JIDS : A Host-based Intrusion Detection System for GNU/Linux

Andrey Ferriyan
Assalaam Teknologi
Yogyakarta, Indonesia
andrey.ferriyan@gmail.com

***Abstract*—Intrusion Detection System (IDS) is application which monitors system or network anomaly. Based on category there are two kinds of IDS, Host-based and Network-based. Jibril Intrusion Detection System (JIDS) is a stand alone Host-based IDS. JIDS propose file integrity checking for detecting anomaly in system. All binary system will be checked using md5 hash. If the hash doesn't match with the database, it will be marked as suspicious file, logged and should be checked.**

Keywords—Intrusion; Intrusion Detection System; Host-based Intrusion Detection

I. Introduction

Nowadays security becomes an important issue. New threats potentially more lethal can cause data leak, data theft or even destroying data. If intrusion from malware or trojans occured, it can compromise confidentiality, integrity and availability from data. At extreme levels a compromise system can shut down a system or a network.

As normal user we don't know whether the binary systems we use are harmful or not. Trojan or backdoor application injecting the binary is very dangerous. For example the backdoor application can run normally as usual but in the same time opening port for communication to the internet. This port is the shortcut for attacker to controlling the infected systems. Without any prevention, detection or proper firewall configuration the attacker can make the infected systems their stepping stone to attack another networks. In worst case the backdoor agent can communicate with other agent to create Distributed Denial of Service attack.

This is why Intrusion Detection System (IDS) is needed and the methods are always improved. IDS become more important. IDS is application which monitors system or network anomaly. Based in category there are two kinds of IDS, Host-based and Network-based[1]. There is also a hybrid, merge between host-based and network-based. A Host-based Intrusion Detection System (HIDS) resides on a host system or specific server and usually works as stand alone system.

Specifically, HIDS focus on specific parts. HIDMN, a host-based and network-based focus on mobile networks[2]. Expert-BSM focus on analyzing audit trails in a real time[3]. [4] focus on log file analysis and [5] focus on process file activity monitoring. For detection technique there are several methods used by researchers, anomaly and misuse detection [6]. In misuse detection, the signature of known attacks usually stored in database. Any similar data if checked is considered attacks. In anomaly detection the intrusion correspond to deviations from the normal activity of system. The anomaly detection system has high false positive / negative alarm rate compared to misuse detection systems. The anomaly detection usually use supervised or unsupervised learning to make high detection rate.

In this paper JIDS use misuse detection offers another method to improve the detection. JIDS using file integrity checking for detection new intrusion. JIDS using md5 hash for integrity checking. Every single file within the system will be recorded and stored its hash code in a database.

II. JIDS Flowchart

JIDS design flowchart can be seen in Figure 1.

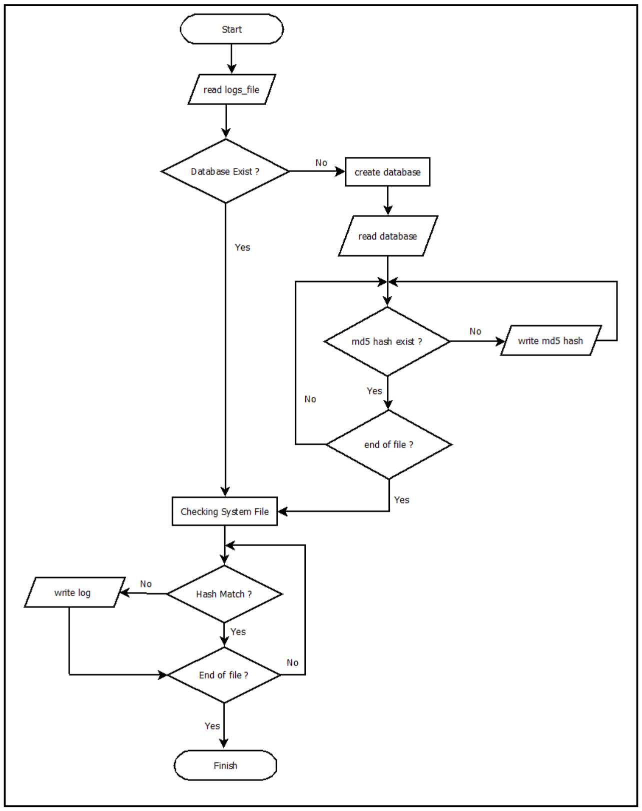


Figure 1. Flowchart

JIDS using database to stored hash from every binary system files. Usually in GNU/Linux or UNIX-like operating system the binary file locate in four different folders. In /usr/bin, /usr/sbin, /bin, /sbin. Therefore, JIDS will monitoring each file within four folders system. JIDS is not like antivirus program that will repair the file but only detecting the different hash from the database. If the hash file is different than the record from the database there are two possibilities. First there must be an update from the system file. Second the system file is compromise. If the system file compromise then it needs further checking.

III. JIDS Architecture

JIDS architecture have three modules. Initialization, checking and logging. Because there is no relation between tables, so it's not necessary creating ERD.

A. *Initialization Module*

This module will be executed by creating database for the first fime. If the database is not exists than JIDS will generate for the first time and record all files in the system folder. If the database is exist than it forward to another module. JIDS using SQLite as the database for stored the hash file. Initialization module will create four tables. One table each for system folders.

After each tables created then next step is to record all the hash in system folder. Table I contain all the hash file from /bin directory in the system folder. Table II contain all hash file from /usr/bin directory and so Table III and Table IV.

TABLE I. Dir_bin Table

Field Name	**Tipe**	**Information**
id_dirbin	Integer	Primary Key
file_name	varchar(20)	
md5sum	varchar(50)	

TABLE II. Dir_usrbin Table

Field Name	**Tipe**	**Information**
id_dirusrbin	Integer	Primary Key
file_name	varchar(20)	
md5sum	varchar(50)	

TABLE III. Dir_sbin Table

Field Name	**Tipe**	**Information**
id_dirusrbin	Integer	Primary Key
file_name	varchar(20)	
md5sum	varchar(50)	

TABLE IV. Dir_usrsbin Table

Field Name	**Tipe**	**Information**
id_dirusrbin	Integer	Primary Key
file_name	varchar(20)	
md5sum	varchar(50)	

B. *Checking Module*

After initialization module run next step is opening database. This module will check every hash file in the four system file folders.

C. *Logging Module*

Every file will be check by checking module and compare it with database. If the hash files being checked is different, than the logging module will record it in the log file.

IV. Implementation

A. *Read Log File*

Initialization module open and read log file using following command in python script. The “today” variable will contains current date when executed.

```
today = time.asctime()
        logdate = "%s\n" % (today)
        try:
         f = open(logs_file,"a")
         f.write("----------- log start -------------\n")
```

If log file doesn’t exist then following command will be executed for creating new log file.

```
except:
         f = open(logs_file,"w")
         f.write("----------- log start -------------\n")
         f.write(logdate)
         f.write("New File :\n")
```

B. *Create Database*

JIDS using sqlite for the database. Using python script sqlite database will be create using following command.

```
con = sqlite.connect(DB_FILE)
cur = con.cursor()
     cur.execute("""
             create table dir_bin
             (
               id_dirbin        integer primary key,
               file_name        varchar(20),
               md5sum           varchar(50)
             )
  """)
```

This command will create dir_bin table that contain every single binary hash file using following command.

```
for ibin in range(lensrcbin):
    join_bin = os.path.join(PATHBIN,SRCBIN[ibin])
     try:
        fbin = open(join_bin,'rb')
    except IOError:
              print "%s direktori... lewati" % (join_bin)
      continue
    databin = fbin.read()
          fbin.close()
          mbin = md5.new()
          mbin.update(databin)
     mbindigest = mbin.hexdigest()
    con = sqlite.connect(DB_FILE)
          cur = con.cursor()
    cur.execute("INSERT                              INTO
    dir_bin(file_name,md5sum)
    VALUES('"+join_bin+"','"+mbindigest+"')")
    con.commit()
```

C. *Checking System*

Checking system then works by comparing each file from directory system using this python script as follows.

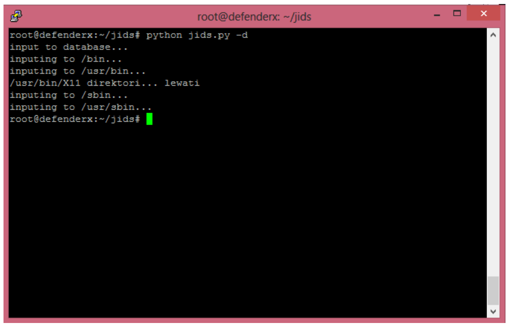
```
join_bin = os.path.join(PATHBIN,SRCBIN[ibin])
binexec = cur.execute("SELECT md5sum FROM dir_bin
WHERE file_name='"+ join_bin +"'")
binresult = cur.fetchone()
```

D. *Write Log File*

If hash file being checked is different, logging module will write in log file as a compromise file.

V. RESULT

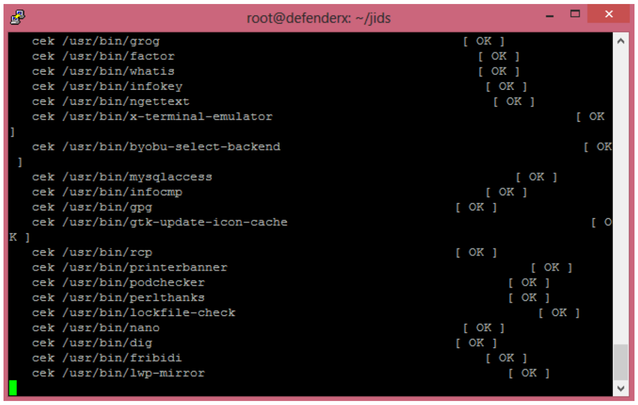
Figure 2 is the screenshots that illustrates initialization in the first time.



```
root@defenderx:~/jids# python jids.py -d
input to database...
inputing to /bin...
inputing to /usr/bin...
/usr/bin/X11 direktori... lewati
inputing to /sbin...
inputing to /usr/sbin...
root@defenderx:~/jids#
```

Figure 2. Initialization process

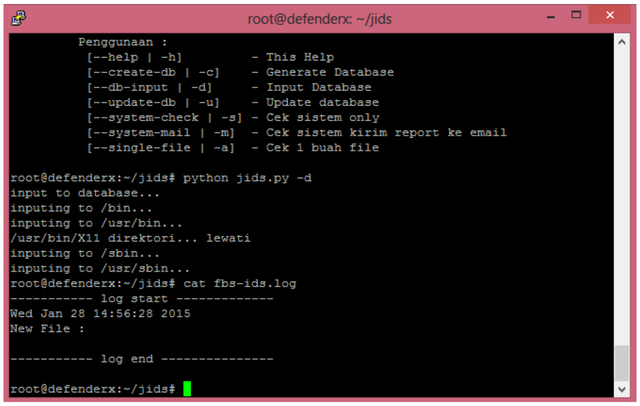
Figure 3 is the screenshots that illustrates checking all binary file from systems.



```
   cek /usr/bin/grog                                          [ OK ]
   cek /usr/bin/factor                                          [ OK ]
   cek /usr/bin/whatis                                          [ OK ]
   cek /usr/bin/infokey                                          [ OK ]
   cek /usr/bin/ngettext                                          [ OK ]
   cek /usr/bin/x-terminal-emulator                                          [ OK
]
   cek /usr/bin/byobu-select-backend                                          [ OK
 ]
   cek /usr/bin/mysqlaccess                                          [ OK ]
   cek /usr/bin/infocmp                                          [ OK ]
   cek /usr/bin/gpg                                          [ OK ]
   cek /usr/bin/gtk-update-icon-cache                                          [ O
K ]
   cek /usr/bin/rcp                                          [ OK ]
   cek /usr/bin/printerbanner                                          [ OK ]
   cek /usr/bin/podchecker                                          [ OK ]
   cek /usr/bin/perlthanks                                          [ OK ]
   cek /usr/bin/lockfile-check                                          [ OK ]
   cek /usr/bin/nano                                          [ OK ]
   cek /usr/bin/dig                                          [ OK ]
   cek /usr/bin/fribidi                                          [ OK ]
   cek /usr/bin/lwp-mirror                                          [ OK ]
```

Figure 3. Checking system process

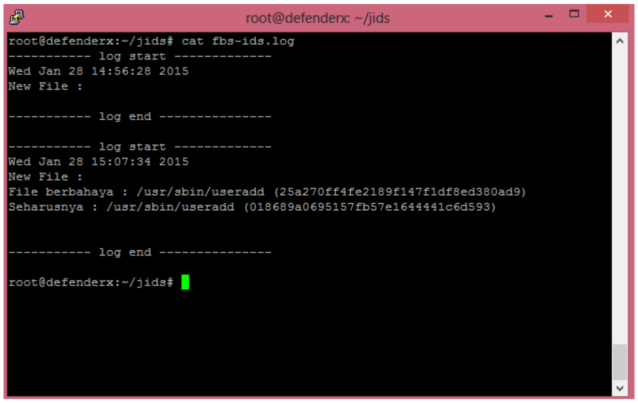
For the simulation we replace a command from /usr/sbin directory and change it with another script for replacement. We choose *useradd* program. *Useradd* is a program for create new user in GNU/Linux system. Figure 4 is the screenshots from log file if there is no intrusion or any new file.



```
         Penggunaan :
          [--help | -h]         - This Help
          [--create-db | -c]    - Generate Database
          [--db-input | -d]     - Input Database
          [--update-db | -u]    - Update database
          [--system-check | -s] - Cek sistem only
          [--system-mail | -m]  - Cek sistem kirim report ke email
          [--single-file | -a]  - Cek 1 buah file

root@defenderx:~/jids# python jids.py -d
input to database...
inputing to /bin...
inputing to /usr/bin...
/usr/bin/X11 direktori... lewati
inputing to /sbin...
inputing to /usr/sbin...
root@defenderx:~/jids# cat fbs-ids.log
----------- log start -------------
Wed Jan 28 14:56:28 2015
New File :

----------- log end ---------------

root@defenderx:~/jids#
```

Figure 4. Log file for no intrusion

Figure 5 is the screenshots illustrates if there is intrusion.



```
root@defenderx:~/jids# cat fbs-ids.log
----------- log start -------------
Wed Jan 28 14:56:28 2015
New File :

----------- log end ---------------

----------- log start -------------
Wed Jan 28 15:07:34 2015
New File :
File berbahaya : /usr/sbin/useradd (25a270ff4fe2189f147f1df8ed380ad9)
Seharusnya : /usr/sbin/useradd (018689a0695157fb57e1644441c6d593)


----------- log end ---------------

root@defenderx:~/jids#
```

Figure 5. Log file for intrusion

VI. Conclusion

There are many proposed method for detecting intrusion. JIDS using file integrity can detect new intrusion from trojan or backdoor application that reside in the systems. Host-based analysis often plays important role as a complement to network-based. The accuracy is high if it combine with task scheduler and run periodically.

REFERENCES

[1] S. Sonawane, S. Pardeshi, and G. Prasad, "A survey on intrusion detection techniques," vol. 2, no. 3, pp. 127–133, 2012.

[2] S. Bijani and M. K. a., "HIDMN: A Host and Network-Based Intrusion Detection for Mobile Networks," *2008 Int. Conf. Comput. Electr. Eng.*, pp. 204–208, Dec. 2008.

[3] S. Solaris, U. Lindqvist, P. A. Porras, and M. Park, "eXpert-BSM : A Host-based Intrusion Detection Solution," 2001.

[4] Y. Lin, Y. Zhang, and Y. Ou, "The Design and Implementation of Host-Based Intrusion Detection System," *2010 Third Int. Symp. Intell. Inf. Technol. Secur. Informatics*, pp. 595–598, Apr. 2010.

[5] N. A. Tien, "Malware Detection by Process ' s File Activity Monitoring Nguyen Anh Tien Keio University Advisors : Professor Hideyuki Tokuda Professor Jun Murai Associate Professor Hiroyuki Kusumoto Professor Osamu Nakamura Associate Professor Kazunori Takashio Assista," 2012.

[6] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey," *EURASIP J. Wirel. Commun. Netw.*, vol. 2013, no. 1, p. 271, 2013.