A framework to 0wn the Web

w3af

Andrés Riancho
andres@bonsai-sec.com

H2HC, Brazil - 2009

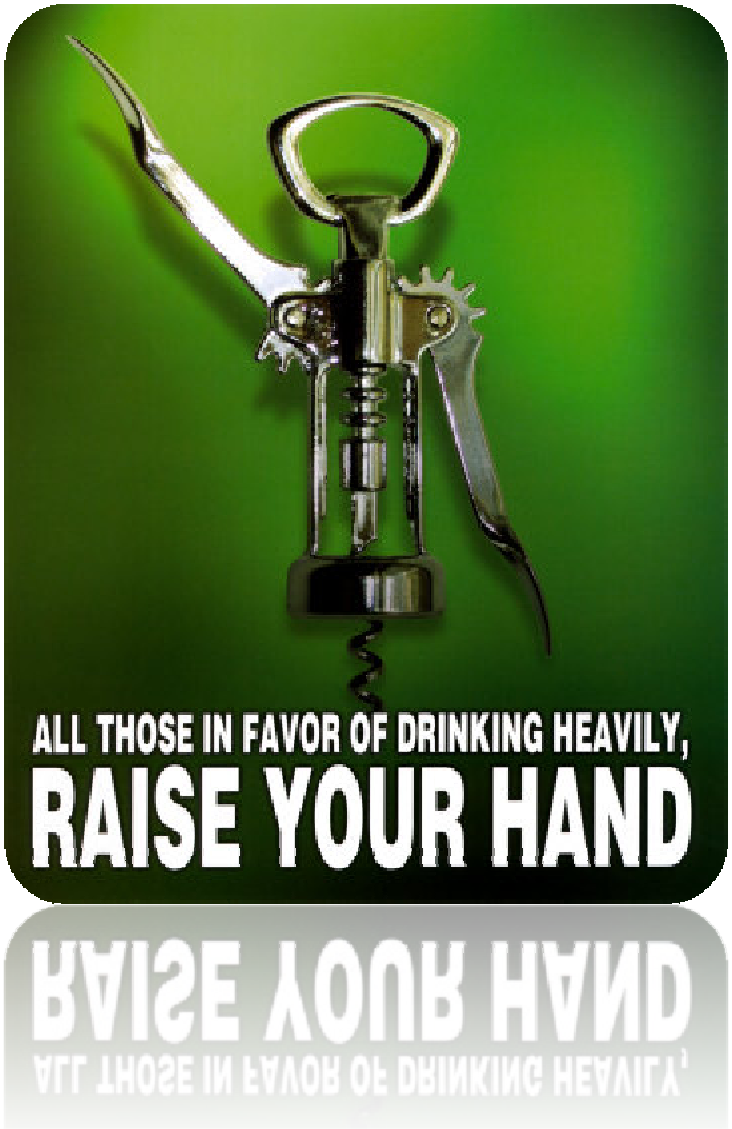
andres@bonsai-sec:~$ whoami

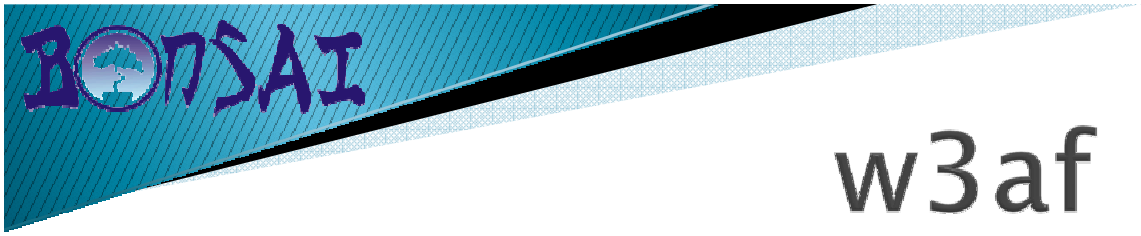
- Web Application Security enthusiast
- Developer (python!)
- Open Source Evangelist
- With some knowledge in networking, IPS design and **evasion**
- w3af **project leader**
- Founder of **Bonsai** Information Security

BONSAI
INFORMATIONSECURITY

Raise your hands!

- Who works with Application Security?
- Who heard about w3af?
- Who used w3af?
- Who tried to read w3af's code?
- Who hacked w3af's code?





w3af

- **w3af** stands for **W**eb **A**pplication **A**ttack and **A**udit **F**ramework
- An Open Source project (**GPLv2**)

- A set of scripts that evolved into a serious project

- **A vulnerability scanner**
- **An exploitation tool**
- **A set of manual analysis tools**

Main features

- **Identifies almost all** web application vulnerabilities using more than 130 plugins.

- Cross platform (written in python).
- **GTK and Console** user interface
- **Really easy to extend**

- Uses Tactical exploitation techniques to discover new URLs and vulnerabilities

- Exploits [blind] SQL injections, OS commanding, remote file inclusions, local file inclusions, XSS, unsafe file uploads and more!

Main features

- **Synergy** among plugins

- WML Support (WAP)
- Broken HTML support
- A smarter fuzzer

- **Manual and automated** analysis web applications
  - **MITM proxy**
  - **Manual request editor**
  - **Fuzzy request generator**

- Least, but not less important: **an active community**

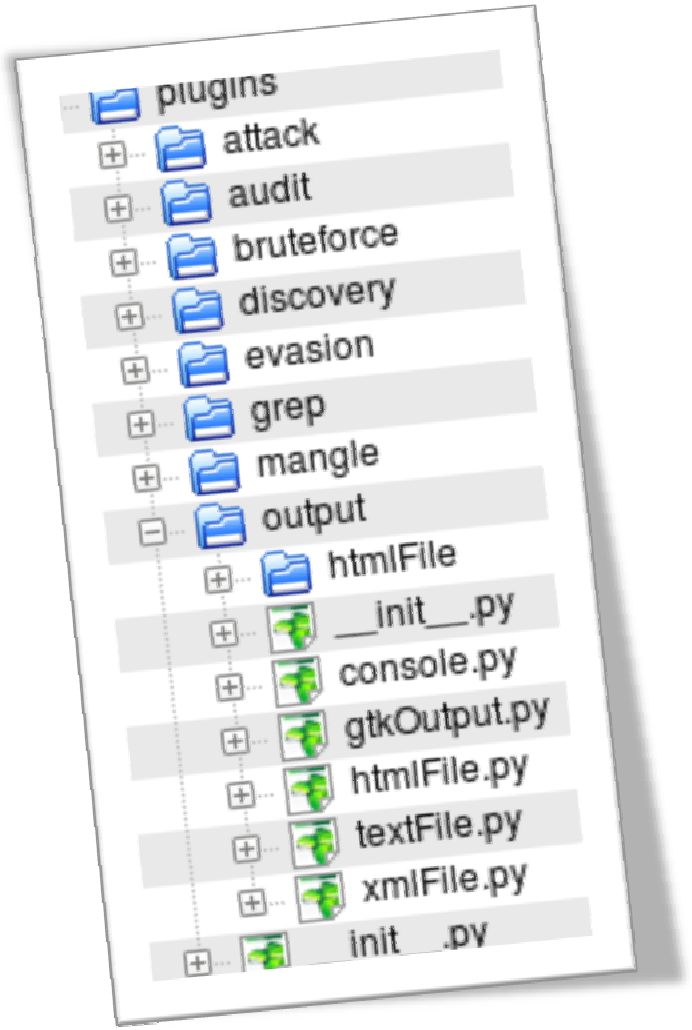
Compressed w3af history

Architecture

- w3af is divided in two main parts, the **core** and the **plugins**.
- The **core coordinates** the process and provides **features** that plugins consume.
- Plugins find the vulnerabilities, and exploit them.
- Plugins share information with each other using a knowledge base.
- *Design patterns and objects everywhere !*

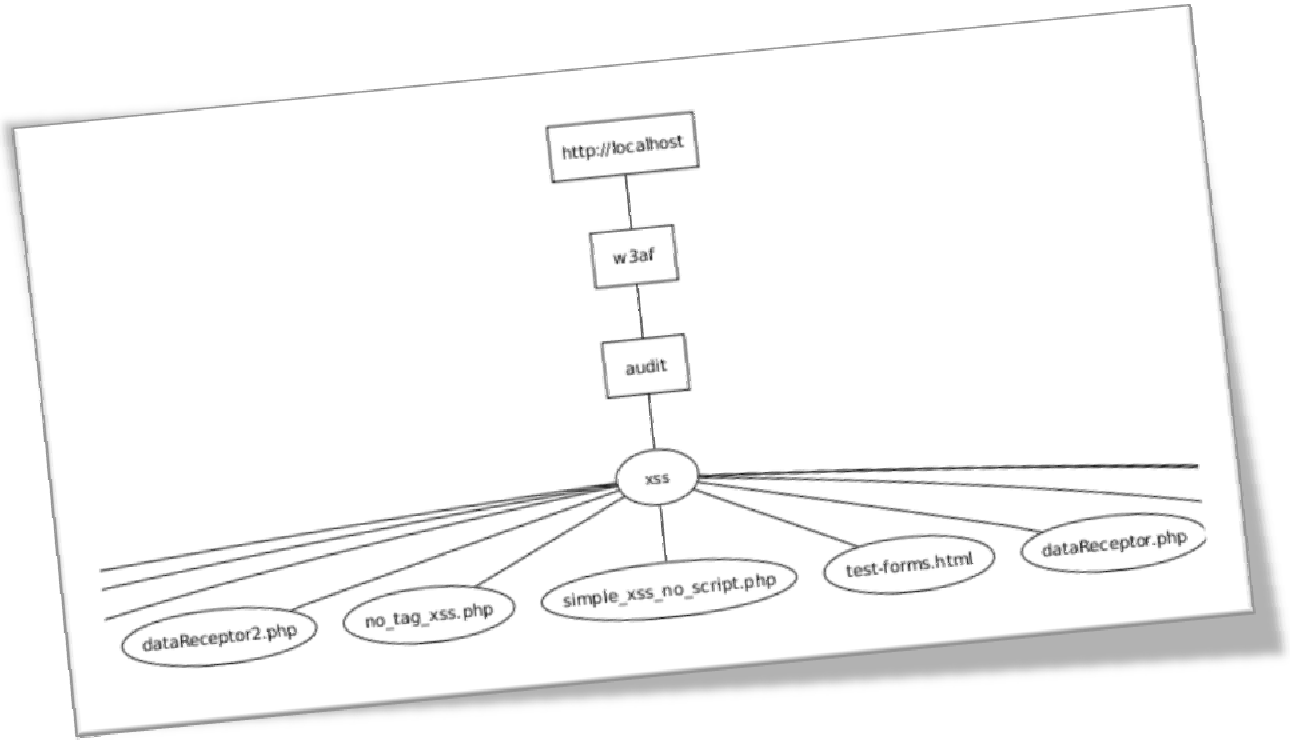
Architecture

- 8 different types of plugins exist:
  - **discovery**
  - **audit**
  - **grep**
  - **attack**
  - output
  - mangle
  - evasion
  - bruteforce



Plugins | Discovery

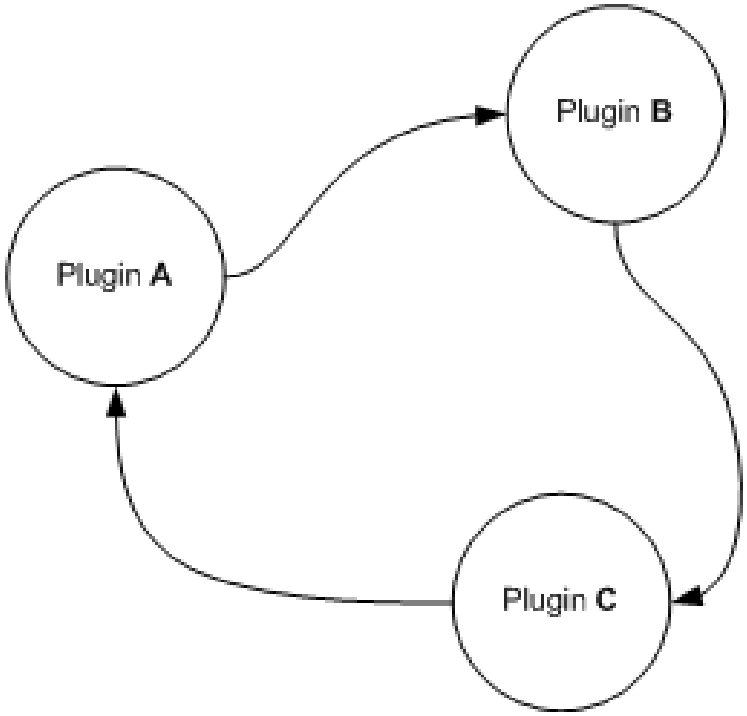
- They **find new URLs , forms**, etc. and create a complete sitemap. The findings are saved in the core as **fuzzable requests.** Examples of discovery plugins are:
  - **webSpider**
  - urlFuzzer
  - googleSpider
  - pykto



Plugins | Discovery

- They are **run in a loop**, the output of one discovery plugin is sent as input to the next plugin. This process continues until all plugins fail to find a new resource.

- This feature increases the **code coverage** of each scan, allowing the audit plugins to find more vulnerabilities.

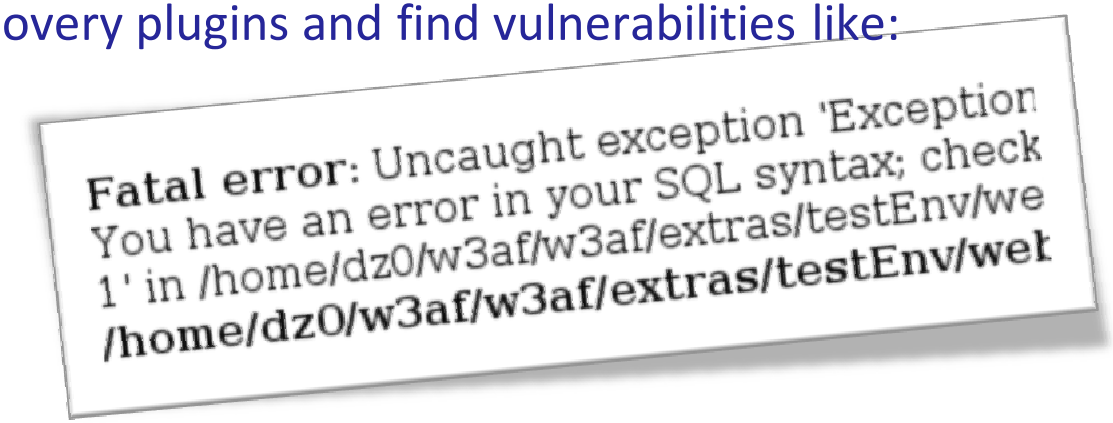


Plugins | Discovery

- Other discovery plugins try to fingerprint remote httpd, verify if the remote site has an HTTP load balancer installed, etc.
  - halberd
  - **hmap**
  - **afd**
  - fingerprint_WAF
- I need some **refactoring…**
  - Crawlers
  - Infrastructure

Plugins | Audit

- They take the output of discovery plugins and find vulnerabilities like:
  - [blind] SQL injection
  - XSS
  - Buffer overflows
  - Response splitting.

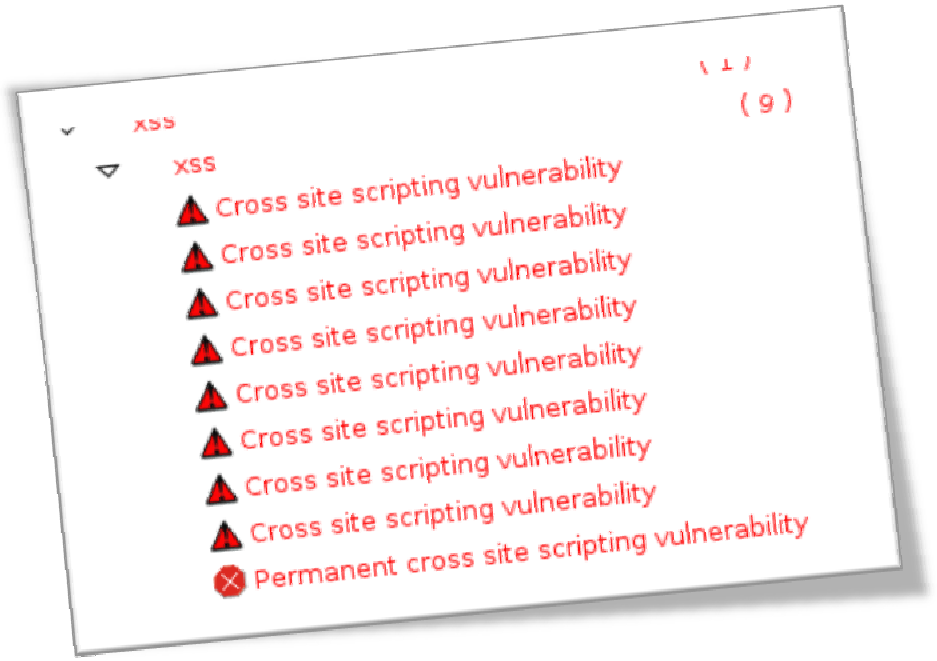


- Vulnerabilities are identified using **different methods**, that vary on the type of vulnerability being identified, but **when possible, all methods are used**:
  - Error based
  - Time delay
  - Creating a new remote file
  - Different responses (AND 1=1 , AND 1=2)

Plugins | Audit

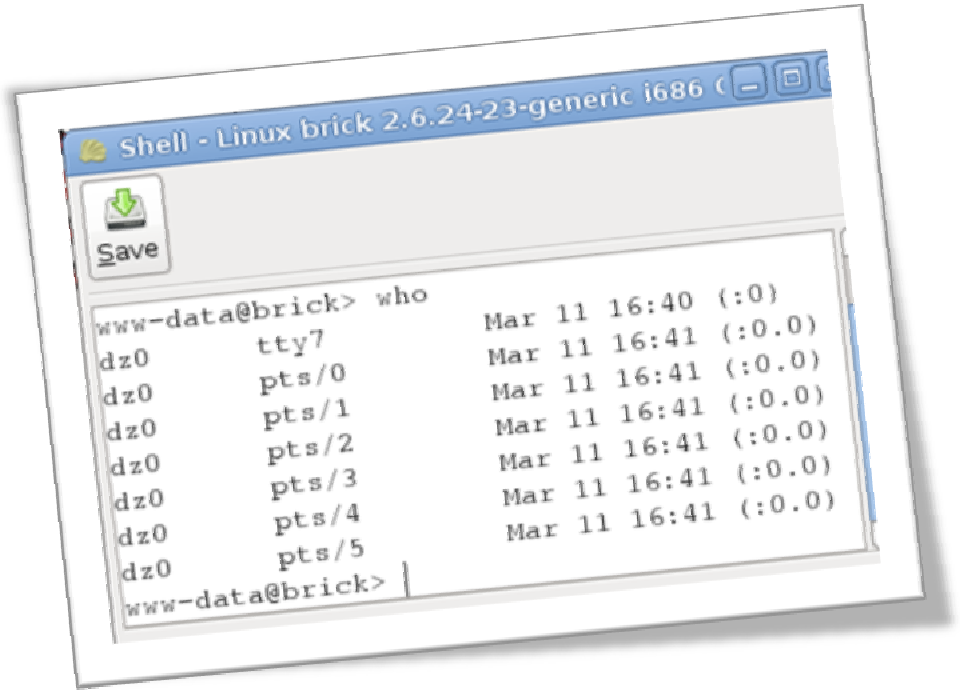
- As vulnerabilities are found, they are saved as **vuln objects** in the knowledge base.

- These vuln objects are then used as the input for attack plugins, that will exploit the vulnerabilities.



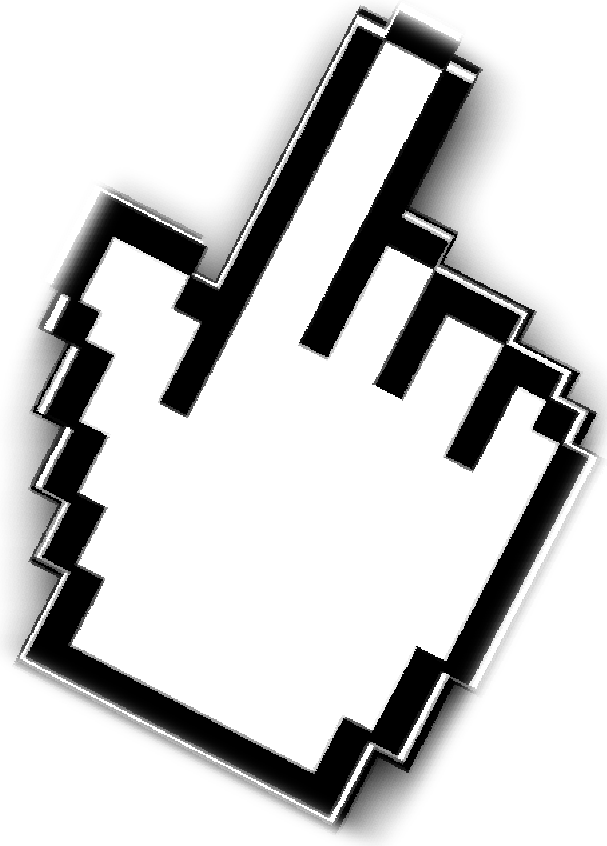
Plugins | Attack

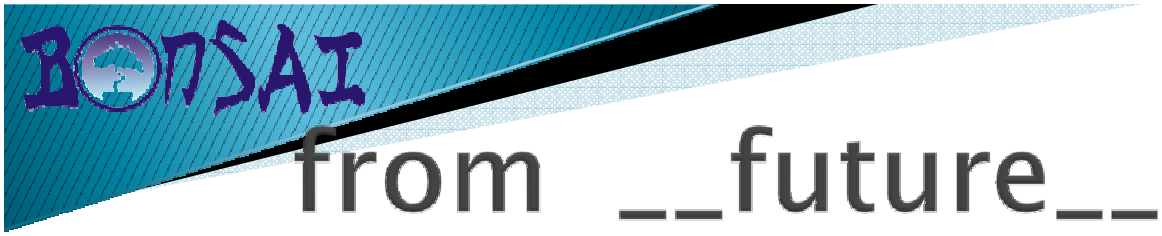
- These plugins read the **vuln objects from the KB** and try to exploit them. Examples of attack plugins are:
  - sql_webshell
  - davShell
  - sqlmap
  - xssBeef
  - remote file include shell
  - OS Commanding shell



Discover, audit and attack!

GUI Tools Demo





from __future__ import *

- **Live scan:**
  - User **browses** the website through w3af
  - w3af parses the requests, and sends them to **audit plugins** in order to find vulnerabilities.
  - The user can **view findings in real time**, while browsing the target website.
- Better management reporting
- **Enhance the MITM Proxy.**
- Releasing **1.0** in a few days (I'm saying this since April)

Conclusions

- w3af is a **growing project**, with many features and huge potential.
- Over the past year, **a lot of bugs were fixed**, making the project **waaaaaaaaaaaaaay more stable.**
- But we still have a lot of known bugs, *and performance enhancements*, that we're going to be working on.
- **New plugins are always welcome!**
- But **what I really want is…**

BONSAI

I WANT YOU!

How to Contribute?

- Project website

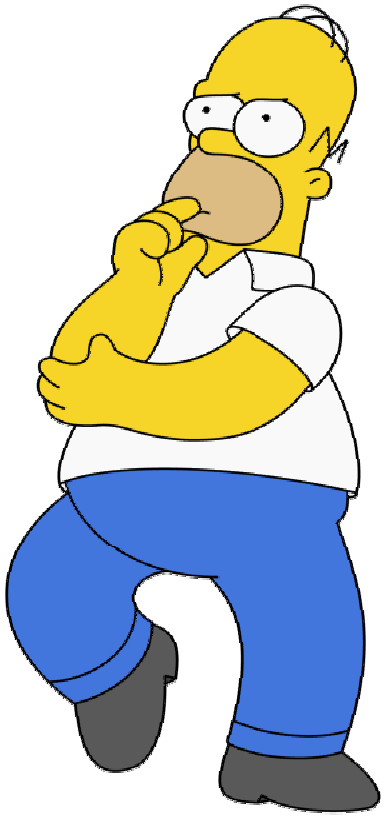
  http://w3af.sf.net/

- Two different mailing lists, **users** and **develop**.
- IRC channel , **#w3af at Freenode.**

- **Attend one of my w3af trainings! Next one is in December @ NYC.**

- If you deliver **Web App Sec Trainings, include this tool!**

¿Questions?



Suggested questions

- I tested it a while ago and **it crashed every time**, what has changed?
- Is **w3af a possible competitor** to AppScan, Cenzic, Acunetix?
- Is w3af going to be **Open Source for ever**?
- Is w3af included in the **Web Application Security training that you're going to deliver on Monday and Tuesday?**

w3af

http://w3af.sf.net/