1. Home . . . 2
- 1.1 1- Setup Database . . . 2
- 1.2 2-Setup Collections and Data Sources . . . 3
- 1.3 3- Import Proxy Logs . . . 4
- 1.4 4- Import Web Logs . . . 4
- 1.5 5- Proxy Log Beacon Detection . . . 4
- 1.6 6- Proxy Log Traffic Analysis . . . 5
- 1.7 7- Web Server SQL Injection and XSS Detection . . . 7
- 1.8 8- Web Server Traffic Analysis . . . 8
- 1.9 9- Index of Suspicion . . . 9
- 1.10 10- Web Interface . . . 10
- 1.11 11- Advanced Querying . . . 17
- 1.12 12- Scaling LATK . . . 18
- 1.13 Bulk Name Resolver . . . 19
- 1.14 Downloads . . . 19
- 1.15 Installation . . . 21
  - 1.15.1 Fedora Installation . . . 21
  - 1.15.2 Installation and Training Manual . . . 22
  - 1.15.3 Mac OS X Installation . . . 23
  - 1.15.4 Ubuntu Installation . . . 24

Home

This is the homepage for the **Log Analysis Tool Kit.**

This is a collection of command line and web based tools for use in incident response and long term analysis use as part of ongoing situational awareness. Often when responding to a security incident the only files available are web server and proxy server logs. The tools here will aid you in detecting odd traffic such as botnet beaconing and SQL Injection attempts. The large amount of data can be overwhelming and the tools in the Log Analysis Tool Kit can be used to parse these files and build a MySQL database for querying.

Currently the log formats supported are:

Proxy Logs:

- Squid
- Bluecoa**t**

Web Server Logs:

- Apache
- IIS

Your feedback is always appreciated. Please report any issues or enhancement requests to the author.

The tools are written in Python3 and PHP. The tool kit has been tested on Mac OSX and Fedora.

Navigate space

1- Setup Database

The first step in using the Log Analysis Tool Kit (LATK) is to create a database. A script is included with LATK to do this.

latk-createdb.py

Usage:

```
giskard:logs joe$ ./latk-createdb.py
OK: Created database LogAnalysisToolKit
OK: Created created tables
```

By default the tools use a local MySQL Server and the user root with no password, you will need to edit the file latk-mysql.conf to match your environment.

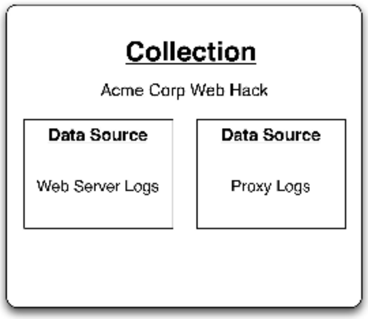
```
[mysql]
user = root
pass =
host = localhost
dbName = LogAnalysisToolKit
dbSock = /var/lib/mysql/mysql.sock
```

You can tune the MySQL database for better performance. On the MySQL server edit the my.cnf file, usually located in /etc. Variables to look at are, key_buffer_size sort_buffer, max_heap_table_size sort_buffer_size, innodb_log_file_size, innofb_log_buffer_size. Some example are below.

```
key_buffer_size=256M
sort_buffer= 64M
max_heap_table_size= 1G
max_allowed_packet = 1M
table_open_cache = 256
sort_buffer_size = 128M
read_buffer_size = 129M
query_cache_size= 32
thread_concurrency = 4innodb_buffer_pool_size = 4G
innodb_additional_mem_pool_size = 256M
innodb_log_file_size = 1G
innodb_log_buffer_size = 256M
```

2-Setup Collections and Data Sources

The Log Analysis Tool Kit works using the concept of collections and data sources. As an example you would create a collection for investigating an incident on WWW Server. Collections and Data Sources can be added through the script latk-setup.py or using the LATK Web Interface.



latk-setup.py

First create a collection, the options are shortname and description in quotes.

```
giskard:logs joe$ ./latk-setup.py collection wwwattack "WWW Server Attack on 6/1/2011"
OK: collections table found
OK: datasources table found
Adding new collection wwwattack with id 1
```

Next add data sources to your collection, the options are shortname, description, log type and the ID of the collection created above.

```
giskard:logs joe$ ./latk-setup.py datasource wwwlogs "Logs from the www server" apache
1
OK: collections table found
OK: datasources table found
Adding new datasrouce wwwlogs with id 1
giskard:logs joe$
```

To display the collections and data sources the show option is passed to latk-setup.py.

```
giskard:logs joe$ ./latk-setup.py show
OK: collections table found
OK: datasources table found
collection_id, name, description
1 , wwwattack , WWW Server Attack on 6/1/2011
datasource name, datasource id, datasource type,  collection name, collection id
wwwlogs , 1 , apache , wwwattack , 1
```

3- Import Proxy Logs

To import proxy logs the tool latk-proxyimport.py is used.

The syntax:

```
giskard:logs joe$ ./latk-proxyimport.py
ERROR: Must supply log file name and dataSourceID.
USAGE: ./latk-proxyimport.py LOG_FILE_NAME datasource_id
```

An example of loading a squid log file would be:

```
giskard:logs joe$ ./latk-proxyimport.py squid/access.log 4
Logfile: squid/access.log
OK: proxyData table found
Creating list of Connections.
20934 records.
```

4- Import Web Logs

To import web logs the tool latk-wwwimport.py is used.

The syntax is:

```
giskard:logs joe$ ./latk-wwwimport.py
ERROR: Must supply log file name and data source id.
USAGE: ./latk-wwwimport.py LOG_FILE_NAME Data_Source_ID
```

An example of loading an apache log is below:

```
giskard:logs joe$ ./latk-wwwimport.py apache/combined.log 1
Logfile: apache/combined.log
OK: webData table found
Creating list of Connections.
Imported 917 records.
Import Complete
```

5- Proxy Log Beacon Detection

The tool latk-proxyreport.py is a python script that attempts to detect "beaconing" behavior in Squid or Bluecoat proxy logs.

The script uses the fields of the logs to rank the connections as beacons. It looks at:

- Time between connections
- Regularity of connections
- Transmission type
- Occurrence of destination host
- Number of connections
- Type of address.

And uses this information to create a score of High, Medium or Low.

To run the program you will need to have python version 3.0 or greater and numpy installed. The syntax for execution is latk-proxybeacon.py *beacongen datasource_id.*

```
giskard:logs joe$ ./latk-proxyrbeacon.py
ERROR: Must supply report type dataSourceID.
USAGE: ./latk-proxyreport.py reportType  datasource_id
USAGE: ./latk-proxyreport.py beacongen X
```

After you have imported the proxy log data, you will need to generate the beacon score using

```
giskard:logs joe$ ./latk-proxybeacon.py beacongen 5
OK: proxyData table found
Data Source Type is:bluecoat
Creaiting Memory Table
Creating list of Unique Source and Dest IPs.
154253 records.
Processing 101 of 154253
Processing 202 of 154253
....
```

The output can be viewed as a chart with results ranked by their score.

```
Client IP |                Dest IP |   Count | Mean Time |     Std Dev |   Score
--------------------------------------------------------------------------------------
-------------------------
     192.168.1.119 |            badguy.com |       2 |      4109 |           0 |
High
     192.168.1.120 |       www.example.com |       8 |   6602.43 |      2.608  |
Medium
     192.168.1.119 |   platform.twitter.com |       4        4528 |      3.55   |
Medium
     192.168.1.119 |       cdn.nflximg.com |       2 |        35 |           0 |
  Low
     192.168.1.120 |           s.ytimg.com |       2 |         2 |           0 |
  Low
     192.168.1.119 |           www.xml.org |      31 |      0.03 |        0.18 |
  Low
     192.168.1.119 |      www.symantec.com |      27 |      0.04 |       0.192 |
  Low
     192.168.1.119 |           www.usa.com |      52 |      0.04 |       0.194 |
  Low
```

6- Proxy Log Traffic Analysis

Proxy logs are very useful for finding summary information of connections to the internet. The tool latk-proxyreport.py can generate two type of traffic reports, summary reports of each internal hosts traffic to the internet and detailed "by connection" reports.

The tool assigns directionality of traffic, whether it is incoming or outgoing, and the amount of data transmitted, in the case of Bluecoat logs it also displays the amount of data both in and out.

The program is run by supplying the type of proxy log, the type of report and the log file name:

```
giskard:logs joe$ ./latk-proxyreport.py
USAGE: ./latk-proxyreport.py reportType  datasource_id
USAGE: ./latk-proxyreport.py (beacongen|beaconshow|summary|detail) X
```

```
giskard:logs joe$ ./latk-proxyreport.py  summary 4
OK: proxyData table found
Data Source Type is:squid
Creating list of Connections.
       Client IP |     Bytes |  # of Hosts
------------------------------------------
   192.168.1.120 |  2837513618 |        65
   192.168.1.119 |    39618709 |       175
   192.168.1.160 |    32202862 |        59
```

A sample bluecoat report as run as follows, note the addition of the bytes in and bytes out field.

```
giskard:logs joe$ ./latk-proxyreport.py summary 5
OK: proxyData table found
Data Source Type is:bluecoat
Creating list of Connections.
           Client IP  |     Bytes In | Bytes Out |Total Bytes|   # of Hosts
----------------------------------------------------------------------
       192.168.1.119 |       2302508 |    375041 |    2677549 |            16
       192.168.1.120 |       1944917 |    307842 |    2252759 |            27
       192.168.1.110 |        247750 |       180 |     247930 |             1
       192.168.1.122 |         19044 |     12990 |      32034 |             2
       192.168.1.123 |          5662 |       346 |       6008 |             1
       192.168.1.150 |          1911 |      1440 |       3351 |             2
       192.168.1.15  |          2831 |       115 |       2946 |             1
       192.168.1.111 |          2831 |       115 |       2946 |             1
```

A detail report shows each connection, the directionality and the amont of traffic in each connection.

A sample squid report is below:

```
giskard:logs joe$ ./latk-proxyreport.py  detail   4
OK: proxyData table found
Data Source Type is:squid
Logfile: squid/access.log
Creating list of Connections.
           Client IP |                                                 Dest IP |
Direction |# of Recs. |Total Bytes |
--------------------------------------------------------------------------------------
----------------------------
     192.168.1.120 |                   netflix903.as.nflximg.com.edgesuite.net |
       in |          483 |    591126658
     192.168.1.119 |                   netflix902.as.nflximg.com.edgesuite.net |
       in |          483 |    588357872
     192.168.1.160 |                                  netflix-897.vo.llnwd.net |
       in |          471 |    583177309
     192.168.1.120 |                   netflix904.as.nflximg.com.edgesuite.net |
       in |          477 |    556416812
```

In the Bluecoat log example below note the addition of bytes in, bytes out, and total bytes.

```
giskard:logs joe$ ./latk-proxyreport.py detail 4
Logfile: bluecoat/access.log
Creating list of Connections.
            Client IP |                                                     Dest IP |
  Direction |# of Recs. |   Bytes In | Bytes Out |Total Bytes |
--------------------------------------------------------------------------------------
--------------------------------------------
       192.168.1.120 |                      netflix903.as.nflximg.com.edgesuite.net |
             in |             2 |     1447932 |          396 |       1448328
       192.168.1.120 |                      netflix902.as.nflximg.com.edgesuite.net |
             in |            11 |      400025 |       161504 |        561529
       192.168.1.121 |                                     netflix-897.vo.llnwd.net |
             in |            17 |      328393 |        16407 |        344800
       192.168.1.120 |                                               www.google.com |
             in |           132 |      239283 |       103439 |        342722
       192.168.1.160 |                                                  www.cnn.com |
             in |            85 |      246838 |        64122 |        310960
       192.168.1.119 |                                                www.yahoo.com |
             in |             1 |      247750 |          180 |        247930
```

7- Web Server SQL Injection and XSS Detection

latk-reportwwwlogs.py is a python script that detects SQL Injection(SQLi) and Cross Site Scripting(XSS) attempts in your web server logs. Currently loginspector.py supports Apache and IIS logs.

**Note - To view the XSS/SQLi report from the web interface it must be run from the command line first.

The syntax is:

```
giskard:logs joe$ ./latk-wwwreport.py
ERROR: Must supply report type and data source id.
USAGE: ./latk-wwwreport.py (sqli|pagediff|traffic) Data_Source_ID
```

Example usage is:

```
giskard:logs joe$ ./latk-wwwreport.py sqli 1
OK: webData table found
------------------------------------------

Alert Type |  Client IP     |  URL
--------------------------------------------------
      SQLi |  192.168.1.119 | i=d'z"0 HTTP/1.1
      SQLi |  192.168.1.119 | i=d'kc"z'gj'"%2A%2A5%2A(((%3B-%2A%60) HTTP/1.1
      SQLi |  192.168.1.119 | i=<%21-- HTTP/1.1
      SQLi |  192.168.1.119 | i=%3Bping+-n+3+localhost HTTP/1.1
      SQLi |  192.168.1.119 | i=%3Bping+-c+4+localhost HTTP/1.1
       XSS |  192.168.1.119 | i=<ScRIpT>fake_alert(String.fromCharCode(UXBY))</SCriPT>
HTTP/1.1
       XSS |  192.168.1.119 | i=<ScRIPT>a=/UXBY/%0Afake_alert(a.source)</SCRiPT>
HTTP/1.1
      SQLi |  192.168.1.119 | i=%3Btype+%25SYSTEMROOT%25%5Cwin.ini HTTP/1.1
      SQLi |  192.168.1.119 | i=<%21--%23include+file="/etc/passwd"--> HTTP/1.1
      SQLi |  192.168.1.119 | i=<%21--%23include+file="C:%5Cboot.ini"--> HTTP/1.1
```

8- Web Server Traffic Analysis

latk-reportwwwlogs.py is a tool that can be run against web server logs(Apache and IIS) to generate various useful bits of investigative information. Please note python 3 is required.

The current functionality of the tool includes:

- Bulk import log files to a database, this allows SQL queries to be run against them.
- Generate list of most popular pages by bytes and hits
- Generate list of top clients by pages and bytes
- On IIS the ability to generate a list of top uploaders.
- Ability to detect new files on a web server

Example usage is:

```
giskard:logs joe$ ./latk-wwwreport.py
ERROR: Must supply report type and data source id.
USAGE: ./latk-reportwwwlogs.py (sqli|pagediff|traffic) Data_Source_ID
```

Traffic Switch

A sample report using the traffic switch produces output like below.

```
giskard:logs joe$ ./latk-wwwreport.py traffic 1
OK: webData table found
Creating list of Connections.
       Client IP |     Bytes |
------------------------------------------
   10.125.67.243 |     1703952
   10.249.71.116 |     1505635
    10.181.18.13 |     1277964
   10.46.195.236 |      784646
   10.14.225.169 |      535381
....
       Client IP |  # of Pages
------------------------------------------
   10.249.71.116 |       382
   10.252.20.193 |        35
   10.245.245.67 |        32
   10.195.114.53 |        26
    10.106.45.12 |        23
   10.124.227.98 |        22
   10.157.251.25 |        19
....
                                                    Page |  # of Bytes
------------------------------------------------------------------------
                                       /~joe/example.mpg | 4345896.0
                                /%7Ejoe/example/foo.html |   4291329
                                         /~joe/index.hml | 3206775.0
                                         /cgi-bin/foo.pl |   1160309
                                /foo.html |    784427
                                                       / |  563940.0
...
```

Pagediff Switch

A sample report using the pagediff switch looks like below:

```
giskard:logs joe$ ./latk-wwwreport.py pagediff 1
OK: webData table found
Current Week Total: 33
Previous Week Total: 32
----------------------------------------------------------------------
New files in the current week
----------------------------------------------------------------------
/Xymsm
/foo.html
/cgi-bin/exampl.pl
/YTCKQ
/pcap.php
----------------------------------------------------------------------
Files in the previous week not present in the most current week
----------------------------------------------------------------------
/file.html
```

9- Index of Suspicion

To five you a starting point to begin your investigation the Log Analysis Tool Kit generates an index of suspicion based on all of the data available to the specific source format.

To run this report use the script latk-ios.py with the datasourceID specified as the only option. Example usage is:

```
giskard:logs joe$ ./latk-ios.py
ERROR: Must supply data source id.
USAGE: ./latk-ios.py Data_Source_ID
```

Example output is below.

```
giskard:logs joe$ ./latk-ios.py 1
OK: webData table found
Data Source Type is:apache
Creating Baseline
Generating Suspicious IP Index
       Client IP |    % TTL Hits |   % TTL Pages |   % TTL Bytes |% TTL XSS/SQLi
|Suspicion Index |
--------------------------------------------------------------------------------------
--------------
     192.168.1.119         67.14 |         90.48 |          33.1 |         95.24
|        285.96
      192.168.1.12         13.31 |          38.1 |         28.72 |             0
|         80.13
       192.168.1.9         10.25 |         33.33 |         35.05 |             0
|         78.63
     192.168.1.117          9.31 |         28.57 |          3.13 |          4.76
|         45.77
```

10- Web Interface

The Log Analysis Tool Kit includes a web front end that works on Mac OS X, Windows and Linux. Data should still be imported on the command line using the python tools. But reporting can be run through the web front end.

Database connectivity is defined in the same file as the python tools /etc/latk-mysql.conf. Edit the following lines to match your environment.

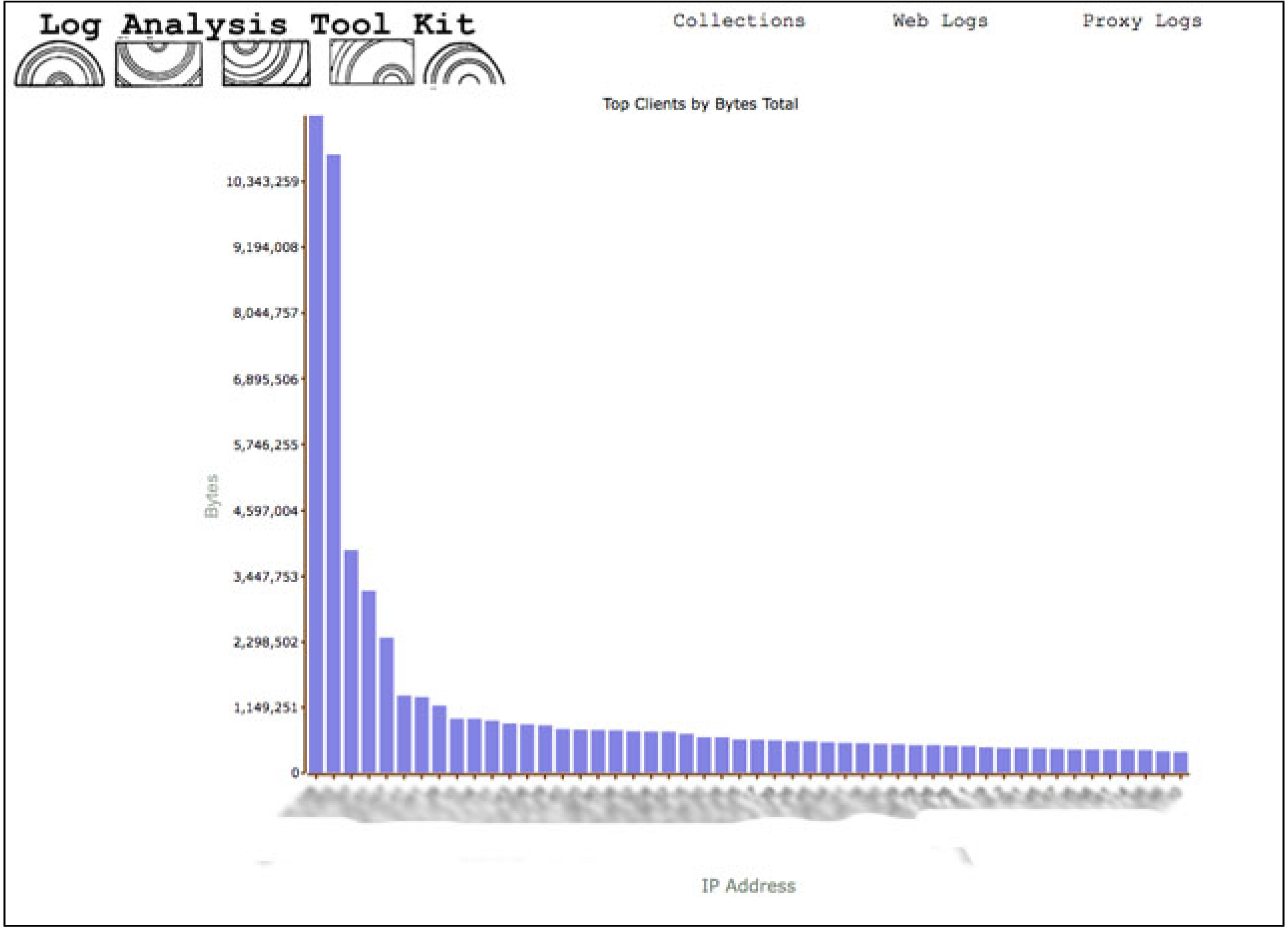
```
[mysql]
user = root
pass =
host = localhost
sock = /var/lib/mysql/mysql.sock
dbName = LogAnalysisToolKit
```

The web interface has two default users, admin/latk and guest/potato. You can change the passwords via sql.

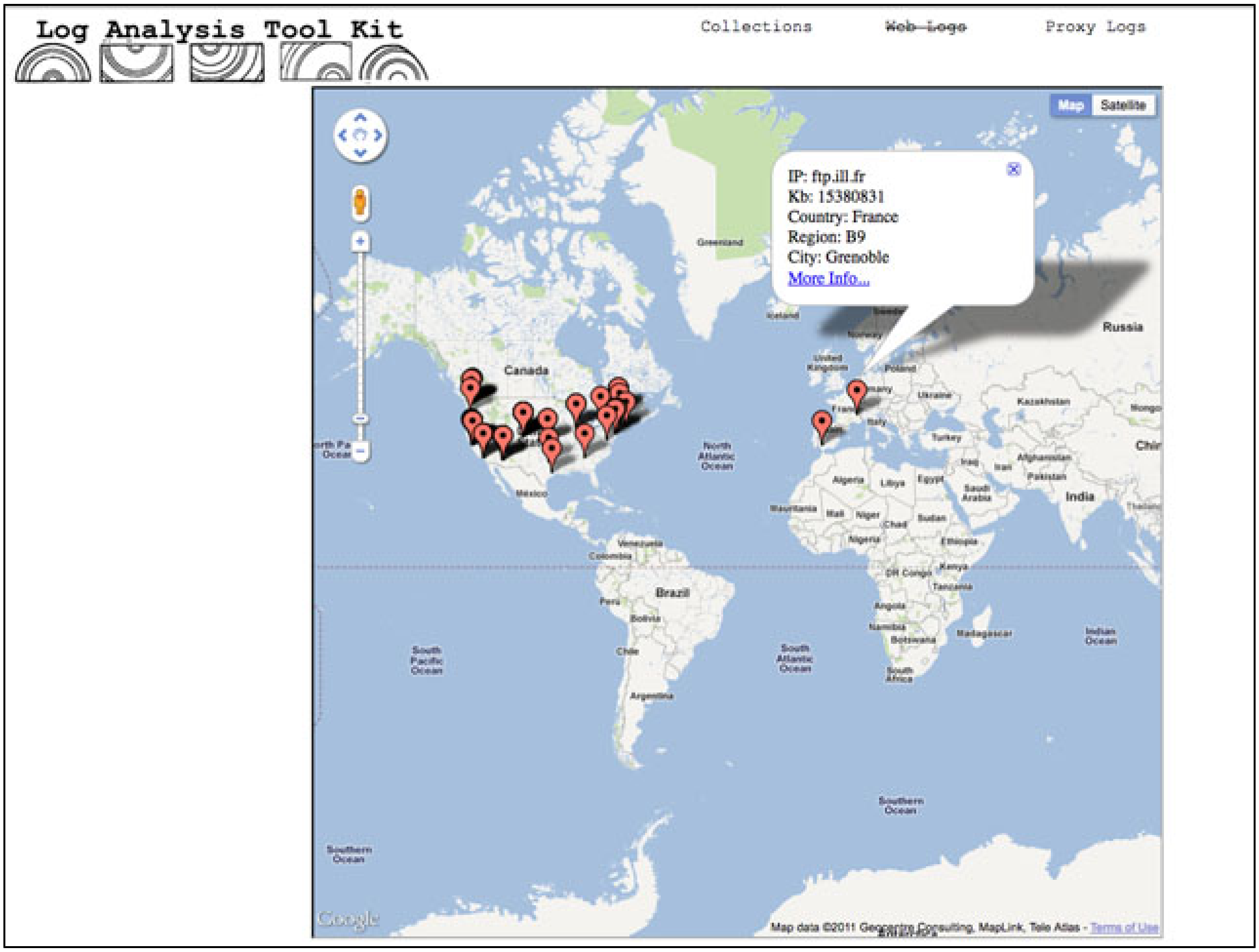
```
[root@localhost /tmp]# mysql -p LogAnalysisToolKit
Enter password:
mysql> update users set password='NewPassword' where username='admin';
Query OK, 1 rows affected (0.02 sec)
Rows matched: 1Â  Changed: 1Â  Warnings: 0
```

Screenshots

Proxy Server Log Analysis Clients by Total Bytes

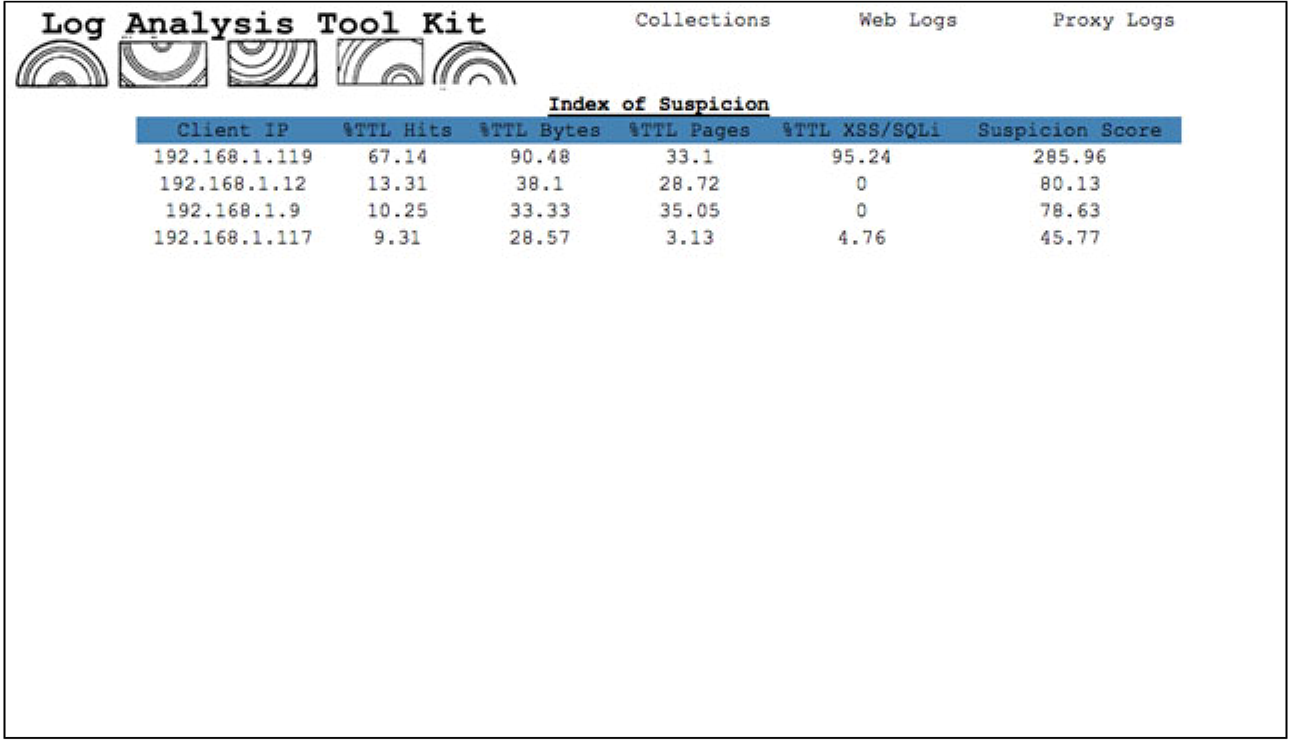


Geographic Map of Log Data



IP: ftp.ill.fr
Kb: 15380831
Country: France
Region: B9
City: Grenoble
More Info...

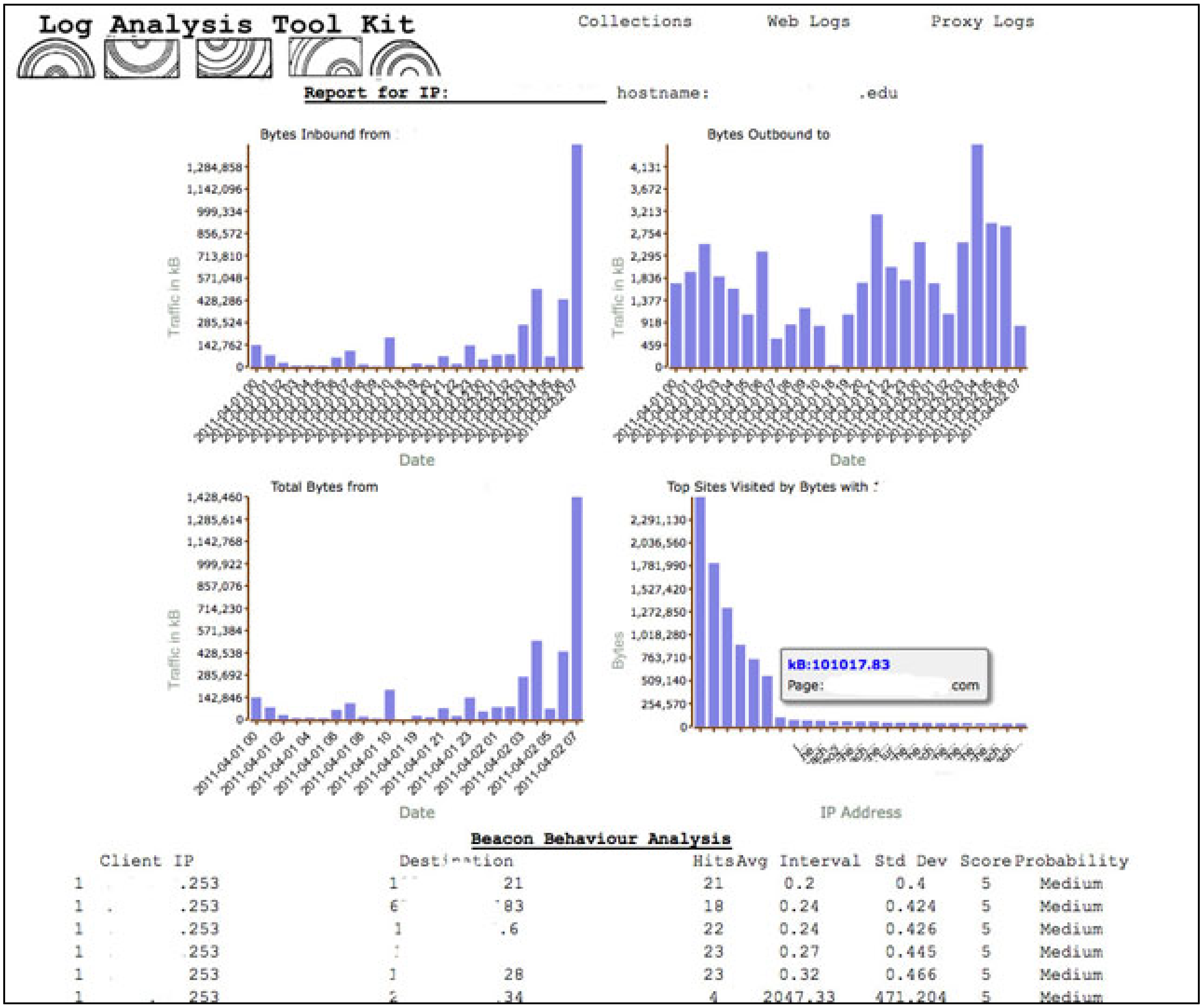
Web Server Log Index of Suspicion Report.



Index of Suspicion

Client IP	%TTL Hits	%TTL Bytes	%TTL Pages	%TTL XSS/SQLi	Suspicion Score
192.168.1.119	67.14	90.48	33.1	95.24	285.96
192.168.1.12	13.31	38.1	28.72	0	80.13
192.168.1.9	10.25	33.33	35.05	0	78.63
192.168.1.117	9.31	28.57	3.13	4.76	45.77

Proxy Server Log Analysis Clients Report

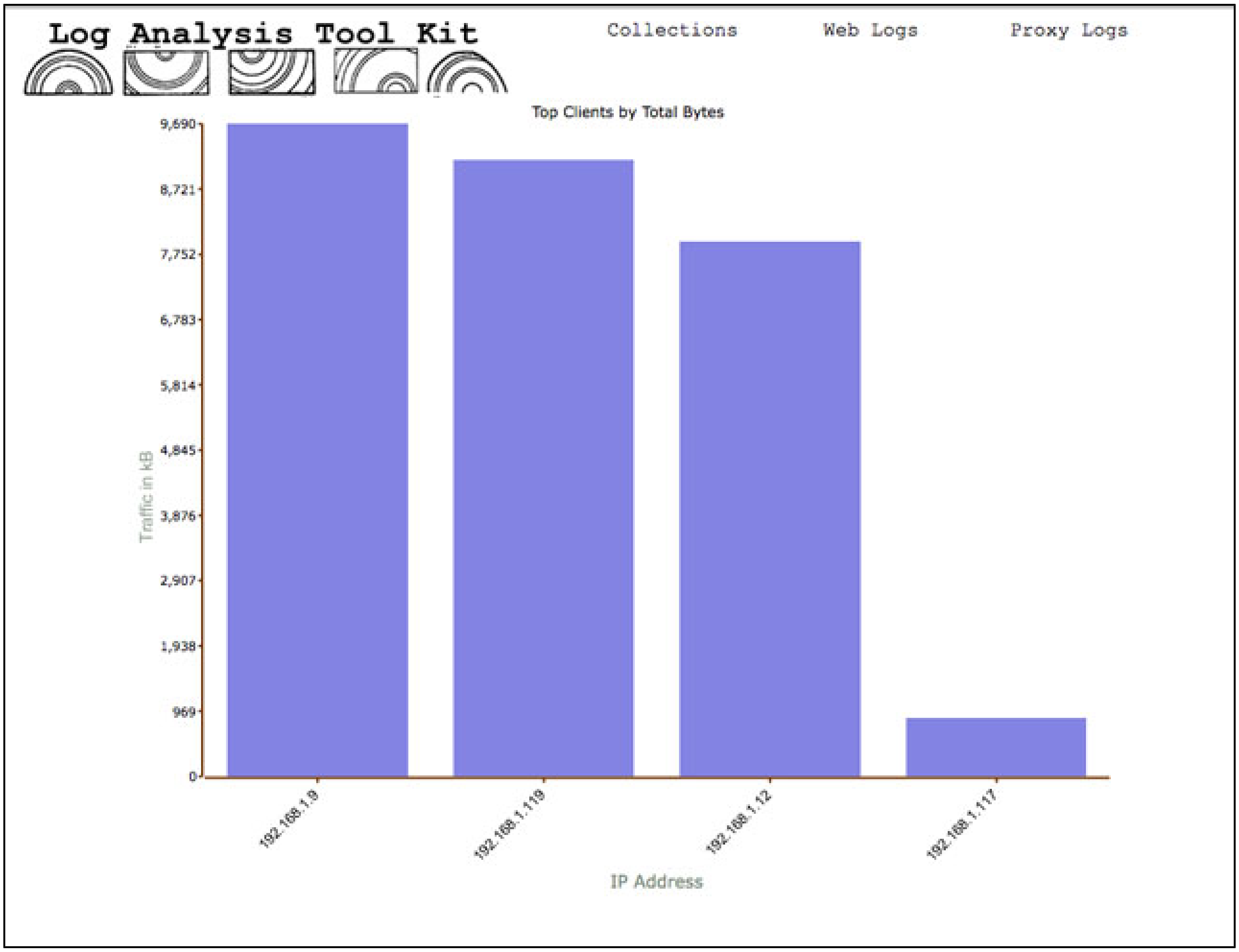


Report for IP: hostname: .edu

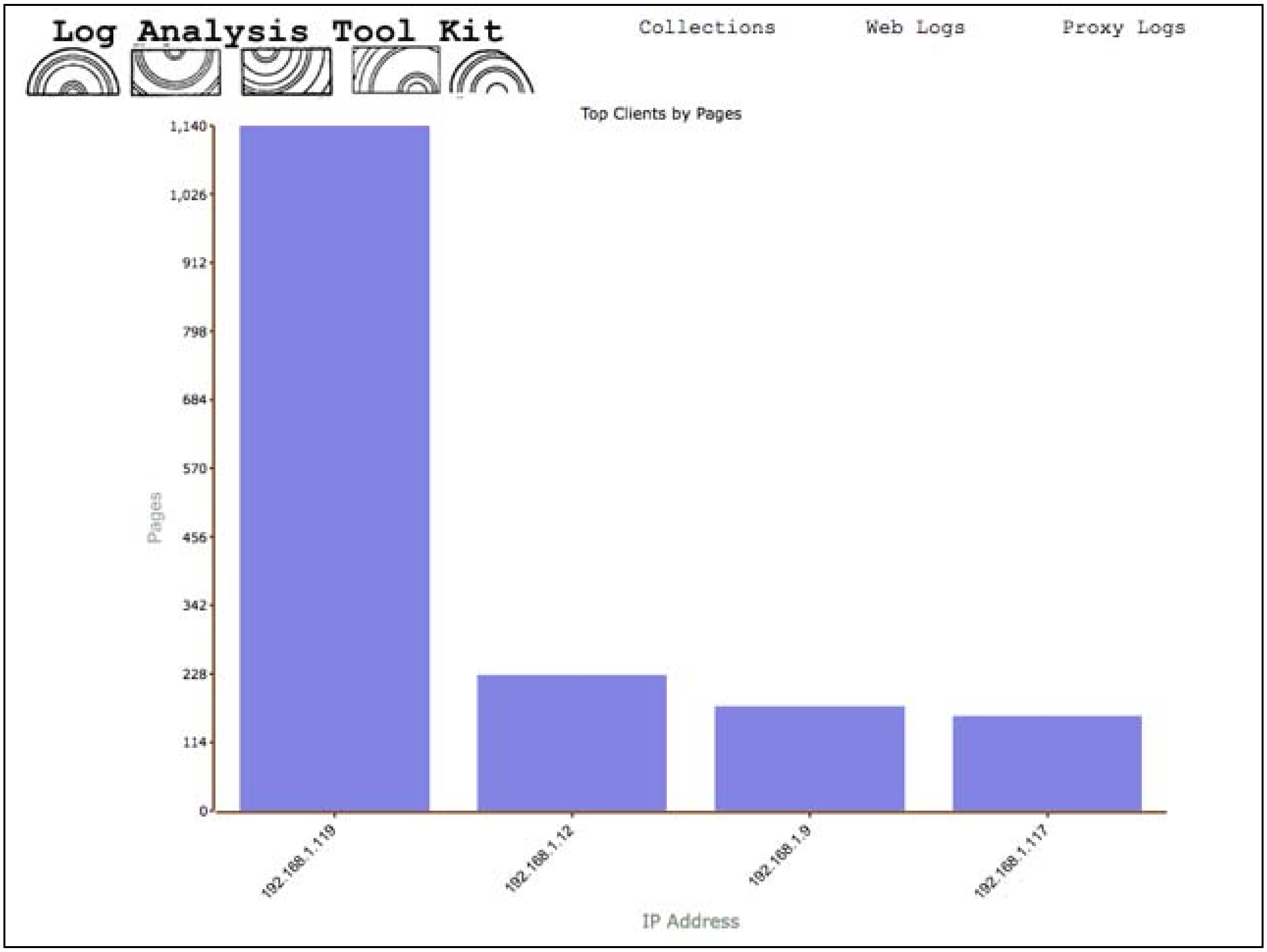
Beacon Behaviour Analysis

Client IP	Destination	Hits	Avg Interval	Std Dev	Score	Probability
1 .253	1 21	21	0.2	0.4	5	Medium
1 . .253	6 83	18	0.24	0.424	5	Medium
1 .253	1 .6	22	0.24	0.426	5	Medium
1 .253	1	23	0.27	0.445	5	Medium
1 253	1 28	23	0.32	0.466	5	Medium
1 . 253	2 .34	4	2047.33	471.204	5	Medium

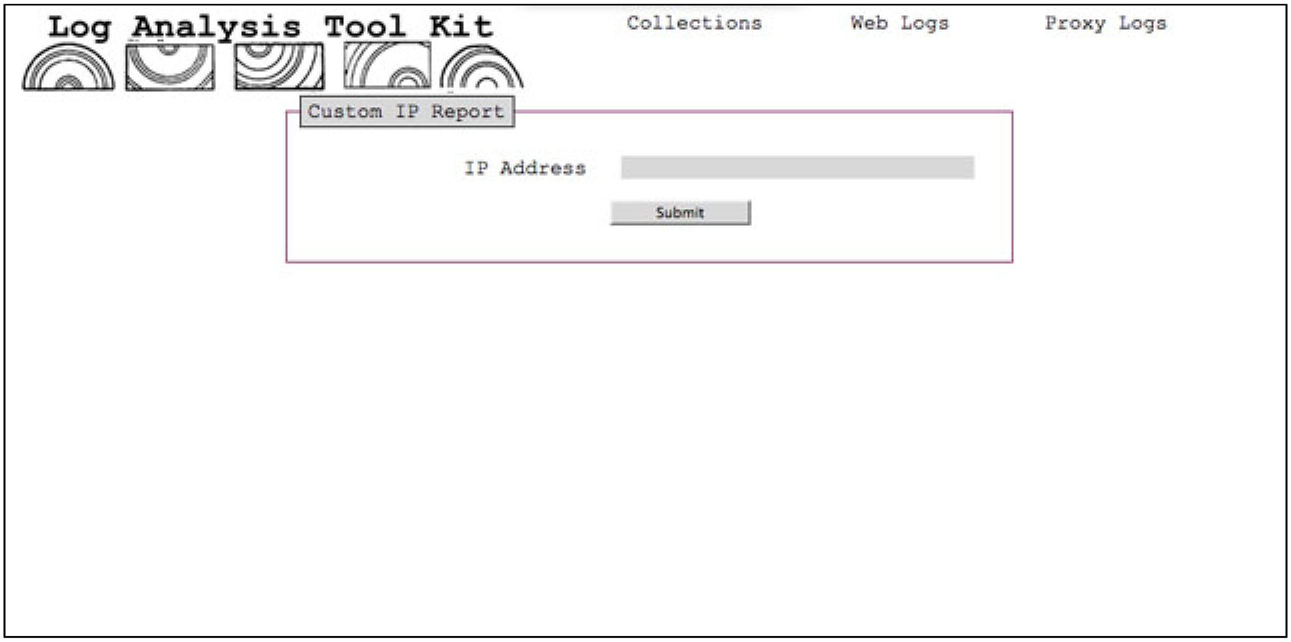
Web Server Logs Analysis of Clients by Total Bytes



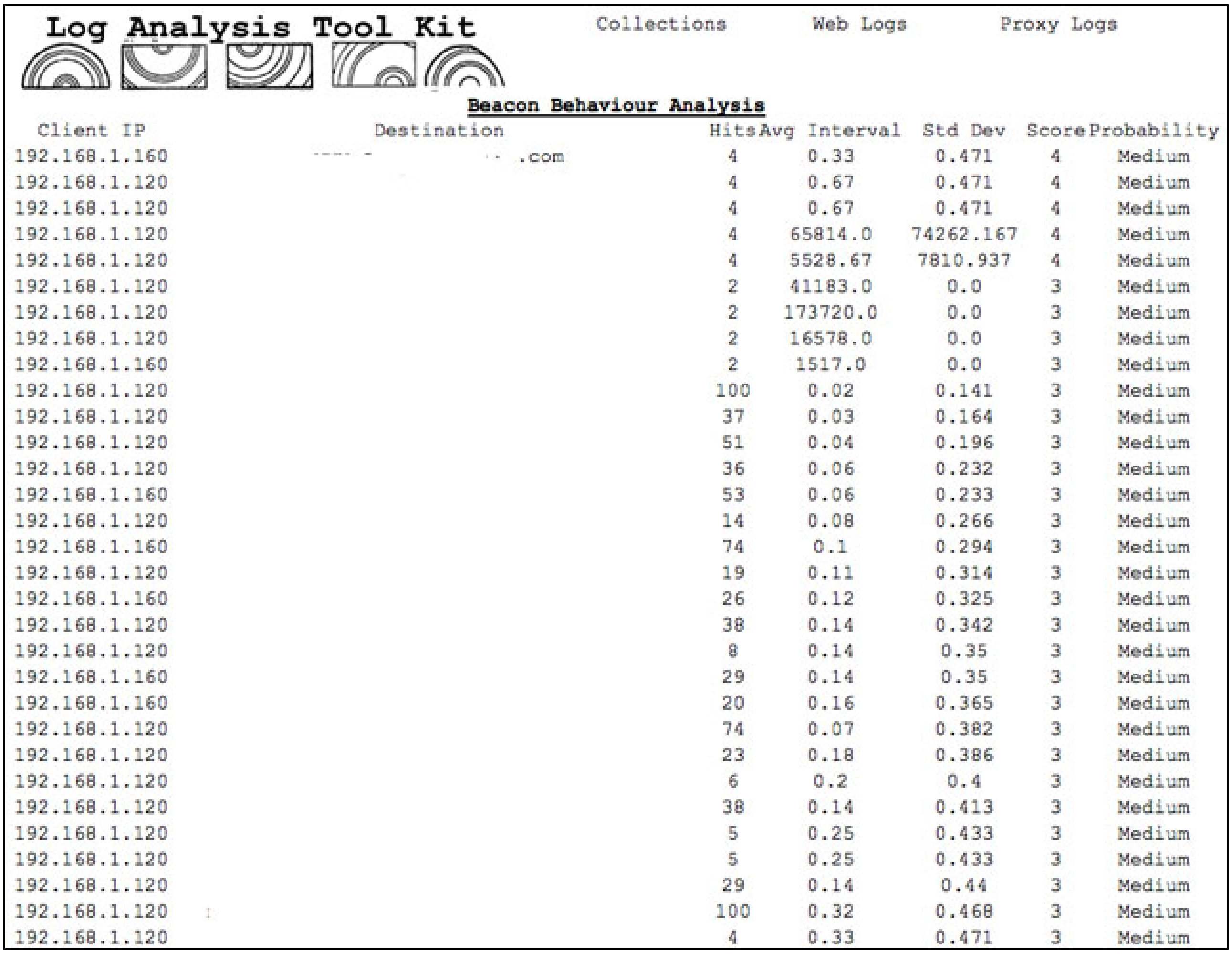
Web Server Log Analysis Total Pages by Client.



A custom search can be run based on IP Address.



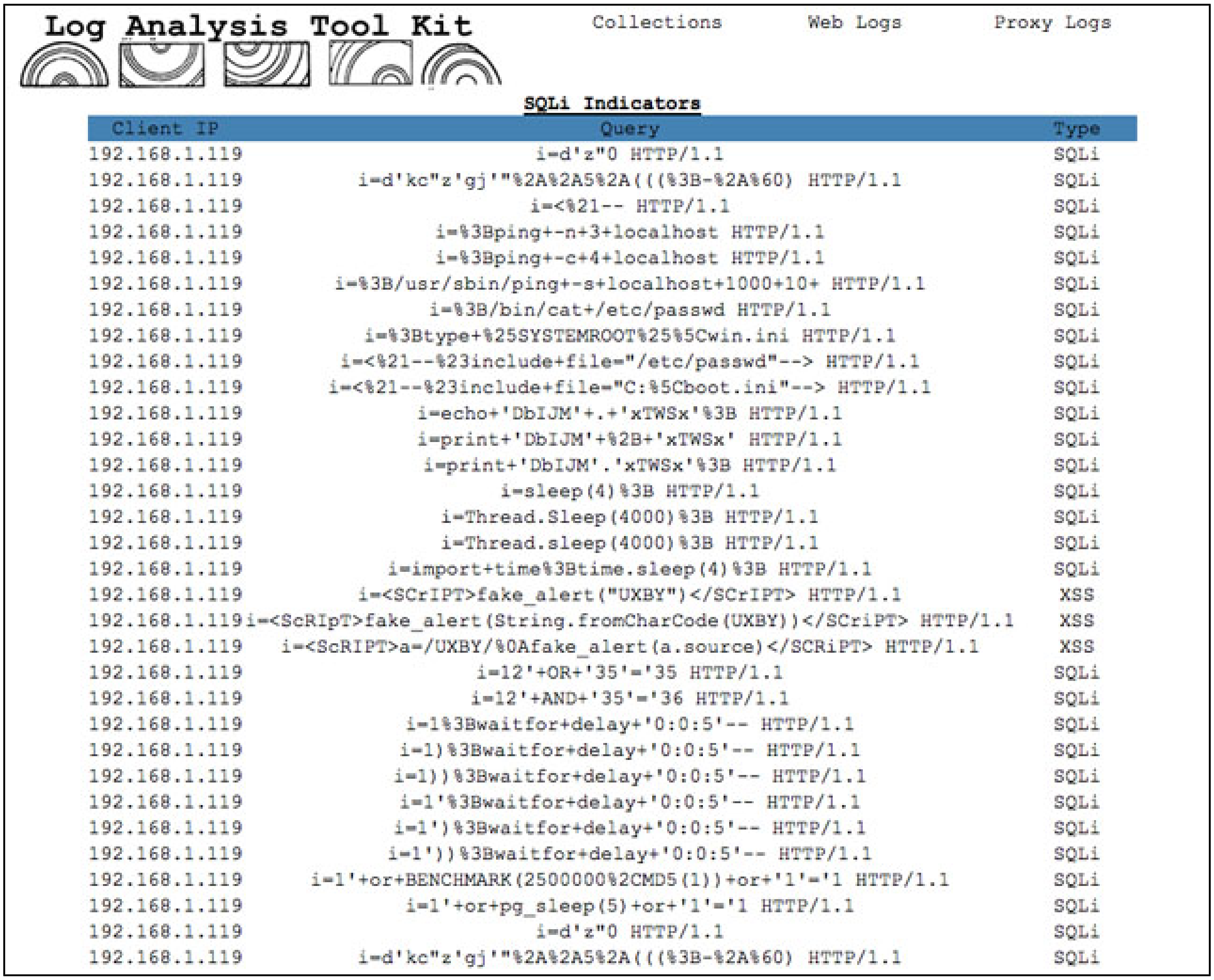
Proxy Server Beacon Detection Report



Beacon Behaviour Analysis

Client IP	Destination	Hits	Avg Interval	Std Dev	Score	Probability
192.168.1.160	.com	4	0.33	0.471	4	Medium
192.168.1.120		4	0.67	0.471	4	Medium
192.168.1.120		4	0.67	0.471	4	Medium
192.168.1.120		4	65814.0	74262.167	4	Medium
192.168.1.120		4	5528.67	7810.937	4	Medium
192.168.1.120		2	41183.0	0.0	3	Medium
192.168.1.120		2	173720.0	0.0	3	Medium
192.168.1.120		2	16578.0	0.0	3	Medium
192.168.1.160		2	1517.0	0.0	3	Medium
192.168.1.120		100	0.02	0.141	3	Medium
192.168.1.120		37	0.03	0.164	3	Medium
192.168.1.120		51	0.04	0.196	3	Medium
192.168.1.120		36	0.06	0.232	3	Medium
192.168.1.160		53	0.06	0.233	3	Medium
192.168.1.120		14	0.08	0.266	3	Medium
192.168.1.160		74	0.1	0.294	3	Medium
192.168.1.120		19	0.11	0.314	3	Medium
192.168.1.160		26	0.12	0.325	3	Medium
192.168.1.120		38	0.14	0.342	3	Medium
192.168.1.120		8	0.14	0.35	3	Medium
192.168.1.160		29	0.14	0.35	3	Medium
192.168.1.160		20	0.16	0.365	3	Medium
192.168.1.120		74	0.07	0.382	3	Medium
192.168.1.120		23	0.18	0.386	3	Medium
192.168.1.120		6	0.2	0.4	3	Medium
192.168.1.120		38	0.14	0.413	3	Medium
192.168.1.120		5	0.25	0.433	3	Medium
192.168.1.120		5	0.25	0.433	3	Medium
192.168.1.120		29	0.14	0.44	3	Medium
192.168.1.120		100	0.32	0.468	3	Medium
192.168.1.120		4	0.33	0.471	3	Medium

Web Server SQLi Injection Report, remember to first run the python script after importing to generate the table.



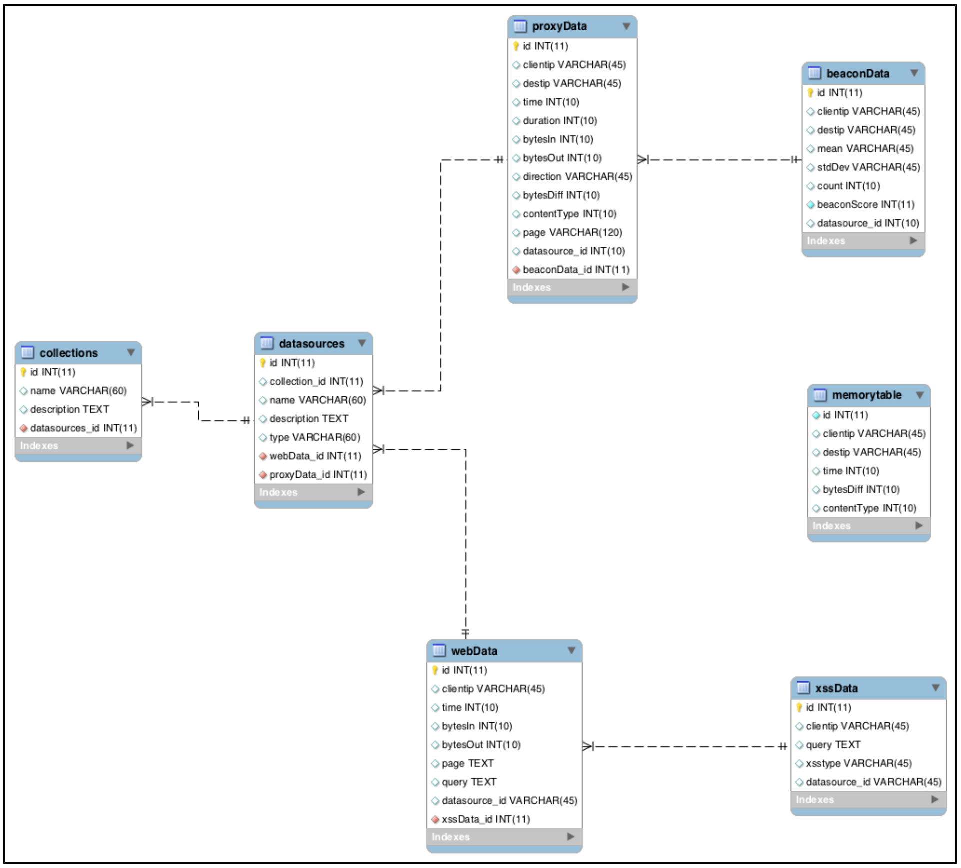
SQLi Indicators

Client IP	Query	Type
192.168.1.119	i=d'z"0 HTTP/1.1	SQLi
192.168.1.119	i=d'kc"z'gj'"%2A%2A5%2A(((%3B-%2A%60) HTTP/1.1	SQLi
192.168.1.119	i=<%21-- HTTP/1.1	SQLi
192.168.1.119	i=%3Bping+-n+3+localhost HTTP/1.1	SQLi
192.168.1.119	i=%3Bping+-c+4+localhost HTTP/1.1	SQLi
192.168.1.119	i=%3B/usr/sbin/ping+-s+localhost+1000+10+ HTTP/1.1	SQLi
192.168.1.119	i=%3B/bin/cat+/etc/passwd HTTP/1.1	SQLi
192.168.1.119	i=%3Btype+%25SYSTEMROOT%25%5Cwin.ini HTTP/1.1	SQLi
192.168.1.119	i=<%21--%23include+file="/etc/passwd"--> HTTP/1.1	SQLi
192.168.1.119	i=<%21--%23include+file="C:%5Cboot.ini"--> HTTP/1.1	SQLi
192.168.1.119	i=echo+'DbIJM'+.+'xTWSx'%3B HTTP/1.1	SQLi
192.168.1.119	i=print+'DbIJM'+%2B+'xTWSx' HTTP/1.1	SQLi
192.168.1.119	i=print+'DbIJM'.'xTWSx'%3B HTTP/1.1	SQLi
192.168.1.119	i=sleep(4)%3B HTTP/1.1	SQLi
192.168.1.119	i=Thread.Sleep(4000)%3B HTTP/1.1	SQLi
192.168.1.119	i=Thread.sleep(4000)%3B HTTP/1.1	SQLi
192.168.1.119	i=import+time%3Btime.sleep(4)%3B HTTP/1.1	SQLi
192.168.1.119	i=<SCrIPT>fake_alert("UXBY")</SCrIPT> HTTP/1.1	XSS
192.168.1.119	i=<ScRIpT>fake_alert(String.fromCharCode(UXBY))</SCriPT> HTTP/1.1	XSS
192.168.1.119	i=<ScRIPT>a=/UXBY/%0Afake_alert(a.source)</SCRiPT> HTTP/1.1	XSS
192.168.1.119	i=12'+OR+'35'='35 HTTP/1.1	SQLi
192.168.1.119	i=12'+AND+'35'='36 HTTP/1.1	SQLi
192.168.1.119	i=1%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1)%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1))%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1'%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1')%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1'))%3Bwaitfor+delay+'0:0:5'-- HTTP/1.1	SQLi
192.168.1.119	i=1'+or+BENCHMARK(2500000%2CMD5(1))+or+'1'='1 HTTP/1.1	SQLi
192.168.1.119	i=1'+or+pg_sleep(5)+or+'1'='1 HTTP/1.1	SQLi
192.168.1.119	i=d'z"0 HTTP/1.1	SQLi
192.168.1.119	i=d'kc"z'gj'"%2A%2A5%2A(((%3B-%2A%60) HTTP/1.1	SQLi

11- Advanced Querying

The LATK uses MySQL to store and normalize log data. You may have specific questions that the command line tools and web interface can't answer.

You may want to use a tool like the MySQL Work Bench of Crystal Reports to access this data.



Consider the following example.

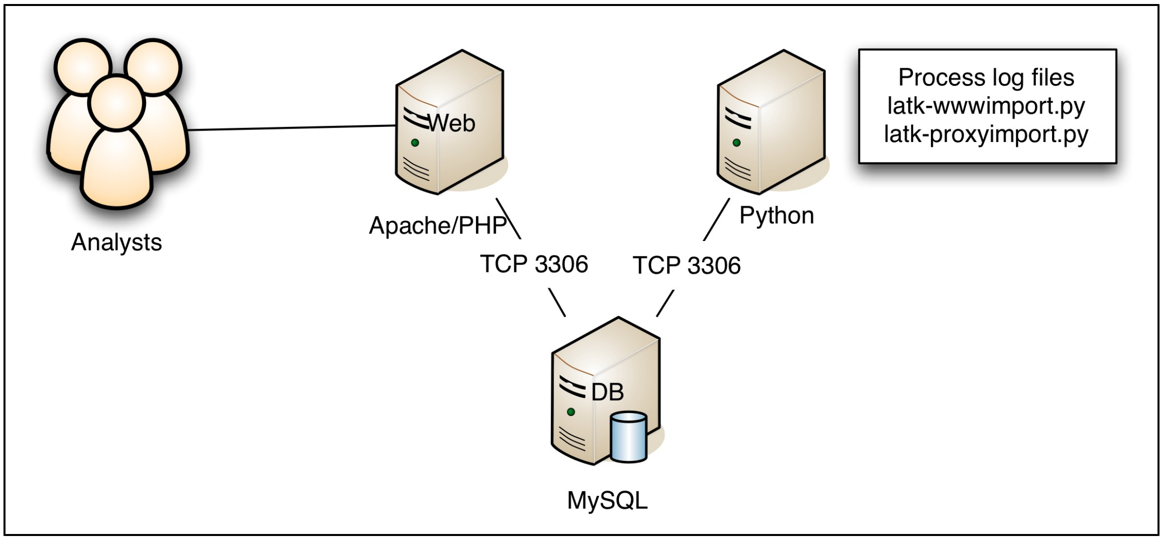
You know a virus was distributed via email after 8:00am and before 10am on 2011/09/07. The email was a spear phishing attack that got users to download a file named '**badpackage.exe'**.

```
mysql> select clientip, page, FROM_UNIXTIME(time, '%Y-%m-%d %H:%i') as accessTime from
proxyDataÂ  Â
Â  Â -> where datasource_id=4
Â Â  -> and time > (UNIX_TIMESTAMP('2011-09-07 08:00'))
   -> and time (UNIX_TIMESTAMP('2011-09-07 10:00'))
   -> and page like '%badpackage.exe%' ;

+--------------+------------------------------------------+------------------+
| clientipÂ Â Â Â  |
pageÂ Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â  |
accessTimeÂ Â Â Â Â Â  |
+--------------+------------------------------------------+------------------+
| 192.168.1.12 | /images/global/badpackage.exe            | 2011-09-07 09:19 |
| 192.168.1.12 | /cb/badpackage.exe              Â Â Â Â Â Â Â  | 2011-09-07 15:14 |
| 192.168.1.12 | /cb/web/badpackage.exe.seam              | 2011-09-07 15:14 |
| 192.168.1.12 | /cb/10/badpackage.exeÂ Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â  |
2011-09-07 15:17 |
+--------------+------------------------------------------+------------------+
```

12- Scaling LATK

Splitting the database off from the web front and the machine that is used to import the data can meet scaling needs.
This is particular useful when dealing with large numbers of users or large amounts of data.



Bulk Name Resolver

You may have a need to resolve names in the MySQL databases created by the other tools in the tool kit. This script will let you run a name resolution on the database and query you pass to the script.The output is on the screen and a CSV query-output.csv

Usage:

```
./latk-resolvnames.py YOUR_QUERY
```

Example:

This will return internal host names from the database created by the overcoat tool that went to google.

```
giskard:logs joe$ ./latk-resolvnames.py "select DISTINCT(clientip) FROM proxyData
WHERE destip LIKE '%google%' and datasource_id=5"
192.168.1.120,laptop.example.com
192.168.1.160,desktop.example.com
```

This will return external client names from the database created by the logminer tool that accessed the site between 2011-04-07 and 2011-04-01.

```
giskard:logs joe$ ./latk-resolvnames.py  \
     "select clientip, page, FROM_UNIXTIME(time, '%Y-%m-%d %H:%i') as accessTime from
proxyData where datasource_id=4 and time > (UNIX_TIMESTAMP('2011-09-07 08:00')) and
time (UNIX_TIMESTAMP('2011-09-07 10:00')) and page like '%access%'"

192.168.1.120,laptop.example.com
192.168.1.160,desktop.example.com
```

Downloads

The Log Analysis Tool Kit is a based on Python and MySQL. The web frontend is written in PHP and Open Flash Chart.

The current version of the Log Analysis Tool Kit tools can be downloaded below.

Downloads

Release 1.7

- Released new version 2012/04/19
    - Archive LogAnalysisToolKit-1.7.tar.gz
    - SRPM LogAnalysisToolKit-1.7-0.src.rpm
    - RPM LogAnalysisToolKit-1.7-0.noarch.rpm
    - Ubuntu DEB
- Added handling for IIS 7 logs

Release 1.6

- Released new version 2012/01/20
    - Archive LogAnalysisToolKit-1.6.tar.gz
    - SRPM LogAnalysisToolKit-1.6-0.src.rpm
    - RPM LogAnalysisToolKit-1.6-0.noarch.rpm
    - Ubuntu DEB loganalysistoolkit_1.6-0_all.deb
- Added authentication to the web app.

Release 1.5-4

- Released new version 2011/11/16
    - Archive LogAnalysisToolKit-1.5.tar.gz
    - SRPM LogAnalysisToolKit-1.5-4.src.rpm
    - RPM LogAnalysisToolKit-1.5-4.noarch.rpm
    - Ubuntu DEB loganalysistoolkit_1.5-4_all.deb
- Minor bug fixes.

Release 1.5-2

- Released new version 2011/11/06
    - Archive LogAnalysisToolKit-1.5.tar.gz
    - RPM LogAnalysisToolKit-1.5-2.noarch.rpm
    - SRPM LogAnalysisToolKit-1.5-2.src.rpm
    - Ubuntu DEB LogAnalysisToolKit_1.5-2_all.deb
    - Fedora 15 VM with Training Data and Slides: LATK-Training-VM.ova
- Minor bug fixes.

Release 1.5

- Released new version 2011/11/01
    - Archive LogAnalysisToolKit-1.5.tar.gz
    - RPM LogAnalysisToolKit-1.5-1.noarch.rpm
    - SRPM LogAnalysisToolKit-1.5-1.src.rpm
    - Ubuntu DEB LogAnalysisToolKit_1.5-2_all.deb
- Added a PHP-Geo maps for Linux and PHP-Pecl maps for others, will attempt to guess the right one.
- Web interface now uses the same database conf file as python; /etc/latk-mysql.conf
- MySQL Socket is now an option in conf file.
- Minor bug fixes

Release 1.4

- Released new version 2011/10/12
    - Archive LogAnalysisToolKit-1.4.zip
    - RPM LogAnalysisToolKit-1.4-0.noarch.rpm
    - SRPM LogAnalysisToolKit-1.4-0.src.rpm
- Added GeoIP maps.
- Actually fixed the custom search in the web search for all log types.
- Fixed issue with xss/sqli discovery.
- Limited IoS web output to 200 lines.
- Changed db connection variables in latklib.php.
- Changed interface to disable reports for log types that are not in use.

Release 1.3

- Released new version 2011/10/04 LogAnalysisToolKit-1.3.zip
- Fixed "custom search" in web interface
- Added an alternate bluecoat log format.

Release 1.2

- Released new version 2011/09/18 LogAnalysisToolKit-1.2.zip
- Changed database connection string to be compatible with python 3.1 and python 3.2.

Release 1.1

- Released new version 2011/08/09: LogAnalysisToolKit-1.1.zip
- Added a ability to searching by keyword in url to web interface.

- Fixed issue with error messages
- Changed default latk-mysql.conf location to /etc.

Release 1.0

- Released new version 2011/07/13: LogAnalysisToolKit-1.0.zip
- Added the Index of Suspicion tool, an automated method for determining the most suspicious ip addresses.
- Packaged web and command line tools together.
- Changed the XSS/SQLi check from dynamic to static, run once. This saves time as the report may be viewed multiple times.

Release 0.3

- Released new version 2011/07/07: LogAnalysisToolKit-0.3.zip
- Released Web Front End to Log Analysis Tool Kit: latk-web-0.3.tar.gz
- Added a central mysql configuration file, so you don't need to edit every python script
- Added threading to wwwlog importing
- Removed extra command line options
- Renamed wwwlog python scripts.

Release 0.2

- Increased speed of proxy import process (700%)
- Increased speed of beacon analysis (10^10% *Not actually measured)
- Fixed issues with poor searches on proxy summary data.

LogAnalysisToolKit-0.2.zip

Release 0.1

- First MySQL Release
- Added data sources and collections.

LogAnalysisToolKit-0.1.zip

Release 0.01

- Original SQLite version.

LogAnalysisToolKit-sqlite.zip

Prerequisites

To run the LATK you will need the Python Mysql Connector, which can be downloaded here: https://launchpad.net/myconnpy

You will also need Numpy available here: http://numpy.scipy.org/

Installation

The Log Analysis Tool Kit has been tested on Fedora, Ubuntu and Mac OSX. It has been run on Windows, but it is only tested on the other OS's.

Detailed instructions are available in the attached slide deck. But everything you should need to know to get up and running quickly is on the installation pages below.

RPMS, Debs and source are all available.

A detailed installation and training manual is available as a PDF here: LATK Training Manual
Expand all   Collapse all

Fedora Installation

Using the provided Fedora RPMs the installation is very straight forward. It will install all of the requirements such as MyConnPy, Apache, PHP, Python etc.

Use YUM to install the rpm.

```
#yum --nogpgcheck install
https://forensics.cert.org/confluence/download/attachments/7340037/LogAnalysisToolKit-
1.5-1.noarch.rpm
```

Start the required services.

```
#service httpd start
#service mysqld start
```

Install the MaxMind GeoIP Light database.

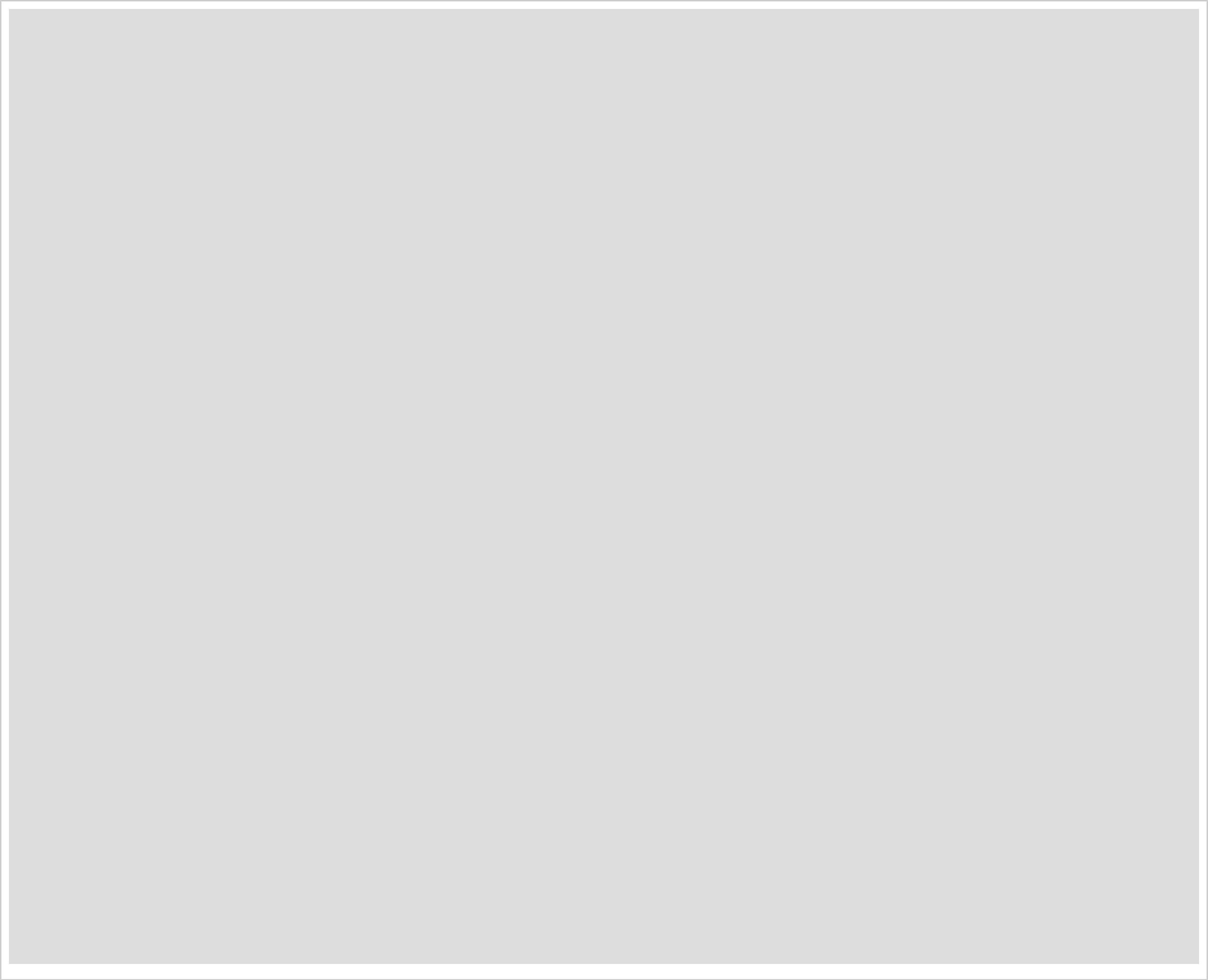
```
#cd /usr/share/GeoIP
#wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
#gunzip GeoLiteCity.dat.gz
#mv GeoLiteCity.dat GeoIPCity.dat
```

If you will be using Fedora to view the web interface, install the flash plugin.

Download the RPM from: http://get.adobe.com/flashplayer/

```
#sudo yum localinstall ~/Downloads/adobe-release-x86_64-1.0-1.noarch.rpm -y
#yum install flash-plugin ây
```

Installation and Training Manual



Download a VM with the Log Analysis Tool Kit installed and follow the training manual.
Skip over the installation instruction and move right to analyzing training data.

LATK Virtual Box Image: https://forensics.cert.org/confluence/download/attachments/13434941/LATK-Training-VM.ova

Mac OS X Installation

LATK installation on OS X is slightly harder than on Linux. You will need to download the source, move it in to place, turn on apache, install mysql, configure apache, install python3, install python3 packages. But it works quite well once this is done.

Download the source.

```
#mkdir tmp
#cd tmp

#wget
hhttps://forensics.cert.org/confluence/download/attachments/7340037/LogAnalysisToolKit
-1.5.tar.gz

#tar âzxvf LogAnalysisToolKit-1.5.tar.gz

#cd LogAnalysisToolKit-1.5
```

Move the binaries in to your home directory

```
#mkdir ~/latk
#cp usr/bin/* ~/latk/

#export PATH=$PATH:~/latk

#echo 'export PATH=$PATH:~/latk' >> ~/.bash_profile

#sudo cp etc/latk-mysql.conf /etc/latk-mysql.conf

#mkdir ~/Sites/latk/

#sudo ln -s ~/Sites/latk /Library/WebServer/Documents/latk

#cp -r var/www/html/latk/*Â  ~/Sites/latk/
```

Configure PHP on Apache

```
#sudo vi /etc/apache2/httpd.confUncomment the line:

LoadModule php5_moduleÂ Â Â Â Â Â Â  libexec/apache2/libphp5.so
```

Start Apache

```
System Preferences ->Web Sharing ->Enable
```

Install MySQL from http://dev.mysql.com/downloads/mysql/

Start MySQL:

```
System Preferences ->MySQL
```

GeoIP is slightly more complex to install, you'll need to use PECL to install the geop library, then download the GeoIP Light database from MaxMind.

Ubuntu Installation

Using the DEB package LATK can be installed very quickly.

```
#wget
https://forensics.cert.org/confluence/download/attachments/7340037/loganalysistoolkit_
1.5-2_all.deb
#dpkg -i loganalysistoolkit_1.5-2_all.deb
```

Start the required services.

```
#service mysql start
#service apache2 start
```

Move the web files to the right directory.

```
#mv /var/www/html/latk/ /var/www/latk
```

Edit the latk-mysql.conf file to point to the correct mysql.sock file.

```
#vi /etc/latk-mysql.conf
sock = /var/run/mysqld/mysqld.sock
```

Install MyConnPy

```
#wget
http://launchpad.net/myconnpy/0.3/0.3.2/+download/mysql-connector-python-0.3.2-devel.t
ar.gz#cd mysql-connector-python-0.3.2-devel

#python3 setup.py install
```

Install Numpy

```
#wget http://downloads.sourceforge.net/project/numpy/NumPy/1.6.1/numpy-1.6.1.zip?r=\
http%3A%2F%2Fsourceforge.net%2Fprojects%2Fnumpy%2Ffiles%2FNumPy%2F1.6.1%2F&ts=13203249
16&use_mirror=iweb

#unzip numpy-1.6.1.zip#cd numpy-1.6.1
#apt-get install python3-dev
#python3 setup.py install
```