# Logging

James Brucker

# KBank admits data hack

The Nation
August 01, 2018 18:47

Pipit Aneaknithi, Kasikornbank president, revealed that on July 25, KBank found that 3,000 names of corporate customers using KBank’s website for the letter of guarantee service might have been leaked.

As soon as KBank detected the irregularity, it said it immediately closed the loophole... The data that may have been leaked was the names and telephone numbers of KBank’s corporate customers using the letter of guarantee service via the website only.

http://www.nationmultimedia.com/detail/business/30351237

# KBank, KTB target in cyber-attacks

Bangkok Post
August 01, 2018 04:00

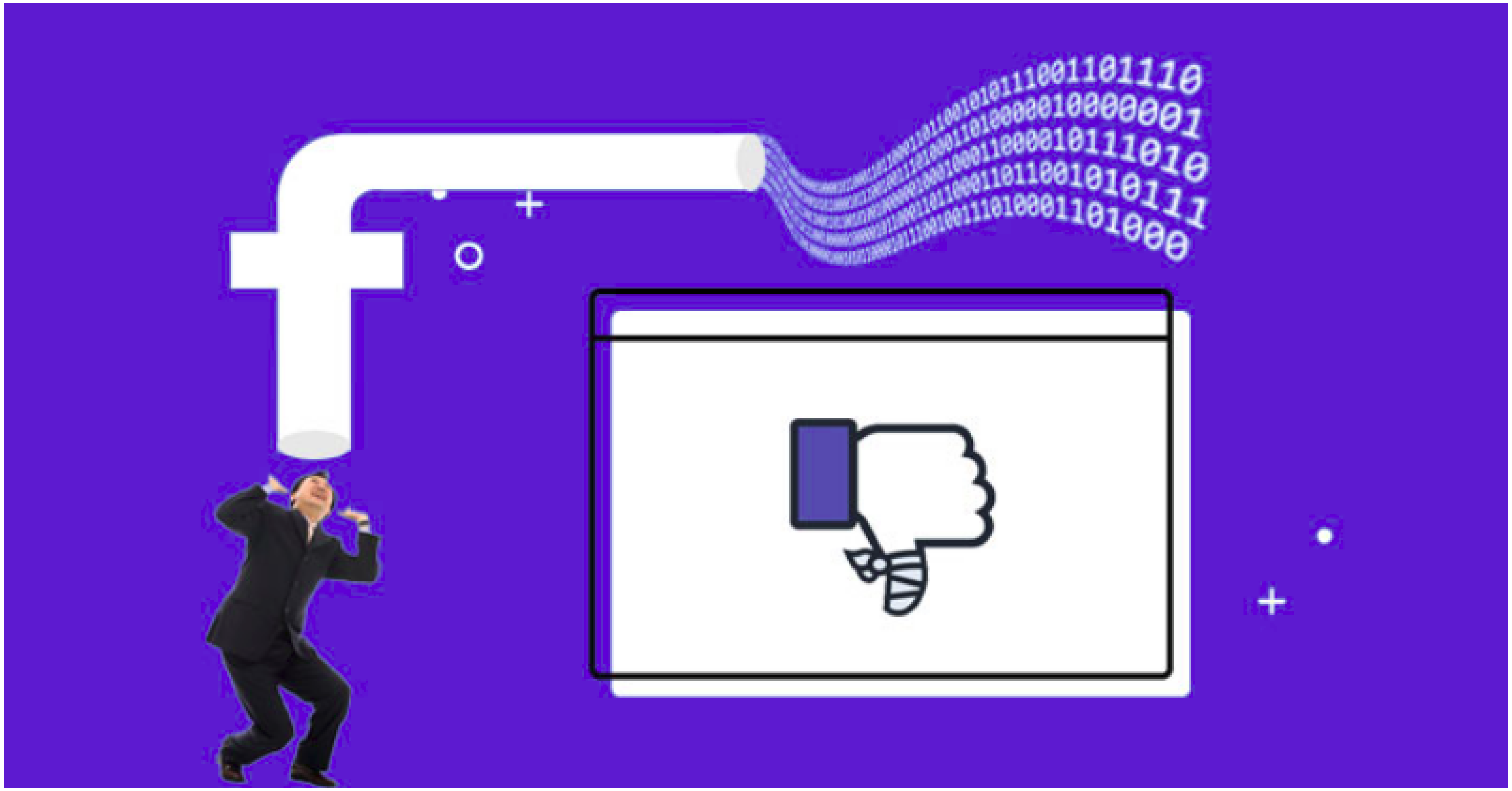
KBank ... found that the names of 3,000 corporate customers using KBank's website ... might have been hacked.

KTB president Payong Srivanich said in a statement that the bank had detected general information from 120,000 retail customers who applied for mortgages and personal loans online ... was hacked in the days leading up the July holidays.

https://www.bangkokpost.com/news/security/1513410/kbank-ktb-targeted-in-cyber-attacks

# 30 Million Facebook Accounts Were Hacked: C
# Them

October 12, 2018 Swati Khandelwal



Late last month Facebook announced its worst-ever security breach that allowed an unknown group of hackers to steal secret access tokens for millions of accounts by taking advantage of a flaw in the 'View As' feature.

# How Facebook was Hacked

The flaw in "View As" feature has been present since July 2017, but first detected on 14 Sep 2018 due to rise in suspicious activity.

Facebook knows exactly which accounts were hacked and you can check your account.

https://www.wired.com/story/how-facebook-hackers-compromised-30-million-accounts/

https://thehackernews.com/2018/10/hack-facebook-account.html

# How Did They Know?

How did KBank know 3,000 customer's data stolen ... and what data was stolen?

How did KTB know 120,000 customers "who applied for a mortgage or loan" had data stolen?

How did Facebook know there was "*suspicious activity*"?

How does Facebook know whose data was stolen?

# LOGGING

They keep "logs" of events and activity.

# Chinese are Attacking My Server!

Unix/Linux keep logs for many services is **`/var/log`**.

On se.cpe.ku.ac.th here is part of /var/log/auth.log:

```
Nov 18 06:29:48 se sshd[6720]: Failed password for root from 116.31.116.16 port 61430 ssh2
Nov 18 06:29:52 se sshd[6720]: message repeated 2 times: [ Failed password for root from 116.31.116.16 port 61430 ssh2]
```

Someone trying to login as root.

Where is 116.31.116.16?

Search Google...

# 116.31.116.16

## 116.31.116.16 | ChinaNet Guangdong Province Network | AbuseIPDB

https://www.abuseipdb.com/check/116.31.116.16

116.31.116.16 has been reported 409 times. ... 116.31.116.16 was first reported on December 3rd 2017 , and the most recent report was 4 hours ago .

## IP List of Brute force attackers

https://report.cs.rutgers.edu/DROP/attackers

... 115.186.147.235 115.249.205.29 116.196.76.135 116.31.116.11 116.31.116.12 116.31.116.14 116.31.116.16 116.31.116.21 116.31.116.23 116.31.116.24 ...

## The Anti Hacker Alliance™ fights against 116.31.116.20

https://anti-hacker-alliance.com/index.php?ip=116.31.116.20

116.31.116.x

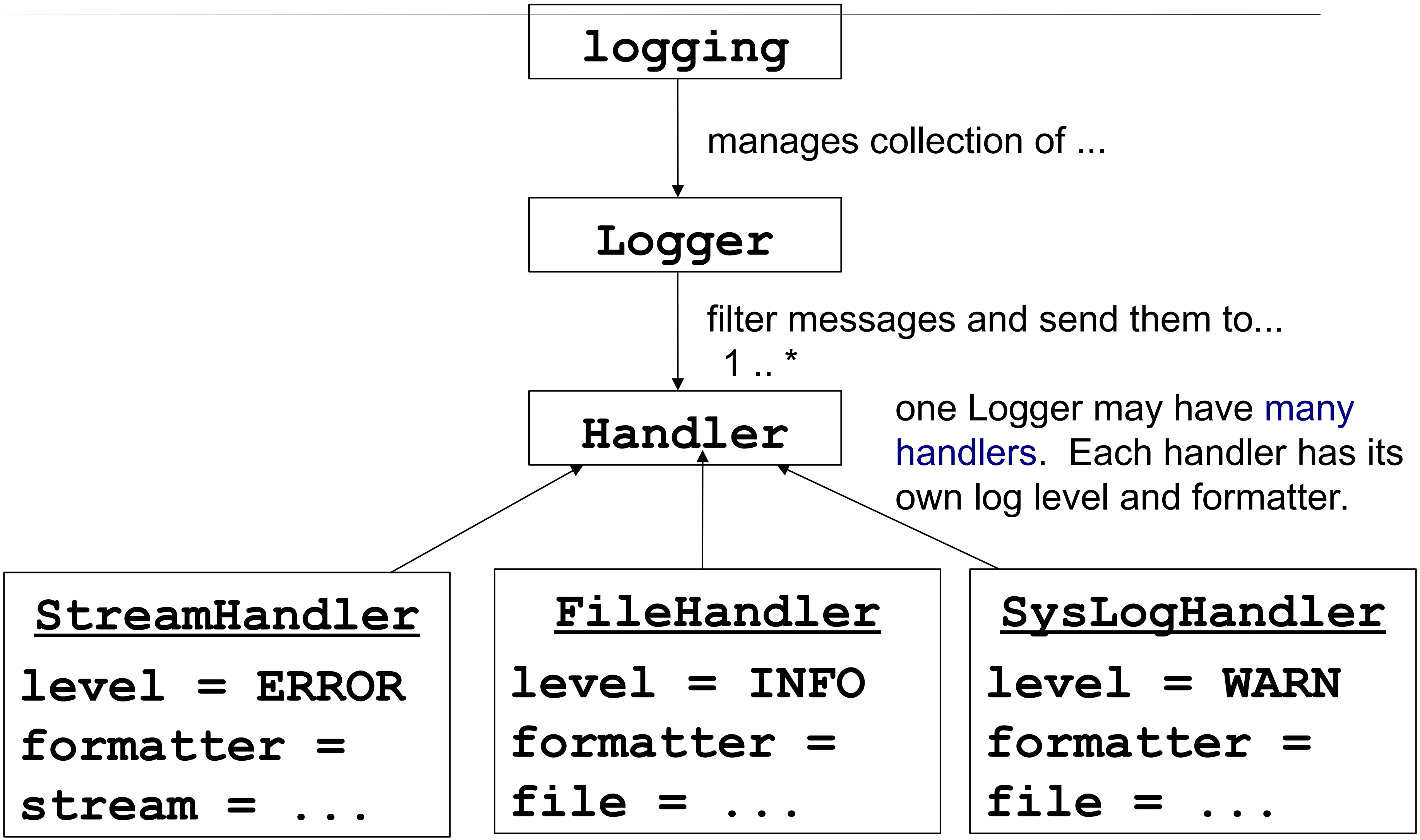
# Python Logging

Python standardlogging package

```
# get a named logger.  Use any logical name
logger = logging.getLogger( __name__ )

# log levels: DEBUG, INFO, WARN, ERROR, CRITICAL
logger.info( "Successful login by " + user.username )
logger.warning( "Failed login by " + form.username )
logger.error( "Poll question has no choices: "+question )
logger.critical( "Connection to database failed" )
logger.log( logging,INFO, "another INFO level mesg" )
```

# Logging Architecture



`logging`

manages collection of ...

`Logger`

filter messages and send them to...

1 .. *

`Handler`

one Logger may have many handlers. Each handler has its own log level and formatter.

```
StreamHandler
level = ERROR
formatter =
stream = ...
```

```
FileHandler
level = INFO
formatter =
file = ...
```

```
SysLogHandler
level = WARN
formatter =
file = ...
```

# getLogger( "name" ) is a Singleton

```
log1 = logging.getLogger( "auth" )
log2 = logging.getLogger( "auth" )
# are they the same object?
log1 == log2
True
id(log1) == id(log2)
True
```

# 5 Log Levels

| Level Names: | Example: |
|---|---|
| CRITICAL = 50 | logger.critical("Can't connect to db") |
| ERROR = 40 | logger.error("Exception raised:", ex) |
| WARNING = 30 | logger.warning("Failed login by wisa on ...") |
| INFO = 20 | logger.info("Successful login by nerd ...") |
| DEBUG = 10 | logger.debug( request ) |
| | |
| Aliases: | logger.log( *level*, *message* ) |
| FATAL = 50 | |
| WARN = 30 | |

# Java Has More Log Levels

java.util.logging

SEVERE

WARNING

INFO

( CONFIG )

FINE

FINER

FINEST

OFF

ALL

(**stupid** names)

Log4J & SLF4J Levels

FATAL

ERROR

WARN

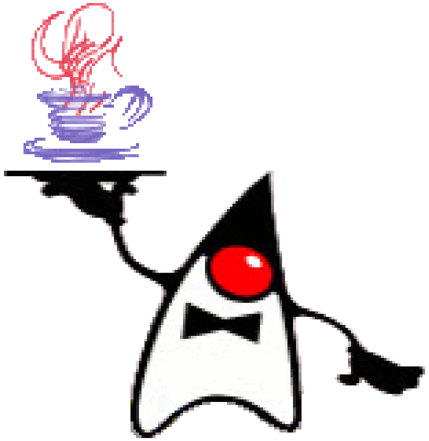
INFO

DEBUG

TRACE

OFF

ALL



# How to Use the 5 Log Levels

What *should* you log to each of these levels?

CRITICAL

ERROR

WARN

INFO

DEBUG

See: *Python Logging Tutorial*

https://docs.python.org/3/howto/logging.html

# Log Handlers and Formatters

Python has many Log Handlers you can choose:

https://docs.python.org/3/library/logging.handlers.html

**Important Handlers:**

logging.StreamHandler(stream=sys.stdout)

logging.FileHandler( *filename* )

logging.RotatingFileHandler( *filename, maxBytes=...* )

logging.TimeRotatingFileHandler

logging.SysLogHandler(*address*=("localhost",port),...)

# Loggers for a Hierarchy

```
rootlog = logging.getLogger( )   - the root logger
loga = logging.getLogger( 'a' )   - child of root
logb = logging.getLogger( 'a.b' ) - child of 'a',
                                     grandchild of root


logb.warn("Warning!")   - sent to logb, then loga, then root
```

# Logging in a Web App

Web Apps have some special concerns:

1. want to log IP address for events and activity

2. Web app may be deployed on many hosts, and may not be persistent.   How can you make separate logs from web app?

3. How to aggregate logs from different parts of app?

# Web App Logging

What *events or activity* should a web app log?

1.

2.

3.

4.

5.

# What Info Should You Log When...

1. A failed or successful login occurs.

>

>

>

2. A user submits a "vote" to the polls application.

>

>

>

# Learn Python Logging

Python Logging Tutorial (Basic & Advanced)

https://docs.python.org/3/howto/logging.html

Logging Cookbook

https://docs.python.org/3/howto/logging-cookbook.html

# Django Logging

Django uses Python Logging, adds some "conveniences".

See: Django User Guide, section 3.16 (only 10 pages)

# Configure Django Logging

```
LOGGING = {
    'disable_existing_loggers': False,
    'handlers': {
        'file': {
            'level': 'DEBUG',
            'class': 'logging.FileHandler',
            'filename': '/path/to/django/debug.log',
        },

    },
    'loggers': {
        'django': {
            'handlers': ['file'],
            'level': 'DEBUG',
            'propagate': True,
        },
    },
}
```

# Configure Django Logging

```
LOGGING = {
    'disable_existing_loggers': False,
    'handlers': {
        'file': {
            'level': 'DEBUG',
            'class': 'logging.FileHandler',
            'filename': '/path/to/myapp.log',
        },
        'console': {
            'class': 'logging.StreamHandler'
        }
     }
     'loggers': {
        'myapp': {
            'handlers': ['console'],
            'level': 'INFO',
            'propagate': False,
            ...
```

# Logging Advice

1. Configure the root logger, but don't use it directly.
2. For deployed web apps, log to console (12FactorApp)
3. Configure logger via config variables.