Zucchetti dataset analysis

UNIVERSITÀ DI PISA

AI for cybersecurity A.A 2021/2022
Department of computer engineering, Pisa
Carlo Leo
Riccardo Gallo

INITIAL DATASET: raw data regarding logs

Attributes:

- *User ID*
- *Timestamp*: when the log occurred
- *Event type*:
  - 1 → successful login
  - 2 → failed login
  - 3 → logout
- *Application*: application acronym (nominal)
- *IP address* (mocked)

OBJECTIVE:
FIND PATTERNS OF SUSPICIOUS LOGS

Preprocessing

Data preprocessing:

- Substituted not-specified IP addresses with UNKNOWN default value (treated as separate IP address)
- Checked data consistency for eventType attribute
- Performed numerosity reduction to delete duplicate rows
- Sorted tuples by date

Login Dataframe

Attributes:

- *userId*
- *date*
- *day*: number from 1 (Monday) to 7 (Sunday)
- *workingTime*: boolean value True if the log occurred between 8am - 19pm
- *eventType*
- *application*
- *Ip (mocked)*

Observations:

- Every log between Saturday and Sunday occurred outside working time
- Whatever kind of event is happened in both working time and non-working time hours
- Failed logins attempts occurred only for application ERM, HRW and TM3

Series

Skimming:

- Delete users having only 0 or 1 login error
- Delete users having all login success after an erroneous login
- Delete users not having at least two login errors in 24h
- Delete logs referring to application with no login errors

Series:

- Sequence of login errors ended either with or without a correct login. The maximum delay between two attempts is 5 minutes.
- Created for:
  - userId
  - IP address
- Each event of the series has been represented as a tuple:
  - *<date, eventType, application,workingTime, day, userId or ip>*

Series Dataframe

Attributes:

- *userId / ip*
- *avgTime*: average time between 2 login attempts
- *failedAttempts*: number of login errors in the series
- *hasSuccess*: value stating if the series ends with a successful login (1) or not (0)
- *application*
- *workingTime*
- *day*
- *ipAmount / userAmount*: number of ip/users in the series

HEURISTIC DEVISED: **Suspicious**

Computed as the product of the following weights:

- *w1*: depends on the week day and the working time
- *w2* : depends on the number of ip/users found in the series
  - 1/(1-amount) if amount > 1 else 0.4
- *w3*: depends on the average time
- *w4* : depends on the number of attempts found in the series

Since each weight ranges in (0,1) to avoid underflow the Graham function has been used (where pi is the wi):

$$p = \frac{1}{1+e^{\eta}}, \text{with } \eta = \sum_{i=1}^{N}[\ln(1-p_i) - \ln p_i]$$

Constants used to weight a series

- Average time range and relative weights
- Failed attempts boundaries and relative weights
- Week day and working time weights

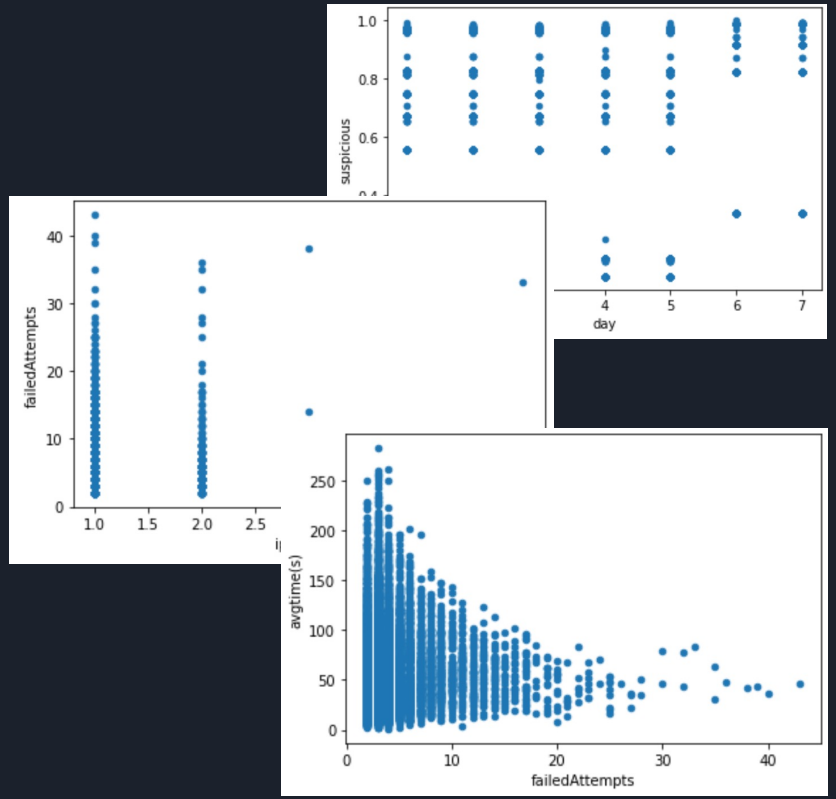
→ changing them you can vary the granularity of the analysis e.g. `MAX_DELAY_ERRORS`

```
# working hours boundaries
START_WORKING = 8
END_WORKING = 19
# time boundaries for series construction
AT_LEAST_ONE_DAY = 24 * 3600 * 1000 # 24 h
MAX_DELAY_ERRORS = 5 * 60 * 1000 # 5min
MAX_DELAY_SUCCESS = 5 * 60 * 1000 # 5min
# attempts boundaries
MAX_ATTEMPTS = 3
MAX_TIME_BOT = 5000 # estimate both maximum time. We consider mocking user typing
RESIDUAL_TIME = MAX_DELAY_ERRORS - MAX_TIME_BOT
```

...

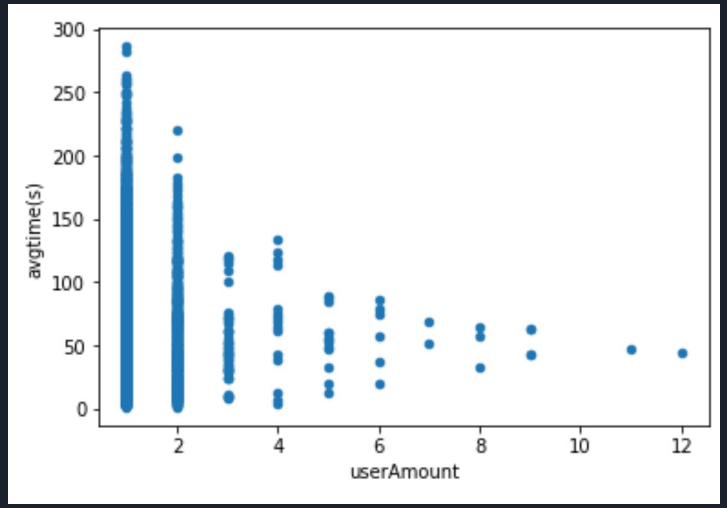
Results – User series

- Suspicious series have been found regardless of working time and day of the week
- Comparing the number of IP and failed attempts came up that: with one or two IPs, the number of failed attempts is bigger
- When the maximum average time between login attempts narrows, the number of failed attempts increases



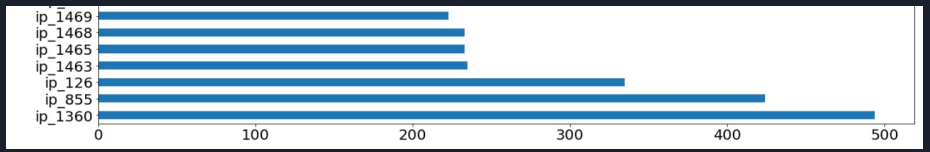
Results – IP series

- When requests coming from the same IP, are for different users, the average time between attempts decreases

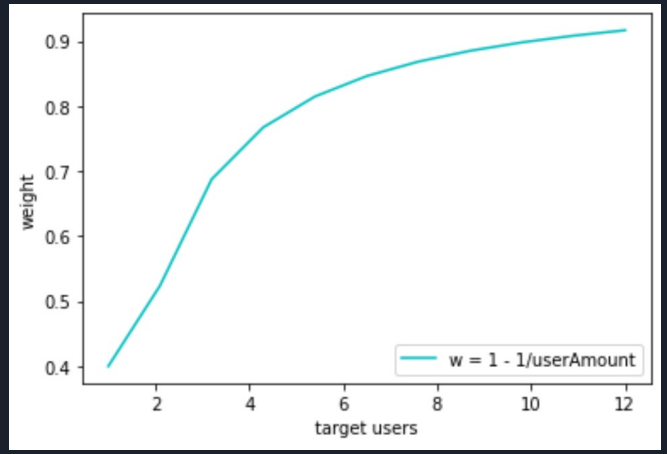


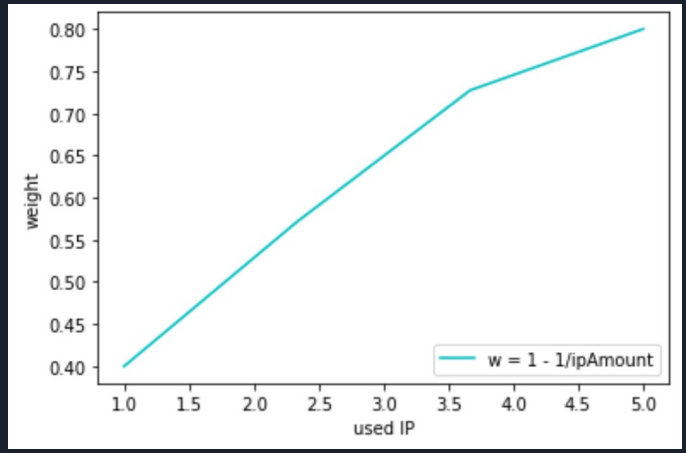
- Looking at the most frequent incoming IP address can be usefull

count	494.000000
mean	0.458877
std	0.335637
min	0.122744
25%	0.122744
50%	0.557377
75%	0.746082
max	0.996529



Results – IP series vs. User series





	userId	avgTime	failedAttempts	hasSuccess	application	workingTime	day	ipAmount	suspicious
35404	31002	4000	11	0	ERM	0	6	1	0.999348
22039	20272	83250	33	0	ERM	1	1	5	0.993082

	ip	avgTime	failedAttempts	hasSuccess	application	workingTime	day	userAmount	suspicious
27027	ip_38662	4000	11	0	ERM	0	6	1	0.999348
34469	ip_5989	4142	8	0	ERM	0	6	1	0.999348

Conclusions

- Following rule of thumb provides meaningfull results, *e.g several attempts in narrow time window*, if computed values are analyzed enough in detail
- **Pay attention** on choosing metrics for searching for suspicious pattern of logs
- Human factor is tedious to foresee

	userId	avgTime	failedAttempts	hasSuccess	application	workingTime	day	ipAmount	suspicious
33203	28878	11000	2	1	ERM	1	2	1	0.122744
33210	28878	7000	3	1	ERM	1	3	1	0.122744
33212	28878	7333	3	1	ERM	1	5	1	0.122744
33214	28878	6000	3	1	ERM	1	4	1	0.122744
33219	28878	8500	2	1	ERM	1	3	1	0.122744
33221	28878	9000	2	1	ERM	1	3	1	0.122744
33224	28878	10000	2	1	ERM	1	4	1	0.122744
33235	28878	14500	2	1	ERM	1	5	1	0.122744
33236	28878	8000	3	1	ERM	1	5	1	0.122744
33238	28878	9000	2	1	ERM	1	5	1	0.122744
33239	28878	8500	2	1	ERM	1	2	1	0.122744
33240	28878	8000	2	1	ERM	1	3	1	0.122744
33242	28878	9000	2	1	ERM	1	4	1	0.122744
33250	28878	14500	2	1	ERM	1	3	1	0.122744
33260	28878	6500	2	1	ERM	1	4	1	0.122744
33262	28878	6500	2	1	ERM	1	5	1	0.122744

Questions?

Thank you!!!!

All work is avaliable on GitHub at:
https://github.com/carloleo/AI-for-cybersecurity