Attack & Defend

Computer Security

Andrew Bellenir, Joshua Hulst, Dr. Mostafa El-Said (advising)

The Scenario

Budget cuts were made at Initech this quarter and it turns out that two sysadmins will have to be let go. There's a good chance these guys that are getting fired are up to something. Management already knows that they modified some bank software to dump a bunch of credit card numbers into a file they've created. The software can be fixed, but it's going to take time. They need to finish out the week then be locked out of the system until the software can be patched.

Our Role

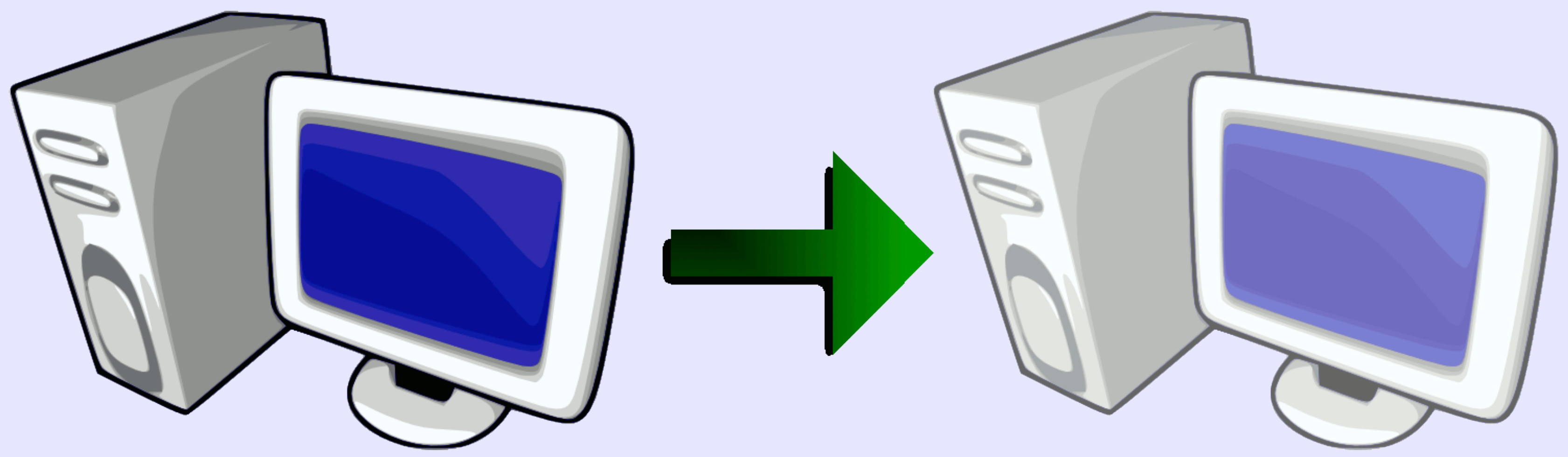
Play out the role of both the disgruntled employee and the defending system administrator. First, attack: given access to an opponent's server, set up several exploits and cover tracks to hide what was done. Second, defend: given a system that an opponent has compromised, try to find and remove all vulnerabilities Finally, attack again: try to remotely hack into the server that was attacked earlier with the goal of stealing a "secret" file that is known to exist.

Attack

Chroot Jail:

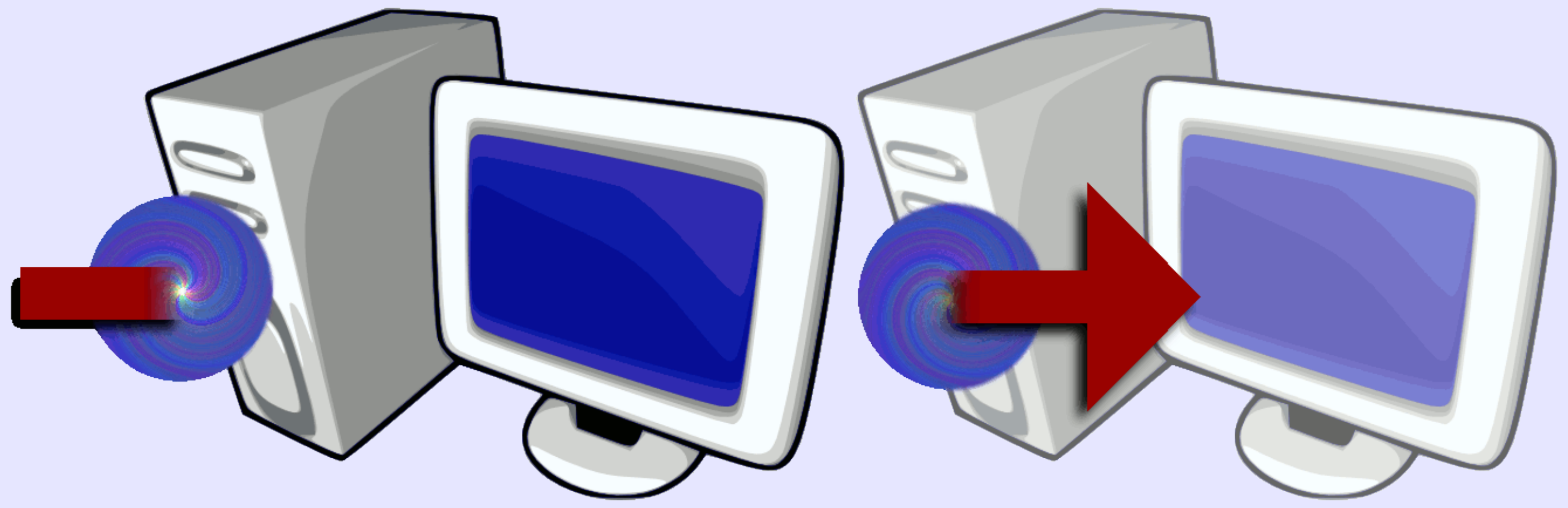
Step 1: Build the Jail

Install a second copy of the operating system on the victim machine and copy user data into it. The new copy will serve as the jail, and will be identical to the original to prevent anyone from noticing the difference. Use links to keep the real operating system and the jail synchronized (password files, running processes, etc.).



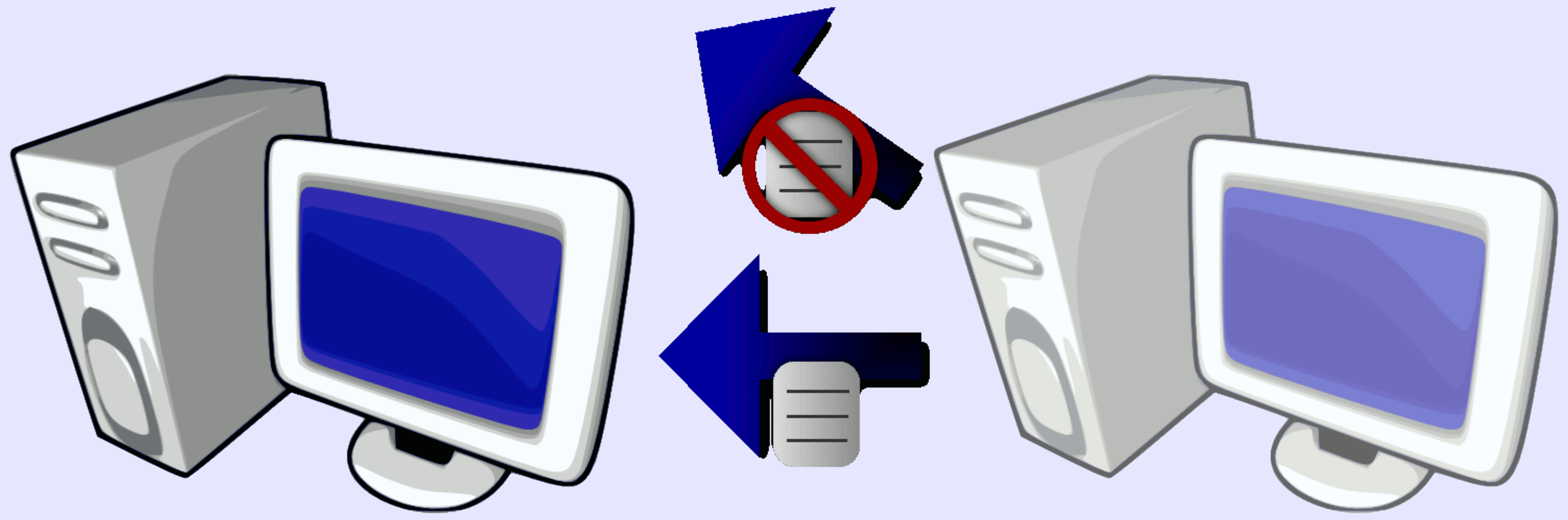
Step 2: Lock Users In

When logging in, all users will be thrown into the jail created in step 1. This can be done by modifying the actual system's log in scripts to include the “chroot” command. While jailed, users will not have access to anything outside the jail, including detection of any changes made to the real system. Traditional defensive measures, such as running antivirus, scanning for root kits, and viewing log files will not be able to detect anything is not as it should be.



Step 3: Set Yourself Free

Most users, when they exit the jail, will be logged out of the system as well. An attacker which has created the jail will want to leave the jail, but stay in the real system. This can be accomplished by having a guard standing at the jail's gate, programmed to recognize authorized users. In the simplest implementations guard software may recognize users either by a “white list” containing the names of authorized users or existence of a token the user created while in jail.



Exploits:

Once the chroot jail is in place other exploits need to be introduced to the victim computer to allow future access. These exploits may range from childishly simple to very complex. Here are some examples...

New user accounts: modify system scripts so that a new user is created with various events, such as server booting up, administrator log in, administrator log out, or restarting some services.

Upgraded user permissions: modify security files particular users (or any users) are allowed to perform any desired action

ICMP backdoor: install a malicious program which waits for specially crafted network messages, which it then interprets to execute any command a remote user wishes

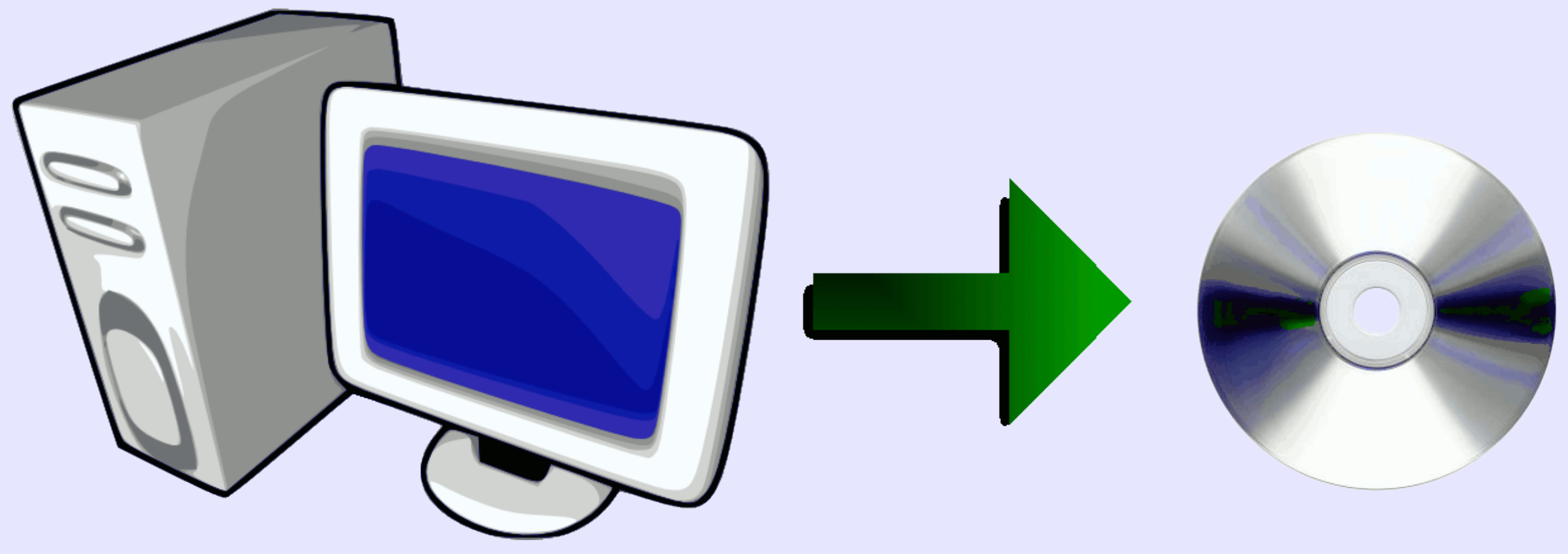
Root kits: install software packages that replace selected built-in applications with alternate versions which have unexpected side effects

Defend

System Restoration:

Step 1: Make Backups

A good system administrator makes regular backups of all essential data and configuration files. If the time of attack can be determined, backups before that time may be used to provide configuration files which are known to not be corrupt.

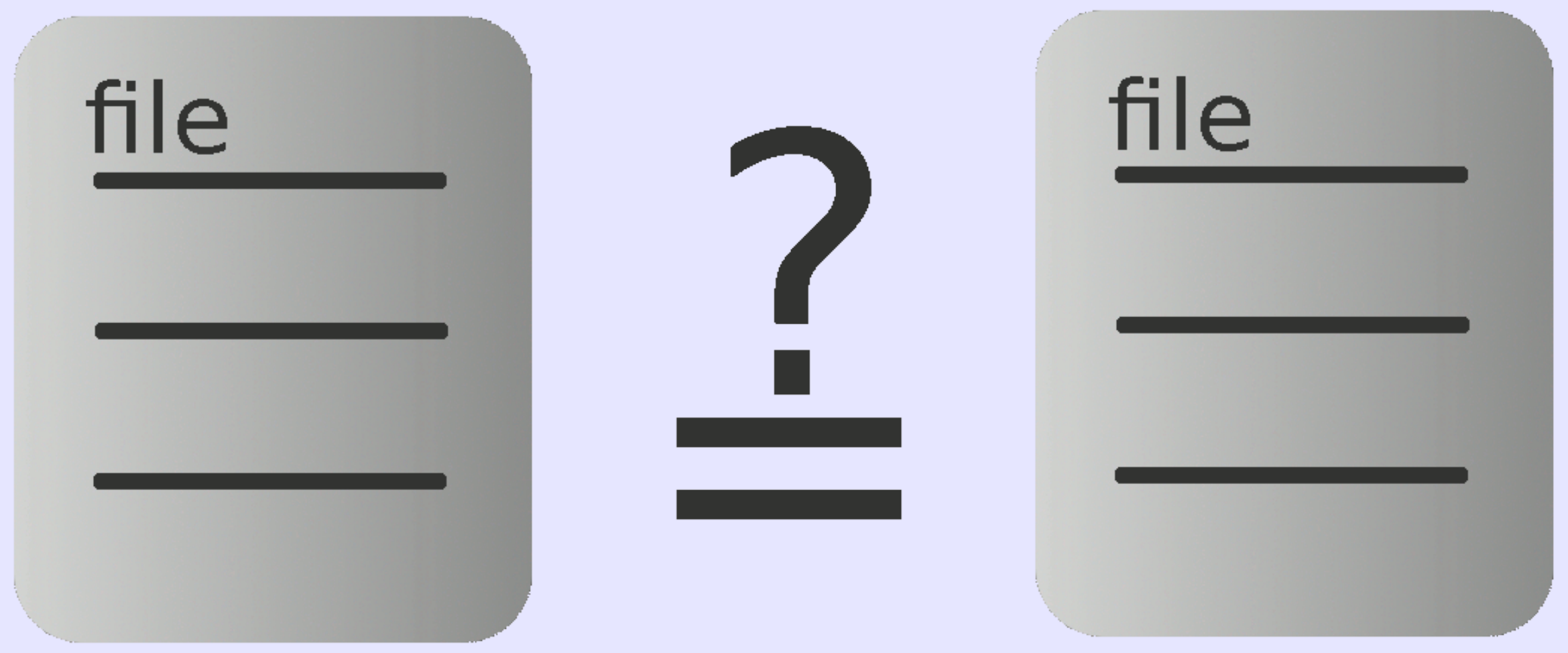


Step 2: Get a Clean Environment

Some exploits cause a compromised server to not behave as expected. Common utilities may be overwritten or the environment itself might provide misleading information (like with the chroot jail). The first step is to start with something that is known to be clean. The simplest way to do this is to use a live CD (a bootable CD with a complete working operating system installed on it).

Step 3: Restore Files

First, determine which files on the hacked computer are different from the files in the backups. One tool that is very good at this is rsync. Given a target directory and a source to compare it to, it can detect if any files have been added, removed, changed in any way, or even just viewed. It is capable of restoring the target directory completely, or just reporting what the differences are. This works very well for finding any changes that have been made to a hacked server and provides valuable information on how to correct any problems.



Alternatives:

For the scope of this project, not all utilities were available for use. In a corporate environment, however, additional precautions may be taken, which would likely prove useful:

tripwire: this application is used to detect any changes made to a computer since installation. it may be configured to inform administrators to a wide range of changes or suspicious user activities.

active defense: use tricks such as the chroot jail in a defensive manner by denying suspicious or unauthorized users access to the real system

firewalls: software tools that watch network traffic to monitor for and block any malicious activity

data encryption: encrypt all sensitive data so it can only be viewed by those who know the key