

Shadow Software Attacks

Angelo P.E. Rosiello
e-mail: angelo@rosiello.org

What is it?

- “Shadow software attacks” are conceptually similar to Shadow Server Attacks.
- At the moment these kind of attacks do not depend on the technological context and can always be applied.
- Even if they are well known in the literature (Rosiello, 2004 – A.Lioy 2001) , and mainly the shadow server attacks, people seem to ignore them...

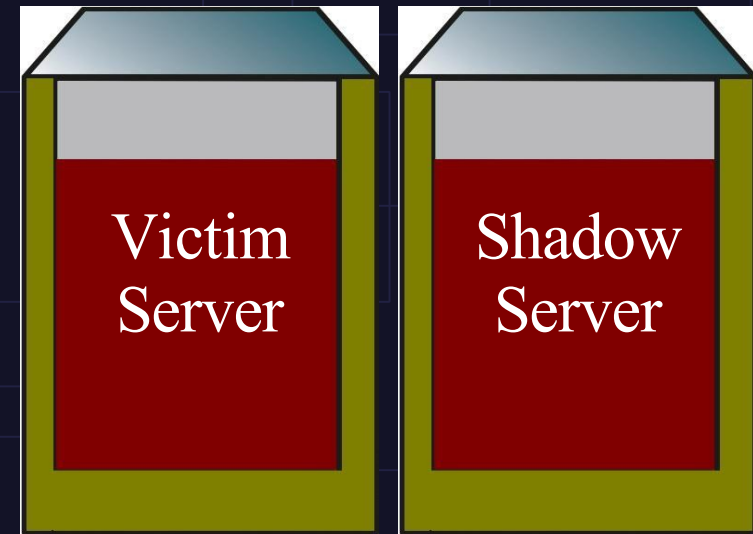
Once upon a time...

- Before facing shadow software attacks we will remind shadow server attacks, introducing some interesting correlations with relative recent phenomena, such as phishing.

Shadow Server Attacks

Definitions

- A shadow server is a copy of a real server.
- Apparently, the shadow server is indistinguishable from the victim server.
- By definition, any service running on the victim server must be available on the shadow one.



A Practical Example

Victim Server

- Operating System: Linux Fedora Core 4
- Web Server: Apache
- FTPD: extreme-FTPd
- etc...

Shadow Server

- Operating System: Linux Fedora Core 4
- Web Server: Apache
- FTPD: extreme-FTPd
- etc...

Are They Really Equals?

- Actually, “copying” a server doesn't mean substituting it.
- Even with more advanced attacking techniques, such as ip spoofing, dns spoofing, we can still distinguish the ordinary server from the shadow one...
- This is an easy task for the administrator but not for normal users.

SSL is Our Friend

- Fortunately SSL let us identify our server.
- An SSL Certificate lets users know that they are really interacting with the true server and that the information they send through the site, such as credit card numbers, online forms, and financial data, is protected from interception or alteration over the Web.

SSL: How Does it Work?

- User contacts the site and accesses a secured URL: a page secured by a Server ID.
- The server responds, automatically sending site's digital certificate, which authenticates the site.
- User's Web browser verifies server's digital certificate and generates a unique "session key" to encrypt all communications with the site.

SSL: How Does it Work?

- User's browser encrypts the session key with site's public key so only the true site can read the session key.
- A secure session is now established and all communications will be encrypted.

What About Other Services?

- In our example the victim server was running also other services, such as an FTP daemon.
- How to secure other services?
- SSL can be run over all other existing application protocols (not only HTTP), avoiding the problem.

The Problem Doesn't Exist!?

- As speaking, most of the servers connected to Internet run services without using secure protocols.
- Usually, users do not pretend the authentication of the server, thus, the exchange of information begins trusting the look-and-feel of the server.
- This is very dangerous since we don't know if the server we are connected to is the real one.

The Problem Does Exist!

- Why?
 - People is lazy.
 - SSL requires a digital certificate.
 - Sometimes it reduces performances (think about a slow gprs connection...)
 - etc.

Phishing (continue...)

- Nowadays phishing has become a popular topic, because of media attention.
- Phishing is a form of online identity theft that aims to steal sensitive information from users such as online banking passwords and credit card information.
- Phishing attacks use a combination of social engineering and technical spoofing techniques to persuade users into giving away sensitive information (e.g. using a web form on a spoofed web page).

Phishing (continue...)

- Now, this kind of attacks sounds pretty familiar, in fact, it is an enhanced shadow server attack!
- Phishing could require faking servers and services (i.e. Shadow server&software attacks) but also an interactive social engineering effort.
- Gartner (Ollman, G. 2004) says that 57 million US Internet users have identified the receipt of e-mail linked to phishing scams and about 2 million of them are estimated to have been tricked into giving away sensitive information!

Shadow Software Attack

Definition

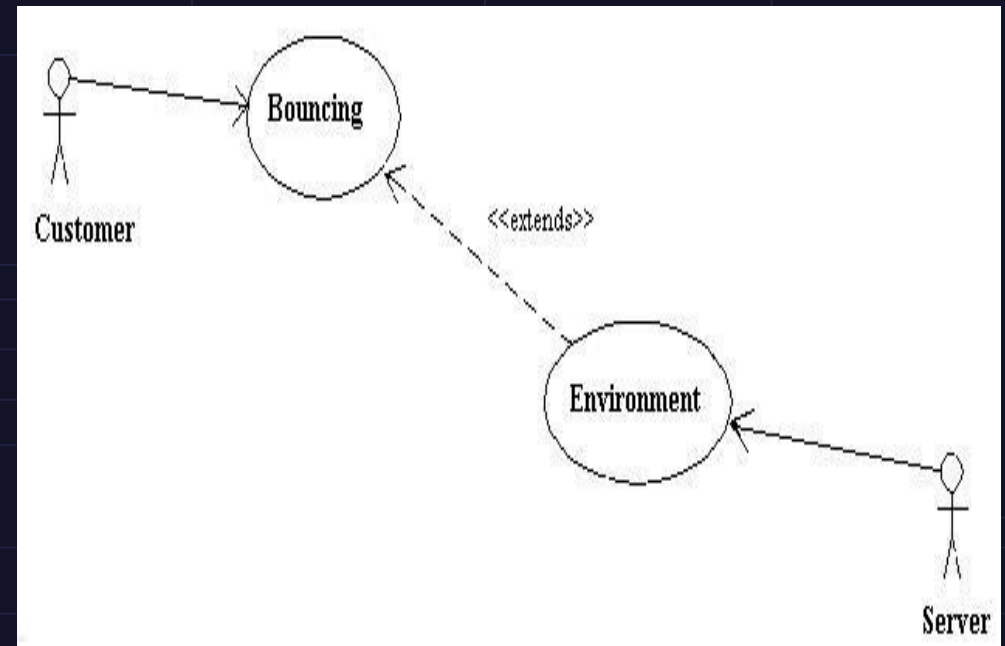
- Shadow software attack can be considered a particular instance of a shadow server attack, where the focus is a well defined software.

A Common Scenario (continue...)

- The chosen actors of the contest are a shell provider, a customer of the shell provider (victim) and the attacker.
- The goal of the customer is to use his account on the provider to launch a bouncer (it could be a proxy for the web, ftp, irc and so on but this attack can be brought to any similar software), for this purpose he is paying the provider.

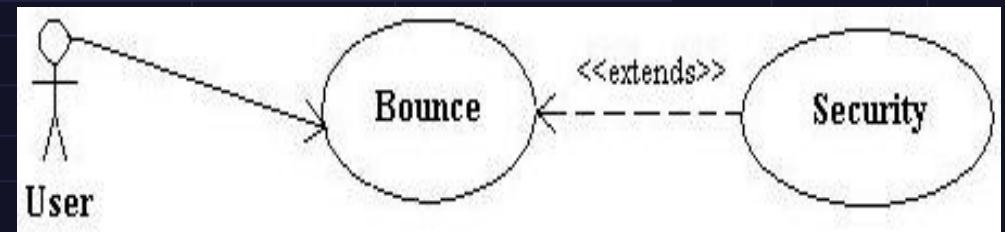
A Common Scenario (continue...)

- The server must provide the environment for the customer to launch his application and this is just a working shell.



A Common Scenario (continue...)

- Now, our user will download, compile, configure and launch his favourite software by his shell account.
- The software that the user is going to download must provide the bounce and *optionally* a secure way of doing it.



A Common Scenario (continue...)

- As shown previously, typically users don't require security or they intend it as an “embedded” property of the provider (since they pay!).
- A careful user should instead require that one of the goals of the bouncer must be to use a <<included>> step of the security.

Possible Attacks

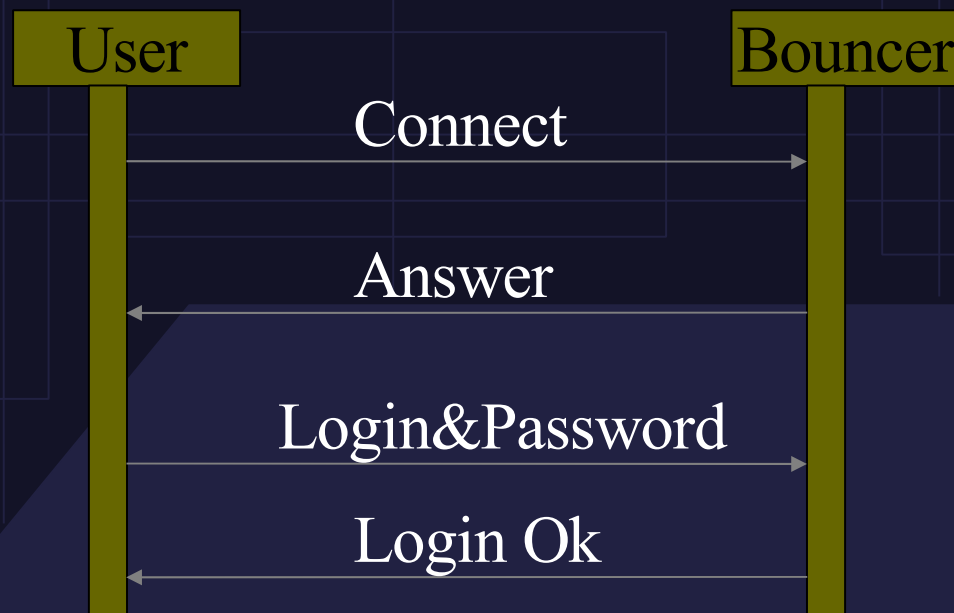
- Now, that we have a background of the situation, let's try to think about a way that could be applied to the previous scenario, in order to steal user's information:
 - Hack the server
 - Shadow server attack
 - Social engineering, phishing
 - etc..

Shadow Software Attack

- What we're going to do is to simulate the software of the victim (in this case a bouncer), with a shadow software, in order to steal his logging data, then his account and password.
- **This pattern does not need any privilege escalation or other requirements.**

Analysis

- A bouncer is a program that listens on a port and requires the authentication of the user to offer him an access.
- The following scenario shows the interactions between the user and his real bouncer.



Attack Requirements (continue...)

- To have success the interaction with the user must be perfect.
- Attacker requirements:
 - Information about the software to be simulated
 - The number of the listening port
 - An account on the same machine of the victim

Attack Requirements

- The previous three requirements can be easily gained, in fact, to observe the software it's enough a connection; to know the listening port a port scan will be useful, or a social engineering solution, and some money will give us an account on the same machine of the victim.

Attack Formulation (continue...)

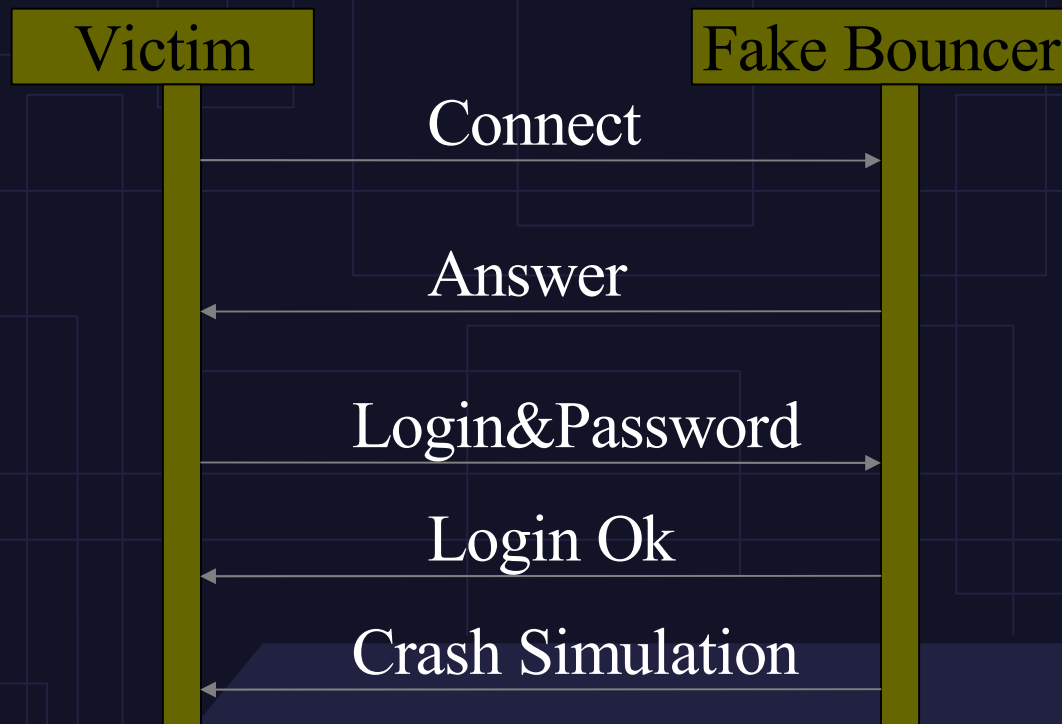
- Let the attacker satisfied all the points discussed above. He can't still launch the fake software because the port is still bound to the victim's bouncer!
- This “hold up” could be resolved in many ways such as crashing the server or the real bouncer or just waiting for a reboot of the machine.

Attack Formulation

- When the victim's software is down (because it was crashed, or the machine rebooted) the victim will not detect it until he won't be able to use it (why should he gets into the shell account?), in this case he will join the shell account to restart it, but this event will not happen because a good attacker will start the shadow bouncer before the user could notice that his bouncer went down.
- At last the attacker could launch the shadow software that will simulate perfectly the bouncer of the victim.

Stealing Victims' Information

- Let's have a look at the sequence Diagram...



Definitely, the user lost his account and password that are in the hands of the attacker.

A Proof of Concept

- An example of shadow software that simulates the psyBNC can be downloaded from <http://www.rosiello.org/archivio/fakepsy.c>
- This kind of attack is really applicable as speaking...

Solutions

Effective Solutions (continue...)

- There are many solutions to avoid a shadow software attack.
- Probably, as discussed at the beginning of the presentation, the better one is to use SSL/TLS, in this way we get an authentication of the server/software we're going to connect to.
- Another possible solution is to use a challenge-response authentication (when authentication is required).

Effective Solutions

- These kind of solutions, and many others, need a “cooperation” among the server admins, softwares developers and also users of the providers.
- It is evident that few users will use a bouncer running SSL....

A Possible Alternative Solution

- We thought about an alternative solution, in order to avoid the shadow software attack, previously described.
- The idea is to assign a range of ports for every user, managing the `bind()` `SYS_call`, this is completely transparent to users.
- Remind that this can be seen as the very reason because the range of ports between 0 and 1024 is reserved to super-users.

Conclusions

Conclusions

- Shadow software attack does not exploit a bug on the server or client but rather the unsecure way to interact with applications, thus, it's important to adopt automatic security systems to detect and prevent it.
- The solution that should be applied depends on the contest, so if you are a system admin think about it, seriously.

Questions&Answers

Thanks for your attention....

Angelo P.E. Rosiello