



NT OBJECTIVES,  
INCORPORATED

# Web App Session Strength

## Measuring Threats to Identity Theft

Mike Shema <mikeshema@yahoo.com>  
BlackHat Las Vegas, July 2004

# Welcome

- Session Strength is only a small part of web application security, but an attack against a poor implementation can cripple the entire site.
  - Harder to identify by automated tools because the most common attacks rely on logical and semantic errors.
  - Harder (impossible?) to defend with application-level firewalls.
- Other popular attacks merely rely on syntax errors that are typically easy to identify.
  - SQL Injection
  - HTML Injection (Cross-Site Scripting)

# Syntax Errors

- Syntax errors are easy to identify and easy to fix.
- In fact, Microsoft is considering the removal of RAW socket support from Windows XP SP2.
  - First, the attacker exploits an unpatched service (buffer overflow), a misconfigured server, insecure default configuration, a poor password, or uses a social engineering technique.
  - Once the system is compromised, the attacker can use RAW sockets to make the attack more evil!!!
- The security community should recommend additional security fixes that would improve web application security.

# SQL Injection – The Solution

- Security Fix: Remove support for ASCII character 0x27 from IIS (or at least Internet Explorer).
- The advantages of blocking apostrophe support.
  - Remove a tool that makes an evil hacker's SQL injection attack easier to accomplish.
  - They're rarely used in general.\*
  - It'd protect insecure web applications from evil hackers.
- Let's lobby Microsoft: [secure@microsoft.com](mailto:secure@microsoft.com)

*\* Over an eight hour period, ASCII 0x27 accounted for less than 4% of the characters present in top stories listed on news.google.com*

# Applying State to a Stateless Protocol

- HTTP is a stateless protocol; the web server receives discrete packets.
  - The protocol does not have a robust capability for tracking a user's activity across multiple requests.
  - Cookies can help fix this problem – provided they're used correctly.
- Session tokens only provide identification.
  - At least, the token's value should *only* contain an identifier.
  - Access rights and roles are associated with the session.
- We'll use the term session “token” instead of cookie because the session can be tracked in multiple places.

# What Are The Threats?

- The goal of a session attack is to obtain access to an application under someone else's credentials.
- Most session attacks only require the attacker to obtain or guess a single value.
  - The attacker does not need to know a valid account name, password, or other “secret” information.
  - The attacker does not need to target a single page of the application.
  - Less likely to be caught by brute-force login countermeasures.
  - *Supposed* to have a larger keyspace.

# Keyspaces

- Username (Alphanumeric, sometimes an e-mail address)
  - 64 symbols, 6 or more characters
  - Keyspace: 68,719,476,736 ~ 36 bits
- Password (Alphanumeric, punctuation)
  - 96 symbols, 6 or more characters
  - Keyspace: 782,757,789,696 ~ 40 bits
- Session ID (Numeric or numeric transform)
  - Defined by bit length (32, 64, 128, 256)
  - Crippled by deterministic content.
    - Incremental
    - Time-based
    - Static (based on user attribute)

# Types of Session Attacks

- Server-directed attacks require interaction with the server.
  - **Analyze** – Inspect the token for deterministic string content.
  - **Predict** – Inspect the token for deterministic string or mathematical content.
  - **Guess** – Inspect the token for deterministic string or mathematical content.
- Server-directed attacks *might* require action from the victim (login to the application).



# User-Directed Session Attacks

- User-directed attacks either require network proximity to either the client or server, or require an application vulnerability.
  - **Sniff** – Use tcpdump, dsniff.
  - **Hijack** – Use dsniff.
  - **Steal** – Use XSS, social engineering.
- Extremely difficult to mitigate from the application layer – not the focus of this presentation.

# Session Attacks

- So how do I figure out which value is used as the session token?
  - Trial and error
  - Educated guesses
  - Use multiple accounts
- We'll look at some real-life web applications.
  - These applications are not necessarily insecure, they are just interesting to examine.
  - Each application implements a different kind of session token.

# Relative Security

- When is a session token secure? Not secure?
- What are the pros and cons of using a static session token vs. a dynamic one?
- Are server-generated tokens adequate?

# Types of Token Management

- Static tokens
  - The user receives the same token value between sessions.
  - Often tied to a user attribute (user name, user ID, password)
- Observations
  - The target pool includes all users of the application.
  - The victim does not have to be active in the application in order for the attacker to guess a valid token.
  - The token cannot be expired, or must have user interaction to be expired.

# Types of Token Management

- Dynamic tokens
  - The user receives a different token value between sessions.
  - Often based on the output of a PRNG.
  - Sometimes tied to a user attribute (user name, user ID, password, connection info)
- Observations
  - Smaller target pool includes all active users of the application (active users < all users).
  - Inadequate token expiration greatly increases the chance of a successful guess.

# Session Attacks

Example of token in a cookie with trivially deterministic content.

# Static Token

ThinkGeek :: My Account - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.thinkgeek.com/brain/account/index.cgi?dsid=b9c2d538t

## private's Spiffy ThinkGeek.com Account

Welcome to your account, No **12f93a9**. err, we mean private private. From here you can view your order history, update your address book, check out special rewards for which you're eligible, and tons of other wonderful things.

<b>Account Info</b>	Edit the name, e-mail address, and subscription options for your account.
<b>Password</b>	Update your password here.
<b>Order Management</b>	View and manage all of your completed and incomplete orders. Cancel newly placed orders. Return recently shipped orders.

Navigate by Interest:

Done

# Static Token

Cookie: s\_cc=true; s\_sq=thinkgeek%3D%2526pid%253D/brain/account/index.cgi%2526pidt%253D1%2526oid%253Dhttps%25253A//www.thinkgeek.com/brain/account/finger.cgi%2526ot%253DA%2526oi%253D199; luckymonkey=12f93a9; spunkymonkey=912ab19cecf67104aff954e1



# Static Token

- Four cookies are set: *s\_cc*, *s\_sq*, *luckymonkey*, *spunkymonkey*
- Two are required to identify a session: *luckymonkey*, *spunkymonkey*
- One is required to identify a user: *luckymonkey*
  - Permanently identifies the user
  - Used to retrieve user profile information

# Observations

- The user identifier and the session identifier are independent
  - This is an authorization problem.
  - A more secure method is to tie the user identifier to the session object on the server.
- The application implicitly trusts client-side data (*luckymonkey* value).
- The *luckymonkey* value is derived from a small pool of numbers.
- Valid numbers can be enumerated at will.
  - No invalid syntax is submitted to the server.
  - Success does not require a large amount of requests in a short amount of time.

# Pool Size & Density

- Pool size (**P**):  $16^7 = 268,435,456$
- Guesses (**n**): Number of values the attacker enumerates; say 10,000
- Speed (**s**): Number of guesses per second submitted by the attacker; say 1
- Density (**D**):  $P / n = 26,844$
- The attacker has ~ 100% chance of successfully guessing a valid token within 3 hours if there are more than 26,844 accounts defined in the application.

# Pool Size & Density

- All this in only 168 minutes of work – 2 minutes to write a script and  $166 (n/(s*/60))$  minutes to wait.
  - The time (**s**) can be increased in order to reduce the chances of setting off real-time alerts.
  - The attack is unaffected by session timeouts and the number of active users.
- If token values are assigned sequentially or in an increasing manner, then this is just an upper bound.
  - Less guesses (**n**) are required to approach 100% success.
  - The target range can be significantly reduced by observing values between two times (one day, one week).

# Pepsi

Example of encoded tokens in an HTML form with descriptive information and time stamp.

# Pepsi

Pepsi-Cola - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://prefctr.ddc.dartmail.net/pepsi/manage\_profile/pepsi\_PrefCtr.asp?INIT\_REG\_TF

## JOIN PEPSI-COLA



### LOGIN

Here you'll be able to update any of your profile information, including your email and mailing address, and change your email newsletter subscription preferences.

Email address:

Password:

**SUBMIT** →

# Encoded Tokens

```
POST /pepsi/manage_profile/pepsi_PrefCtr.asp?cmd=MainEntry
HTTP/1.0
Host: prefctr.ddc.dartmail.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.5)
Accept: text/xml,text/html,text/plain,image/png,image/jpeg,*/*
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Referer:
http://prefctr.ddc.dartmail.net/pepsi/manage_profile/pepsi_PrefCtr.
asp?INIT_REG_TREATMENT_CODE=21
Cookie: ASPSESSIONIDQSSDQATT=BHCBNDACPKKIKLJNDKCBKDOB;
ASPSESSIONIDQSSRCATS=IABIIDPBMOGKDDGIPPPNCKAB
Content-Type: application/x-www-form-urlencoded
Content-Length: 148

FTAFemailaddr=&FTAFmessagename=&FTAFforIAF=&GenerateNewPassword=0&
EMAIL=arbogoth%40yahoo.com&PASSWORD=nicetry&image1.x=83&image1.y=
12
```

# Observations

- None of the ASPSESSIONID\* cookies are required for the request.
- You must have valid credentials.
- You cannot target other users without guessing their username and password – this presents a significant challenge to the attacker.
- The session integrity appears to be well-kept, but session confidentiality is another matter...



# Encoded Tokens

```
<input TYPE="hidden" name="CURRENT_EMAIL"
value="arbogoth@yahoo.com">
<input TYPE="hidden" NAME="FTAfemailaddr" VALUE="">
<input TYPE="hidden" NAME="FTAfmessagenam" VALUE="">
<input TYPE="hidden" NAME="FTAforIAF" VALUE="">
<input type="hidden" name="RequestNewPassword">
<input type="hidden" name="oldpassword" value="nicetry">
<input type="hidden" name="SessionId"
value="MDAwMDAwMD1jb3JlMjRkbTEwMDAwMDAwMjIyMDAwMDAwMDg3OTk5Mjc4ND
AwMDAwMDAwMDAwMDAwMDE4YXJib2dvdGhAeWFob28uY29tMDAwMDAwMTAxMDc1Nzg
xNTg3">
<input TYPE="hidden" NAME="CONTEST_WINNER" VALUE="">
<input TYPE="hidden" NAME="STATE_select" VALUE="">
<input TYPE="hidden" NAME="AFFINITY_select" VALUE="11">
<input TYPE="hidden" NAME="PHONE">
<input type="hidden" name="Birth_Date" VALUE>
<input type="hidden" name="US_RESIDENT_IND" value>
<input TYPE="hidden" NAME="dob_mn_select" VALUE="01">
<input TYPE="hidden" NAME="dob_dy_select" VALUE="01">
<input TYPE="hidden" NAME="dob_yr_select" VALUE="1901">
```

# Encoded Tokens

- Examine the SessionId value

```
MDAwMDAwMD1jb3JlMjRkbTEwMDAwMDAwMjIyMDAwMDAwMDg3OTk5M  
jc4NDAwMDAwMDAxMDAwMDAwMDE4YXJib2dvdGhAeWFob28uY29tMD  
AwMDAwMTAxMDc1NzgxNTg3
```

- Base64 decode

```
00000009core24dm1000000022200000008799927840000000100  
0000018arbogoth@yahoo.com000000101075781587
```

- Some fields
  - [arbogoth@yahoo.com](mailto:arbogoth@yahoo.com) = e-mail address
  - 1075781587 = time stamp

# Additional Observations

- The encoded SessionId value does not contain any sensitive information that isn't already included elsewhere in plain text.
- This is an example of static management of session tokens.
  - The user must submit a valid username and password.
  - The tokens are derived from a large pool (based on e-mail address and password complexity).
  - The user can affect the token (password).

# Hotmail (Passport)

Example of encrypted tokens in cookies with static content.

# Encrypted Values

- Two cookie values required to maintain session state and user identification.
  - MSPAuth
  - MSPPProf

# Encrypted Values

- MSPProf=5DfgRUYOKD1kMnECRoOcHPDKfgA7I7Zm  
Ooh0\*YmZTdgIQBmi2LQ1m2xMaNIR1DoEDwzhP6R!3  
UNu20gYn1GtCGeou2xfVMqM5v\*XykC18L7jj6KU9DC  
wSwL0JiG83hsq3P1gUxSac2!zDhR3HHGuoQjHHjEHv  
p39zEbBqQj8vP3XirIWn6i04hgCyPPK\*Zc3TibXZwRwJ  
QaqwN9gJo2VcEAQ\$\$
- MSPAuth=55gQxyNTIDuABLif4JgWUNYXpgtdfcm2Zc  
SIUFrFqxuTNiyXA8qEkaTiXcj9ovqtLLoPVMMg9NNHs  
SLp43LZnZRA\$\$

# Observations

- Example of static token management
  - Tokens are encrypted with a secret key
  - Tokens remain valid for a long period
- Threats
  - Chosen plaintext attack if the cookie structure can be determined.
  - Compromise secret key
  - Replay

# iTunes Music Store

Example of static and random tokens in HTTP Headers.



# iTunes Music Store

The screenshot shows the iTunes Music Store interface. At the top, there's a menu bar with 'File', 'Edit', 'Controls', 'Visualizer', 'Advanced', and 'Help'. Below the menu bar are playback controls (play, stop, next) and a search bar with the Apple logo. The search bar contains the text 'Search Music Store' and 'Browse'. The main content area is titled 'Search Results for: pink floyd' and shows 'Top Albums' and 'Top Songs'. The 'Top Albums' section features two albums: 'Dark Side of the Moon' by Pink Floyd (Genre: Rock, \$16.99) and 'The Wall' by Pink Floyd (Genre: Rock, \$25.74). The 'Top Songs' section lists several songs, including 'Another Brick in the Wall, Pt. 1', 'Comfortably Numb', 'Money', and 'Wish You Were Here'. Below these sections is a table of search results. The table has columns for 'Song Name', 'Time', 'Artist', 'Album', and 'Relevance'. The results list songs from 'The Wall' album, such as 'In the Flesh?', 'The Thin Ice', 'Another Brick in the Wall, Pt. 1', 'The Happiest Days of Our Lives', 'Another Brick in the Wall, Pt. 2', 'Mother', and 'Goodbye Blue Sky'. The bottom of the window shows a status bar with '158 songs' and various control icons.

Source

- Library
- Radio
- Music Store
- Shopping Cart
- My Top Rated
- Recently Played
- Top 25 Most Played

Search Results for: pink floyd

Account: mikesHEMA@yahoo.com

**Top Albums**

- Dark Side of the Moon**  
Pink Floyd  
Genre: Rock  
\$16.99 [ADD ALBUM](#)
- The Wall**  
Pink Floyd  
Genre: Rock  
\$25.74 [ADD ALBUM](#)

**Top Songs**

- Another Brick in the...
- Comfortably Numb
- Money
- Wish You Were Here
- Another Brick in the...

Song Name	Time	Artist	Album	Relevance
In the Flesh?	3:20	Pink Floyd	The Wall	
The Thin Ice	2:29	Pink Floyd	The Wall	
Another Brick in the Wall, Pt. 1	3:10	Pink Floyd	The Wall	
The Happiest Days of Our Lives	1:51	Pink Floyd	The Wall	
Another Brick in the Wall, Pt. 2	3:59	Pink Floyd	The Wall	
Mother	5:36	Pink Floyd	The Wall	
Goodbye Blue Sky	2:48	Pink Floyd	The Wall	

158 songs

# Static & Dynamic Tokens

```
GET /WebObjects/MZStore.woa/wa/com.apple.jingle.app.store.DirectAction/shoppingCart HTTP/1.1
Accept-Language: en-us, en;q=0.50
X-Token: 30303030303031303831333030373836
User-Agent: iTunes/4.2 (Windows; U; Microsoft Windows 2000 Professional Service Pack 4 (Build 2195)) DPI/96
X-Dsid: 96723735
Cookie: countryVerified=1
Host: phobos.apple.com
```

# Static & Dynamic Tokens

- Looking at the headers

```
X-Token: 30303030303031303831333030373836
         0 0 0 0 0 0 1 0 8 1 3 0 0 7 8 6
                                     1,081,300,786
                                06/04/2004 06:04:46 GMT
```

X-Dsid: 96723735

- The X-Token value uses a 16 digit value, but the actual values are time-based (86,400 seconds per day)
- The Dsid is a random 8-9 digit number ( $\sim 2^{30}$  values)
- In order to impersonate an account, you must guess *both* values – this isn't easy. ( $> 2^{46}$  potential combinations)

# Slashcode 2.2.6

Example of encoded tokens in cookies with static content.



# Slashcode 2.2.6

MacSlash: A daily dose of Macintosh News and Discussion - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.macslash.com/

# Mac Slash

A Daily Dose of Mac News and Discussion

### MacSlash

- » [FAQ](#)
- » [Discussions](#)
- » [Journals](#)
- » [Messages](#)
- » [Topics](#)
- » [Authors](#)


---

- » [Preferences](#)
- » [Older Stuff](#)
- » [Past Polls](#)
- » [Submit Story](#)

### Interactive RsyncX v2.0 Webcast

posted by [acaben](#) on Friday April 16, @06:09AM  
from the **coming-up-next** dept.

Anonymous Coward writes "*The MacOSXLabs.org group will be presenting an "RsyncX 2.0" WebCast on Tuesday, April 20 at 1:00 pm EDT. This WebCast will introduce version 2.0 of RsyncX, a suite of tools for file backup and distribution for Mac OS X. A case study demonstrating the functionality of RsyncX in an enterprise environment will be presented by The University of Michigan School of Art and Design. For more information on how to watch this WebCast, please visit the*



### Login

Nickname:

Password:

[ [Create a new account](#) ]

### Poll

Gmail addresses I'm waiting to grab

1

2

Search MacSlash:

# Encoded Tokens with Static Content

- Cookie value:  
**%2531%2533%2530%2536%2538%253a%253a%257b%2599%2593%252d%257b%25a5%257b%2552%2587%259a%2561%25bc%2544%253d%25bf%2503**
- After replacing URL-encoded characters, equivalent to:  
**%31%33%30%36%38%3a%3a%7b%99%93%2d%7b%a5%7b%52%87%9a%61%bc%44%3d%bf%03**
- After replacing URL-encoded characters, equivalent to:  
**13068::7b99932d7ba57b52879a61bc443dbf03**
- Reveals this format (just a guess based on content and length):  
**<id>::<md5sum>**

# Observations

- Keyspace is highly influenced by the user (password hash).
- The token cannot be regenerated without user (or administrator) interaction.
- A compromised token provides full access.
- A token can be attacked off-line.

# Static Tokens

- The attacker is not inhibited by session expiration.
- The victim does not have to be active (or recently active) within the application.
- The probability of a successful guess increases with the total number of users.
- Static content is more likely to enable the attacker to target a specific victim.
- If a static token is compromised, then it is likely that a user-initiated action is required to change the token.
- Static tokens often employ user-generated entropy (humans != good entropy generators)



# Dynamic Tokens

- The attacker is inhibited by proper session expiration.
- The victim must be active (or recently active) within the application.
- The probability of a successful guess increases with the total number of concurrent users.
- If a dynamic token is compromised, then the scope of the compromise *might* be limited to a short period.

# Token Prediction

- The strength of a PRNG can be measured to some degree of confidence.
  - Entropy
  - FFT (frequency analysis, correlation)
  - Non-linear time series
- The PRNG should generate values from a large enough pool to limit the risk of brute-force attacks.

# Token Prediction

Examples of useful algorithms (Entropy, FFT, Time-series analysis)

# Random Token Pool Considerations

- How does the application use the token?
- How many concurrent sessions are expected?
  - A large amount of concurrent sessions increases the target space.
- What is the lifespan of the token?
  - A longer lifespan increases the target space.
- Is the token keyed to an attribute of the user?
- Seed the PRNG early and seed regularly.

# Server-Generated Values

- Many web servers and application engines provide methods for generating random session IDs.
  - Also used to track the session object
  - Provides methods for manipulating the data related to a session.
- Need to be able to trust the session ID
  - How “random” is it?
  - Does the token contain user-related data?

# Server-Generated Tokens

- A quick comparison of cookies generated by common servers.
- Comparing similar values between dissimilar sources.

# Countermeasures

- Input validation is important for protection from other attacks, but does not prevent session guessing or impersonation attacks.
- Strong passwords and authentication systems do not prevent session attacks, but they can reduce the impact of a successful attack.
- Remember, a session attack uses legitimate data as part of a legitimate request made by an unauthorized user.
  - It's difficult to create IDS rules to watch for these attacks.
  - Good logging by the web server will catch these attacks, but not as a proactive measure.

# Countermeasures

- Temporarily identify the user with a session token.
  - Identification does not have to be based on an attribute of the user.
  - Random values
- Do not describe the user with a session token.
  - Profile information (e-mail address, username)
  - Privilege information (user, admin)
  - Sensitive information (SSN)
- Use server-side session objects to describe the user.
  - Site privileges
  - Data privileges
  - The session object remains on the server and cannot be directly manipulated by the user or sniffed by an attacker.



# Countermeasures

- The session should expire under several circumstances
  - When the user logs out of the application.
  - When a predetermined amount of time has passed, regardless of user activity.
  - When the user's session is inactive for a predetermined amount of time.
- Benefits
  - Further protects user in environments where computers are shared or physical access is difficult to control.
  - Decreases the amount of valid sessions concurrently available, which makes token guessing more difficult.

# Countermeasures

- How long is a “predetermined amount of time?”
- Determine how users interact with your application.
  - What is the average time a user would normally spend on the site?
    - An HTTP e-mail service or a forum would probably encourage users to be on for a long time, perhaps 8 hours (a work day).
    - An on-line banking application has more threats to consider, perhaps 30 minutes is appropriate (check account balance, perform a trade).
  - What type of information is stored in the application?
    - Financial and personal information pose a greater risk than an e-mail address and viewing preferences.

# Countermeasures

- Observe and model user behavior to improve session management.
  - Average time active
  - Percent of users who use logout feature
  - Percent of users who use “remember me” feature

# Countermeasures: Data Security

- Mask sensitive information to prevent exposure even if the account is compromised.
  - Credit card numbers  
xxxx-xxxx-xxx5-4321
  - Personal ID numbers  
xxx-xx-x321 (Social Security Number)
  - Passwords  
xxxxxxxx
- Require re-authentication before executing sensitive transactions.
  - Change password
  - View, edit financial profile

# Summary

- Examine the token generation method
  - Static
  - Dynamic
- Examine the token content
  - Does temporarily identify the user
  - Does not permanently describe the user
- Examine the token transformations
  - Encode
  - Encrypt
  - Hash

# References

- Tools
  - Netcat, [http://www.atstake.com/research/tools/network\\_utilities/](http://www.atstake.com/research/tools/network_utilities/)
  - OpenSSL, <http://www.openssl.org/>
  - Paros Proxy, <http://www.proofsecure.com/>
  - Scilab, <http://www.scilab.org/>
- Books
  - *Hack Notes: Web Security*
  - *Hacking Exposed: Web Applications*
- Whitepapers & Resources
  - <http://www.sans.org/rr/papers/60/480.pdf>
  - <http://www.pdos.lcs.mit.edu/papers/webauth:sec10.pdf>
  - <http://www.cgisecurity.com/lib/SessionIDs.pdf>
  - <http://www.mpipks-dresden.mpg.de/~tisean/>
  - <http://razor.bindview.com/publish/papers/tcpseq.html>

# Some Helpful Tips

- Use Perl to guess MD5 hash content:  

```
use Digest::MD5 qw(md5 md5_hex md5_base64);  
$digest = md5($data);  
$digest = md5_hex($data);  
$digest = md5_base64($data);
```
- Use Perl to guess SHA1 hash content:  

```
use Digest::SHA qw(sha1 sha1_hex sha1_base64);  
$digest = sha1($data);  
$digest = sha1_hex($data);  
$digest = sha1_base64($data);
```
- You cannot decrypt a SHA1 or MD5 hash, but you can compare values.

# Some Helpful Tips

- Base64 data show up quite often; be aware of alternate symbol schemes.
- Quick Base64 decoder (Perl)
  - `#!/usr/bin/perl`  
`use MIME::Base64;`  
`print decode_base64($ARGV[0]);`
  - `./bd64.pl aGVsbG8gd29ybGQh`  
`hello world!`
- Quick Base64 Encoder (Perl)
  - `#!/usr/bin/perl`  
`use MIME::Base64;`  
`print encode_base64($ARGV[0]);`
  - `./be64.pl 'hello world!'`  
`aGVsbG8gd29ybGQh`



# Some Helpful Tips

- Timestamps show up even more often
  - Convert date & time to epoch time
    - `date +%s`  
1082139295
    - `date "Apr 27 12:00:00 GMT 2004" +%s`  
1083067200
  - Convert epoch time to date & time (Perl)
    - `perl -e 'use Time::localtime; print ctime(1083067200)'`  
Tue Apr 27 08:00:00 2004
- Timestamps may also include milliseconds.