**Rozbalil som vzorky a reporty vo virtualke**

lzop -d nazov suboru  
tar -xf nazov suboru

--------------------- pre pôvodný dataset nemusím robiť

**Rozdelil som dáta na vzorky a VT reporty:**kod je v script0

**Odstránil som vzorky ktoré nemali reporty a naopak:**

Odstránenie json koncovky: find . -type f -exec sh -c 'mv "$0" "${0%.json}"' {} \;  
Uložil som zoznam súborov: find . -type f -exec basename {} \; > ../reports.txt

Použil som ho na odstránenie súborov z druhého priečinku:

Kód je v script1, ale príkaz sa mení na:

if ! grep -qxFe "$i" /home/remnux/Downloads/malware/report; then

Nakoniec som pridal koncovky: find . -type f -exec sh -c 'mv "$0" "${0%}.json"' {} \;

**Odstránil som príliš veľké súbory (>5mb):**

find . -size +5M -delete //v priečinku zložiek. Potom som odstránil aj reporty ktoré patrili k odstráneným vzorkám.

**Odstránil som súbory ktoré neboli PE (boli MS-DOS)**zistil som typ súborov: find . -type f -exec file {} \; > ../types

Osobitne som uložil tie, ktoré neboli PE: sed -n '/PE32/!p' types > notpe

Upravil som ich aby ostali len názvy: sed 's/ .\*//' notpe | sed 's/^..\(.\*\).$/\1/' > names  
vymazal som ich z oboch priečinkov s upraveným script1.

**Odstránil som vzorky ktoré boli false vo vybraných AV**kód je v script2. Odstránil som ich, aby nevznikla „false“ trieda.

**Odstránil som súbory ktorých reporty nemali vybrané AV**

Kód je v script4. Vymazal som ich cez upravený script1.

**Odstránil som súbory ktoré mali mali viac ako polovicu false hodnotení**

Kód je v script12. Vymazal som ich cez upravený script1.

**Odstránil som súbory ktoré nemali kompletné reporty**

Kód je v script6. Vymazal som ich cez upravený script1.

-------------------------------

**spustil som labels.py**has\_empty = False

labeling\_type = "consensus"

for\_prediction = False

Cez skript1 som odstránil malware ktorý nebol v triedach

potom som získal disasemblované súbory:  
Objdump – script10. Niektore nespracoval, tak som ich zaznamenal:   
find . -size -5k -exec basename {} \; > ../empty

Potom som ich cez skript1 odstránil z disassembled, reports, malware

Po získaní empty súboru som spustil **labels.py**:

has\_empty = True

**Zabalil som malware a poslal do windows virtuálky a zistil entropiu**zabalenie: tar c Win32\_EXE | lzop - > malware.tar.lzo

Rozbalenie: lzop.exe -df C:\Users\BPD\Desktop\malware.tar.lzo

Zistenie entropie: sigcheck64 -a -c C:\Users\BPD\Desktop\Win32\_EXE > C:\Users\BPD\Desktop\entropy.csv

**V linux virtuálke som upravil výstup zo sigcheck**

Súbor sa uložil v UTF16, preto: iconv -f UTF16 -t UTF8 entropy.csv > entropy

sed -r -i 's/.{33}//' entropy.csv

sed -i 's/"//g' entropy.csv

awk -F ',' '{print $1, $NF}' entropy.csv > entropy

dos2unix entropy

**Získal som hexadecimálnu reprezentáciu, textové reťazce**  
Hexdump – script9  
strings – script11, min. 5 znakov  
všetky dáta presuniem do priečinka projektu

--------------------------------------------------------------------------------------

**Extrakcia**for\_prediction = False

First\_time = True

**Preprocessing**metóda standardize má posledný atribút False

**Klasifikácia**mode = "preselection" //nezbehlo, malo RAM

Pozor pri metódach na num\_class !!

**Selekcia**Pozor pri metódach na num\_class !! aj v klasifikácii

Najprv: is\_standard = False   
potom: is\_standard = True  
premenuj selection\_times podla poctu atributov

**Klasifikácia**mode = "selection\_check"  
premenuj classification\_selected podla poctu atributov

Pozor pri metódach na num\_class !! aj

**Klasifikácia**mode == "prediction\_train"

**Results**

For\_prediction = FalseAk chcem pouzit feature\_intersection, musim najprv odstranit z groups riadky bez atributov a nechat len riadky ktore koresponduju s najlepsimi vysledkami  
Pre feature\_intersections\_difference musim v groups priecinku urobit dve subory (rovnake ako pri feature\_intersection), ale v prvom budu najlepsie atributy zo stromov v jednom riadku a v druhom budu v riadku najlepsie atributy zo svm

Premenuj groups podla poctu atributov

----------------------------------------------- predikcia

**Labeling**

Zalohuj povodne data v priecinku subory a povodne reporty  
for\_prediction = True

has\_empty = False  
Rovnako ako v predoslom, po ziskani has\_empty suboru dam has\_empty = True

**Extrakcia**vymaz subory z features a podpriecinkov ale nechaj ich prazdne

for\_prediction = True

first\_time = True

**Klasifikácia**

Pozor pri metódach na num\_class !!

mode == "prediction"

**Results**

For\_prediction = True

**Poznámka k výsledkom**

Pre SVC najprv štandardizujem celý dataset, potom urobím selekciu, ale uložím neštandardizované dáta. Pred trénovaním pre predikciu urobím novú štandardizáciu len na selektovaných dátach a na túto uložím spolu so scalerom (a natrénovaným modelom). Pri získaní atribútov pre predikciu použijem scaler a získam štandardizované dáta ktoré dám natrénovanému modelu na predikciu.

ALE štandardizácia bola iná ako pri selekcii a tieto hodnoty už nemusia dobre deliť priestor atribútov – výsledky môžu rapídne klesnúť aj keď je štandardizácia zhodná pri tréningu aj predikcii.

Takisto môžu klesnúť výsledky keď hlavičku zo SVC selekcie použijem v inej predikcii.