# Supportting materials

## The algorithm of metrics calculation

Algorithm 1: The calculation of the four metrics

**Input**

*CS*: The call sequence of the software under test

**Output**

*P*: possibility; *S*: severity; *M*: manipulability; *C*: crypticity

**Begin**

/\* Collect and filter the system calls of the tested software application.\*/

**for** *c* **in** *CS*

/\* if c does not satisfy the condition, drop it from *CS*. \*/

**if** **not** satisfy(*c*) **then**

drop(*c*, *CS*)

/\* Trigger PPN *pn* with filtered system call sequence. \*/

**else**

trigger(c, pn)

/\* insert the move path of token into *BPS* \*/

insert(*instc*.*proc*, *BPS*)

/\* Calculate the metrics by formula (1) to (4) \*/

[*P*, *S*, *M*, *C*] = [*P*(*BPS*), *S*(*BPS*), *M*(*BPS*), *C*(*BPS*)]

**End**

Algorithm 1 evaluates the system calls of the tested software application, and outputs the four quantitative metrics. First, we collect the system calls by monitoring the execution of the software on our experiment platform. Next, we check the collected system call sequence by two kinds of conditions. One is to check whether the call name is in the typical PPN modules. The other is to consider whether the parameters are related to our definition in the variable sets. Third, we trigger the PPN with the filtered system call sequence, perform the predefined operations. Fourth, the calls are processed, the moving trace of the tokens forms the BPS of the tested application. Finally, we calculate the four metrics based on the BPS.

## Quantification algorithm of overall privacy leak degree

The main steps involved in PLEAS are as follows.

Firstly, we put four metrics of different software into a matrix *M* as presented in Fig. 11, then normalize the data. For each element *mij* in *M*, we calculate the normalized value  and vector as in (5):

|  |  |
| --- | --- |
| , i = 1, 2, 3, 4 and j=1, 2, … , n | (5) |

Where , .

Secondly, we calculate the covariance matrix *R* =  from *M*:

|  |  |
| --- | --- |
| , i, j = 1, 2, 3, 4 | (6) |

from which the candidate principal component (PC) with the eigenvalue and eigenvector of R is calculated. Assuming that the eigenvalues are *λ1 ≥ λ2 ≥ λ3 ≥ λ4*≥ 0, and the corresponding eigenvectors are *u1*, *u2*, *u3*, *u4*, where *ui* = [*u1i*, *u2i*, *u3i*, *u4i*]T, then the candidate principal components are:

|  |  |
| --- | --- |
|  | (7) |

Thirdly we select principal components from the candidates. The number of principal components is less than or equal to the number of original variables. This selection is from the information proportion *bi* of the *pci*. The first principal component is the most informative, and each succeeding component in turn has a less variance possible. *bi* is calculated by eigenvalues:

|  |  |
| --- | --- |
|  | (8) |

and the cumulative information proportion *ap* is calculated based on *bi*:

|  |  |
| --- | --- |
|  | (9) |

when *ap* approaches 1, we select *pc1*, *pc2* ... *pcp* as the principal components.

Finally we evaluate the values *X* of each alternative with the principal components selected in step 7 as in (10):

|  |  |
| --- | --- |
|  | (10) |

*O* is the ranking value to evaluate the overall degree of privacy leak of the tested software application.The whole procedure is presented in Algorithm 2.

Algorithm 2: PLEAS

**Input**

*A1,A2,… ,An*: metrics tuple of *n* tested software applications, where *Ai* = (*pi*, *si*, *mi*, *ci*).

**Output**

*O1,O2,… ,On*: overall privacy leak degree from *A1* to *An*

**Begin**

/\* Normalize metrics data by formula (5). \*/

*M* = [*A1,A2,… ,An*]T

*M’*= normalize(*M*)

/\* Calculate the covariance matrix by formula (6). \*/

*R* = covariance(*M*)

/\* Calculate the eigenvalue and eigenvector of R. \*/

*U* = [*u1*, *u2*, *u3*, *u4*] = eigenvector (*R*)

*Λ* = [*λ1, λ2, λ3, λ4*] = eigenvalue(*R*)

/\* Select principal component by formula (7) to (9). \*/

**for** *i* = 1 **to** 4

*bi* = *bi* + *λi*

*pci*= *ui* ∙ [*x1*, *x2*, *x3*, *x4*]T

*p* = *i*

/\* The proportion threshold of principal component is 80% \*/

**if**  **then** **break**

/\* Evaluate the alternatives by formula (10). \*/

**for** *i* = 1 **to** *n*

**for** *j* = 1 **to** *p*

*Oi* = *Oi* +*λi* ∙ *pci*

**End**

The PCs found by PLEAS can assign the optimal weights of different metrics without the subjective assumption to maximize the variance of the metric vector. Therefore, it can effectively evaluate the overall degree of privacy leak based on the four metrics. Proposition 1 is a formal proof of this characteristic.

**Proposition 1.** *Let X = [x1, x2, … , xp] be variable vector which denotes the metrics of privacy leak behavior, W = [w1, w2, … , wp] be the weight factor of each metric, λ1 ≥ λ2 ≥ … ≥ λp ≥ 0 be the eigenvalue of the variance matrix Var(X), the corresponding eigenvector is* *u1, u2, … , up. We have.*

**Proof:** Assume that M =. Letbe the Jordan Decomposition of Var(X), then = (*u1*, *u2*, *…* , *up*) diag(*λ1*, *λ2*, *…*, *λp*). Let *Y=.* Because YTY = = WTW= 1, we have

M =

=

=.

The maximum is obtained when *Y* = [1, 0,…,0]T, thus . Theorem 2 holds.

*WTX* is the first principal component selected based on Equation (7). Similarly, we can select other principals using Equations (8) and (9).