Logging

James Brucker

KBank admits data hack

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.

The Nation   August 01, 2018 18:47

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

KBank, KTB target in cyber-attacks

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.

Bangkok Post - August 01, 2018 04:00

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

30 Million Facebook Accounts Were Hacked: Ch Them

📅 October 12, 2018   👤 Swati Khandelwal
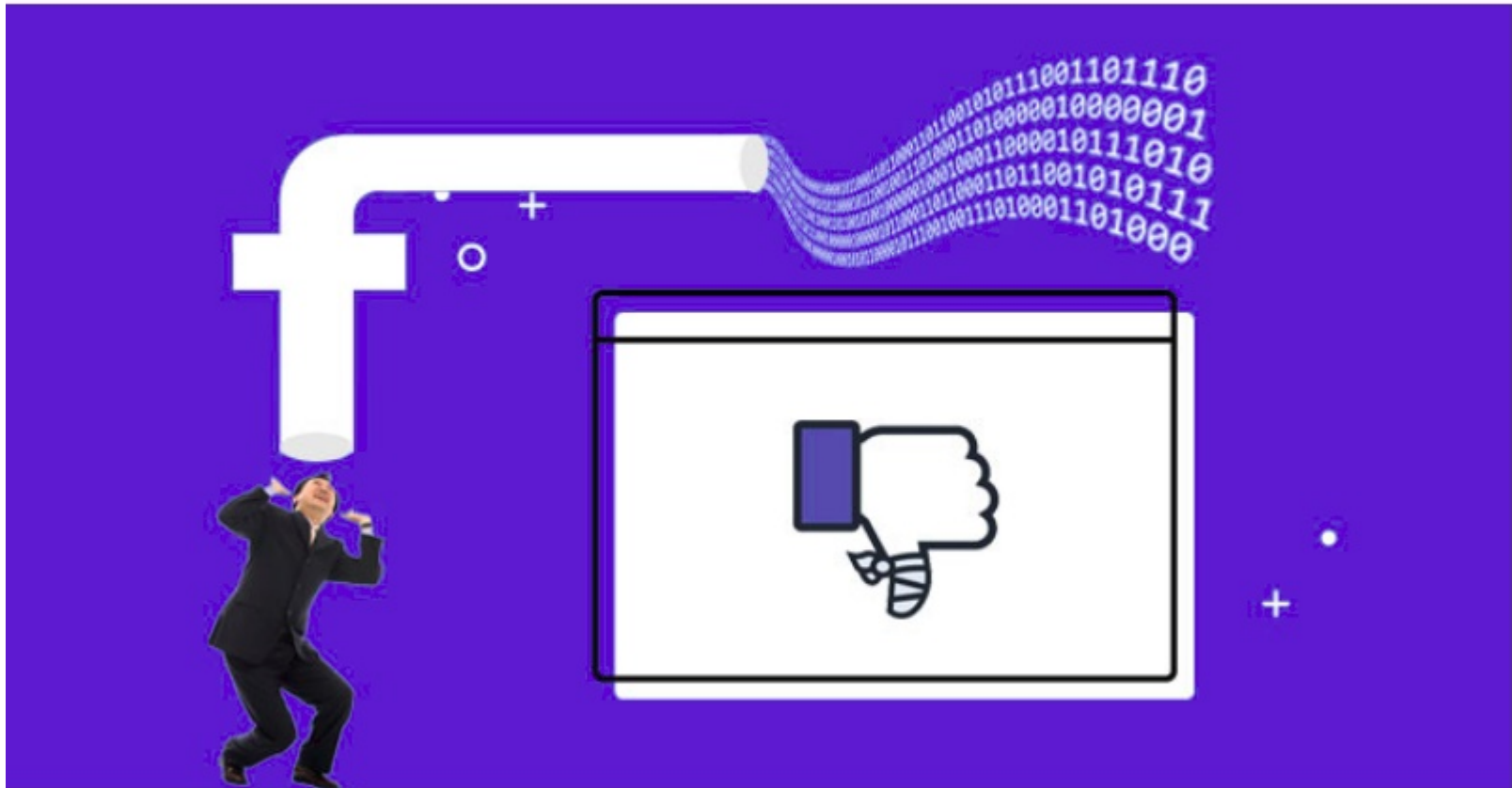
Late last month Facebook announced its worst-ever security breach that allowed an unknown group of hackers to steal secret access tokens for millions of accounts by taking advantage of a flaw in the 'View As' feature.

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 <u>what</u> 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 <u>whose</u> data was stolen?

LOGGING

They keep "logs" of events and activity.

Linux Logs Almost Everything

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

Typical logs are:

`auth.log` - authentication related (login, sudo)

`boot.log` - system start-up (boot) activity

`dpkg.log` - package install and configuration messages

`kern.log` - messages from the kernel

`lastlog` - most recent login by each user

`ufw.log` - firewall messages

Log files are automatically rotated every 1 - 7 days, so they do not become too large.

Chinese are Attacking My Server!

In `/var/log/auth.log` on se.cpe.ku.ac.th:

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 is 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

The "Fix"

1. ssh was already configured to deny root login.
2. add firewall rule to deny all traffic from IPs in China.

Python Logging

`logging` - Python standard logging package

# get a named logger. Use any logical name
testlog = logging.getLogger("testing")

# log messages at different "log levels"
testlog.debug("a message for developers.")
testlog.info("this is a boring info message")
testlog.warning("a warning you should pay attention to")
testlog.error("An error occurred.")
testlog.critical("A critical error or failure")
testlog.log(logging,INFO, "another INFO level mesg")

5 Log Levels

Level Names:

CRITICAL = 50

ERROR = 40

WARNING = 30

INFO = 20

DEBUG = 10

Aliases:

FATAL = 50

WARN = 30

Example:

logger.critical("Can't connect to db")

logger.error("Error rendering template")

logger.warning("Failed login by ...")

logger.info("Successful login by ...")

logger.debug(request)

logger.log(*level*, *message*)

What do Log?

```python
# log some events
log.info( "Successful login by " + user.username )
log.warning( "Failed login by " + form.username )
log.error( "Poll question has no choices: "+question )
log.critical( "Connection to database failed" )
log.log( logging,INFO, "another INFO level mesg" )
# log an exception
try:
    q = Question.objects.get(id=1)
except Exception as ex:
    log.exception("Someone stole my question", ex)
```

You Can Control Logging

1. Turn logging on, off, or filter:

   "Only print messages of level WARNING or higher"

   ```
   logging.setLevel( logging.WARNING )
   ```

   "... but for 'polls' loggers, print INFO messages too"

   ```
   logging.getLogger('polls').setLevel( logging.INFO )
   ```

2. Redirect the log messages to a file

3. Change the format of log messages

Use Log Level to Control Output

```python
# Default settings only print WARNING or higher
log = logging.getLogger( )
log.info( "This message is not printed" )


log.warn("This is a warning")
```
```
This is a warning
```
```python
# set message threshold level to INFO (INFO or higher)
log.setLevel( logging.INFO )
log.info( "This message IS printed" )
```
```
This message IS printed
```
```python
log.debug( "this is not printed" )
```

Loggers Form a Named Hierarchy

Root logger: root = logging.getLogger( )

'polls' logger: log1 = logging.getLogger('polls')

'polls.models' log: log2 = logging.getLogger('polls.models')

Loggers Inherit from Parent (Ancester) loggers

Configure the root logger.

Other loggers get those settings by default.

Log Message Propagation

If `logger.propagate = True` (the default)
then events logged to this logger will be passed to
handlers of higher level loggers, in addition to handlers
of this logger. Threshold & filters of ancestor loggers
are ignored.

[see "class logging.Logger" in Python Docs]

root = logging.getLogger( )   - the root logger

loga = logging.getLogger('a')   - descendent of root

logab = logging.getLogger('a.b') - descendent of 'a' & root

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

You must try this yourself in order to understand what it does.

Use module name for logger name

You can use any name you want for logger.

Standard is to use module name or package name

Using a good name helps you track source of messages.

1. A logger named for <u>this</u> module

```
log = logging.getLogger( __name__ )
```

2. A logger for "polls" app

```
log = logging.getLogger( "polls" )
```

What do Log?

```python
# log some events
log.info( "Successful login by " + user.username )
log.warning( "Failed login by " + form.username )
log.error( "Poll question has no choices: "+question )
log.critical( "Connection to database failed" )
log.log( logging,INFO, "another INFO level mesg" )
# log an exception
try:
    q = Question.objects.get(id=1)
except Exception as ex:
    log.error("Someone stole my question", ex)
```
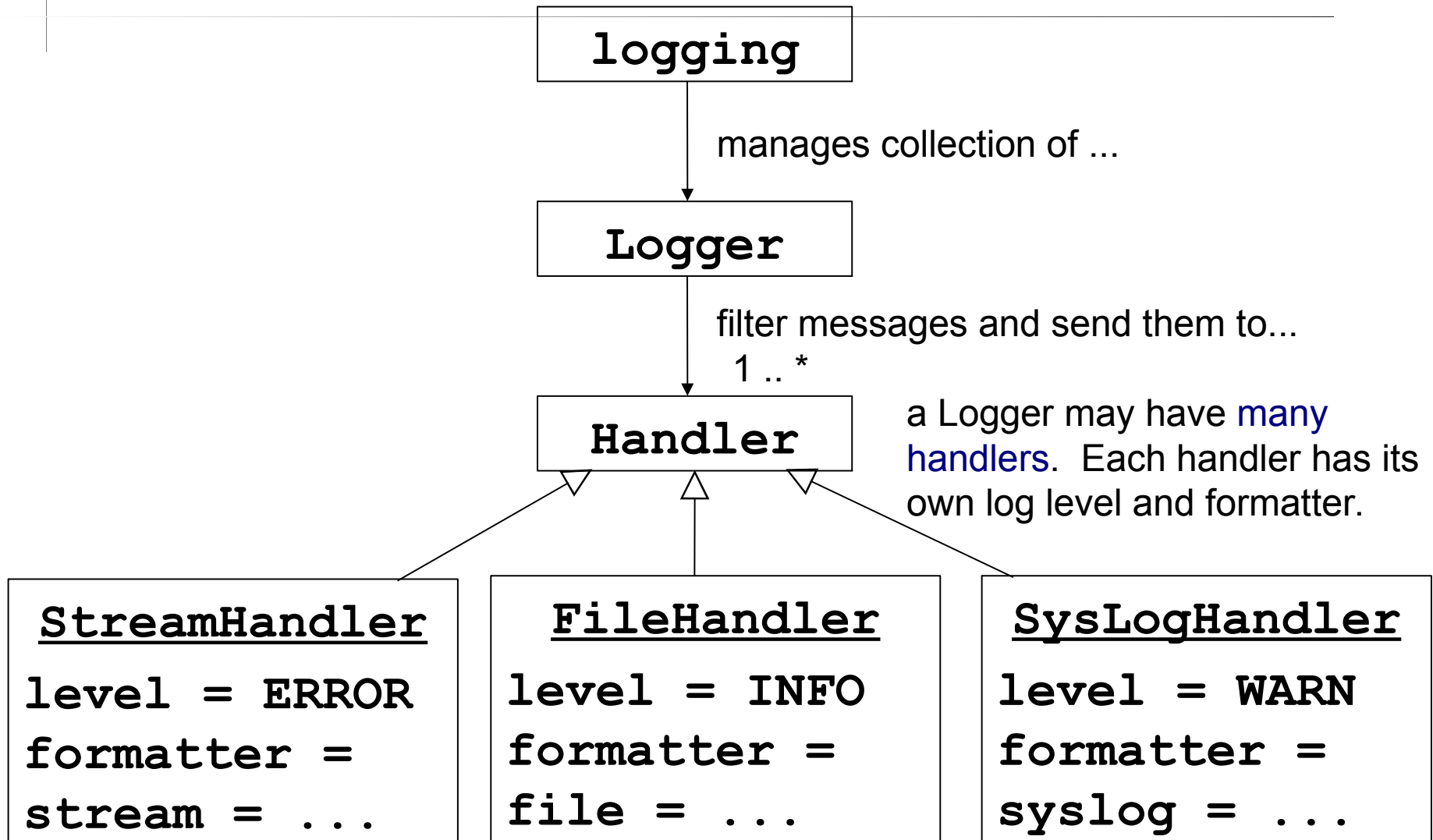
Logging Architecture

logging

manages collection of ...

Logger

filter messages and send them to...
1 .. *

Handler

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

StreamHandler	**FileHandler**	**SysLogHandler**
level = ERROR	level = INFO	level = WARN
formatter =	formatter =	formatter =
stream = ...	file = ...	syslog = ...

getLogger("name") is a Singleton

Only one instance for each named logger.

Good! *We can get a logger whenever we need it.*

```
log1 = logging.getLogger( "auth" )
log2 = logging.getLogger( "auth" )
log3 = logging.getLogger( "AUTH" )
# are they the same object?
log1 is log2
True
# is the logger name case sensitive?
log1 is log3
False
```

How to Use the 5 Log Levels

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

CRITICAL

ERROR

WARNING

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),...)

Web App Logging

Web Apps have some special concerns:

1. want to know 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. Login - username, IP address, date-time

2. Logout

3. Errors and exceptions

4. Deployment

5. User activity - at least all activity that changes something

6. Invalid requests

Logging Done Wrong

Facebook stored 200 - 600 Million users' passwords in **plain text** in log files for years.

https://krebsonsecurity.com/2019/03/facebook-stored-hundreds-of-millions-of-user-passwords-in-plain-text-for-years/

What Info Should You Log When...

1. A failed or successful login occurs.

> username

> IP address

> date/time

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

> question and choice he voted

> which session or IP address he voted from

>

Learn Python Logging

Logging HOWTO in the Python Library docs

- Use guide in the Advanced Tutorial.

- <u>Don't</u> use the Basic Tutorial (static log methods)

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

Logging - Logging Facility for Python

- in the Python library docs

- configuration, using formats, and handlers

How to Configure Logging

- `https://docs.python-guide.org/writing/logging/`

- 3 ways: .INI file, a dict or JSON file, function calls

Learn Python Logging

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 on Logging (only 10 pages with many examples)

Configuration: Django uses JSON-format text to configure loggers in `settings.py`.

Configure Django Logging

```python
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, not settings.py.

   - OK to partially configure in settings.py but get details from configuration file

Java Has More Log Levels

java.util.logging

SEVERE

WARNING

INFO

(CONFIG)

FINE   - stupid name

FINER - stupider

FINEST

OFF

ALL

Log4J & SLF4J Levels

FATAL

ERROR

WARN

INFO

DEBUG

TRACE

OFF

ALL