Hugo González

@hugo_glez

http://atit.upslp.edu.mx/~hugo/

10 años UNIVERSIDAD POLITÉCNICA DE SAN LUIS POTOSÍ

UNIVERSIDAD POLITÉCNICA DE SAN LUIS POTOSÍ

NWeSP

IEEE COMPUTER SOCIETY

International Conference on Next Generation Web Services Practices (NWeSP'07)

First approach to Desing a Web Service Honeypot, Wspot

Hugo Francisco González Robledo
Universidad Politécnica de San Luis Potosí
hugo.gonzalez@itslp.edu.mx

Who ?

- Professor in computer department on UPSLP.
- SCSA for Solaris 10
- LPI level 1
- Honeynet Project
- IEEE
- ACM
- Linux, security

Agenda

- Abstract
- Web Services
- Honeypots
- Security concepts
- Attacks for WS
- WSpot
- Conclusions

Abstract

- Web Servies technology steel growing around the world. One of the hotest topic about it it’s security.
- New technology, new kind of attacks, new kind of mitigation and responds. There are a lot of iniciatives about security in Web Services, all of them try to eliminate the risk, or at least, reduce it. WS-Security standards are all about how to reduce the vectors of attack, but at this time, there aren’t real data about how the attacks happend.
- The WSpot, it’s and old concept apply to new technology to track, discover, know and mesure the tools, tachtics and motivations of the attackersi

Web Services

- Web Services are new techonology for the machine to machine interaction, it permits communicaton between two or more end points. It uses strong standars and well know technology to permit the interaction. There are specific format that permit machines understand and process the information for the Web Service (the WSDL file). The other point uses this specification to interact trow SOAP messages over HTTP protocol using XML serialization

Honeypots

- A honeypot is a informatic resource, real o simulated, that values reside in the information about misuse that. The honeypots are resources used recently to discover new threat of attacks, it moves on the way of attackers, since servers, malware, clients and now web applications with some projects.
- The are several kinds of honeypot. Low interaction honeypot only emulate the services they are offering, there are no real service behind them.

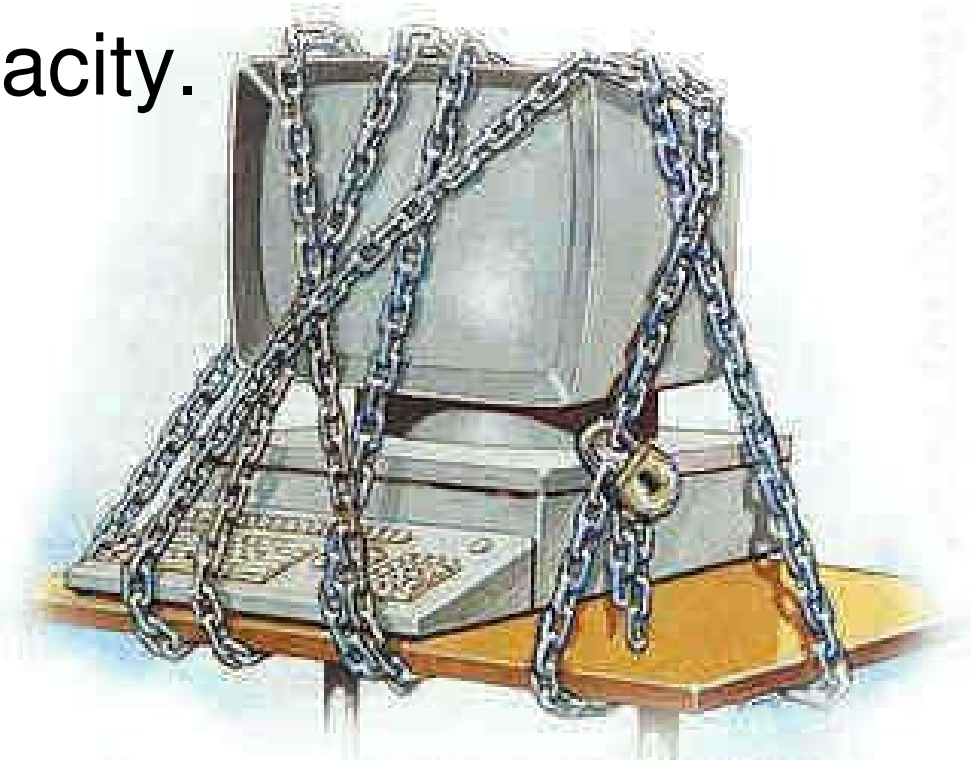
- For example honeyd, or kfsensors. This type of honeypot only can collect a limited amount of information, but they are simple to maintain and install, secure and require less resources.

- Wspot is inspired on honeyd

Honeynet
méxico Project

Security Concepts

- Identify, Autentication,
- Autorization, Auditory,
- Integrity, Confidentiality,
- Non repudiation and Privacity.



Attacks

There are four layers in wich an attack can occur:

1. **In transit**. The message go on wire, it’s not on the begin nor end of the communication.
2. **Engine.** Is part of the application server, it’s vendor controlled, not the user.
3. **Deployment.** It’s about configuration and installation, it’s user controlled.
4. **User code.** It’s the binaries, the code for the application. Sloppy coding practices open door for attackers. It’s user controlled. A lot of attacks usethis layer.

Attacks on layer 4

- Parameter tampering
- WSDL probing
- SQL/LDAP/XPATH/OS command injection
- Virus/Spyware/Malware injection
- Bruteforce
- Data type mismatch
- Content spoofing
- Session tampering
- Format string
- Information leakage
- Authorization

WSpot

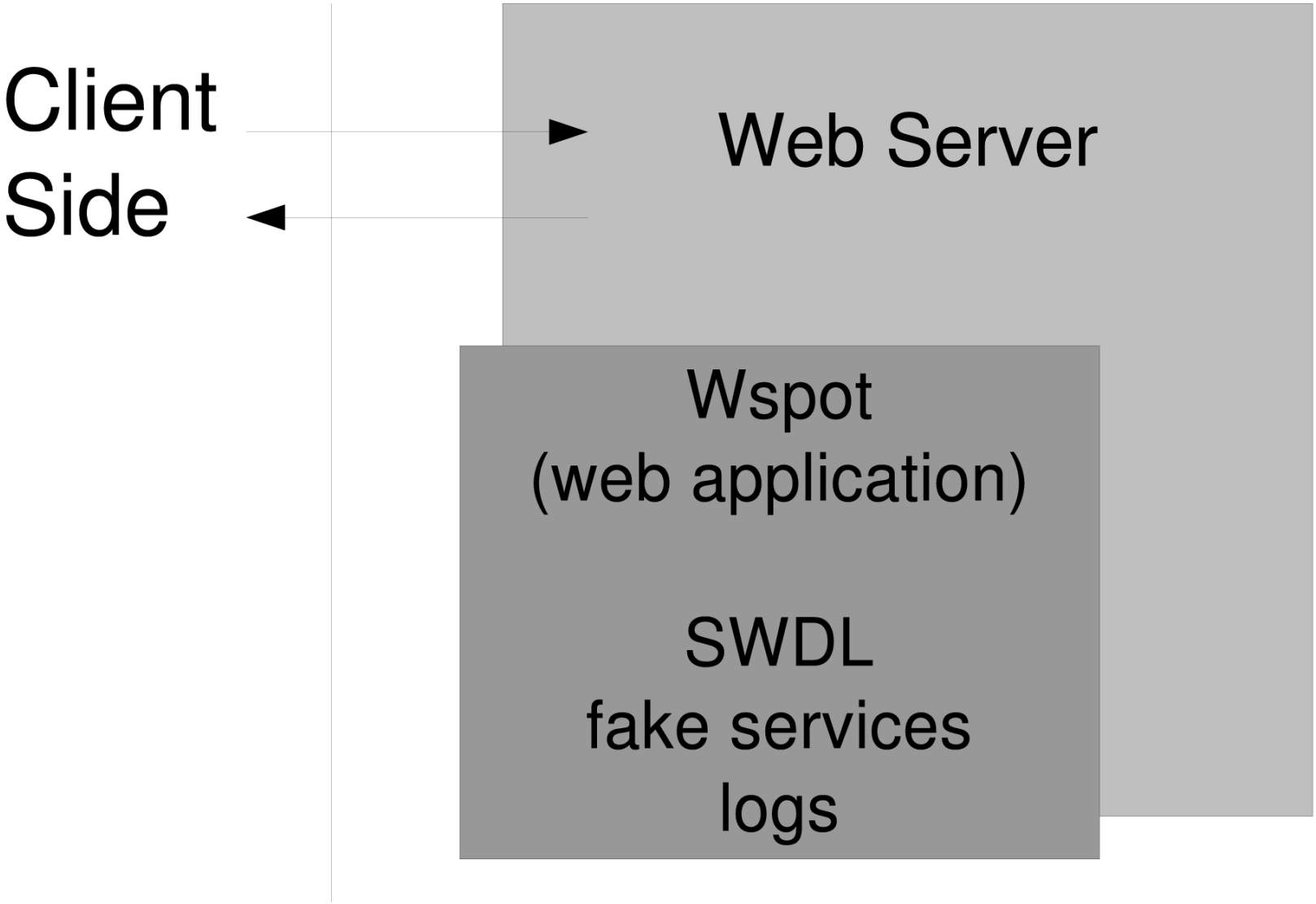
- Objetives:

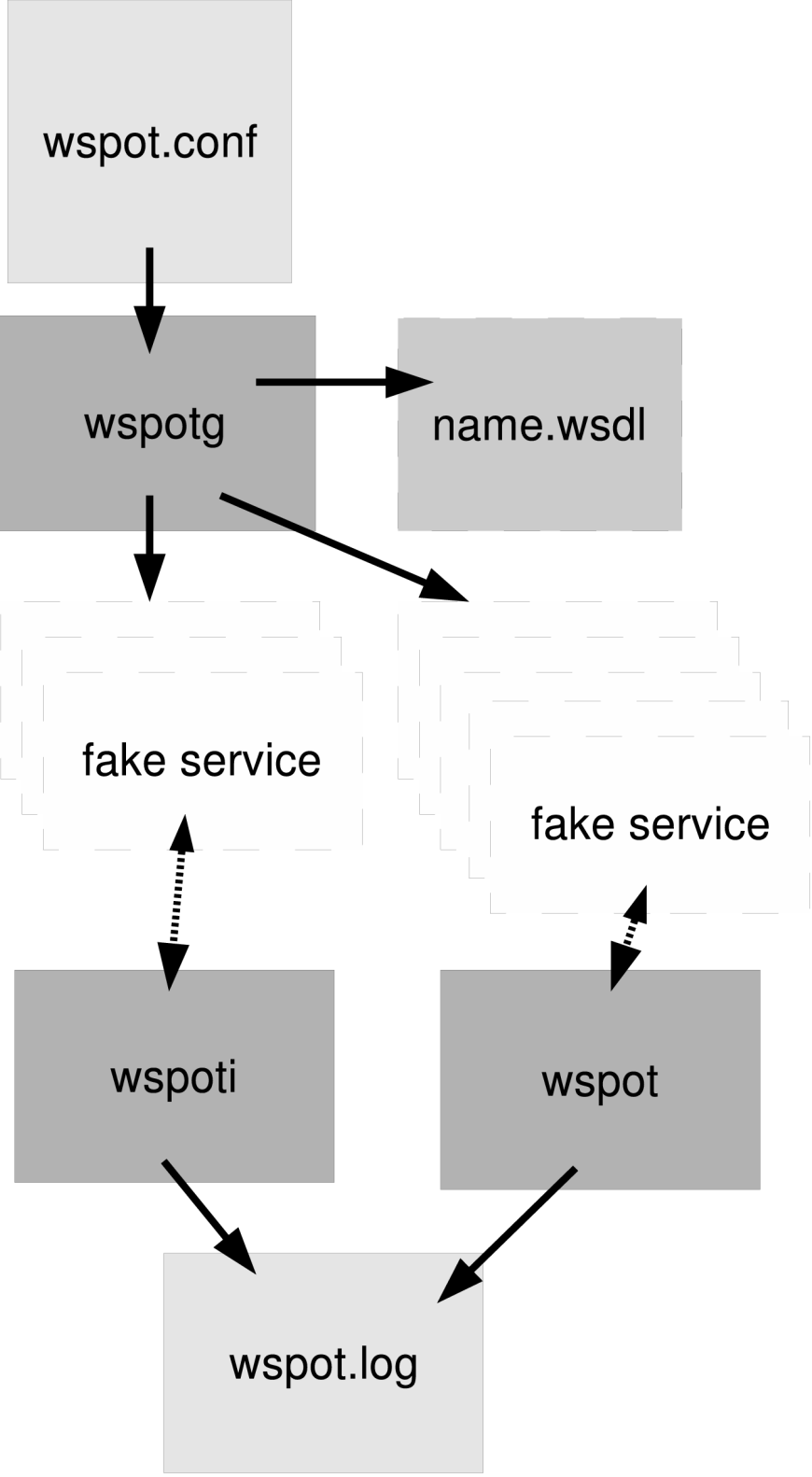
a. Emulates a SOAP based web service,

b. Log and register all the activites on it.

c. Offers a complete and easy to understand log.

- With this aproach we could have a lot of information on attacks to the web service at the layer four and have an opportunity to discover new types of attacks or new tools used to compromise web services, even we could demostrate that actually web services are targest for the black hats.


Client
Side
Web Server
Wspot
(web application)
SWDL
fake services
logs



There are five main components for the wspot:

a. wspot.conf file: it contains the values for the configuration variables used by deploy, the name of the fake web service, the names of the functions and type of parameters are included too. This is the base for the wspot function.

b. wspotg: this program use the wspot.conf file to generate the “webservicename.wsdl”, and create the soft links for each function defined in the config file. This links points to the wspot and wspoti program.

c. wspot: this program makes the interaction for the general functions, the results of this are always errors about not verified user. It logs all the information recived to the log file. It would detect parameter tampering, type mismatch, session tampering and so.

d. wspoti: this program is similar of wspot, but it works like identification module, the parameters are almost user and password, so it logs the information to log file too. The result are always error about username and password. It would detect enumeration, bruteforcing, session tampering and so.

e. wspot.log file: it records all the information gathered by the programs.

Current and future work!

- This WSpot only emules the servcies, the next approach will be on a real architecture of web services, a limited function an extended description for it, and logs all the things happen.
- But the most importan work will be deploy Wspot to the real world.
- In the first step it will be inside a honeynet, to prevent and monitor all the behaivor of WSpot. Then it will deployed without any extra infraestructure.

Conclusions

- Here has been shown the first approach of desing for a web service honeypot called WSpot. With a simple and clear design it is able to record all the activities on it and it’s possible to face different on every deployment. Through the honeypot services, and the data collected, we are able to learn in advance for the new faces of attackers, new tools or methods used against web services. It’s important to mesure the attacks, for develop convenient methodologies and tools to deal with the insecurity, and focus on the critical four layer

Thanks to

- NWeSP’07 – Organization
- Shreeraj Shah for technical review
- UPSLP for supporting the trip