# w3af – A framework to own the Web

Andrés Riancho
ariancho <at> cybsec.com

*Ekoparty 2007*
*Buenos Aires, Argentina*

# who am i ?

- Security Consultant at *Cybsec*
- Programmer
- Open source evangelist
- Web Application security enthusiast
- Background in networking, IPS design and evasion

# w3af

- w3af stands for **W**eb **A**pplication **A**ttack and **A**udit **F**ramework
- An Open Source project (GPLv2)
- A script that evolved into a serious project
- A vulnerability scanner
- An exploitation tool

# Main features

- Finds common and uncommon web application vulnerabilities.
- Cross platform ( written in python).
- Uses Tactical exploitation techniques to discover new URLs and vulnerabilities
- Web and console user interface

# Main features

- Web Service support
- Exploits [blind] SQL injections, OS commanding, remote file inclusions, local file inclusions, XSS, unsafe file uploads and more!
- WML Support (WAP)
- Really easy to extend
- **Synergy** among plugins

# Main features

- Ability to find vulnerabilities in query string, post data, URL filename ( http://a/f00_injectHere_b4r.do ), headers, file content (when uploading files with forms) and web services. JSON support is almost ready!
- Number of plugins: 115 and growing
- w3af is **smart**, more on this later ;)

# 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 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 and create the corresponding 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 fuzzable request.

Other discovery plugins try to fingerprint remote httpd, allowed HTTP methods, verify if the remote site has an HTTP load balancer installed, etc.

# Plugins | Audit

They take the output of *discovery* plugins and find vulnerabilities like:

- [blind] SQL injection
- XSS
- Buffer overflows
- Response splitting.

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

# Plugins | Grep

These plugins grep every HTTP request and response to try to find information.
Examples of *grep* plugins are:

- findComments
- passwordProfiling
- privateIP
- directoryIndexing
- getMails
- lang

# Plugins | Attack

These plugins read the *vuln objects* from the KB and try to exploit them. Examples of *attack* plugins are:

- mySqlWebShell
- davShell
- sqlmap
- xssBeef
- remote file include shell

# Plugins | Others

- *Output:* They write messages to the console, html or text file.
- *Mangle:* They modify requests and responses based on regexs.
- *Evasion:* They modify the requests to try evade IDS detection.
- *Bruteforce:* They bruteforce logins.

# Tactical Exploitation

What w3af does about tactical exploitation:

- vhost search in MSN
- search for mail address in Google,MSN and MIT PKS.
- password profiling
- halberd
- archive.org search
- search Google, MSN, Yahoo

# Discovery demo

This demo will show:

- fingerPKS, fingerMSN, fingerGoogle
- bruteforce using collected usernames, and dynamically generated passwords:
  - username
  - target site ( www.domain.com ; domain.com ; domain )
  - passwords generated by the password profiling plugin

# Discovery demo (contd.)

Let's rock...

# Virtual daemon

- Ever dreamed about using metasploit payloads to exploit web applications ? NOW you can do it !
- How it works:
  - I coded a metasploit plugin, that connects to a virtual daemon and sends the payload.
  - The virtual daemon is runned by a w3af attack plugin, it receives the payload and creates a tiny ELF / PE executable

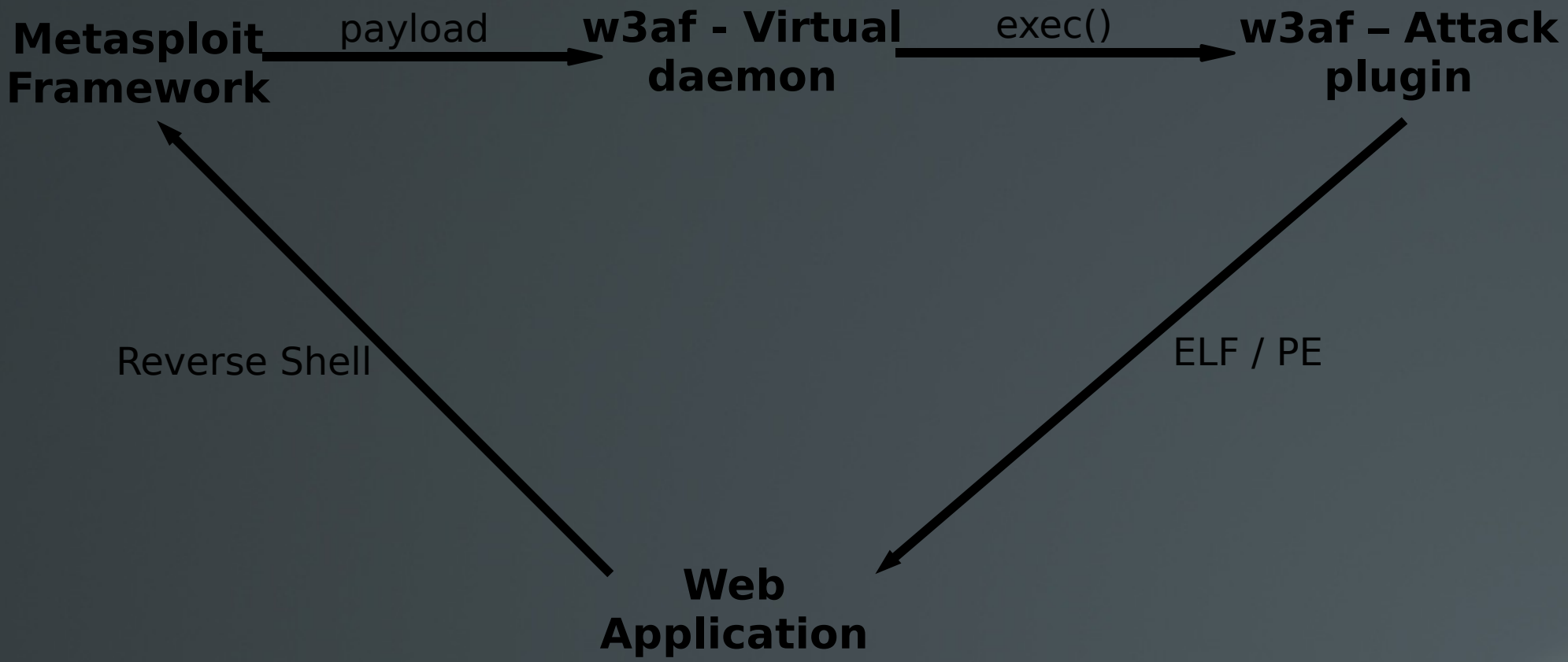
# Virtual daemon

- How this works (contd.):
  - The attack plugin knows how to exec remote commands, and the virtual daemon knows how to upload the ELF/PE using “echo” or some other inband method.
  - A new scheduled task is created to run the payload, and the metasploit plugin is ordered to wait
  - The payload is run on the remote server.

# Virtual daemon

- How this works (contd.):
  - Normal communication between metasploit and the exploited service follows.

# Virtual daemon



# Virtual daemon

demo!

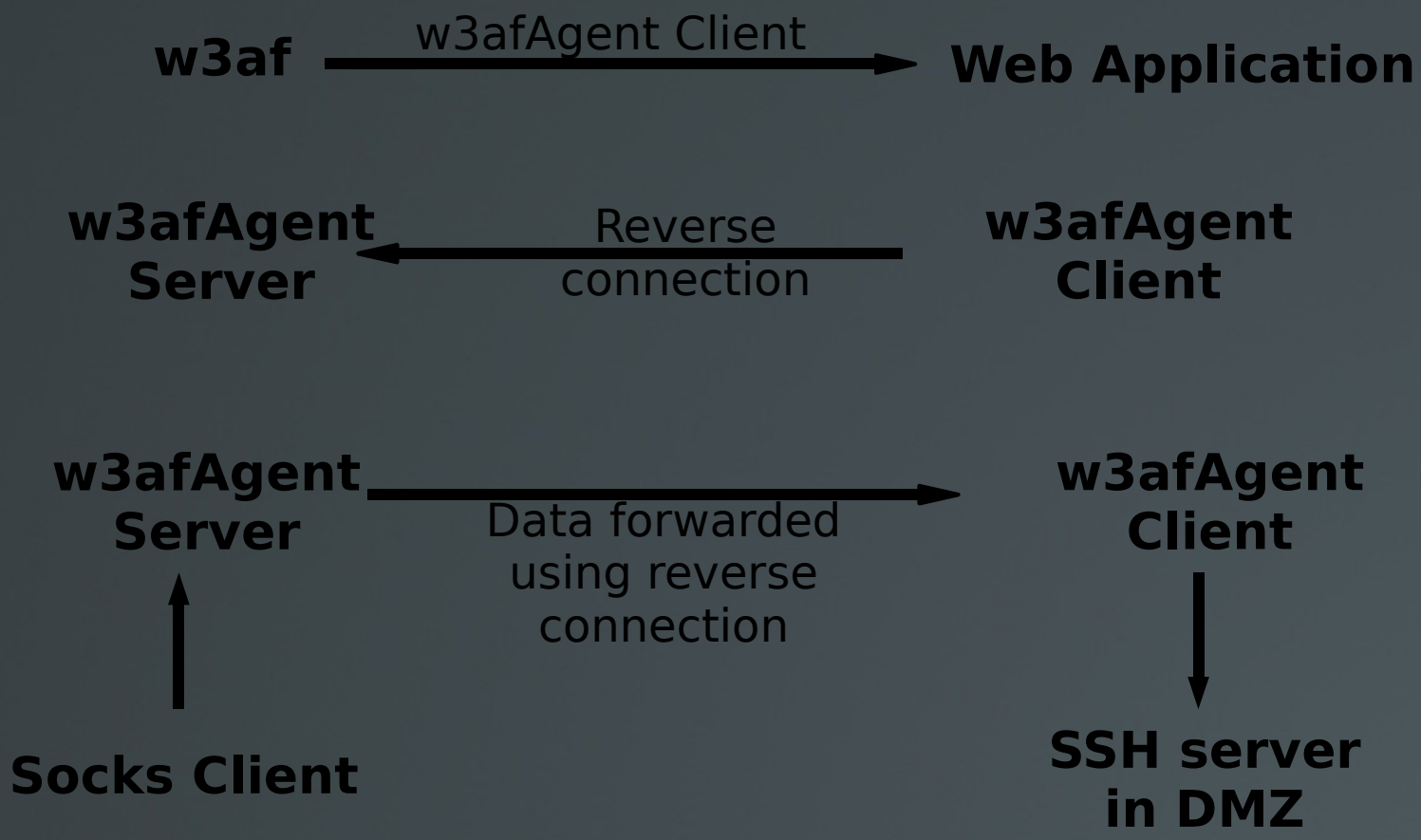
# w3afAgent

- A reverse “VPN” that allows you to continue intruding into the target network.
- How does it work?
  - I send the w3afAgent client to the target host using a transfer handler (wget, tftp, echo)
  - The client connects back to w3af, where the w3afAgent server runs a SOCKS daemon.

# w3afAgent

- How does it work? (contd.):
  - Now the user can use any program that supports a SOCKS proxy to route connections through the w3afAgent Server.
  - All the traffic is forwarded to the w3afAgent Client, where a new TCP connection is created.

# w3afAgent


w3af
w3afAgent Client
Web Application
w3afAgent Server
Reverse connection
w3afAgent Client
w3afAgent Server
Data forwarded using reverse connection
w3afAgent Client
Socks Client
SSH server in DMZ

# w3afAgent

- Things that don't work but could:
  - UDP traffic
- Things that won't work:
  - Raw sockets
  - Sniffing

# import __future__

- Javascript support
- More stable core
- More attack plugins, refactoring of attacks.
- Better webUI
- Better management report generation
- Long descriptions for vulnerabilities
- “Endless” discovery-audit-exploit loop

# import __future__

- Replace SOAPpy with ZSI
- And maybe...
  - Static code analysis of scripting languages (integration with Orizon? http://orizon.sf.net/ )
  - Apache / IIS log analysis

# Project information

- Site
  - http://w3af.sf.net/
- Mailing list and sourceforge home
  - http://sourceforge.net/projects/w3af/
- It's open source, ***you*** should contribute!
- Project leader contact
  - andres.riancho <at> gmail.com
  - ariancho <at> cybsec.com

# Project sponsor


- 11 years experience in information security
- Clients in LATAM, USA and Europe
- Based in Argentina
- Professional Objectivity
- Research friendly ;)

# Questions?

Feature requests ? ideas?Bug reports? contributionsRants about web2.0?i want flash support! Web Services hacking.