Offensive & Defensive & Forensic Techniques for Determining Web User Identity

Part 3

Zachary Zebrowski
zak@freeshell.org
Approved for Public Release: 12-3046.
Distribution Unlimited

All materials is licensed under a Creative Commons “Share Alike” license.

- http://creativecommons.org/licenses/by-sa/3.0/

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Forensics Outline

- Forensic Database Analysis
- Forensic Log Analysis
- Introduction to Real Time Analysis
  - Not my forte, thus an “introduction”. Included for completeness.
- Caveats & Questions

Basic Outline of Approach

- Triage the data sets you're provided and look for "most important" databases.
  - We assume that you get more than one database and that the databases are related…
- Host & convert the database to Unicode.
- Create a new summary database of important fields & process the information
- Search tool suggestions
- Dealing with Murphy

Triage

- Make sure the data “makes sense” and that you were given the right data files.
- Identify important databases
- Identify important tables within the databases

Hosting the data

- Remember, you have *no* control of the data.
  - You can't ask for advice why some settings were used.
- The data can come in fast and furious, or slow
- The data can come in many formats
  - SQL dumps or "Raw" data files.
    - Oracle
    - MSSQL
    - MySQL
  - MS Access
  - MS Excel files
  - Csv / Text / Other…
- The data may have viruses. **Do not run antivirus on the data until it's been processed so you can extract info anyways from the files that may have valuable content in addition to the viruses.**
- ***Be sure you are on an isolated network.***
- Importing the data into MySQL is good, because it's a fast server.

Hosting the data, continued.

- First, be sure to attribute where the data came from & why ***before*** you start processing it.
  - Nothing is worse than having a non-attributable copy of data. You don't want to be the one who tells the Boss you cannot use the data in court because you don't know where it came from.
- Next, backup the data
  - Backup the data to another location one you've copied it locally to your data staging computer.

Hosting the data, continued.

- Attempt to host the data on a database server.
  - The data might be in an old data format. Try to use a system that handles this.
    - Note that for MySQL data, the *Linux* version of the MySQL server is a lot more forgiving than the Windows version of the MySQL server.
      - Not really sure why.
  - The data might be in a character set that you are not expecting. You might need to use special commands to import it.
  - You may need to "repair" tables if they don't contain rows.
  - After you initially load the data, if you determine something is majorly wrong, don't be afraid to re-import the data.
  - You may need to fix permissions on the database depending on the database version.

Hosting the data, continued…

- Allow for bogus data.
- In the absolute worse case… there is always vi (or emacs or notepad) to manually extract data.
  - It actually works.

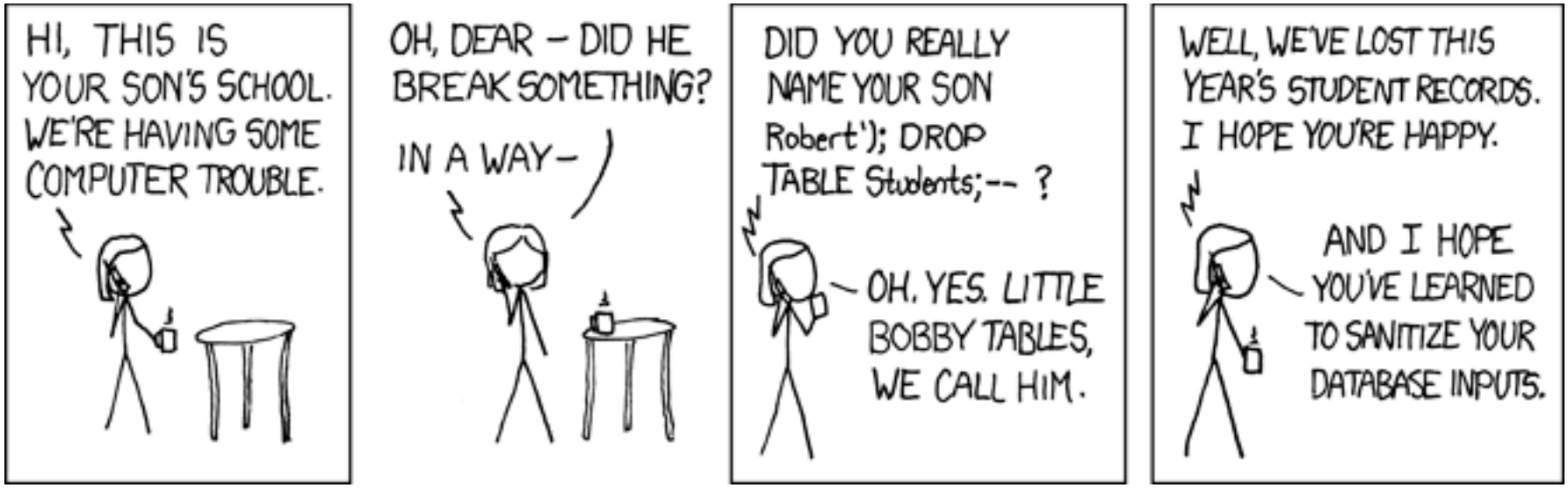
Convert to Unicode

- Why convert the data to Unicode?
  - Unicode allows multiple character sets to be treated as one.
- Now that you've hosted the data, convert it to Unicode, via convert.pl script.
- What's the character set of a given document?
  - **Ask your web browser to render it.** It might guess correctly.
  - If it doesn't, the data might be corrupted. You may need to reload the data into the database using a different character set. Alternatively, you need to pre-process the data to convert using a different character set. Or try changing the OS to a different one.

Processing information from the Unicode database to a new database

- Extract the priority fields only and import them into a new database.
  - Store or just note where the secondary information is in a full text index of some sort. (Consider Lucy library on CPAN.)
- Index the data as the users have requested.
- Don't be afraid to create computed indexes…
  - Soundex, lower case only, etc.
- Create a "secondary" view using html for browsing the entire database. Also useful for sharing information external to your organization.
  - When this is a priority.

Recover deleted information



Recover Deleted Information

Student ID	First	Last
1	Lisa	Simpson
2	Bart	Simpson
3	Millhouse	Van Houten

Student ID	Quiz ID	Grade
1	1	100
2	1	75
3	1	85
1	2	100
2	2	50
3	2	78

Recover Deleted Information

- Assume that Bart's name is deleted, you still have the grades for him. Thus, you can probably figure out who is who, for certain cases.
  - This is not easy to do, and it's even worse when there keys are reused. But, it's possible, and can contain valuable information.

Search Tool Suggestions

- Do not make the search engine overly complex.
  - Search the data first. Everything else (properties of files, images, etc.) is secondary.
- Let the user quickly search through as many types of data as needed as fast as possible. (Think Google search).
- Use existing technology when possible.
  - Use available API's to existing search engines or write a script... Allow the user to stay in your app as much as possible.
- Use "mail box" methodology to allow the user to view results as results become available.
- Multiple servers are good.
  - One per major type of data
  - One for "constant" files – e.g., JavaScript or images.
  - One for full text index server
  - One for database server.
- Uninstall mlocate.

Murphy

- So many different stories…
- Try to make sure people are on the same page before the crisis hits, so you know which systems are critical to avoid powering off the wrong system over the weekend.
- Tape down any loose wires as needed.
- Assume file systems will fail (running out of inodes, bad hdd…)
- Move “one off” systems to a production facility as soon as feasible.
- Assume hard drives will fail – have backups.
- Assume CPUs will fail – have backups.
- Assume power will fail – have (working) UPS’s.
- Assume code will break – have dev, staging, and production copies of data & apps.
- Don’t short change development / staging / backup hardware.

Sanity Checking

- Sanity check at various points.
- Check the encodings to make sure it makes sense at critical points.
- Use IE or another browser to verify that it looks ok
- If you are not sure, run text through translation software as a sanity check that it could be valid text.
- When first looking at a schema, make sure it makes logical sense to you. If possible, ask someone who knows what the data is.
- When you have finished processing, run a sanity check script before releasing to users.

Generalizing

- Be careful to start generalizing from the start. Only start after you have enough data.
- You may be generalizing the wrong thing.

Let's take a break!

- Questions?

- Next up: Log Analysis & Real Time Analysis

Forensic Log Analysis

- Much of the summaries in the previous section pertaining to databases remain the same
  - Triage & Tracking
  - Convert to Unicode
  - Search Tool Suggestions
  - Murphy
  - Sanity Checking
- Let's focus on what's different.

Log Analysis – What's different

- You're given data that won't immediately import into a database
  - If you need to see just what a particular user does, consider using grep.
  - Otherwise, consider writing a database schema for simple log formats.
  - Details follow!

Using Grep

- Problem – you want to see all activity from a given IP Address from logs
- You can use grep to extract all lines that contain that ip address from your log files
- Eg: grep 130.215.10.1 *
  - For all files in your current directory, return all lines that have string 130.215.10.1 in the line

Using Grep

- You can also extracting a user's activity via basic authentication username
- Eg. grep zaz *
  - For all files in the current directory, return all lines that have zaz in the line.

Creating a SQLite db of logs

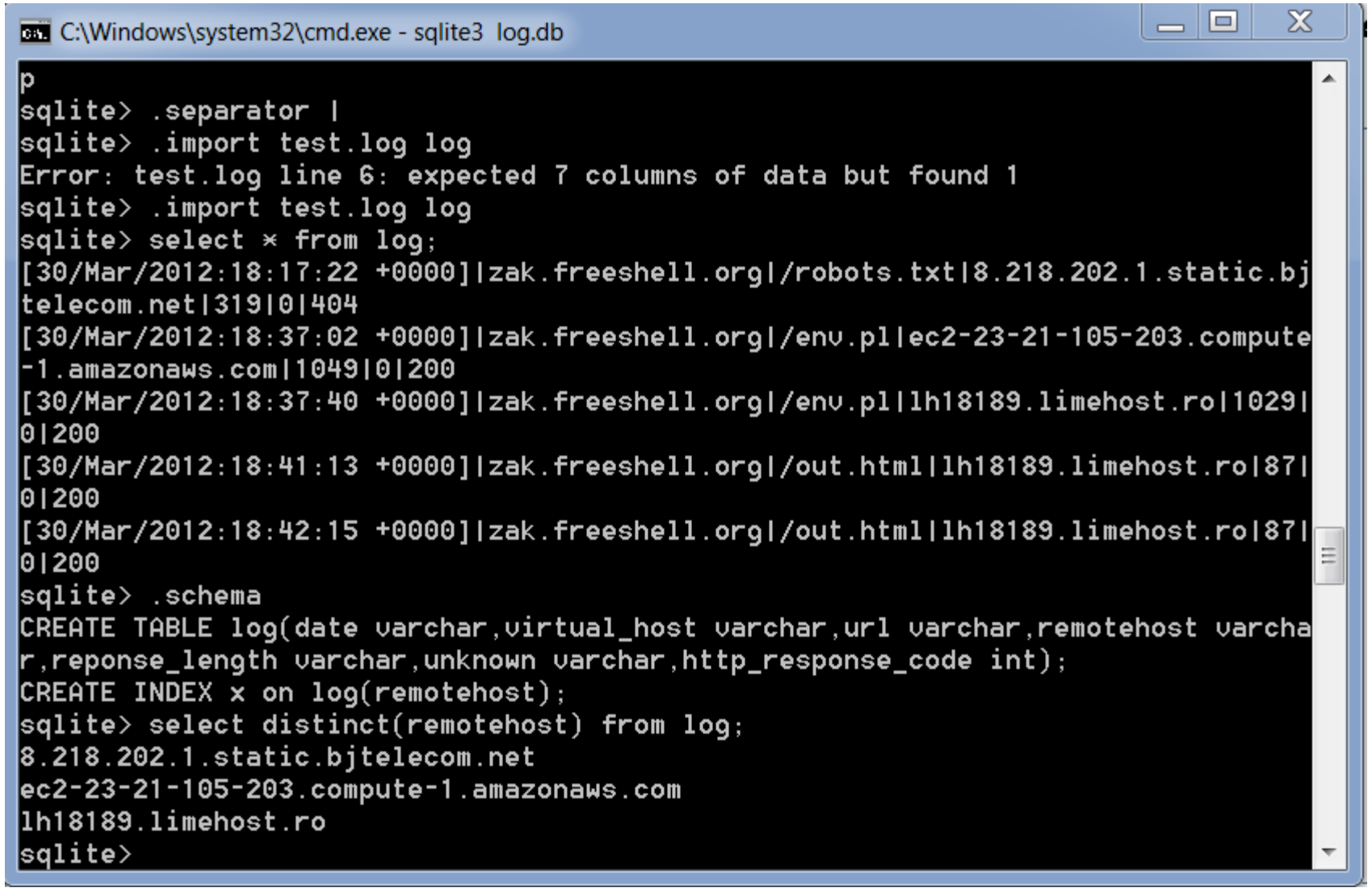
- First download SQLite (http://www.sqlite.org).
- Start: sqlite3 log.db
- For a web log:

```
CREATE TABLE LOG(
date varchar (50),
Virtual_host varchar(50),
url varchar(50),
Remotehost varchar(50),
Response_length varchar(50),
Unknown varchar(50),
http_response_code int
);
CREATE INDEX X on LOG(remotehost);
```

Creating a SQLite db of logs, part 2

- .separator |
- .import log.txt LOG
- Select distinct(remotehost) from log; # or other SQL query

Sample


C:\Windows\system32\cmd.exe - sqlite3 log.db
p
sqlite> .separator |
sqlite> .import test.log log
Error: test.log line 6: expected 7 columns of data but found 1
sqlite> .import test.log log
sqlite> select * from log;
[30/Mar/2012:18:17:22 +0000]|zak.freeshell.org|/robots.txt|8.218.202.1.static.bj
telecom.net|319|0|404
[30/Mar/2012:18:37:02 +0000]|zak.freeshell.org|/env.pl|ec2-23-21-105-203.compute
-1.amazonaws.com|1049|0|200
[30/Mar/2012:18:37:40 +0000]|zak.freeshell.org|/env.pl|lh18189.limehost.ro|1029|
0|200
[30/Mar/2012:18:41:13 +0000]|zak.freeshell.org|/out.html|lh18189.limehost.ro|87|
0|200
[30/Mar/2012:18:42:15 +0000]|zak.freeshell.org|/out.html|lh18189.limehost.ro|87|
0|200
sqlite> .schema
CREATE TABLE log(date varchar,virtual_host varchar,url varchar,remotehost varcha
r,reponse_length varchar,unknown varchar,http_response_code int);
CREATE INDEX x on log(remotehost);
sqlite> select distinct(remotehost) from log;
8.218.202.1.static.bjtelecom.net
ec2-23-21-105-203.compute-1.amazonaws.com
lh18189.limehost.ro
sqlite>

Real Time Analysis

- It’s rare that you can do this legally.
  - Generally, companies / university's prohibit this.
    - Privacy issues.
- Note – There are courses that go into much deeper analysis than what we’re doing.

tcpdump / wireshark analysis

- Wireshark is a graphic tool that allows you to visualize tcpdump data.
  - You can visually decode various protocols (http / pop, etc).
- tcpdump is a Unix tool that allows you to look at all traffic on a given Ethernet port.
  - This assumes that you have permission to run this command – generally, it needs to be installed & configured as an administrator.
  - See http://danielmiessler.com/study/tcpdump/

TCP dump info

- You'll see real time incoming / outgoing packets for your host
- You'll need to store the output to packets
- You'll need to search the captures.
- You can then search for a (random packet) in the noise.
  - The previous methods for searching various techniques (database, grep, etc.) to find that packet

Commercial Tools

- There are multiple commercial tools that do "real time" analysis. Google for various tools.
- Some tools (at various price points):
  - NetWitness
  - Solera

Questions?

- Ask other people for answers that are better than mine.
- Only 2 more slides, no break! ☺

Caveats

- Internet security is a huge field.
- There will always be new attacks & defenses
  - I was unable to share an attack that I wrote with you because newer version of Tor successfully prevent the attack.
- There will always be security conferences.

Questions?