Logging: not just a good idea

Eddy Vanlerberghe

owasp@evilwan.be

OWASP
October 23, 2008

The OWASP Foundation
http://www.owasp.org

Introduction

- Logging often not formally planned or designed
- Frequently insufficient in case of incidents
- Implemented by developers “as they go”
- Registered in insecure locations
- Relevance of logged information inadequate

Definition

"Information produced by an application that is not strictly required for its core functionality."

Border Case: User Visible Error Messages

- Volatile nature: not permanently recorded
- Usually contents not intended for end-user
- May reveal too much information for attackers
- Often result of insecure configuration at server-side
- Sometimes due to undocumented “features” of third-party components

Different Interested parties

- Developer
- System Administrator
- Marketing
- Audit
- alt.hackers.malicious
- ...

Developer's Interest

- "If an error occurs, I want to know what to modify in which lines of which files."
- Personal angle: "Look how quickly I can fix any bug!"
- Security angle: minimize downtime, fix errors as soon as possible

System Administrator's Interest

- “Do we need bigger iron/network pipes?”
- “Why is the system reacting so slow today?”
- “Where did that daemon come from and who changed my root password?”
- Security angle: confidentiality, integrity and availability

Marketing Interest

- “Why are people skipping that super-duper flash movie we payed big bucks for?”
- Security angle: ???

Audit Interest

- “It wasn't our fault and here is the proof!”
- Security angle: non-repudiation, accountability

Hacker's Interest

- “So, what is the name of that table containing the creditcard details in their database?”
- Security angle: information leading to successful attacks, destruction or obfuscation of proof pointing in their direction

Web Server Logs

- Timestamp
- Remote IP address
- Requested resource
- Request result status and return length

```
127.0.0.1 - - [25/Jul/2008:14:59:20 +0200]
  "GET
  /dokuwiki/lib/exe/js.php?edit=0&write=1
  HTTP/1.1" 200 16902
127.0.0.1 - - [25/Jul/2008:14:59:21 +0200]
  "POST /dokuwiki/doku.php HTTP/1.1" 302 -
```

Web Server Logs (cont.)

- Full request content not available: no cookies, no POST-ed parameters
- Response content not available: no cookies being set, only total length of response
- IP address does not equal “Jane Doe, 1600 Pennsylvania Ave NW, Washington, DC 20500”
- Are **ALL requests recorded? (can errors cause logging to be skipped?)**
- **IP address is often the internal address of a load balancer, reverse proxy or WAF**

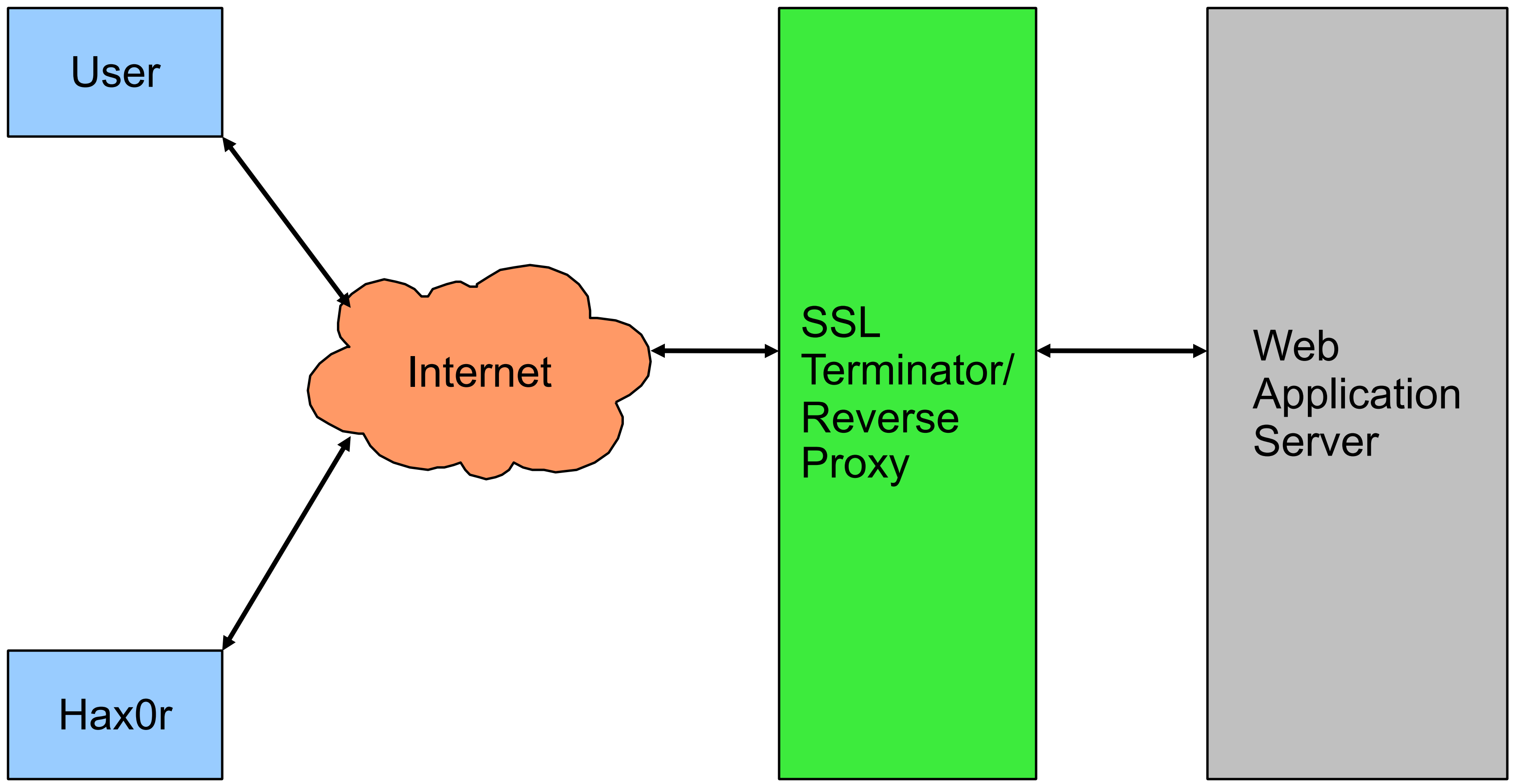
Typical Application Logs

- Are usually intended for developers only (e.g. "`13/10 12:13:14 Tx 88944890 started`")
- Not always taking multithreading issues into account: three consecutive log entries can be from two different threads, and information of different threads may not be in chronological order
- Often not part of up-front design, especially with respect log management (backups, log rotation, access rights,...)

Transaction Related Logs

- Intended to be used for official actions such as settling disputes, input for accounting (e.g. number of transactions executed per month) etc.
- Part of up-front design
- Should be reviewed for intended purposes:
  - Is logged information sufficient for intended purpose?
  - Is the logged data stored securely?
  - What are the policies and procedures for handling backups? (off-site, encrypted,...)

Example Setup



Data Flow

- Web service uses one URL for all transaction requests (“`/doTransaction.jsp`”)
- User sends cookie containing account number
- Back end server executes transactions on behalf of account specified in cookie
- Back end logs transaction data: time, source account, destination account, amount, description, IP address reverse proxy
- Reverse proxy logs “POST” requests
- Clocks of proxy and web server are not sync'd

Log Contents

Proxy:
```
1.2.3.4 - - [2008-07-11:14:59:20] "POST
http://webserver/doTransaction.jsp HTTP/1.1" 200 1234
5.6.7.8 - - [2008-07-11:14:59:20] "POST
http://webserver/doTransaction.jsp HTTP/1.1" 200 1122
```

Web Server:
```
10.0.0.2 - - [2008-07-11:14:57:33] "POST /doTransaction.jsp
HTTP/1.1" 200 1234
10.0.0.2 - - [2008-07-11:14:57:33] "POST /doTransaction.jsp
HTTP/1.1" 200 1234
```

Application Log:
```
10.0.0.2 2008-07-11:14:57:33 123456789012 210987654321 5000
Electricity
10.0.0.2 2008-07-11:14:57:33 123456789012 111222333444 5000
Electricity
```

Typical Questions To Be Answered

Logging with security in mind: questions that need answers based on available logged information:

- When?
- Who?
- What?
- Where?
- How?
- Why?

When?

- Can be required to determine the "Who"? (typically dynamic IP addresses are re-used by multiple persons over time)
- Often used to link information from different logging sources (e.g. for building timelines during forensic investigations)
- Importance of accurate system clocks across all systems involved

Who?

- Ask yourself: if something happens, do I have enough information to identify the culprit?
- Physical person? Organization?
- Remote IP address (beware of reverse proxies, load balancers or WAFs)
- Indication of open WiFi being abused?
- Application level identification? (usernames, account numbers,...)
- May need help from law enforcement for resolving IP address in owner information

What?

- Ideally: all traffic going in and out
- Often not realistic
- Minimum:
  - Time
  - Remote IP
  - Resource accessed + parameters supplied
  - Result status + most important info returned
  - Diagnostics generated during handling of request
  - Application specific required electronic evidence (digital signatures, ...)

Where?

- Identify which component generated the log entry (WAF filter? Application digital signature verification?...)
- Location of intruder?
  - Insider? (involve human resource departement?)
  - Domestic attacker? (case for local LE?)
  - Foreign attacker? (block entire countries from site?)

How?

- Investigate how an intrusion occurred
- Which weaknesses were abused?
- Can the incident occur again? (e.g. if an old server, with old software was replaced as part of the containment, the new situation may be more secure)
- What would be the most effective ways to block the intrusion from happening again? (helps to prioritize new protective measures)

Why?

- Can be used to prevent attacks being launched by taking away the reason why they occurred
- If disgruntled customer: keep them happier?
- If disgruntled employee: look at ways to keep employees happier?
- "Because I can": not much to do against that motive except building a fortress

“Secure Logging”

- Implement chain-like functionality:
  - line counters
  - (signed) hashes of previous record(s)
- Use independent, isolated log servers in a physically controlled environment
- Use write-once devices
- Include digital signatures on each line provided by dedicated “notarial” systems

If Push Comes To Shove...

- Court case: in Belgium the goal is to convince the judge(s) that you are right and the other party is wrong
- Electronic evidence is different compared to paper documents
- Make up for possible uncertainty by:
  - Redundant logging by independent systems
  - Show how logging is produced by automated processes
  - Keep several generations of backups in physically different, but secured, locations

Proactive Usage of Log Info

- Implement monitoring on generated log data
- Define thresholds for “interesting” events
- Create progressive escalation infrastructure
- Block suspected malicious outsiders
- Dangers:
  - False positives
  - Blocking of legitimate users
  - Too many escalation alerts erode their effectiveness

Handling Log Data

- Can contain confidential information
- Plan to be able to quickly look at part of logged data (timeframe, origin based, ...)
- Make backups
- Plan on long-term storage
- Beware of potential dangerous contents (e.g. XSS attack as part of requested URL, referrer or user-agent string containing XSS,...)

Conclusions

- Logging is an important part of non-repudiation: record not only approvals/hashes/signatures, but also the entire process
- Record sufficient information to reconstruct the path from user to database
- Beware of time stamps from different systems and reverse proxies
- Log data can contain confidential information and should be protected as such
- Proactive measures can have undesired side effects

Questions?