Host-based responses to detected misuse

Christopher Wee

wee@cs.ucdavis.edu

23 April 1998

keywords: host-based response, misuse, intrusion detection

1. Introduction

Some host-based responses not considered in this report, ... yet. (suggested by Jim Hoagland)

- Increase monitoring,
- alert administrator,
- transfer to jail system,
- replace sensitive files with look-alikes.

2. Suspending a user and his privileges

During the course of an investigation, it may be necessary to suspend the accounts of a user suspected of misuse, i.e., the misfeasor. We want to prevent the misfeasor from accessing the system and interfering with the investigation by removing evidence, continuing the misfeasance, creating cover stories (i.e., falsified evidence that diverts responsibility to a different user) or initiating a new attack.

In this report, we describe how to temporarily suspend and then restore the access privileges of a user. We consider both NT and Unix.

2.1 What do we mean by “suspend”?

The most ovious effect of suspending a user account is to prevent interactive logins. Though the user cannot login, he may have processes already executing or tasks that are scheduled to start automatically at a later time. Many activities may occur on the system even after the user account is suspended that are accountable to the user. The user may continue to receive e-mail, which is deposited into his mail file. He may also access the web server and the FTP server anonymously.

There are different aspects that may be considered part of the definition “to suspend”.

- No more interactive logins. The user may be allowed to copy his files with a special restricted copy shell.
- Existing processes are suspended, i.e., marked as non-runnable by the scheduler.
- Existing processes are terminated
- E-mail is held in an escrow account, but not delivered to the user’s mail file.
- E-mail is marked as “non-deliverable” and returned to the sender,

Suspending an account is equivalent to deleting an account for a temporary period.

2.2 Current processes

We suspend or terminate the processes that belong to the user. If the operating system does not support process suspension (some don’t), we dump an image of the process to disk to preserve any incriminating evidence, The `ps` command reports which processes belong to the user. The `kill` command terminates the user's processes either gracefully or forcefully. The `top` tool displays processes (like `ps`) and can kill them (like `kill`) with proper privileges.

The Windows NT task manager does not identify the accountable user of the process. NT was conceived as a single-user, multi-tasking operating system, so all processes were assumed to be accountable to the user currently logged in. When we terminate the user’s desktop session (i.e., log out), all his processes are terminated. (Why doesn’t it require user authentication to log out?)

Neither Unix nor NT permit a user’s processes to be restored. NT stores the positions of open Explorer windows and restores those when the user next logs in. Solaris CDE can save the names of programs and the positions of windows which are restored when the user next logs in. However, the state of application programs are not stored. Some applications may record the

2.3 Suspending login in Unix

There are several pre-requisites for a user to successfully log into a Unix system. The user must have a valid account with a valid shell. The shell executable must exist and be executable. The daemon processes that offers interactive login must also be executing, often inetd. A home account is not necessary to login, most shells default to ‘/’.

Removing any of these requirements will prevent the user from logging in, but terminating the daemons or removing the shell executable prevents other users from logging in as well. Instead, the user’s account name-password entry in `/etc/passwd` or `/etc/shadow` is invalidated so the user cannot successfully authenticate himself. To invalide the password, replace the encrypted password with an invalid token, ‘“*” or “NP”. We also replace the shell (e.g., `/bin/csh`) with `/bin/false`.

If the user attempts to log in, he will be unable to authenticate himself because ‘*’ does not decrypt to any password string. If the user attempts a remote login (rexec, rlogin, or rsh) and use transitive trust between hosts (i.e., `/etc/hosts.equiv` or .rhosts file), the `/bin/false` shell replacement will terminate.

The last attempt may be activity that is attributable to the user, so the overall objective -- no user attributable activity, may not be met. The alternative may be to suspend all login daemons, affecting other users as well.

2.4 Suspending logins in Windows NT

To disable logins in Windows NT, use the User Manager (or User Manager for Domains) tool. Select the user and check the “Account Disabled” box. The next time the user attempts to log in, the message “Your account has been disabled. Please see your system administrator” is displayed. However, the user’s processes that are already running are left running.

Disabled accounts are used as templates for new accounts or to suspend user’s access. The built-in Administrator account cannot be disabled. To restore the user’s account, the “Account Disabled” box is unchecked.

2.5 Preventing FTP or WWW access in Unix

Since the FTP or web server may be configured to permit anonymous access, it will be impossible to discriminate between the suspended user and other users. If the user is guilty, he may use such anonymous access to cover his tracks, add data to the system to explain or cover his previous misfeasance.

2.6 Existing FTP sessions in Windows NT Internet Information Server

The NT IIS can terminate existing FTP connections. From the HTML-based administration screen, select FTP, then select “Current Sessions”. A list of existing sessions are displayed. Select those belonging to the user and click on the close button. That terminates the users sessions immediately.

2.7 Automatic jobs

The cron(1) facility allows the user to program tasks that are executed on the user's behalf at specific times. The at(1) facility also allows the user to delay the execution of commands. These automatic jobs must also be examined and possibly suspended. Using cron, the user may place a time-bomb script that removes incriminating evidence if it notices that the user has not logged in for a certain number of days. However, the cron tasks may perform tasks that are beneficial to the system and its other users. In that event, the sysadmin may consider transferring these tasks to a different user account with just the appropriate privileges to complete the job and not interfere with the investigation.

We can suspend a user's scheduled jobs by placing his username into the cron.deny and at.deny files.

2.8 Incoming connections

The user may still receive e-mail directed to his account. Using a .forward file, the user's e-mail may trigger some activity on the system. E.g., a particular subject line may delete some files. To suspend even the reception of e-mail on the affected system, the e-mail may be discarded, returned as non-deliverable or simply held in a separate account (use the alias feature).

2.9 Discussion

We propose a system design that easily controls user activities with full accountabilty. All computation must be accountable to a user. A user is a high-level semantic abstraction.

Server programs that perform tasks on behalf on clients must be programmed with full recognition of the accountable user.

The processor requires a user-token for each quanta of computation. The user account manager creates these tokens from validated user credentials. User tokens must not be forgeable, neither are user credentials.

In Unix, the setuid and seteuid system calls create new user credentials.

To suspend a user's access:

- We need to revoke existing credentials. Without user credentials, no activity can be accounted to the user.
- We need to suspend the ability to make new credentials.

To restore a user's access:

- We restore the ability to make new credentials.

3. Taking a snap-shot of user files

3.1 Capturing ssh keys

4. Stop all outgoing network connections from a host

From a particular user

5. Harden a system

5.1 compile /etc into hardwired code

6. Trace network connections

6.1 What is the policy for a spam site?

7. Conclusions

8. Acknowledgements

Thanks to James Hoagland for helpful comments on this paper.