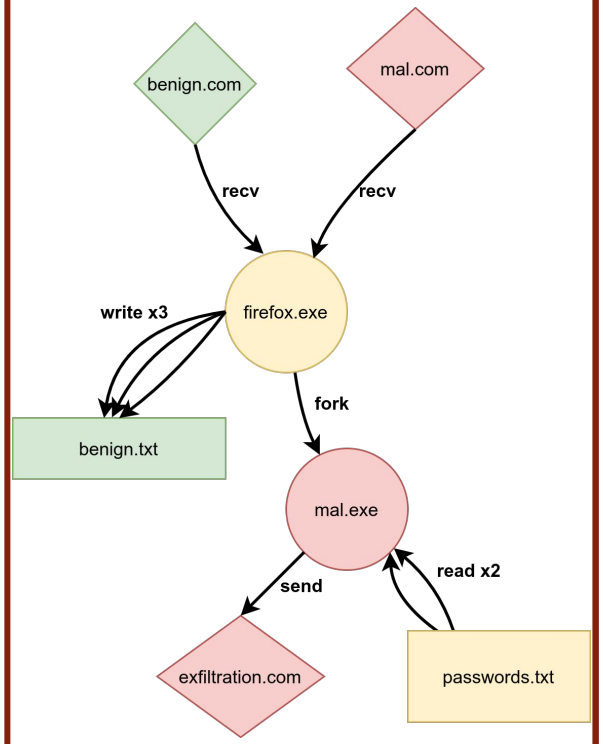
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information

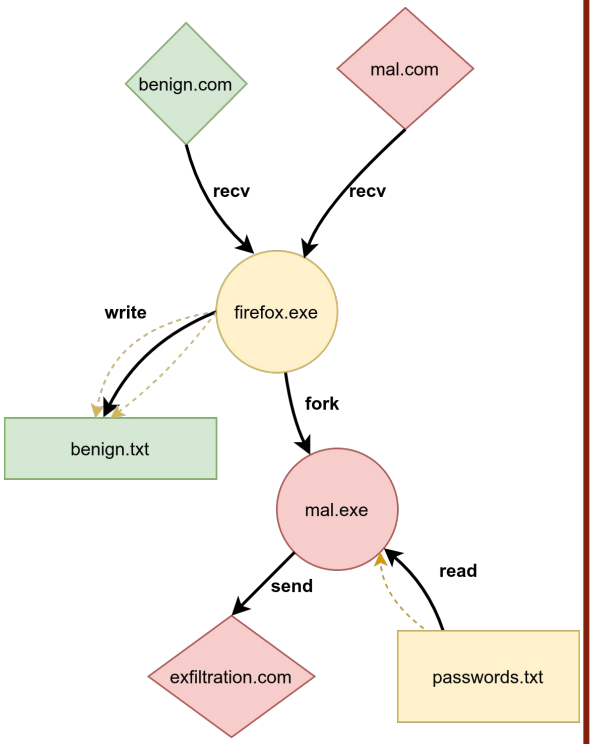


Causality-Preserving

(based on Xu et. al.[1])

Threat Model:
Abides by system level abstractions

Preserves:
Information flow



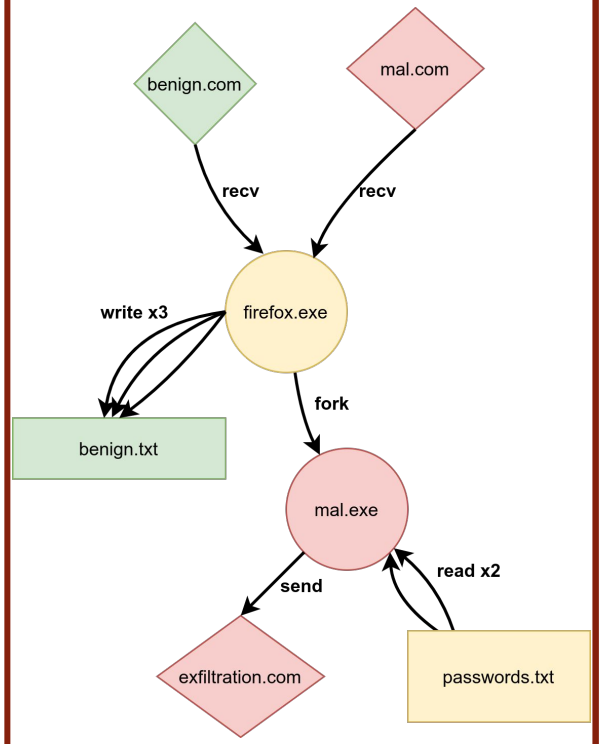
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information

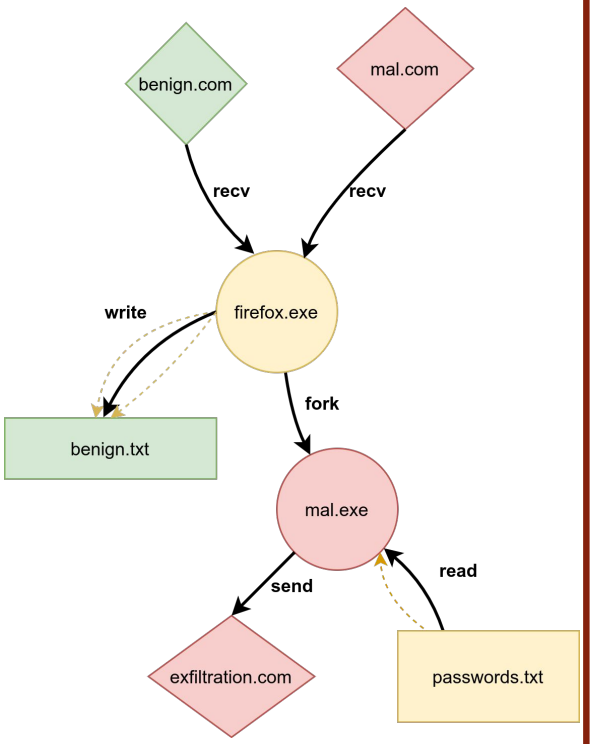


Causality-Preserving

(based on Xu et. al.[1])

Threat Model:
Abides by system level abstractions

Preserves:
Information flow



Attack-Preserving

Threat Model:
Abides by system level abstractions

Preserves:
Uniquely Malicious information flow

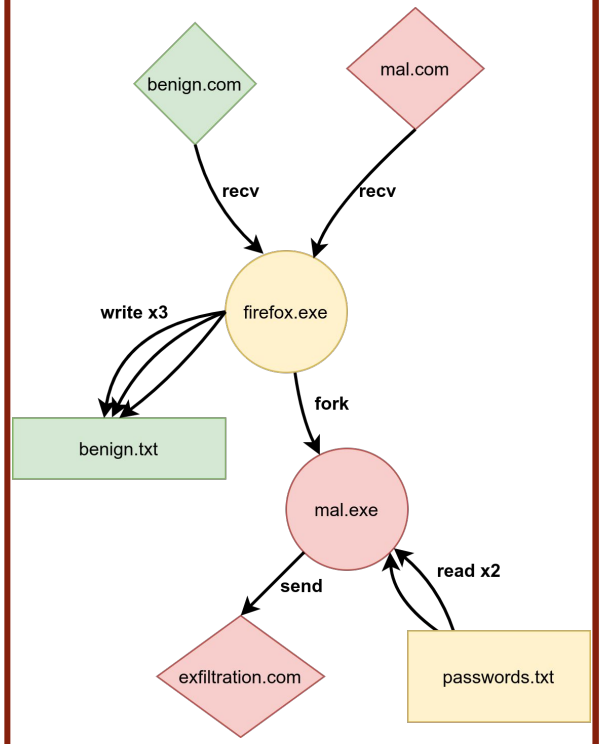
[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

Lossless

Threat Model:
Diverges from system level abstractions

Preserves:
All Information

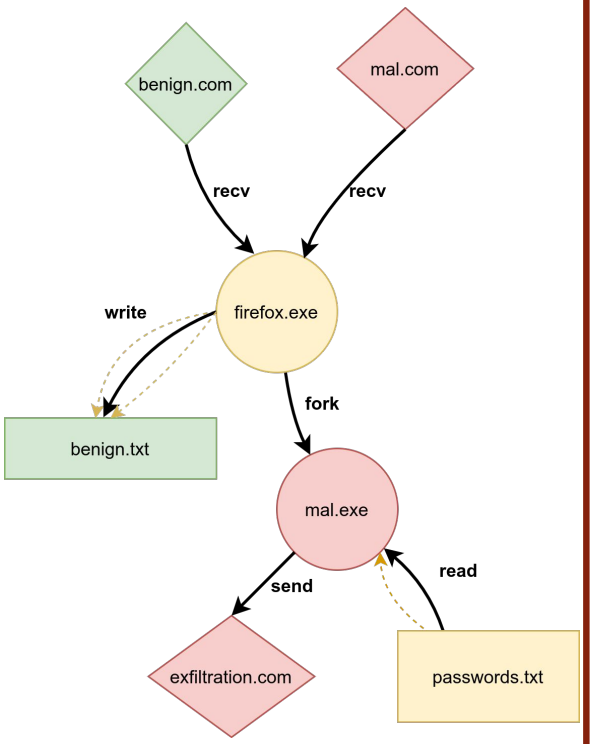


Causality-Preserving

(based on Xu et. al.[1])

Threat Model:
Abides by system level abstractions

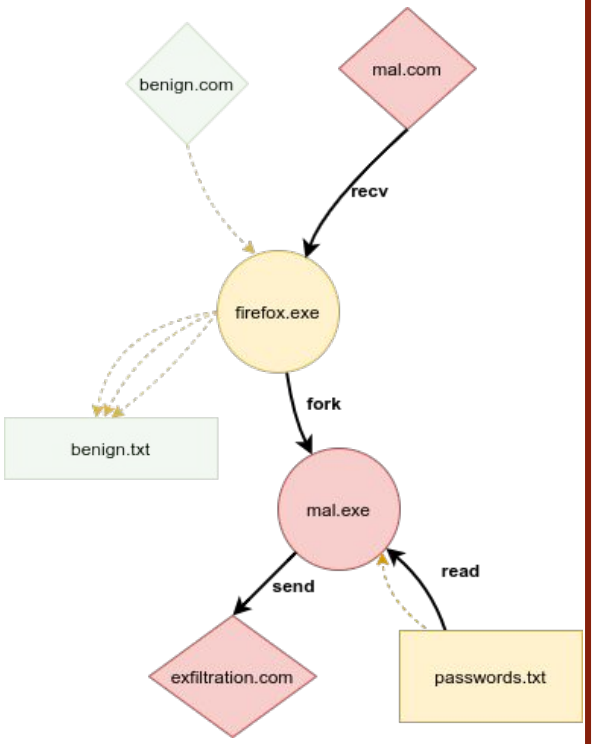
Preserves:
Information flow



Attack-Preserving

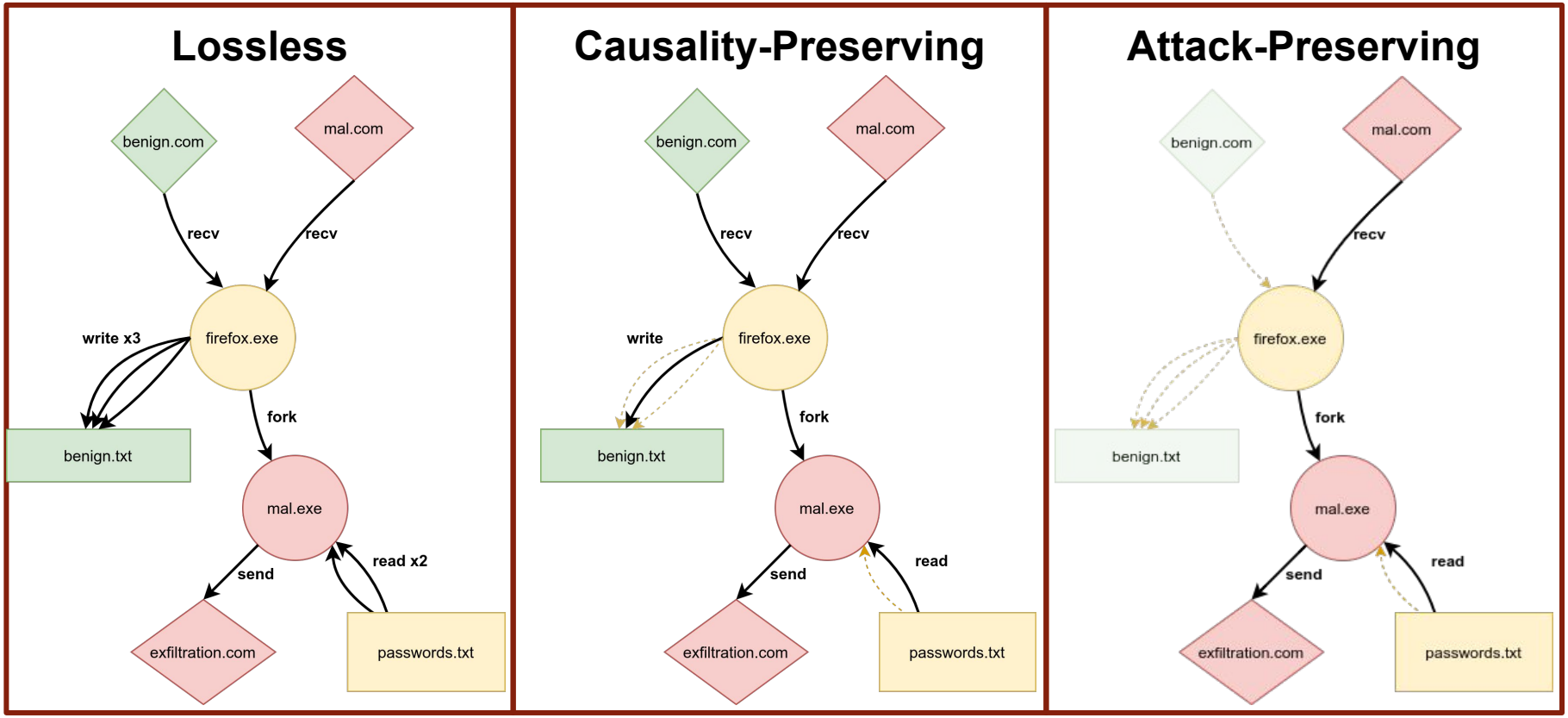
Threat Model:
Abides by system level abstractions

Preserves:
Uniquely Malicious information flow

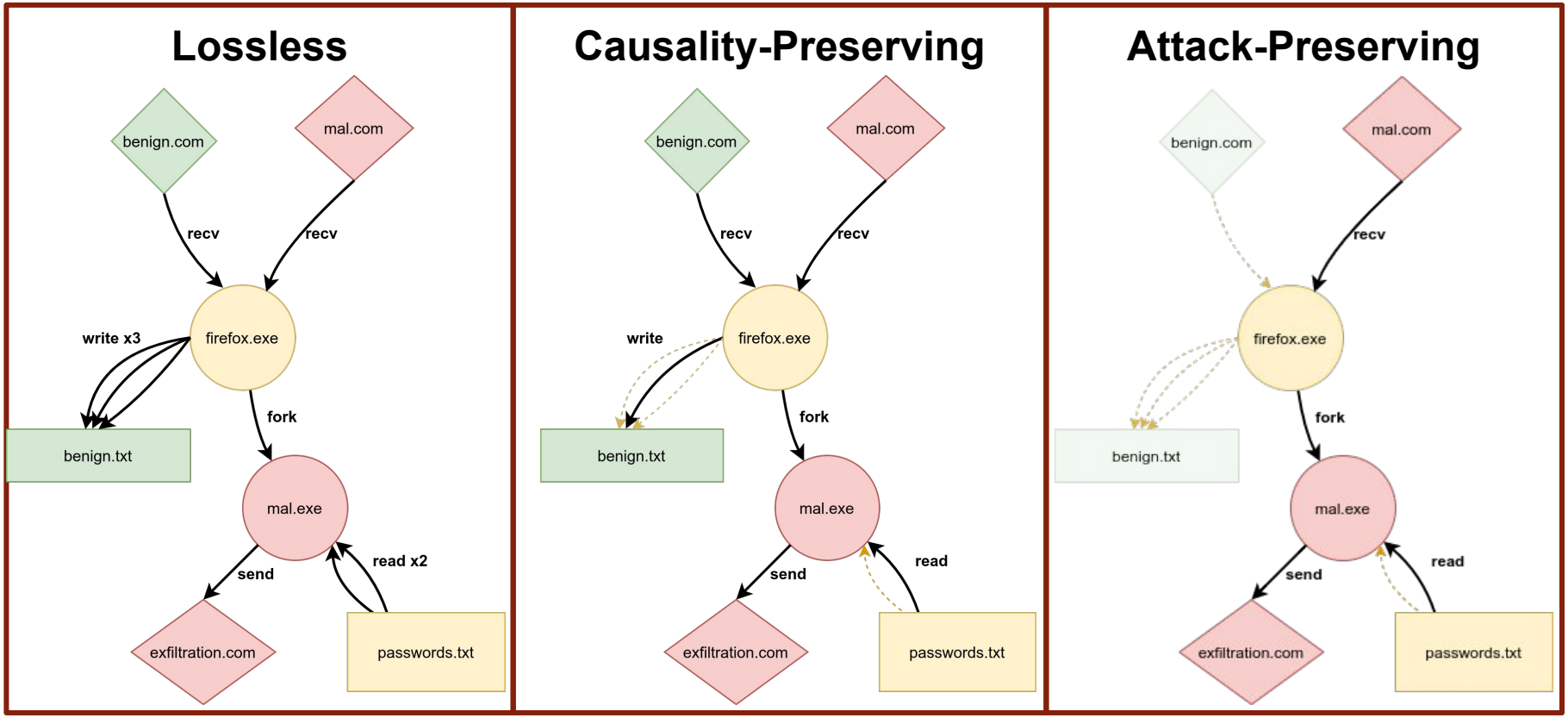


[1] Zhang Xu et. al. 2016. High Fidelity Data Reduction for Big Data Security Dependency Analyses. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security

Formalizing Forensic Metrics

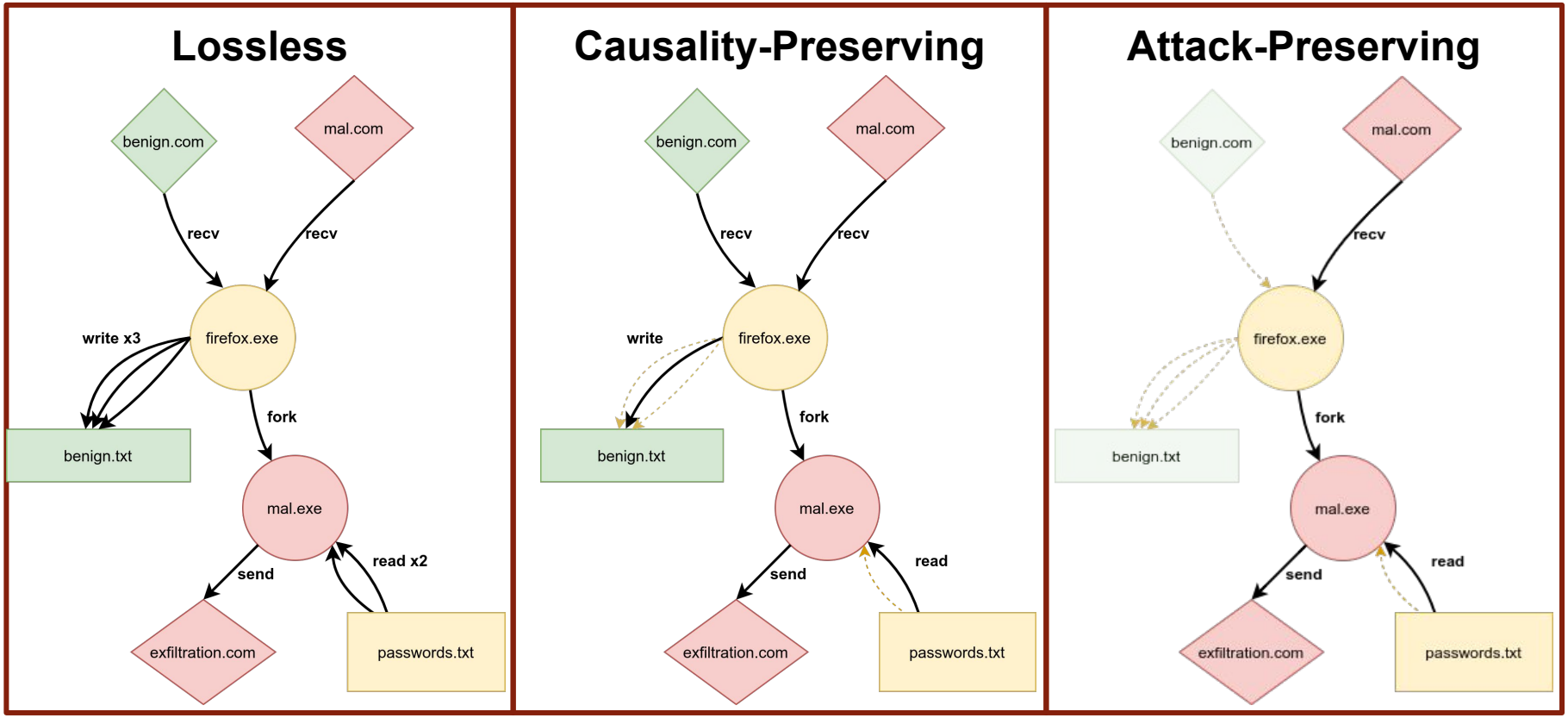


Formalizing Forensic Metrics



$$Reduction_i(G) = G_i$$

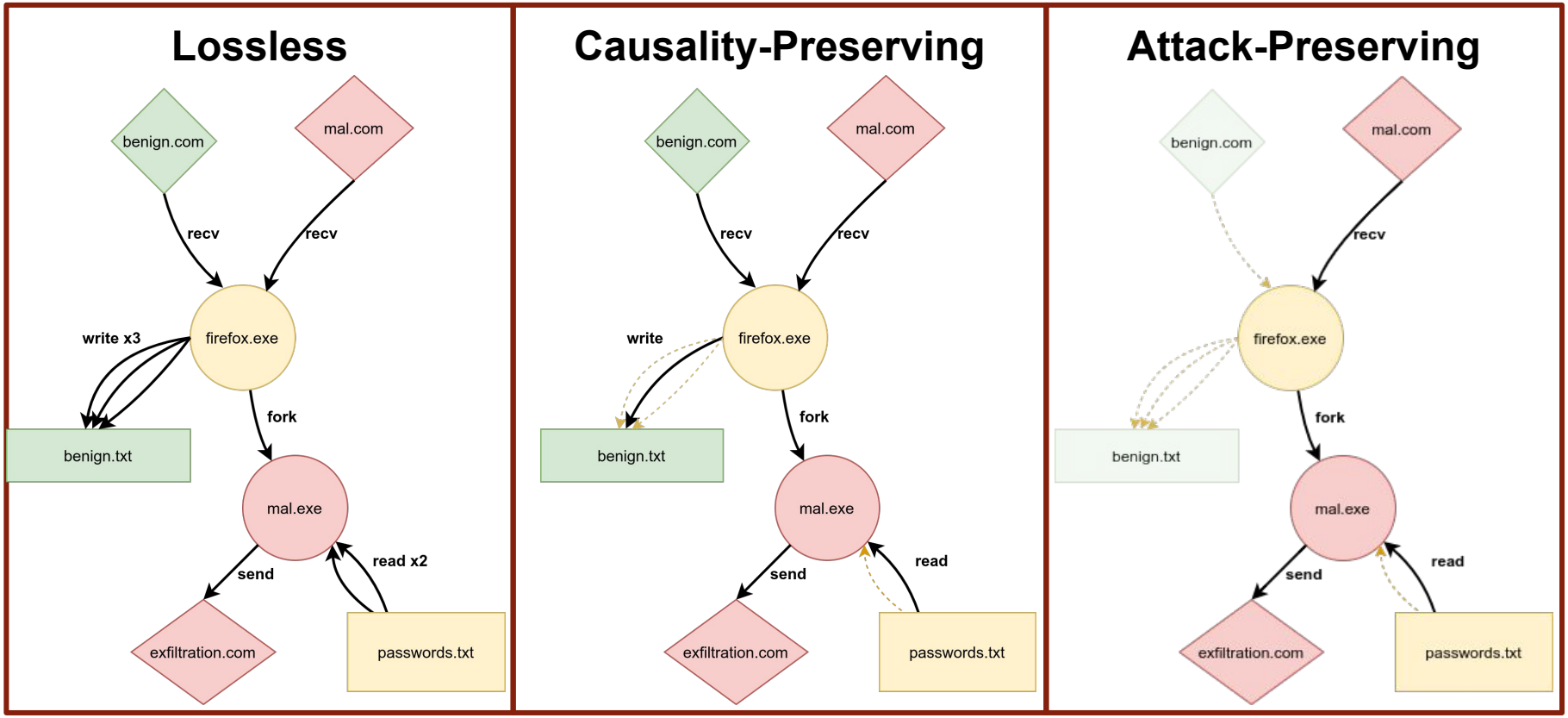
Formalizing Forensic Metrics



$$Reduction_i(G) = G_i$$

$$Ideal_m(G) = G_m$$

Formalizing Forensic Metrics

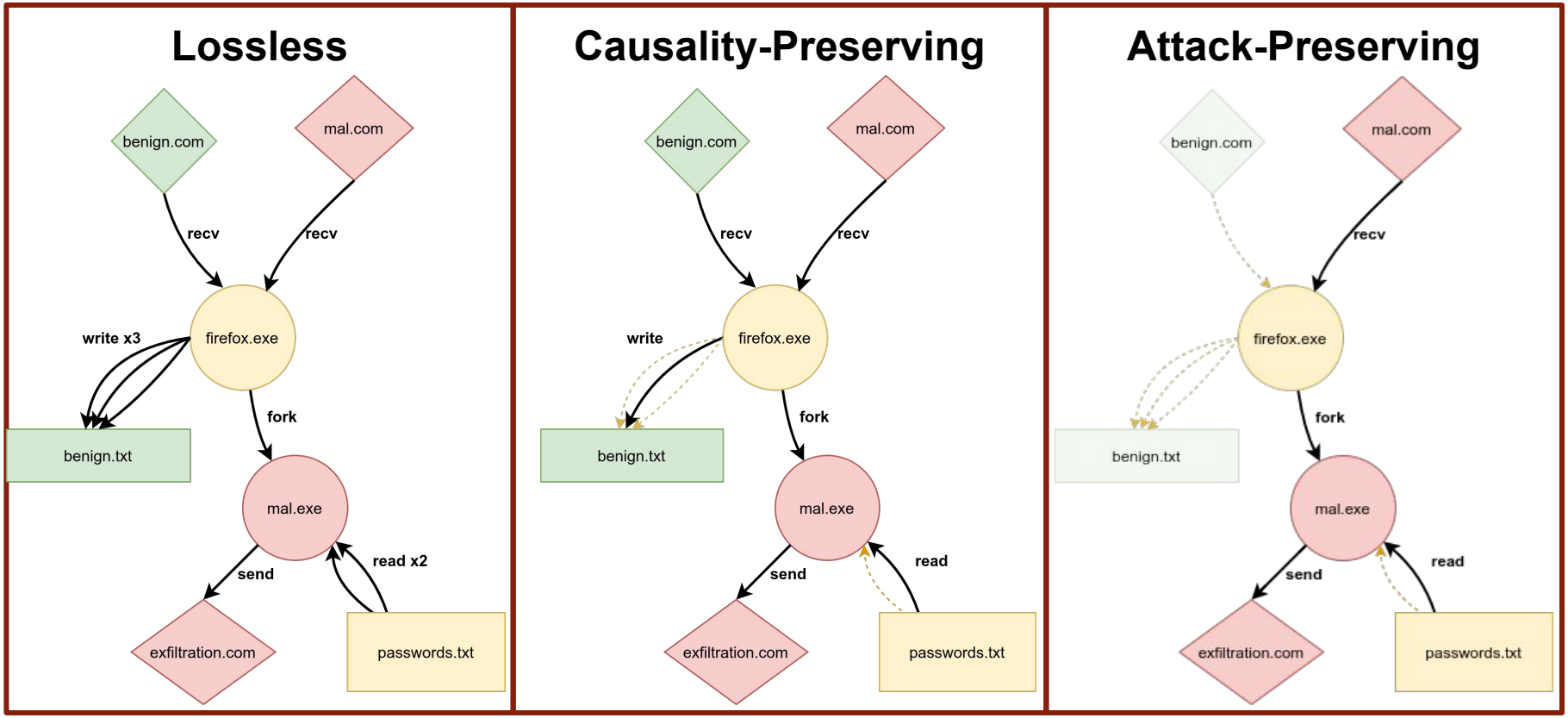


$$Reduction_i(G) = G_i$$

$$Ideal_m(G) = G_m$$

$$E(G) = \{e\}$$

Formalizing Forensic Metrics



$$Reduction_i(G) = G_i$$

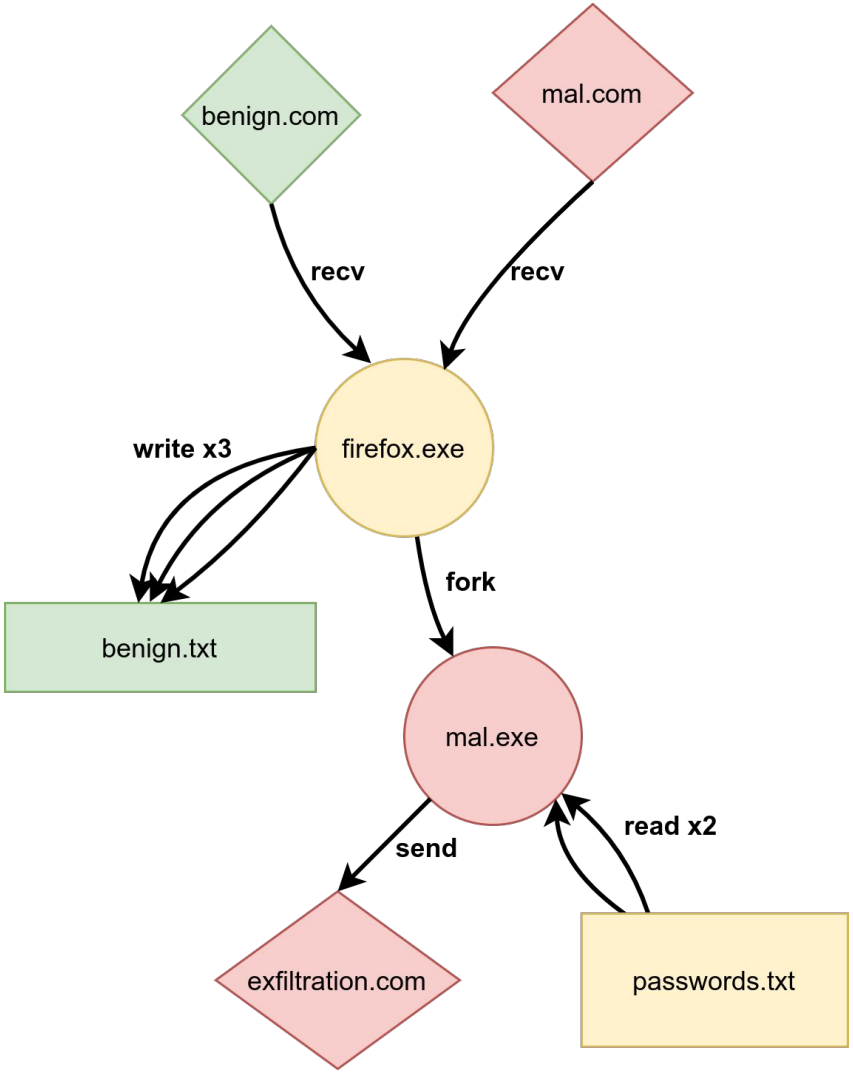
$$Ideal_m(G) = G_m$$

$$E(G) = \{e\}$$

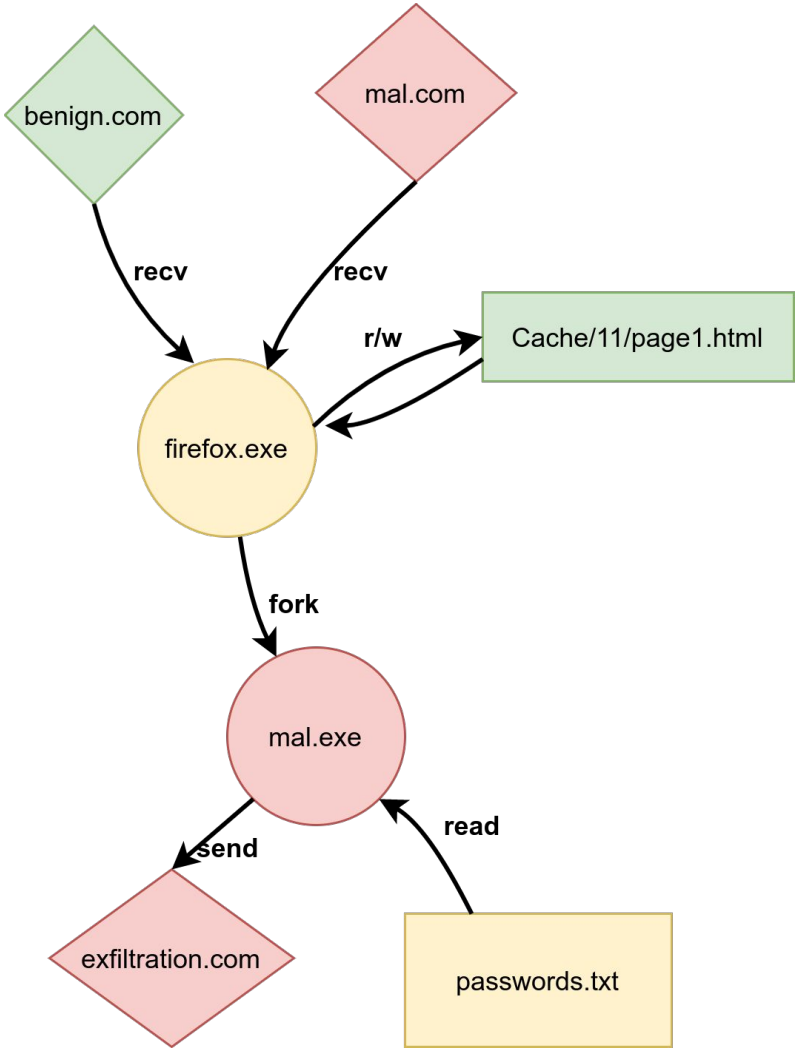
$$\text{Forensic Validity}_{i,m} = \frac{|E(G_i) \cap E(G_m)|}{|E(G_m)|}$$

LogApprox

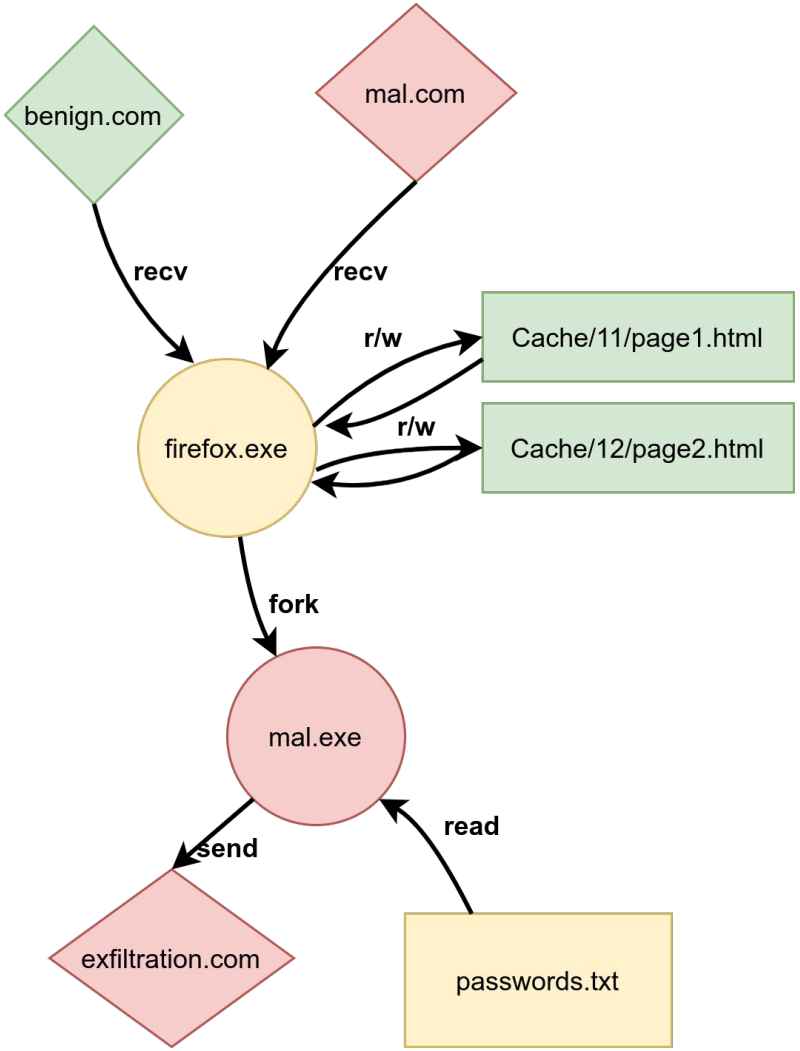
LogApprox


benign.com
mal.com
recv
recv
firefox.exe
write x3
benign.txt
fork
mal.exe
send
read x2
exfiltration.com
passwords.txt

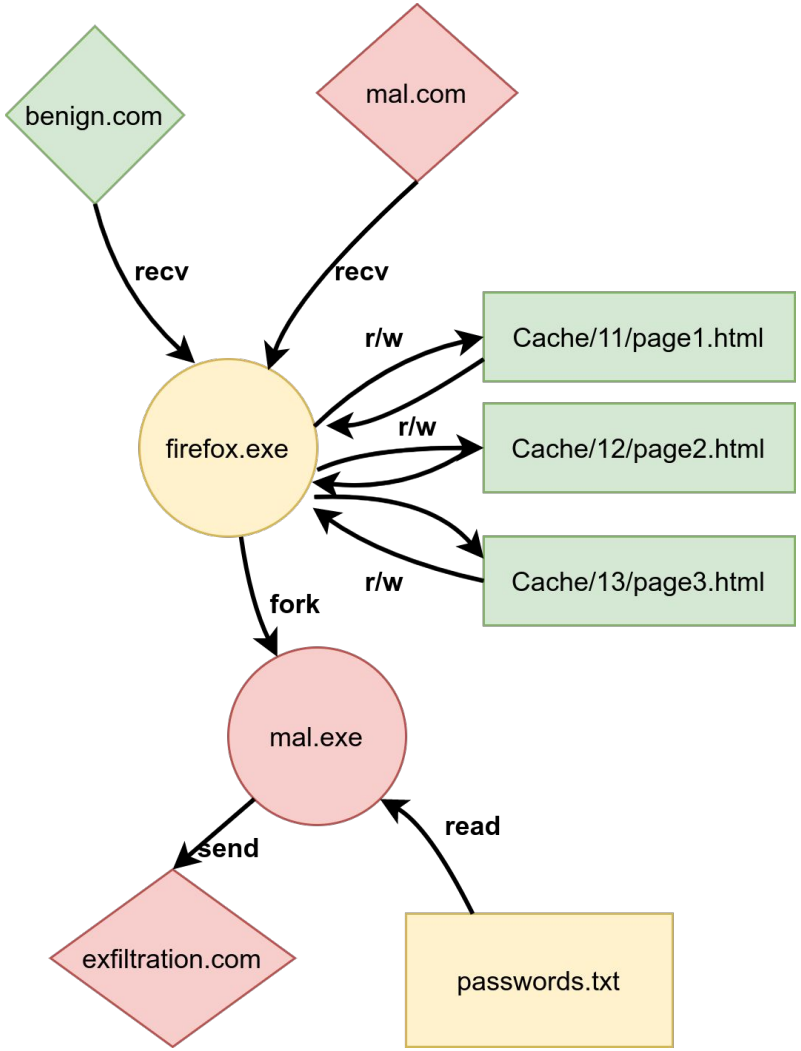
LogApprox


benign.com
mal.com
recv
recv
r/w
Cache/11/page1.html
firefox.exe
fork
mal.exe
send
read
exfiltration.com
passwords.txt

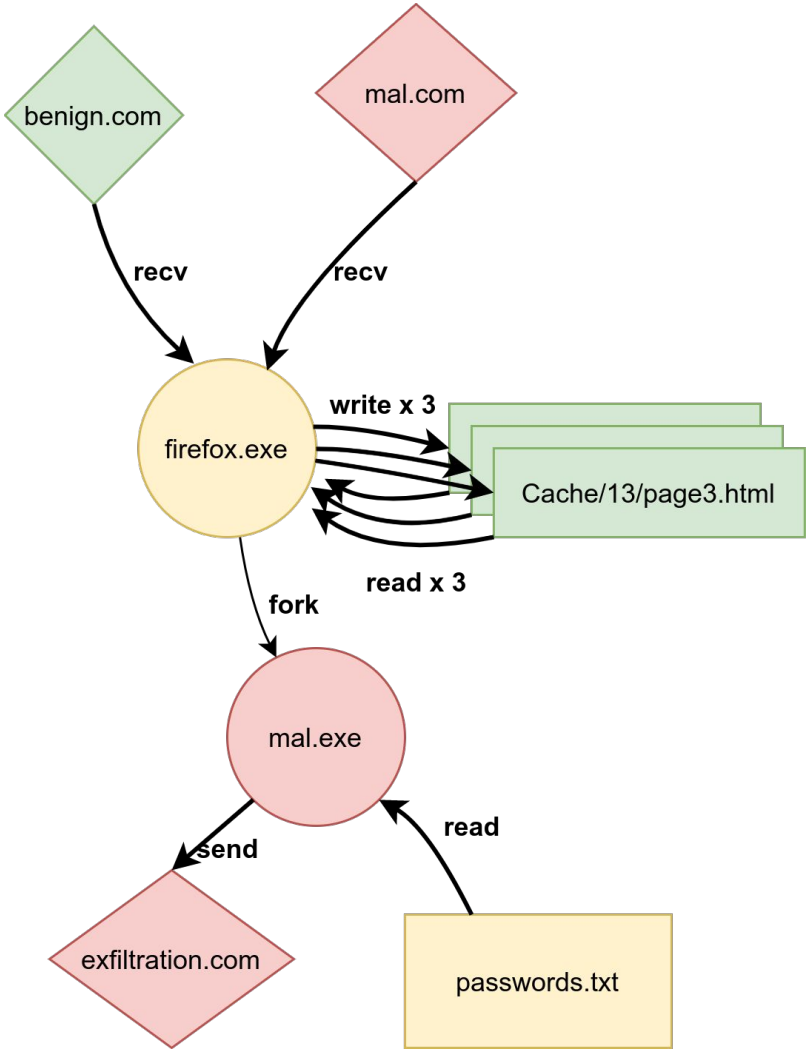
LogApprox


benign.com
mal.com
recv
recv
r/w
Cache/11/page1.html
r/w
firefox.exe
Cache/12/page2.html
fork
mal.exe
read
send
exfiltration.com
passwords.txt

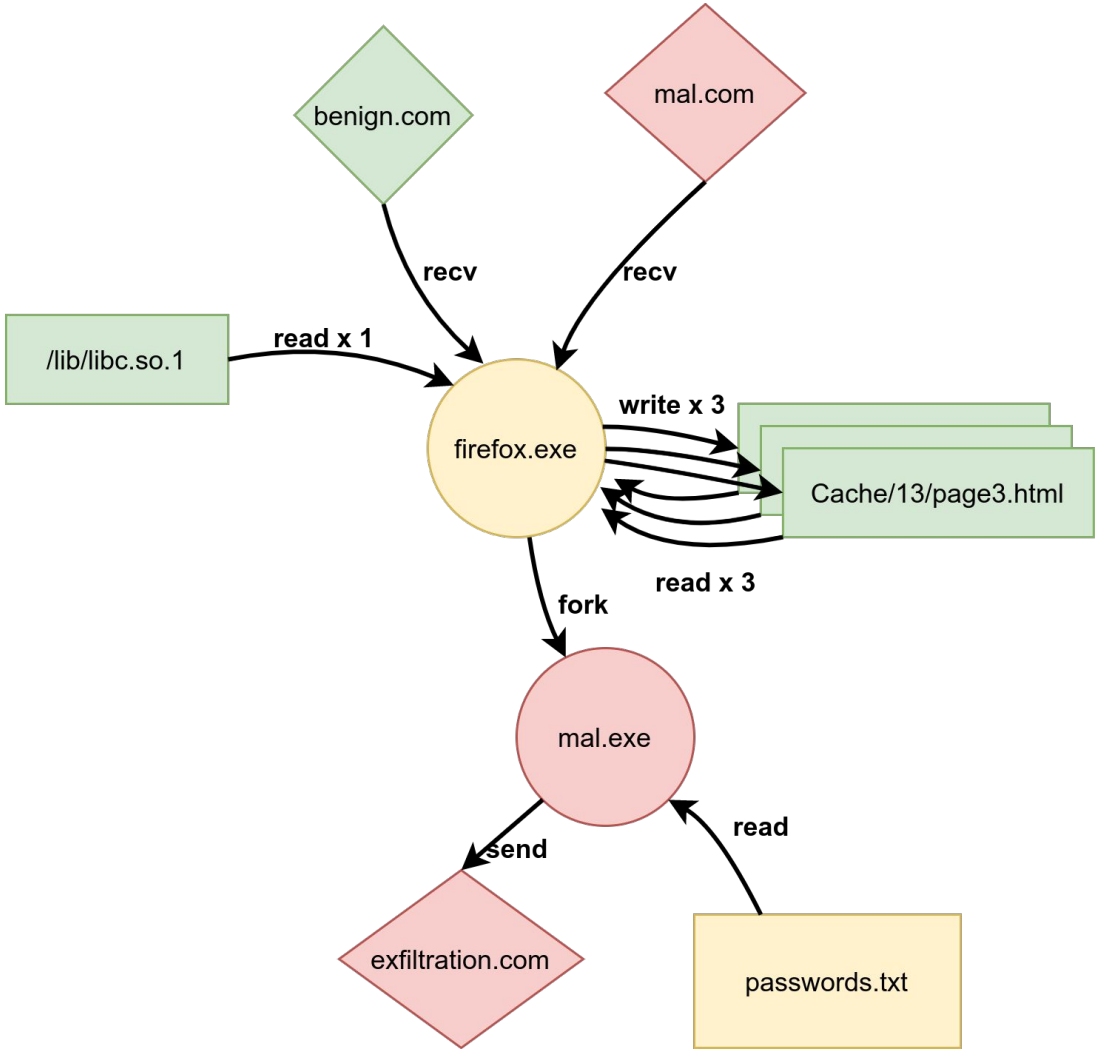
LogApprox


benign.com
mal.com
recv
recv
r/w
Cache/11/page1.html
firefox.exe
r/w
Cache/12/page2.html
r/w
Cache/13/page3.html
fork
mal.exe
read
send
exfiltration.com
passwords.txt

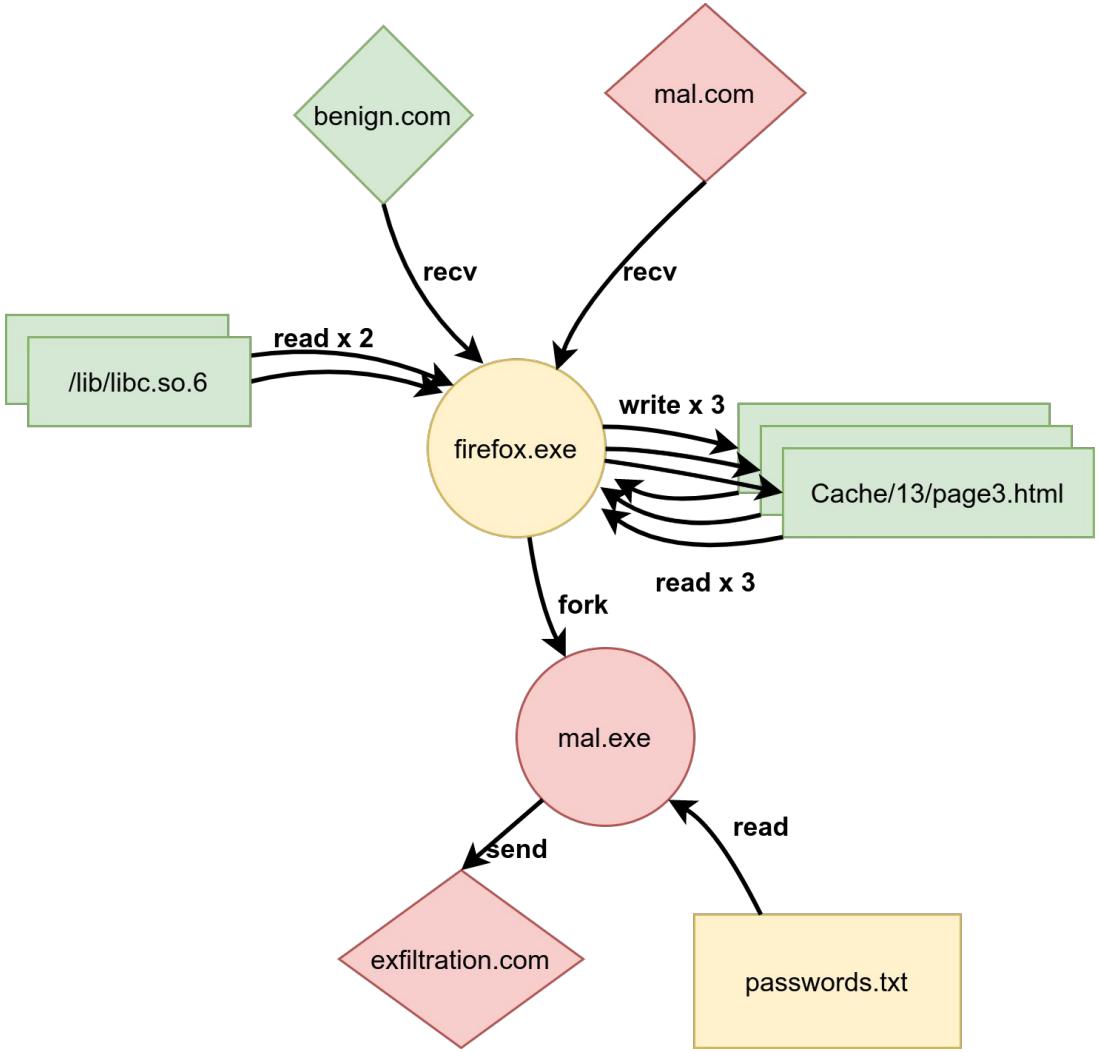
LogApprox


benign.com
mal.com
recv
recv
firefox.exe
write x 3
Cache/13/page3.html
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

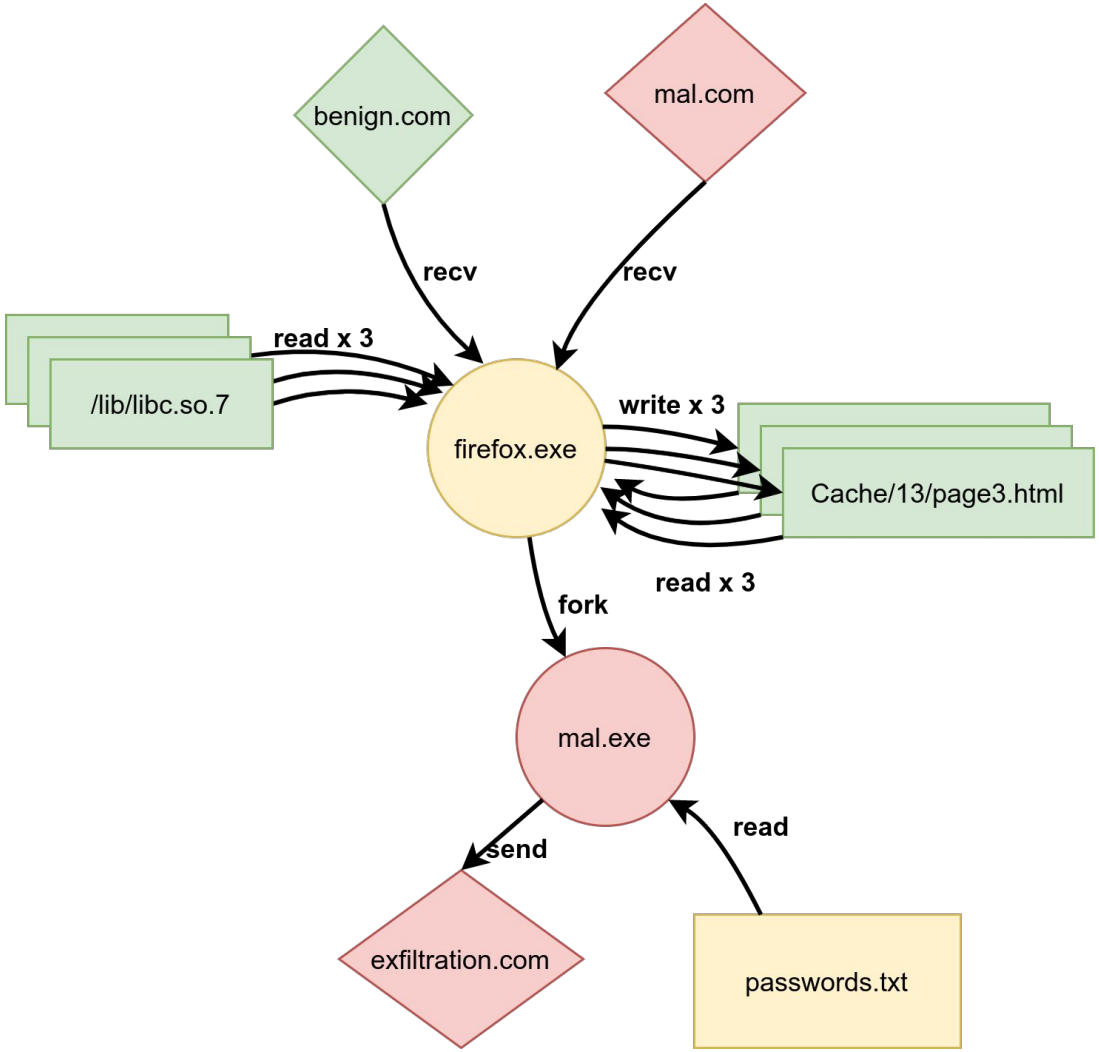
LogApprox


benign.com
mal.com
recv
recv
/lib/libc.so.1
read x 1
firefox.exe
write x 3
Cache/13/page3.html
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

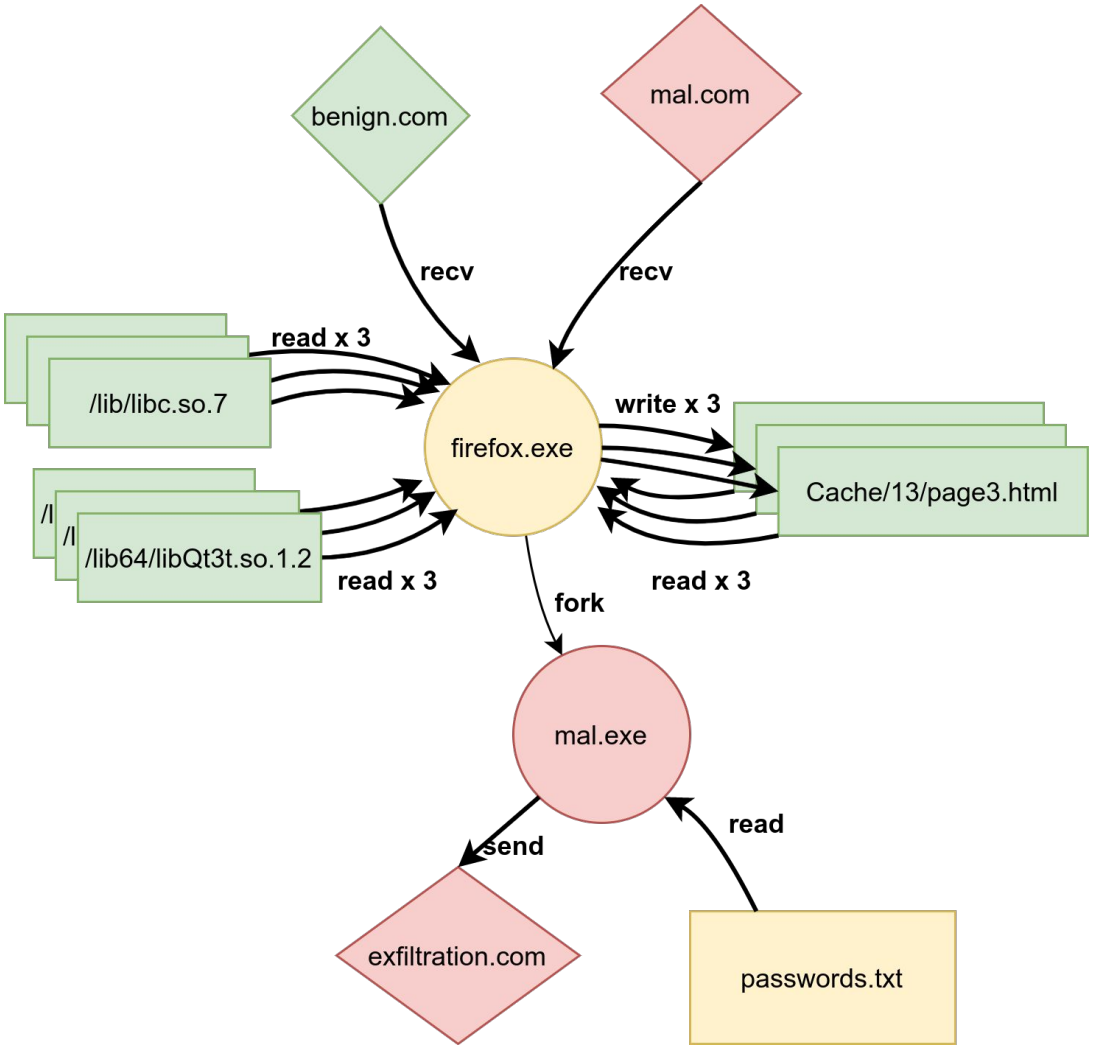
LogApprox


benign.com
mal.com
recv
recv
read x 2
/lib/libc.so.6
firefox.exe
write x 3
Cache/13/page3.html
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

LogApprox


benign.com
mal.com
recv
recv
read x 3
/lib/libc.so.7
firefox.exe
write x 3
Cache/13/page3.html
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

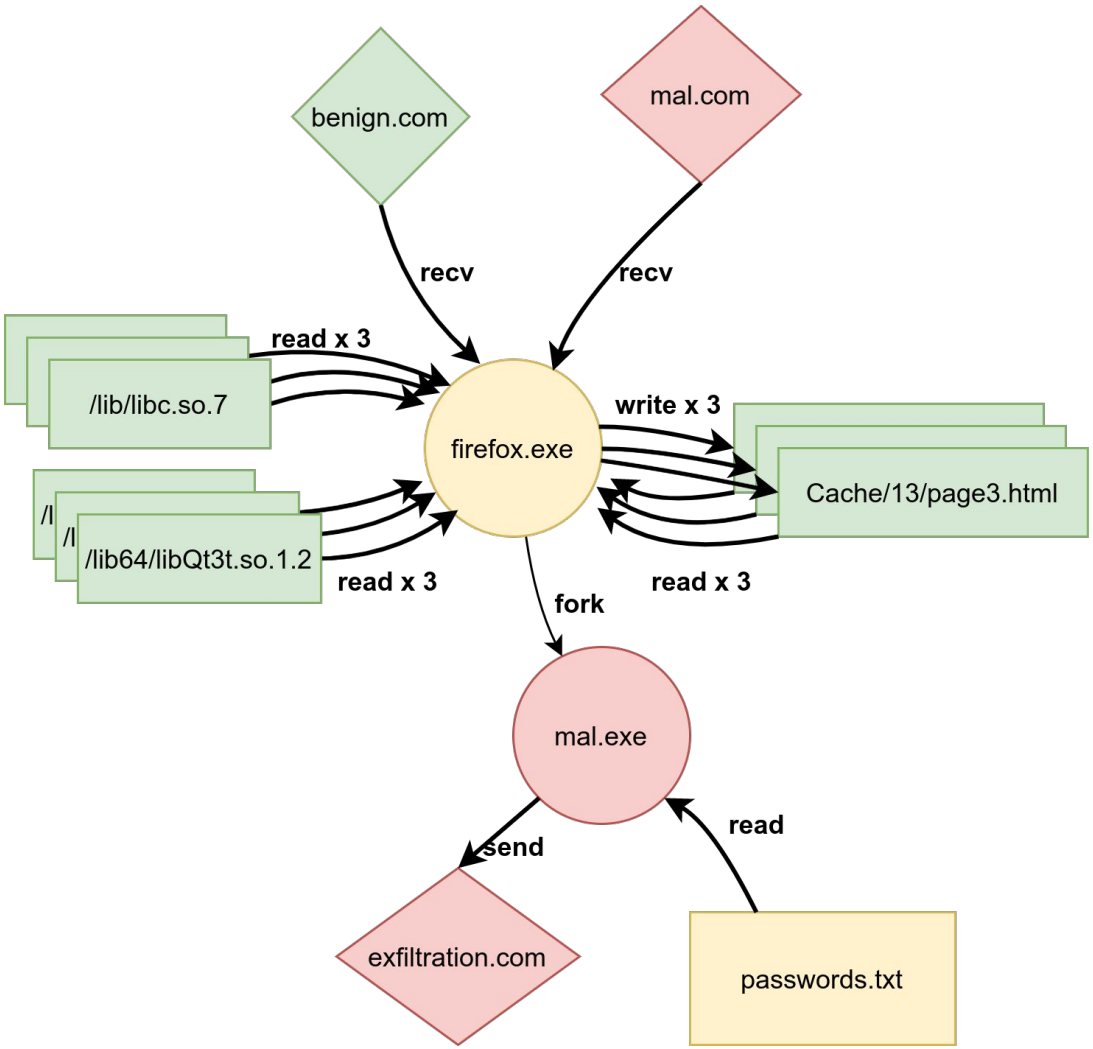
LogApprox


benign.com
mal.com
recv
recv
read x 3
/lib/libc.so.7
firefox.exe
write x 3
Cache/13/page3.html
/lib64/libQt3t.so.1.2
read x 3
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

LogApprox

Reduction Opportunities

- Most system events are file IO events!
  - Related files unable to be causally reduced



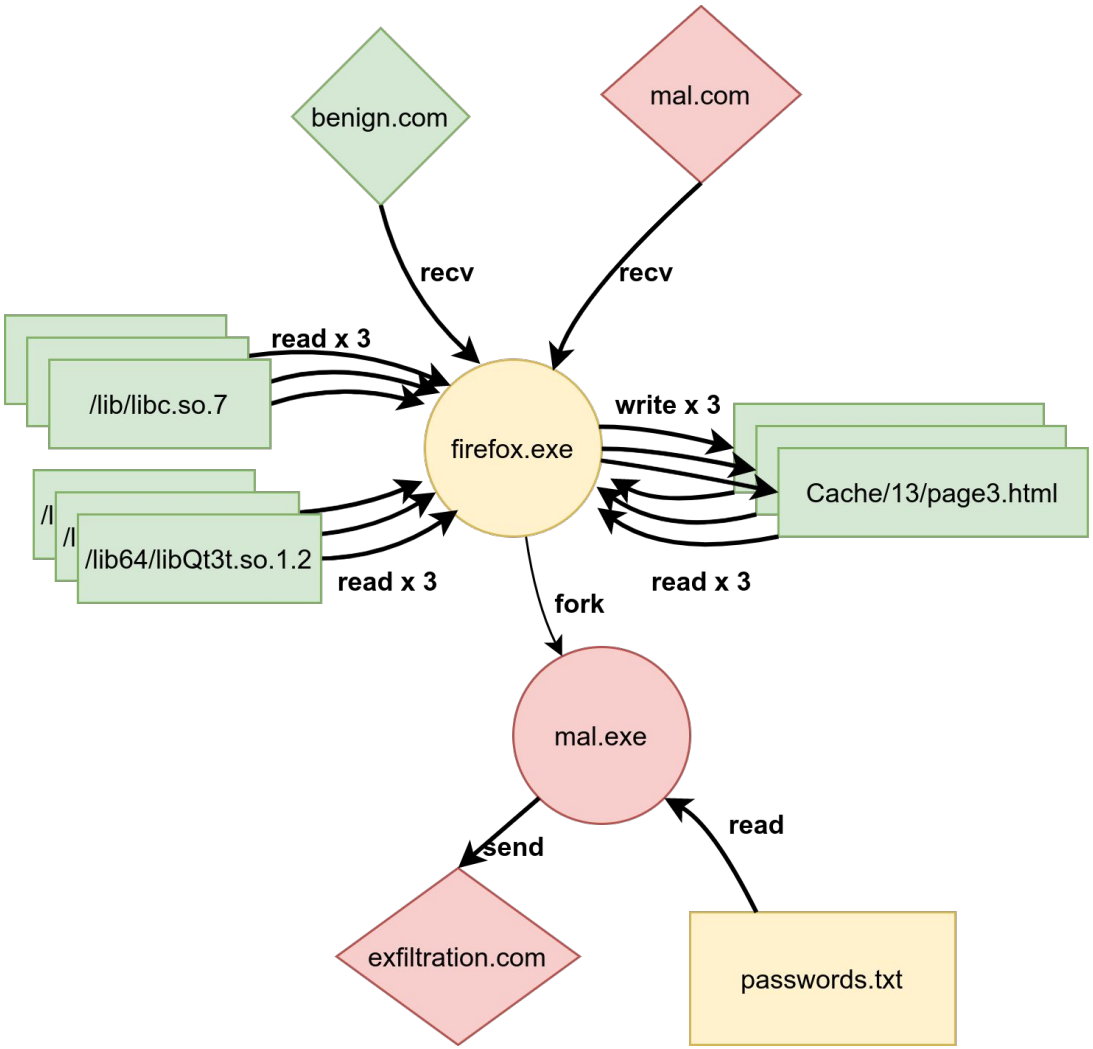
LogApprox

Reduction Opportunities

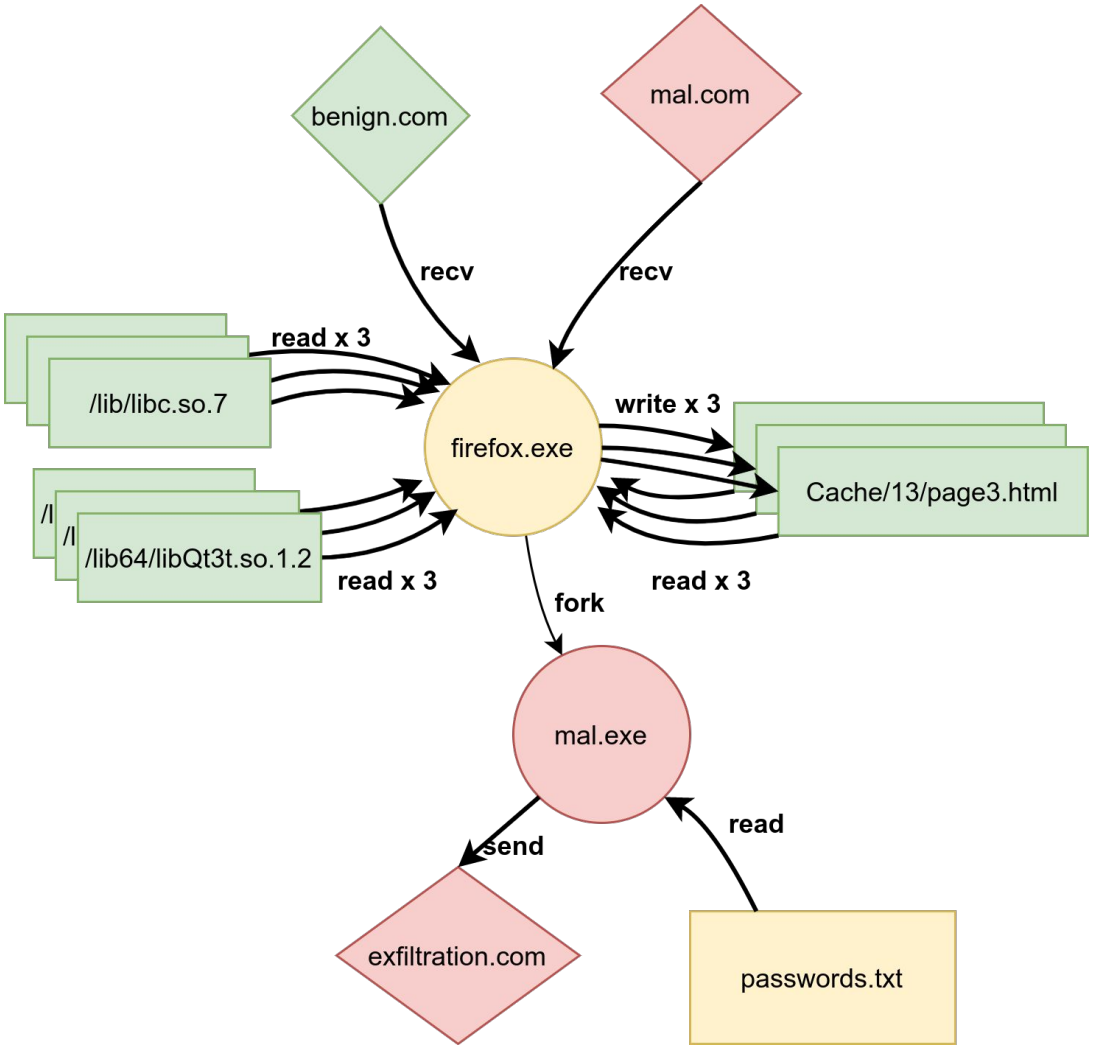
- Most system events are file IO events!
  - Related files unable to be causally reduced

LogApprox Reduction:

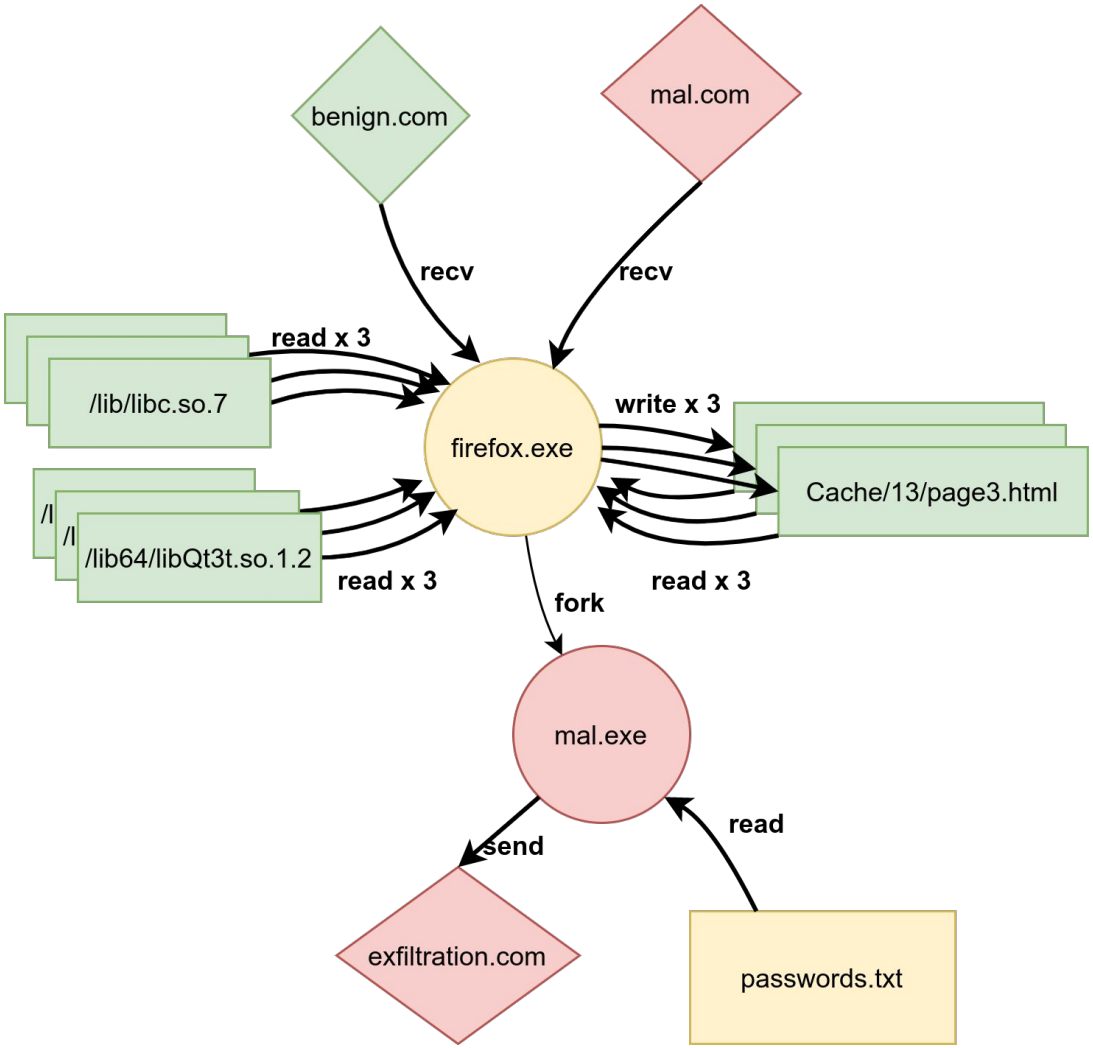
- Coalesce repetitive IO activity via <u>regexes</u>



LogApprox


benign.com
mal.com
recv
recv
read x 3
/lib/libc.so.7
firefox.exe
write x 3
Cache/13/page3.html
/lib64/libQt3t.so.1.2
read x 3
read x 3
fork
mal.exe
send
read
exfiltration.com
passwords.txt

LogApprox

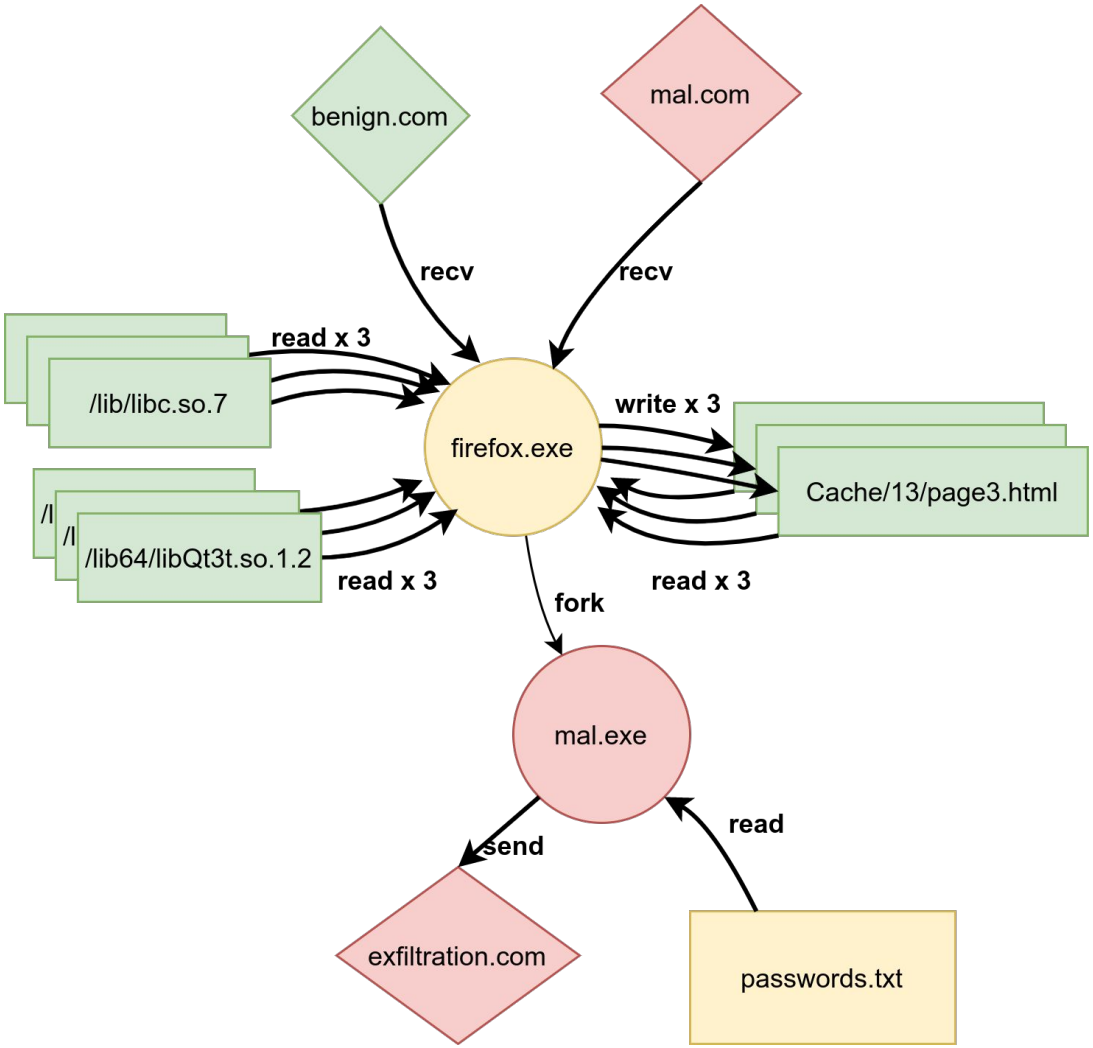


Filepaths for Firefox.exe

/Cache/11/page1.html
/Cache/12/page2.html
/Cache/13/page3.html
/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

LogApprox

Group 1:
/Cache/11/page1.html



Filepaths for Firefox.exe

/Cache/12/page2.html
/Cache/13/page3.html
/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

LogApprox

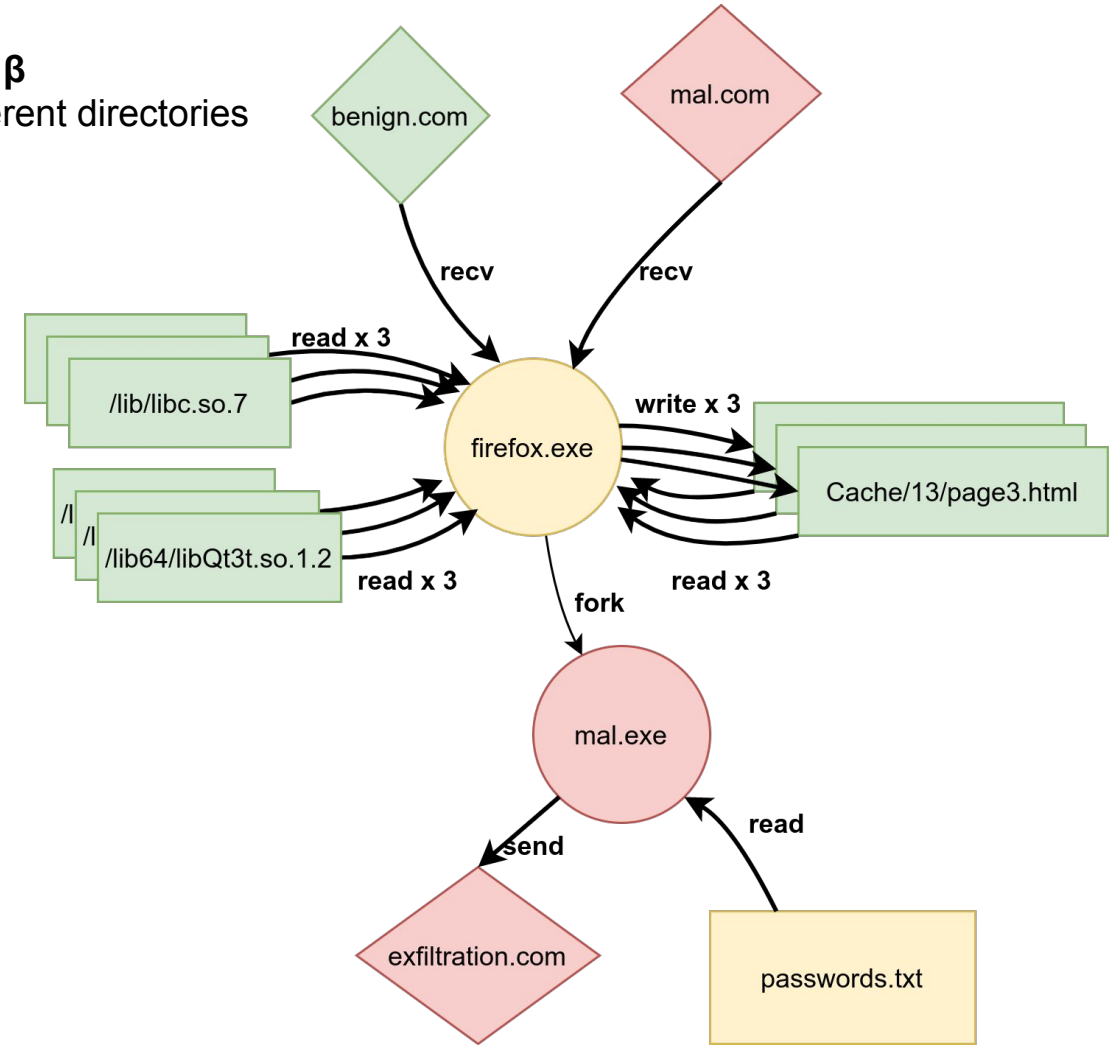
Group by:

Filename Similarity: **A**
Levenshtein Edit Distance

Path Distance: **β**
Number of different directories

Group 1:

/Cache/11/page1.html



Filepaths for Firefox.exe

/Cache/12/page2.html
/Cache/13/page3.html
/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

LogApprox

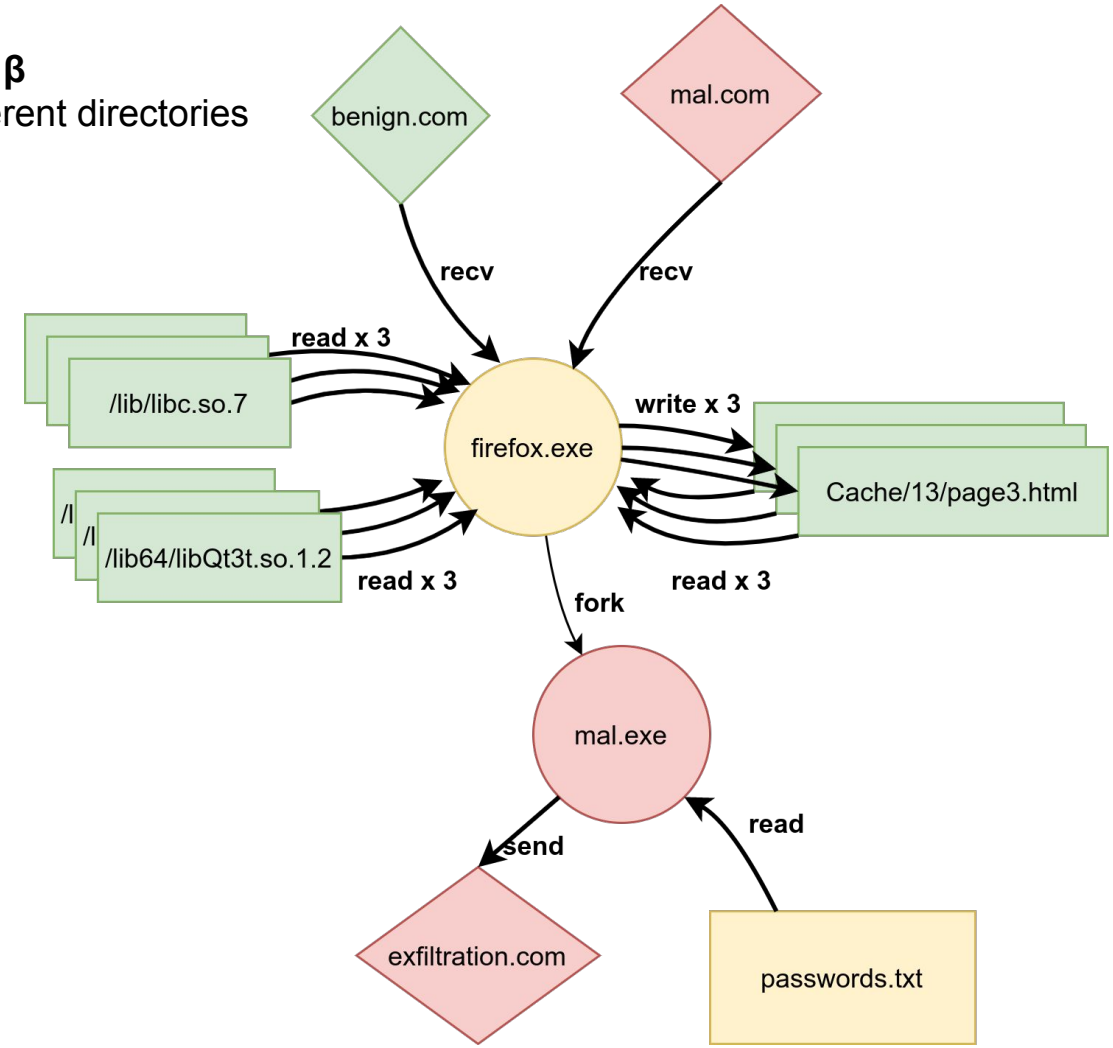
Group by:

Filename Similarity: **A**
Levenshtein Edit Distance

Path Distance: **β**
Number of different directories

Group 1:

/Cache/11/page1.html
/Cache/12/page2.html
/Cache/13/page3.html



Filepaths for Firefox.exe

/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

LogApprox

Group by:

Filename Similarity: **A**
Levenshtein Edit Distance

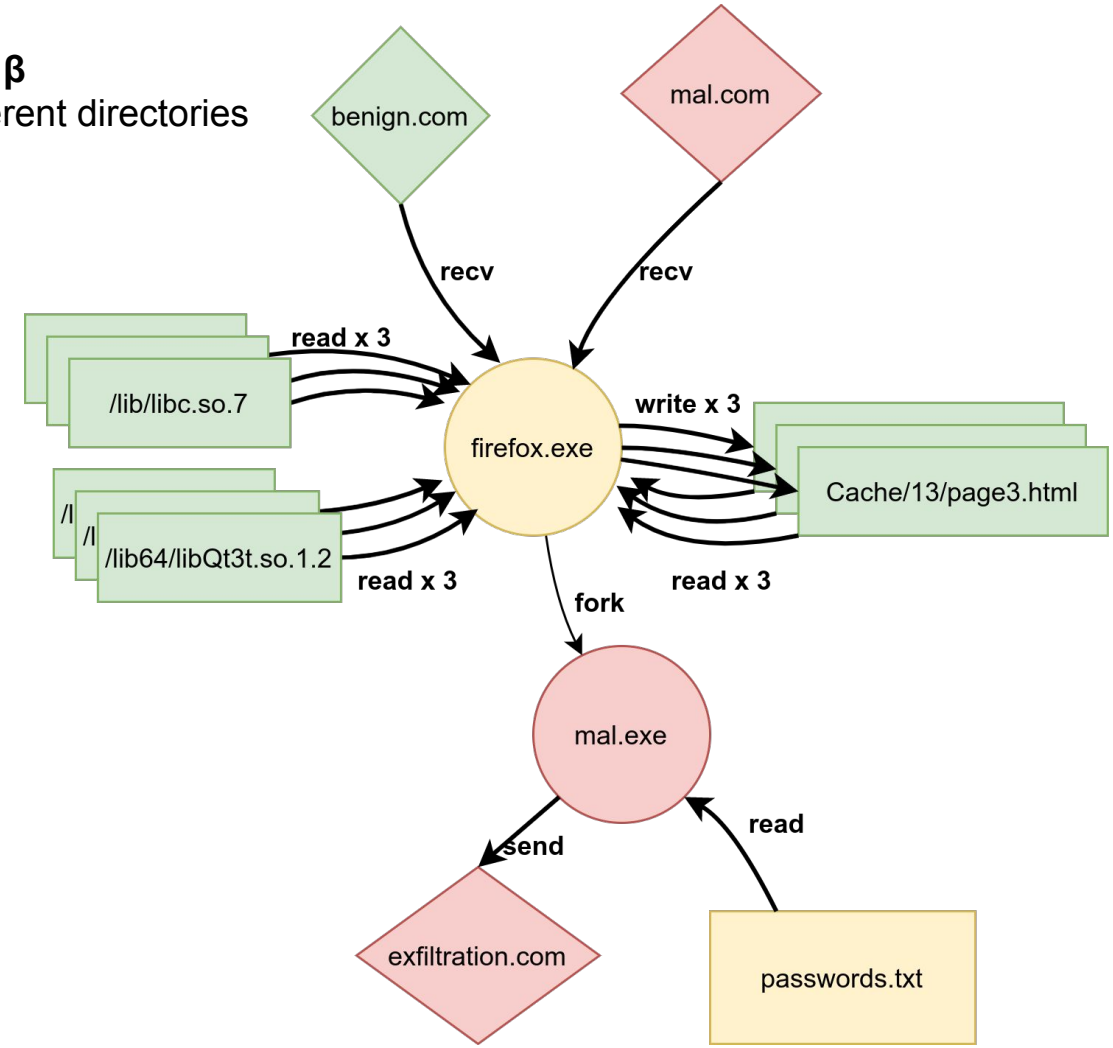
Path Distance: **β**
Number of different directories

Group 1:

/Cache/11/page1.html
/Cache/12/page2.html
/Cache/13/page3.html

Group 2:

/lib/libc.so.1



Filepaths for Firefox.exe

/lib/libc.so.6
/lib/libc.so.7
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

LogApprox

Group by:

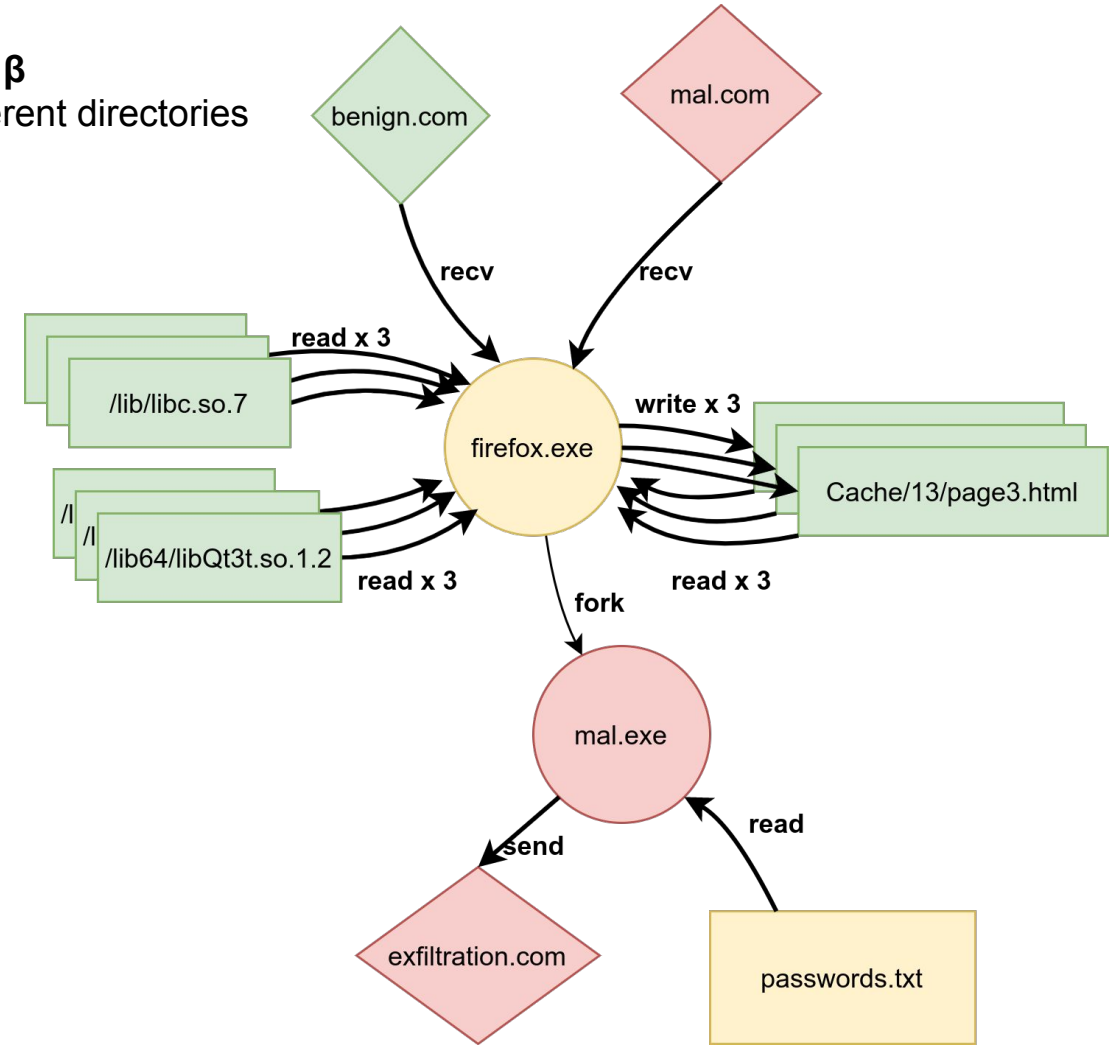
Filename Similarity: **A**
Levenshtein Edit Distance

Path Distance: **β**
Number of different directories

Group 1:
/Cache/11/page1.html
/Cache/12/page2.html
/Cache/13/page3.html

Group 2:
/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7

Group 3:
/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2



LogApprox

Group by:

Filename Similarity: **A**
Levenshtein Edit Distance

Path Distance: **β**
Number of different directories

Group 1:

/Cache/11/page1.html
/Cache/12/page2.html
/Cache/13/page3.html

/Cache/*/page*

Group 2:

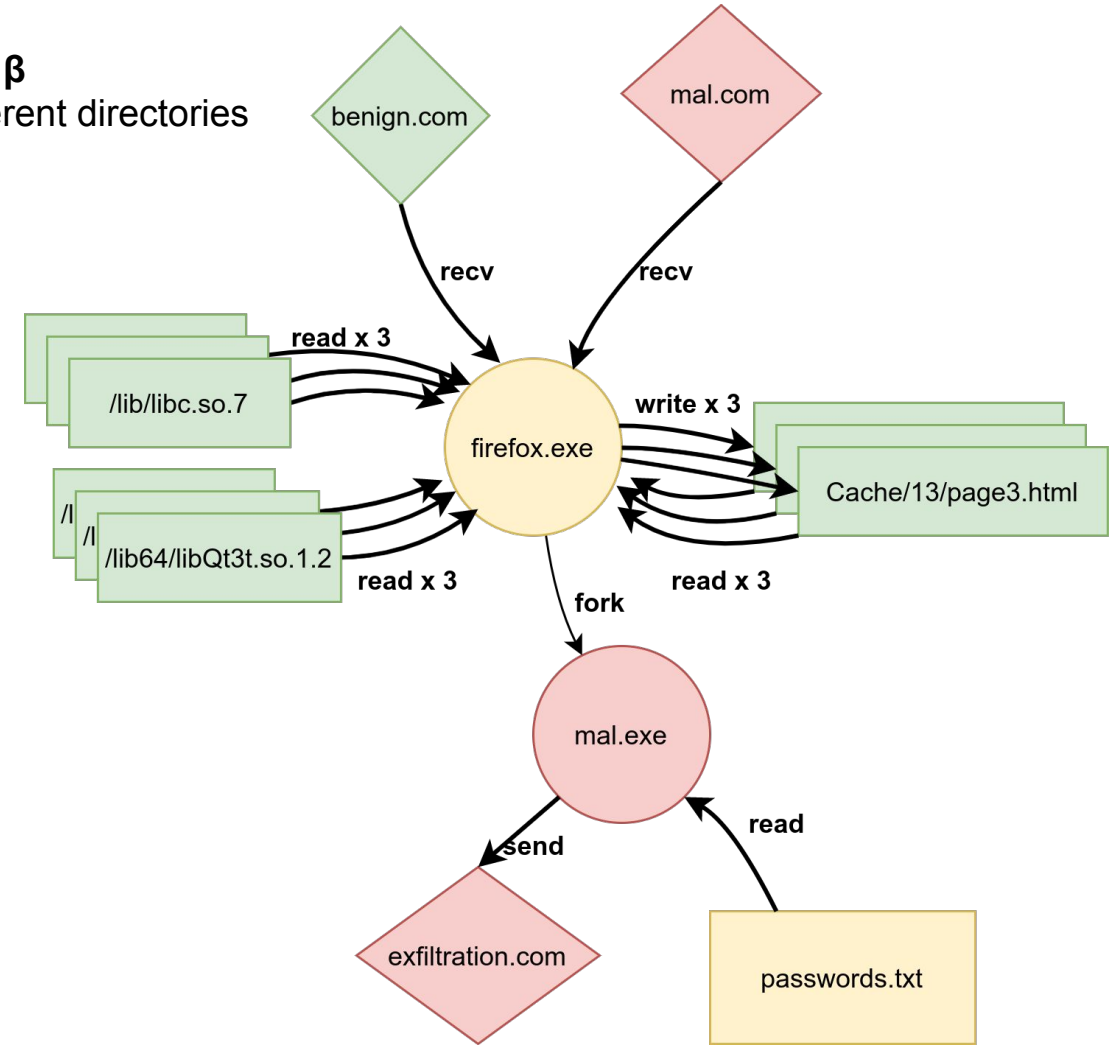
/lib/libc.so.1
/lib/libc.so.6
/lib/libc.so.7

/lib/libc.so.*

Group 3:

/lib64/libQt3t.so.1
/lib64/libQt3t.so.1.1
/lib64/libQt3t.so.1.2

/lib64/libQt3t.so.1*



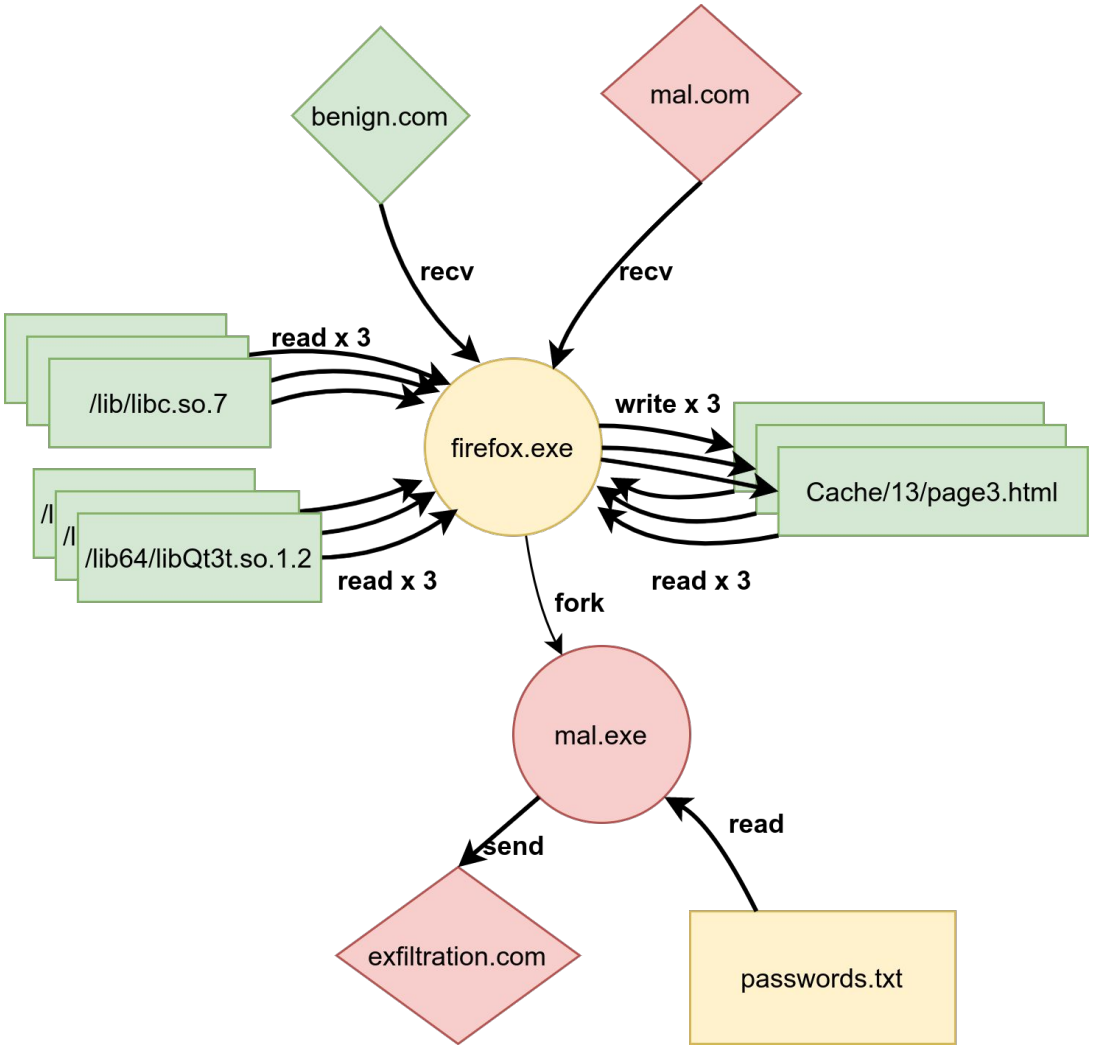
LogApprox

Firefox IO Templates:

/Cache/*/page*

/lib/libc.so.*

/lib64/libQt3t.so.1*



LogApprox

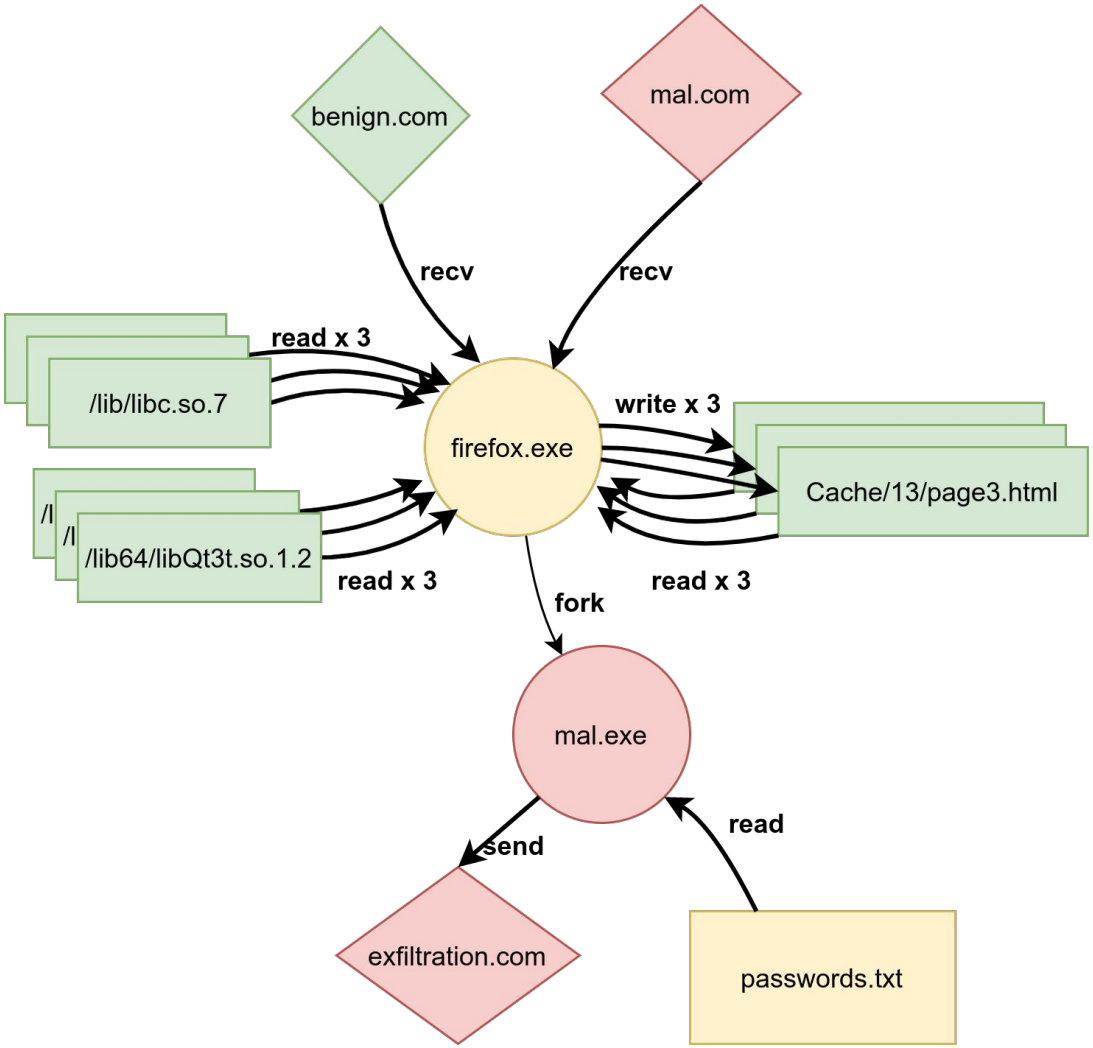
Firefox IO Templates:

/Cache/*/page*

/lib/libc.so.*

/lib64/libQt3t.so.1*

APPLY



LogApprox

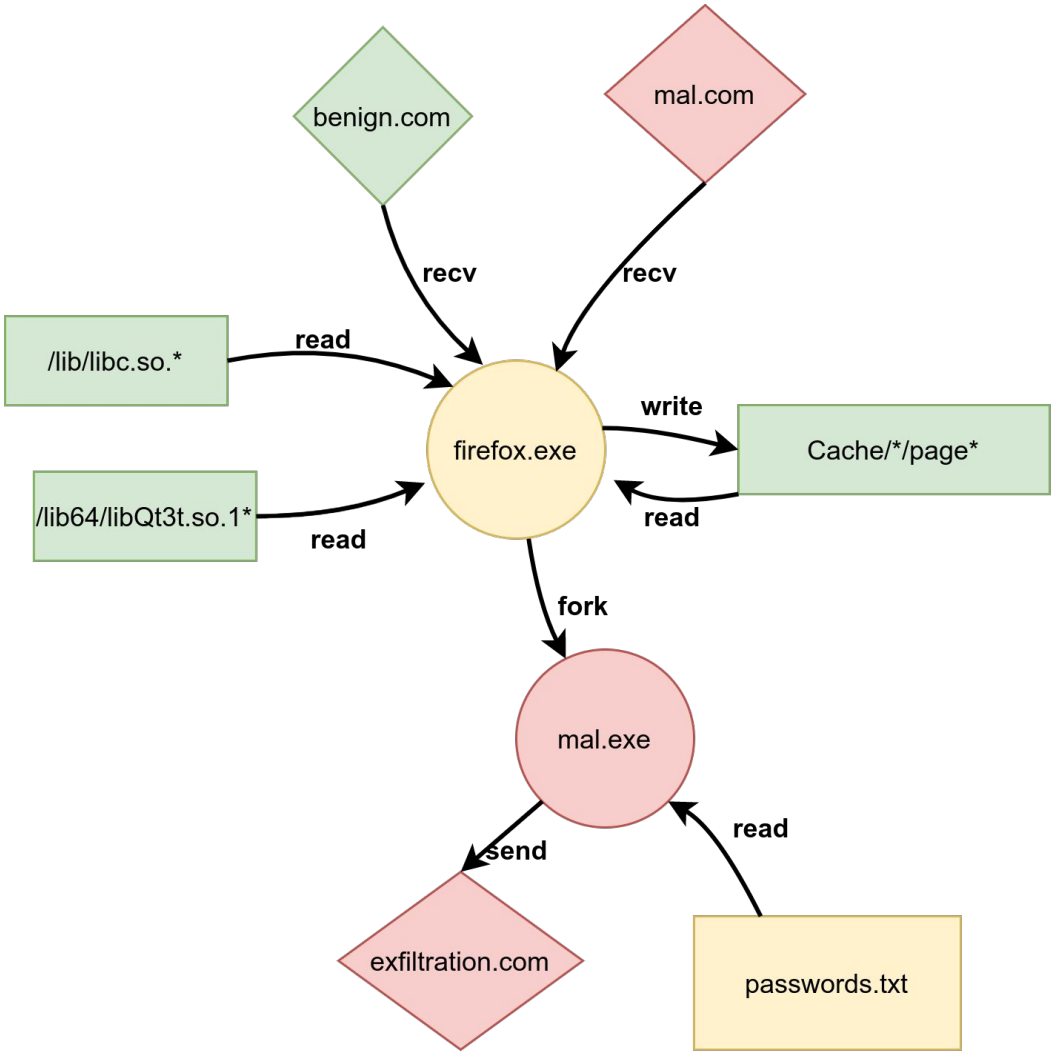
Firefox IO Templates:

/Cache/*/page*

/lib/libc.so.*

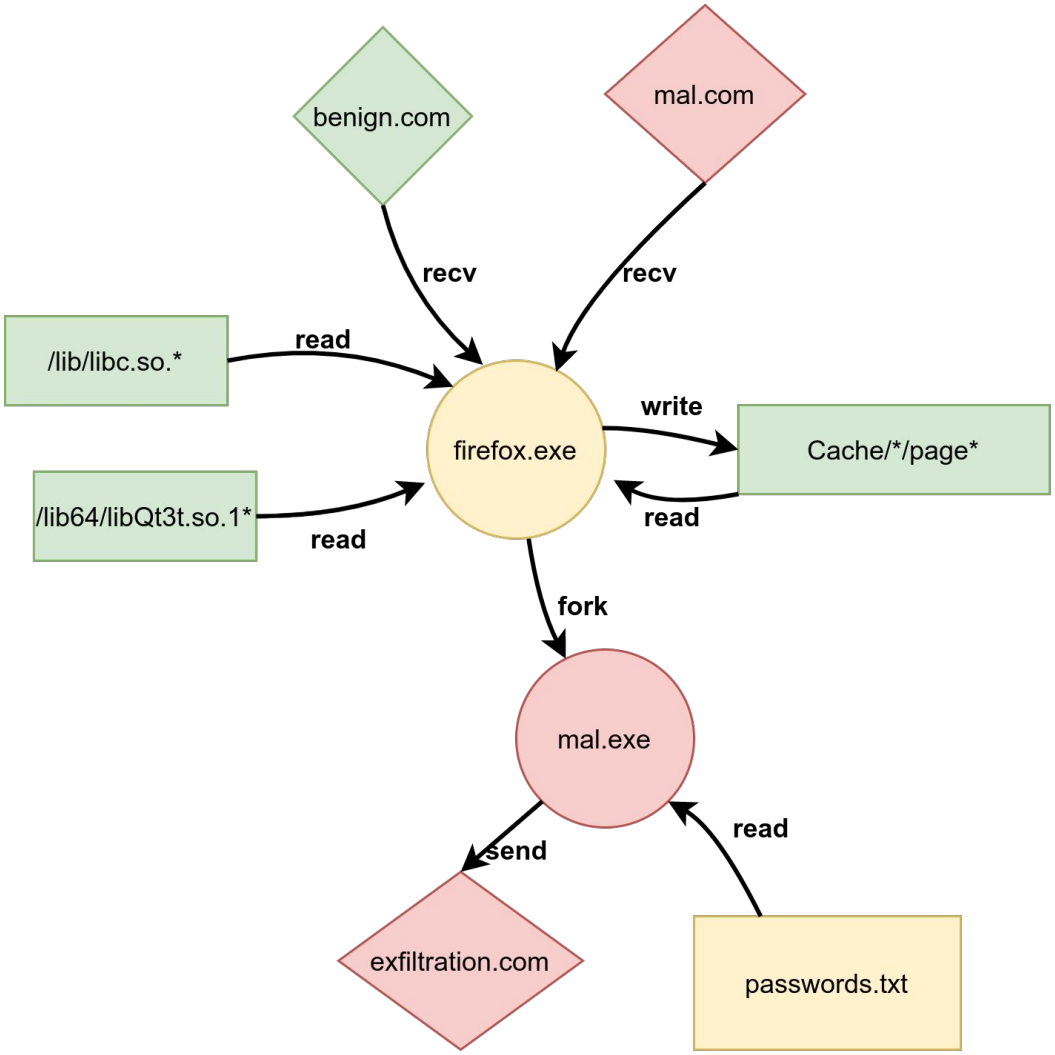
/lib64/libQt3t.so.1*

APPLY



LogApprox

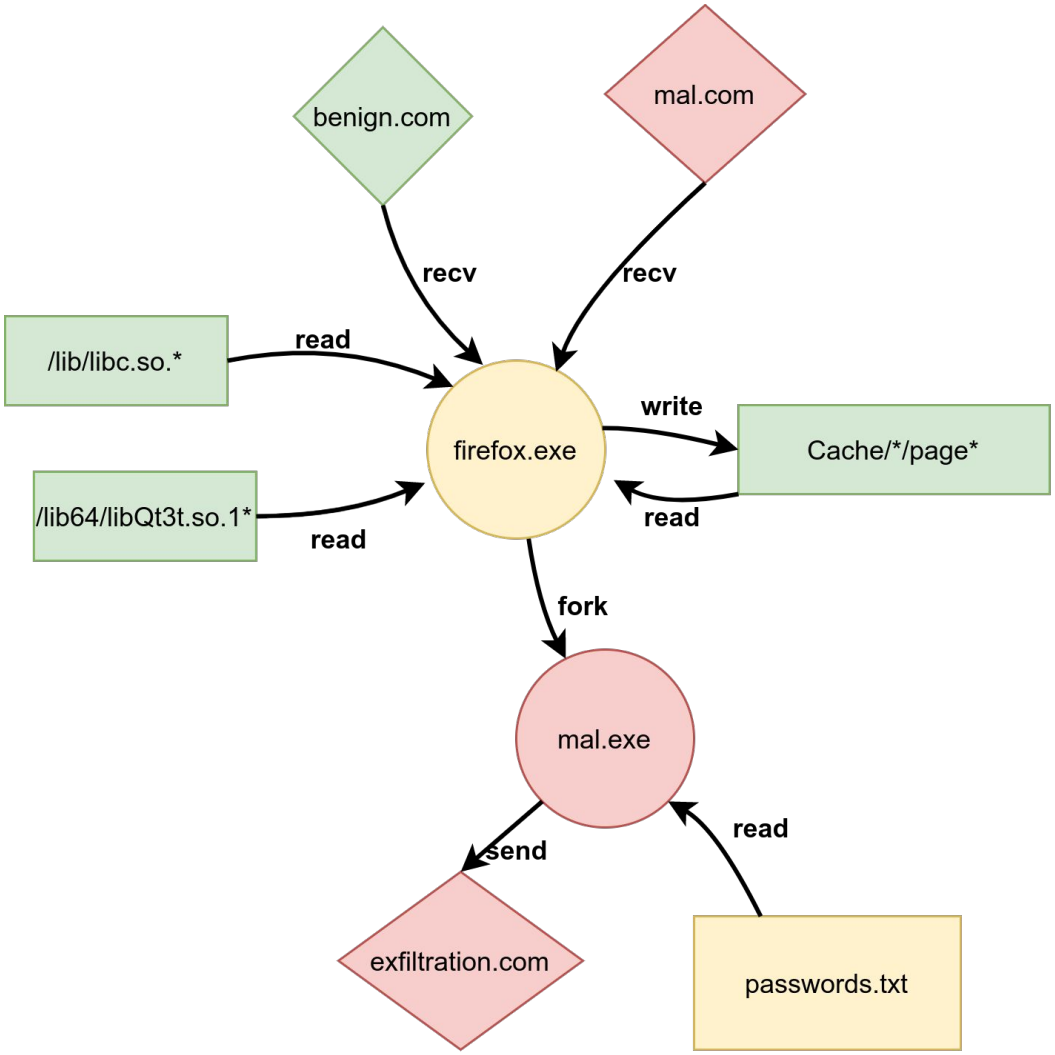
Properties



LogApprox

Properties

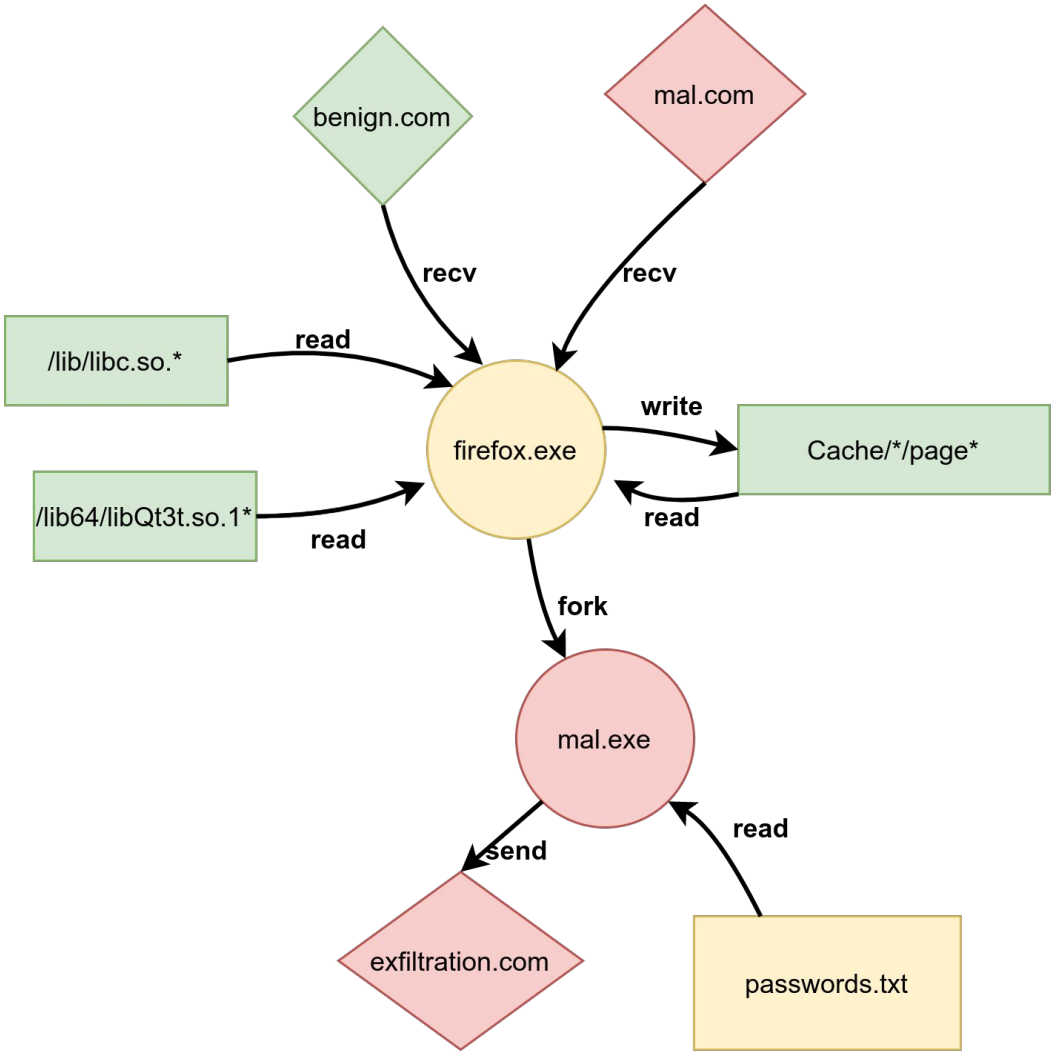
- Only reduces repetitive ***local*** file IO



LogApprox

Properties

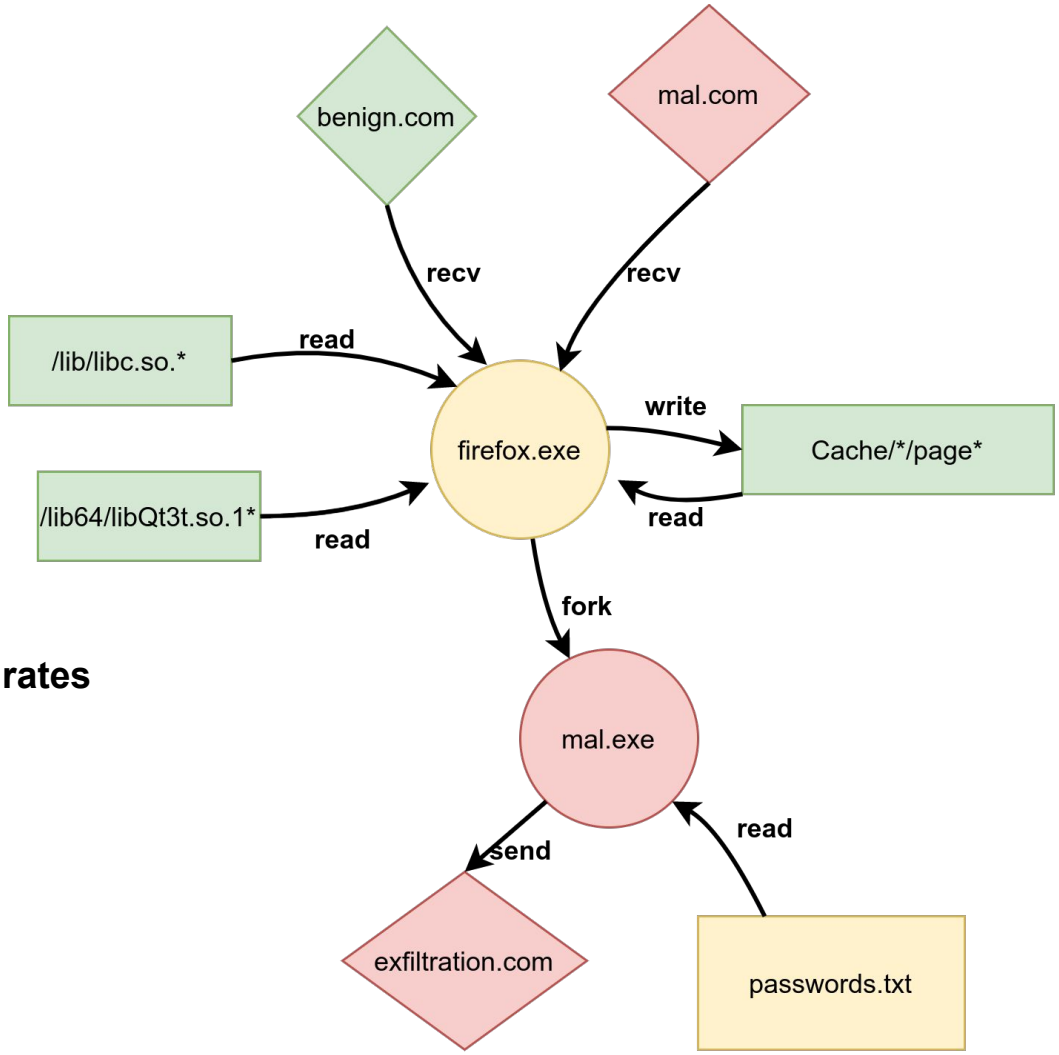
- Only reduces repetitive ***local*** file IO
- IO is only ever ***approximated***



LogApprox

Properties

- Only reduces repetitive ***local*** file IO
- IO is only ever ***approximated***

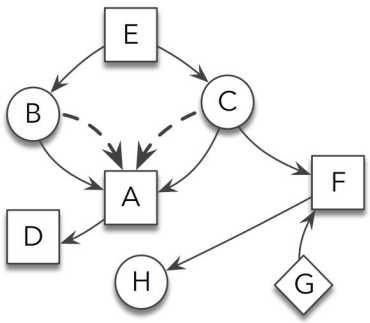


LogApprox can receive high reduction rates while preserving anomalous behavior!

Evaluation against Exemplar Reduction Techniques

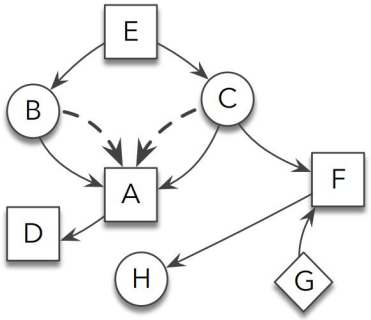
Evaluation against Exemplar Reduction Techniques

Causality-Preserving Reduction
by Xu et. al.

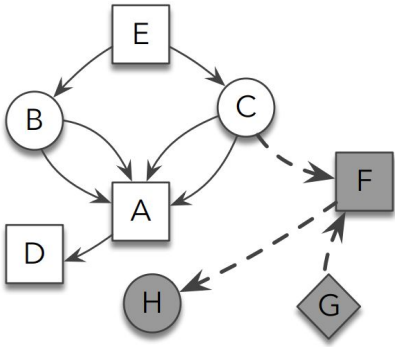


Evaluation against Exemplar Reduction Techniques

Causality-Preserving Reduction
by Xu et. al.

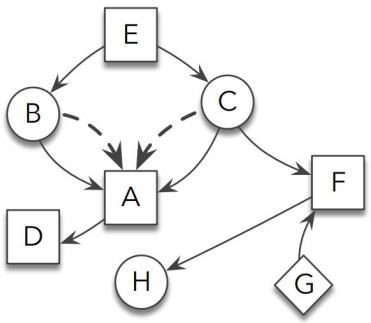


LogGC
by Lee et. al.

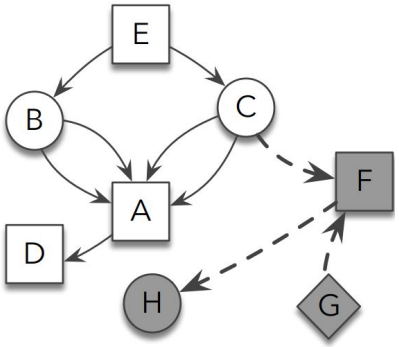


Evaluation against Exemplar Reduction Techniques

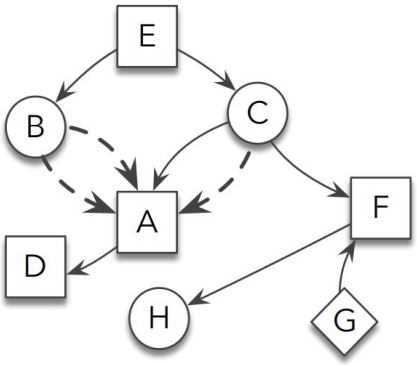
Causality-Preserving Reduction
by Xu et. al.



LogGC
by Lee et. al.

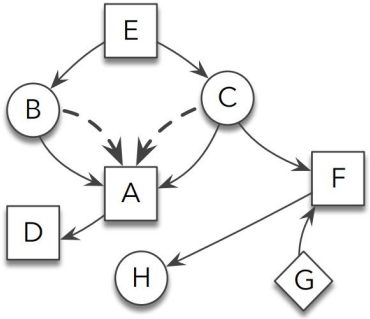


Full and Source Dependence Preserving Reduction
by Hossain et. al

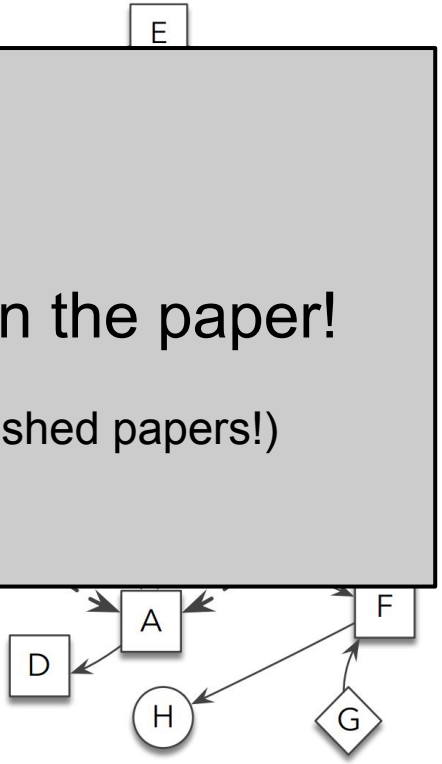


Evaluation against Exemplar Reduction Techniques

Causality-Preserving Reduction
by Xu et. al.



LogGC
by Lee et. al.



Details of each algorithm in the paper!

(and within their respectively published papers!)

Full and Sou
Reduction
by Hossein et. al

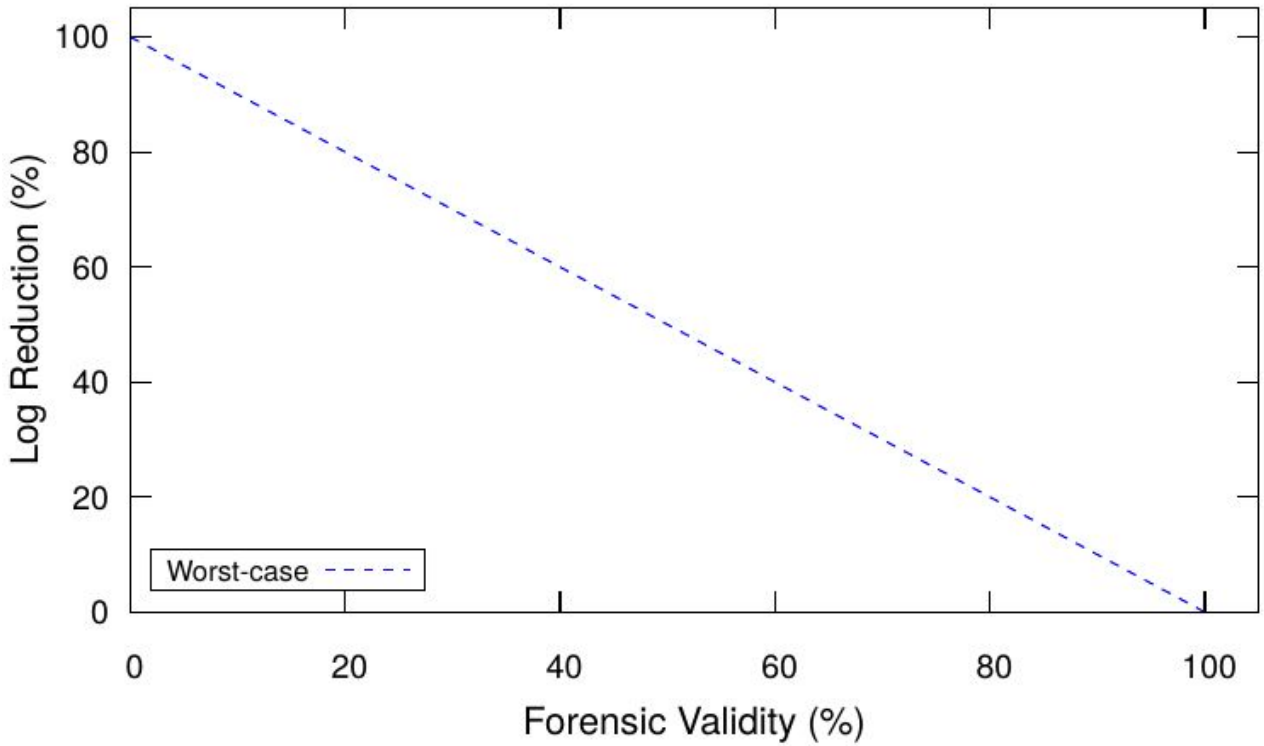
Forensic Evaluation

Curated set of real-world vulnerabilities and exploits:

- unrealircd[1] : IRC Server
- vsftpd[2] : FTP Server
- webmin[3] : System Configuration Tool
- Wordpress[4] : Content Management System
- PHP Webshell[5] : Generic Web Server
- Firefox[6] : Web Browser

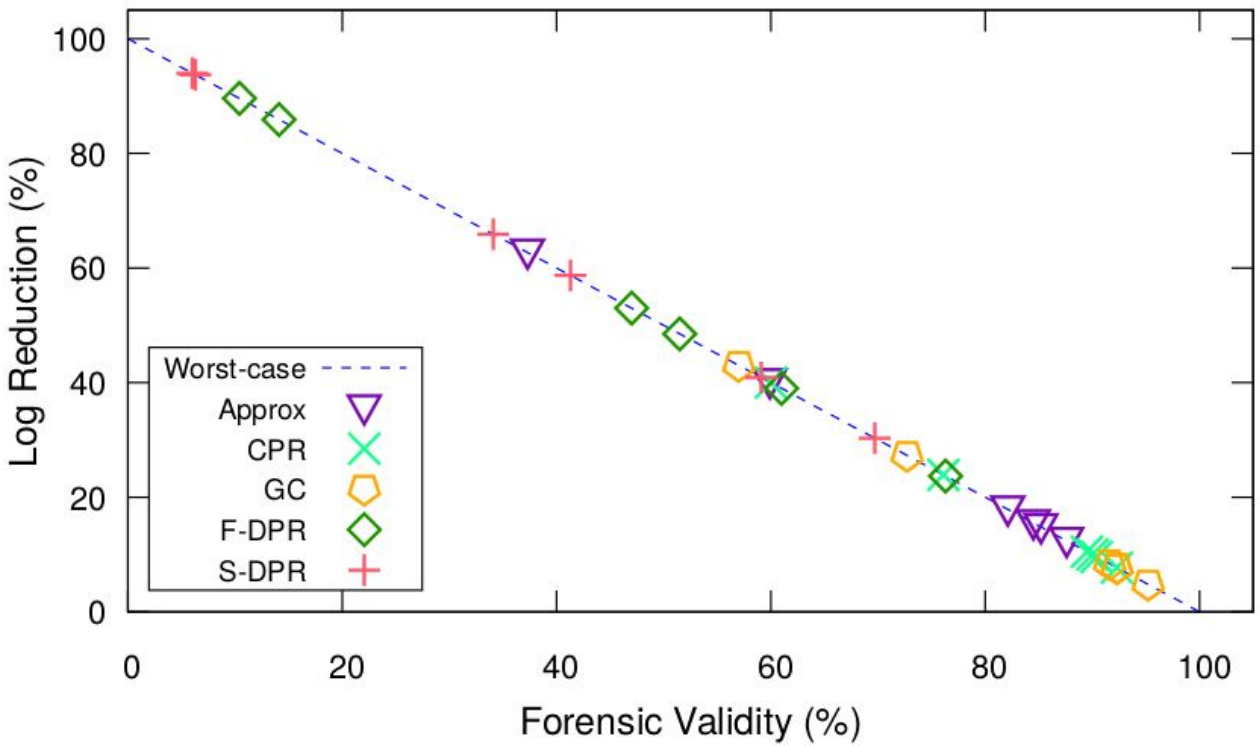
[1] Exploit-DB. 2010. UnrealIRCd 3.2.8.1 - Backdoor Command Execution. [2] Exploit-DB. 2010. UnrealIRCd 3.2.8.1 - Backdoor Command Execution. [3] Exploit-DB. 2019. Webmin 1.920 - Unauthenticated Remote Code Execution [4] Rapid7. 2018. WordPress Admin Shell Upload. [5] Mitre, Server Software Component: Web Shell. Retrieved from https://attack.mitre.org/techniques/T1505/003/, 2019
[6] A. D. Keromytis, “Transparent computing engagement 3 data,” https://github.com/darpa-i2o/Transparent-Computing, 2018,

Results


100
80
60
40
20
0
Log Reduction (%)
Worst-case
0
20
40
60
80
100
Forensic Validity (%)

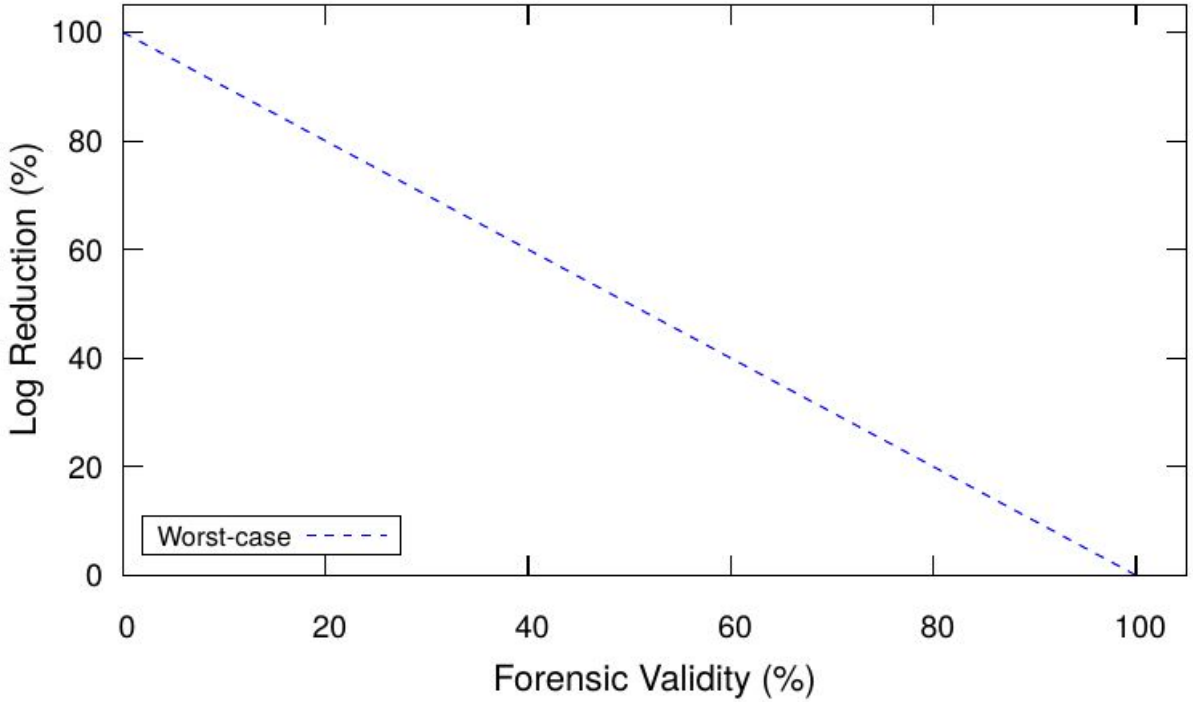
Results

Lossless Forensics



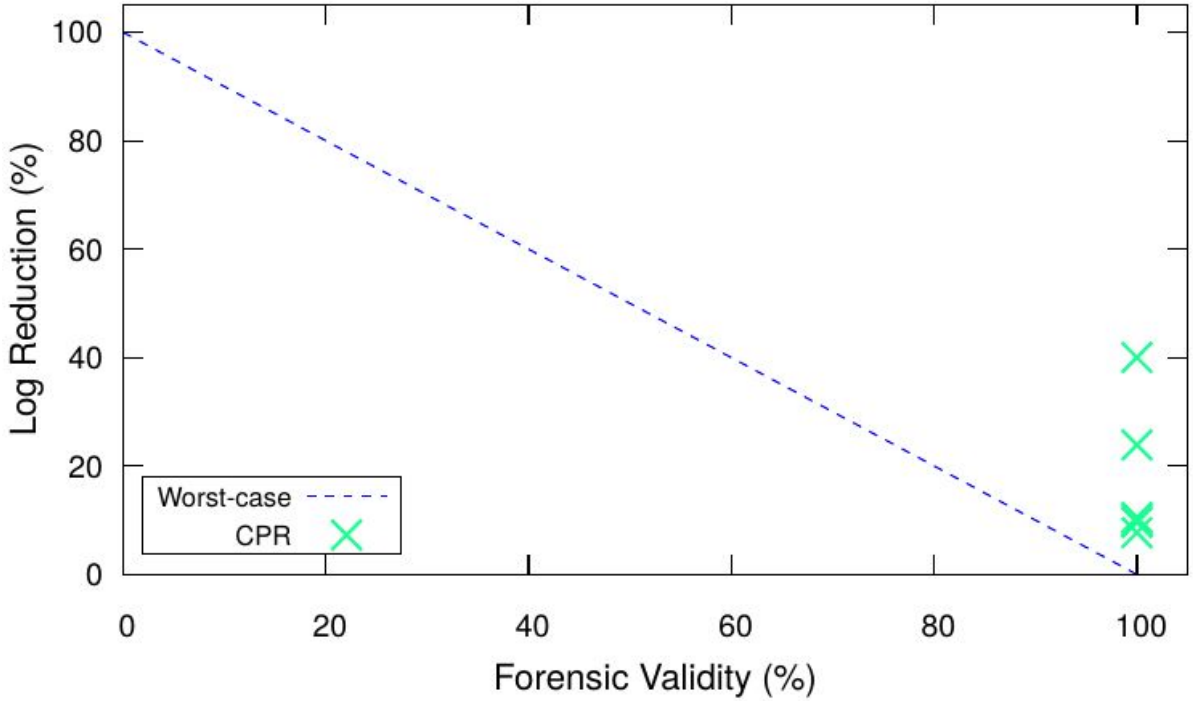
Results

Causality-Preserving Forensics
(all information flow)



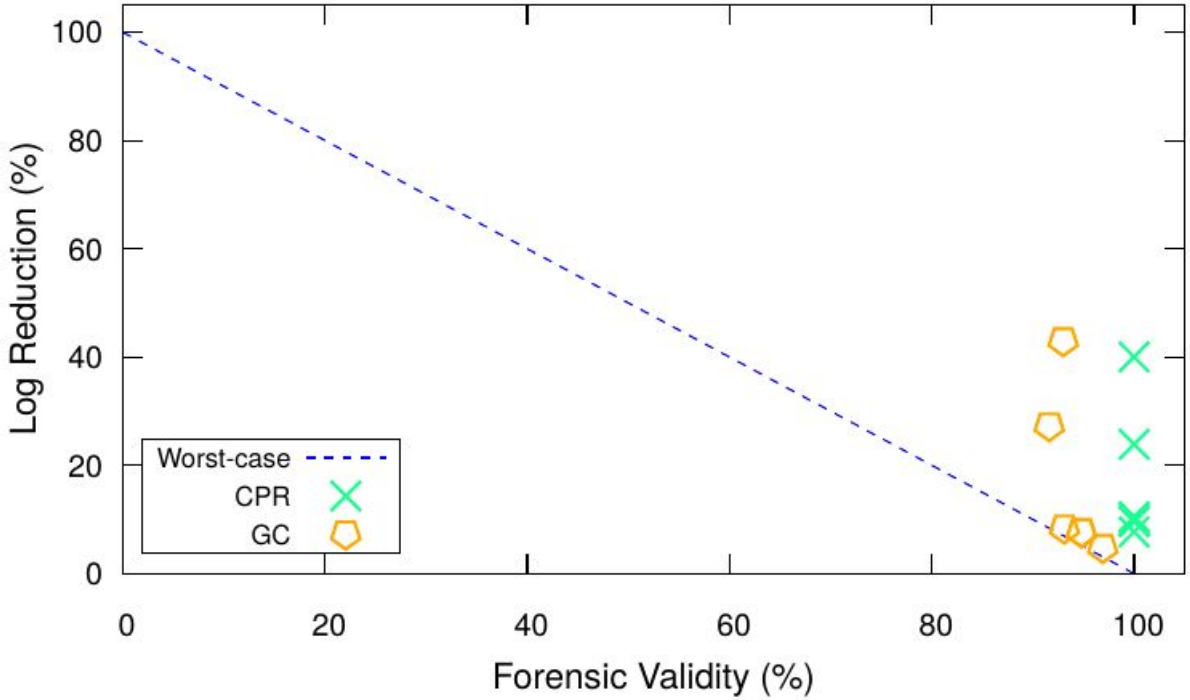
Results

Causality-Preserving Forensics
(all information flow)



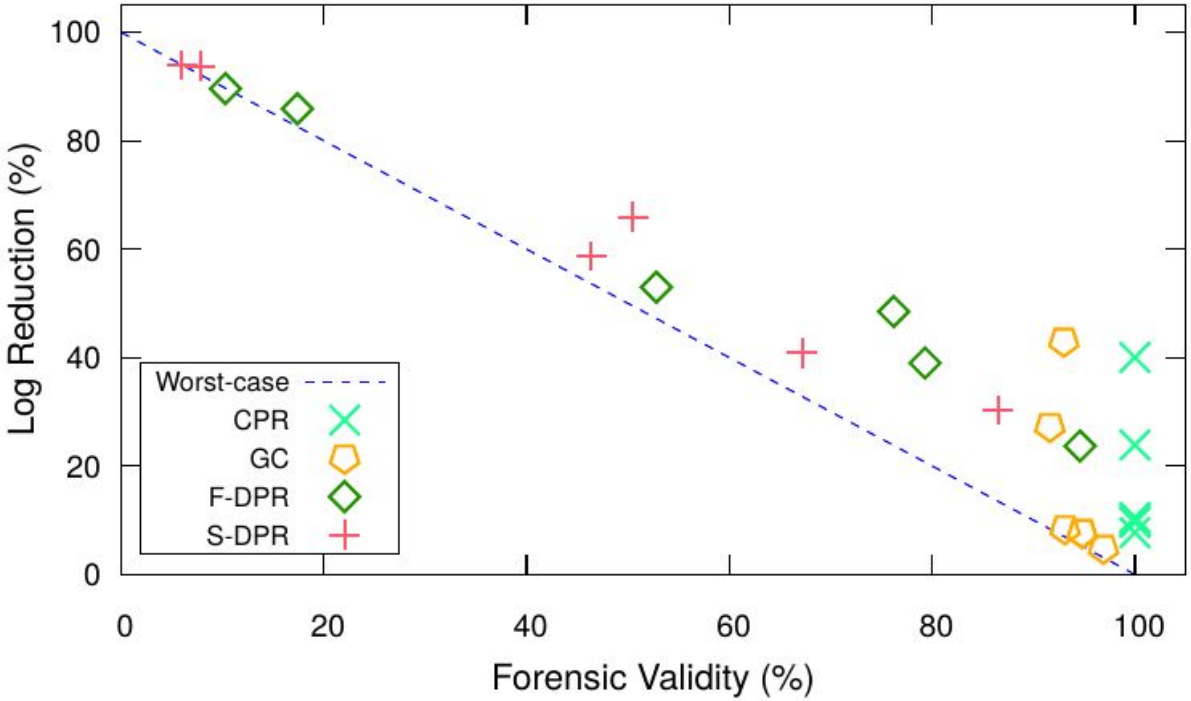
Results

Causality-Preserving Forensics
(all information flow)



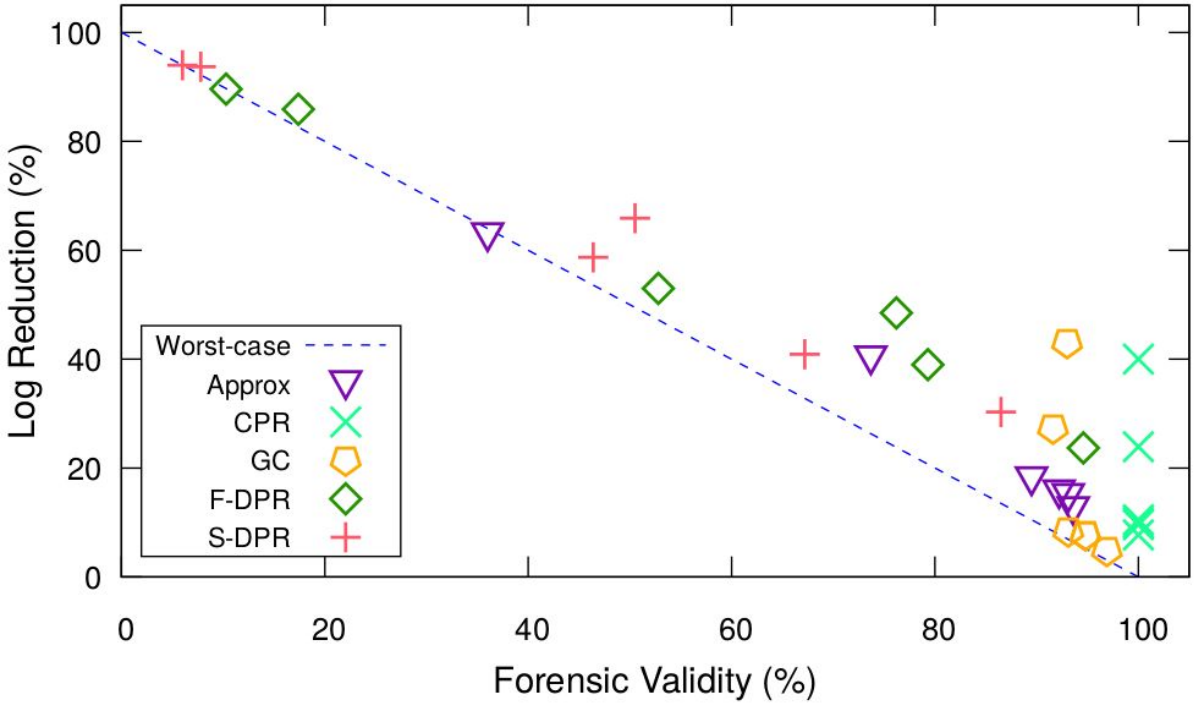
Results

Causality-Preserving Forensics
(all information flow)



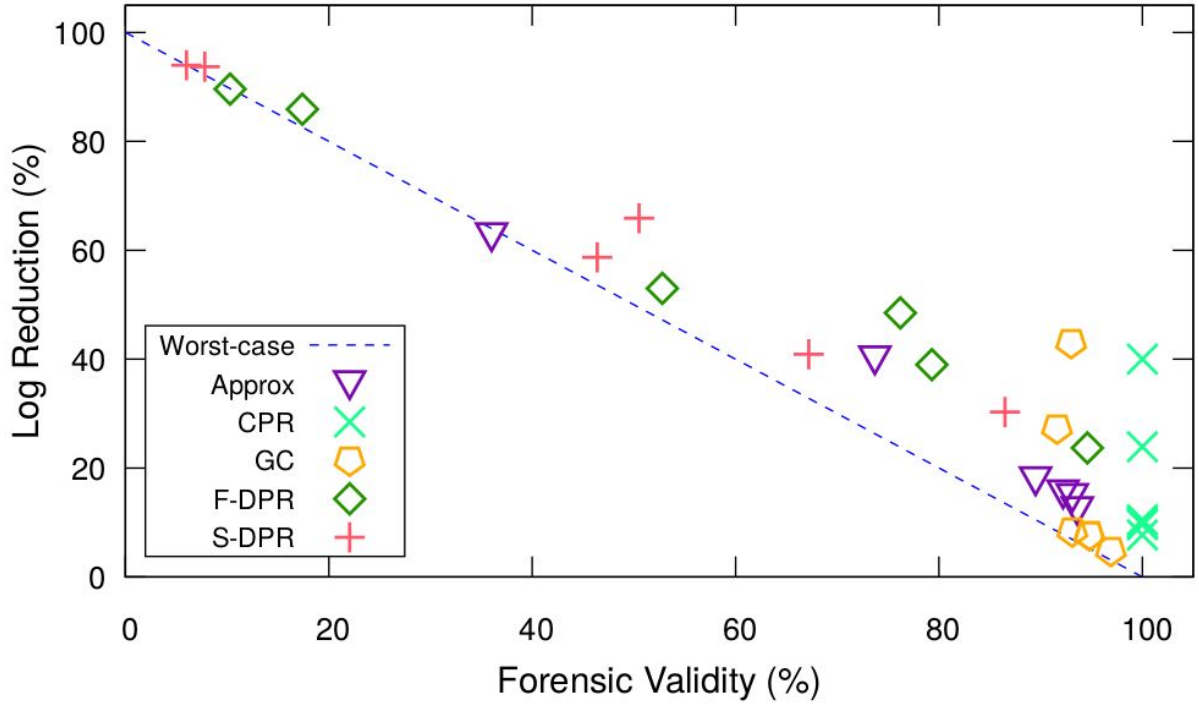
Results

Causality-Preserving Forensics
(all information flow)

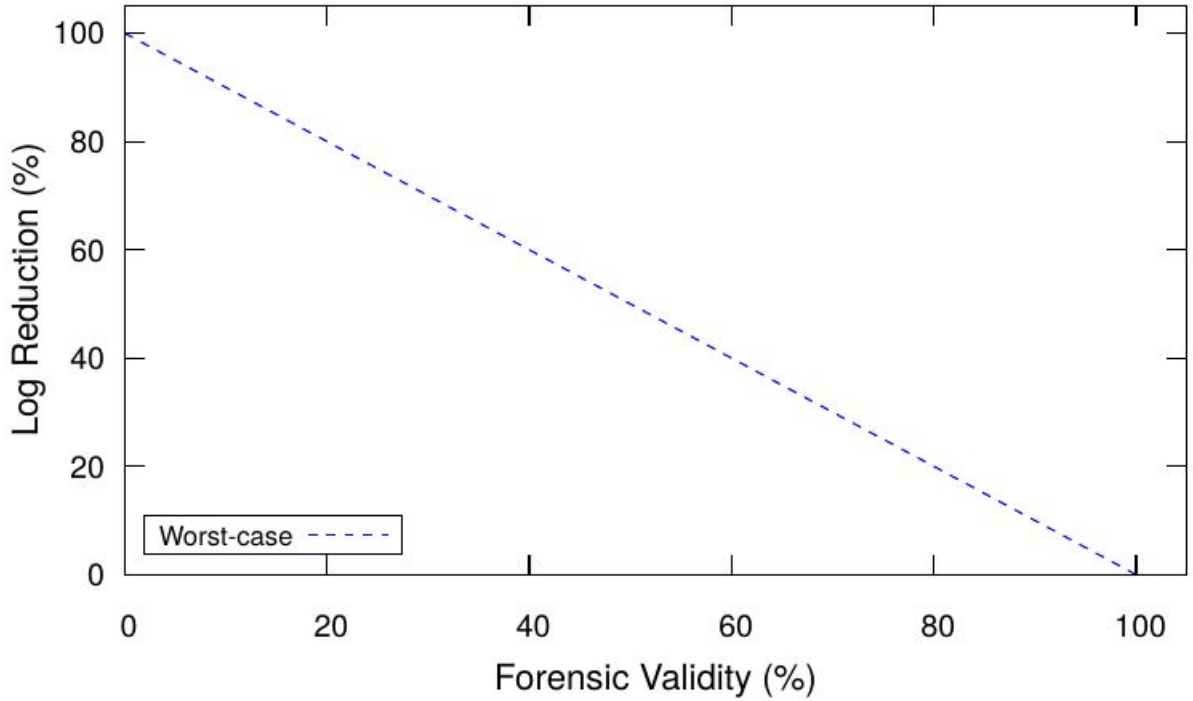


Results

Causality-Preserving Forensics
(all information flow)

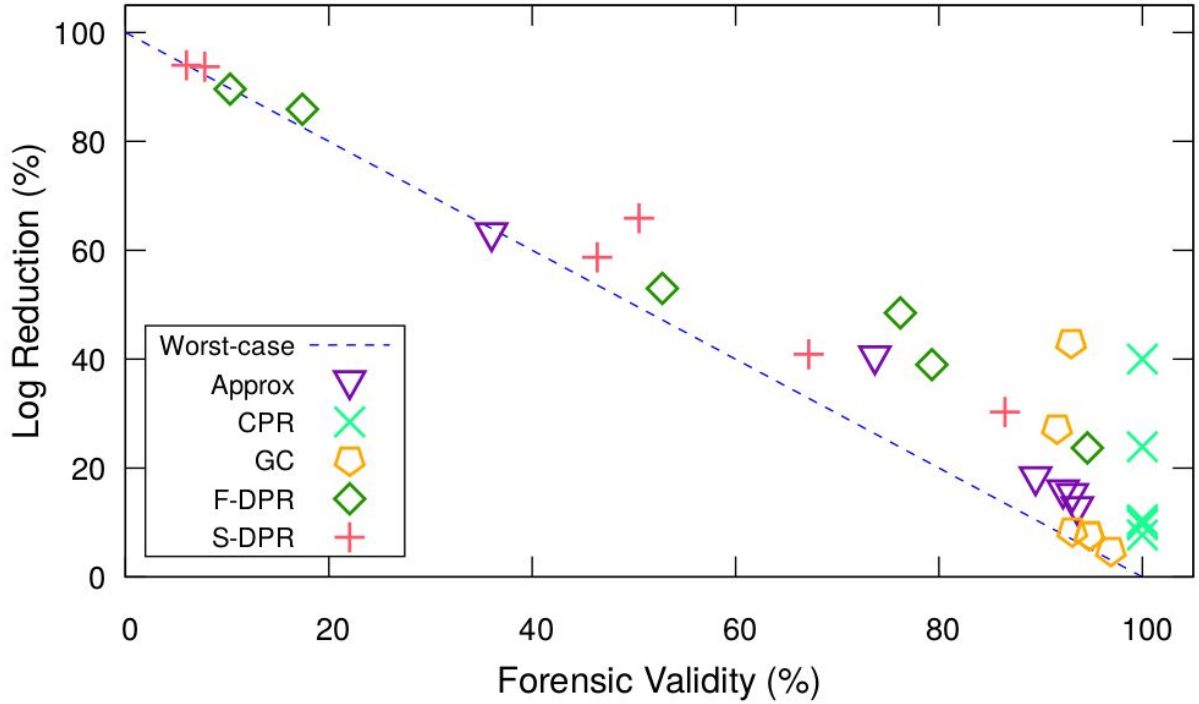


Attack-Preserving Forensics
(uniquely malicious information flow)

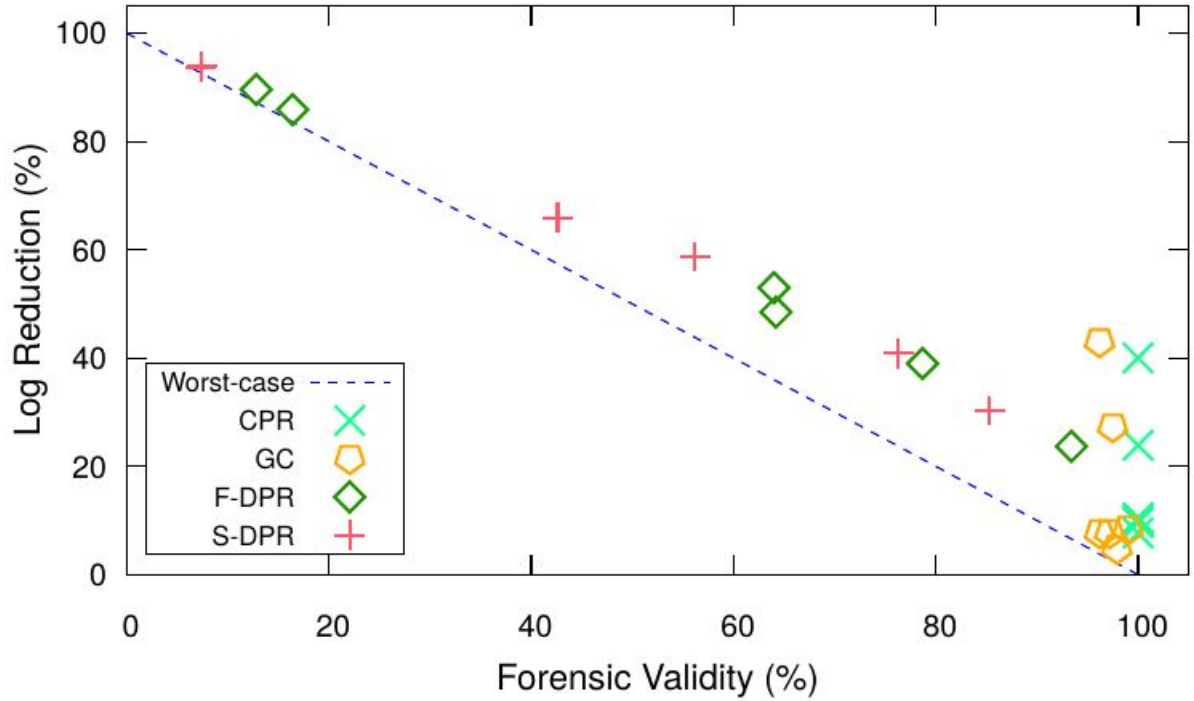


Results

Causality-Preserving Forensics
(all information flow)

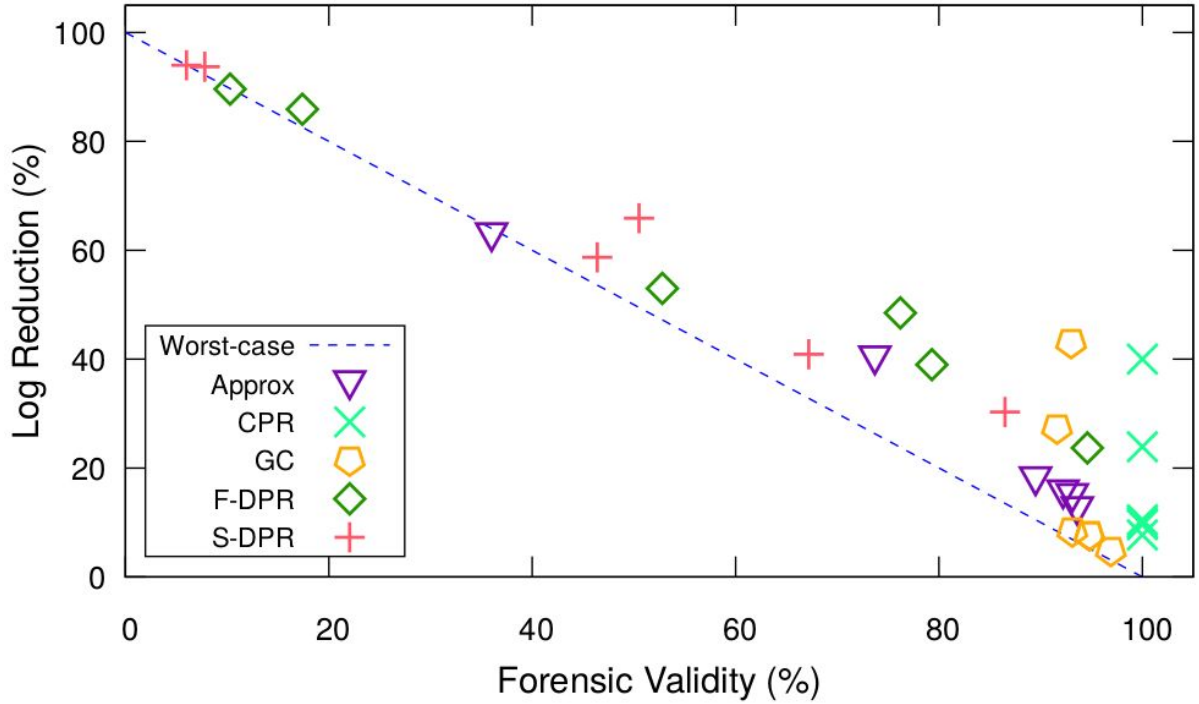


Attack-Preserving Forensics
(uniquely malicious information flow)

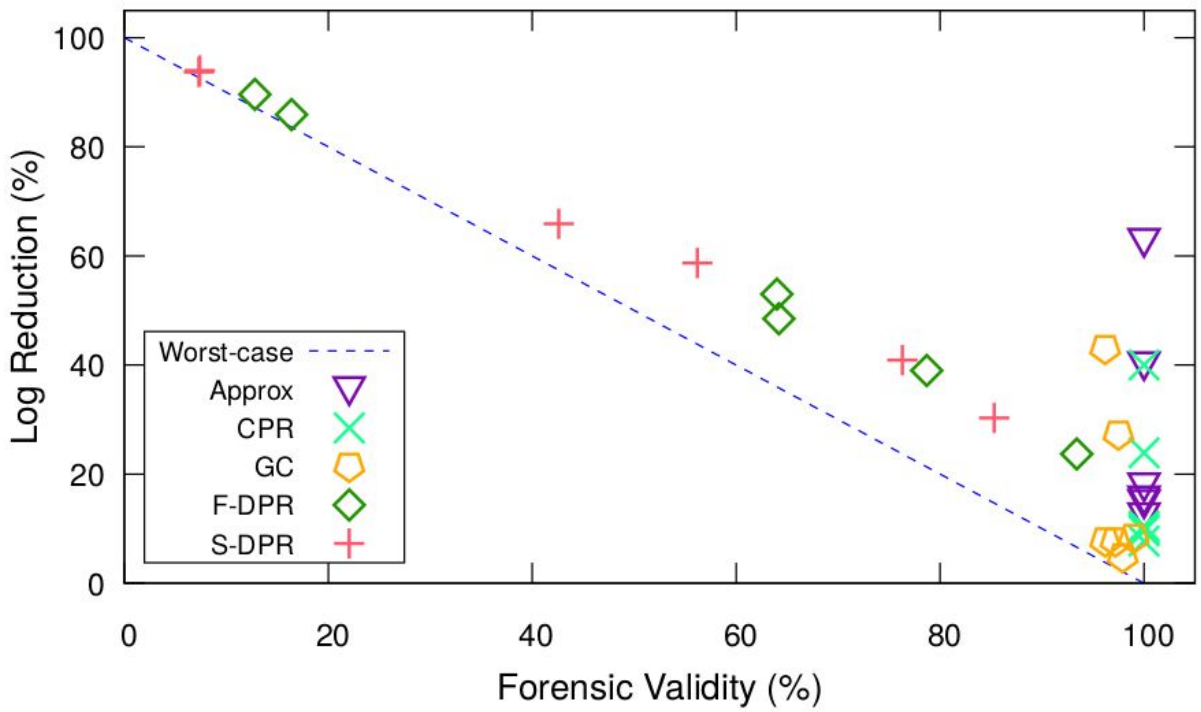


Results

Causality-Preserving Forensics
(all information flow)



Attack-Preserving Forensics
(uniquely malicious information flow)



Takeaways

Takeaways

Validity of reduced logs should **not** be based on anecdotal studies

- Depends on **task** and **threat model**
- Providing a continuous metric for arbitrary queries is a step in the right direction

Takeaways

Validity of reduced logs should **not** be based on anecdotal studies

- Depends on **task** and **threat model**
- Providing a continuous metric for arbitrary queries is a step in the right direction

Reduction techniques can be tailored to specific tasks and threats

- Tasks: Source and Full Dependency Preserving
- Threat Models: **LogApprox**

Thank You!