Event Triggered Execution: Unix Shell Configuration Modification, Sub-technique T1546.004 - Enterprise

Archived: 2026-04-05 13:01:57 UTC

Adversaries may establish persistence through executing malicious commands triggered by a user's shell. User Unix Shells execute several configuration scripts at different points throughout the session based on events. For example, when a user opens a command-line interface or remotely logs in (such as via SSH) a login shell is initiated. The login shell executes scripts from the system (`/etc`) and the user's home directory (`~/`) to configure the environment. All login shells on a system use /etc/profile when initiated. These configuration scripts run at the permission level of their directory and are often used to set environment variables, create aliases, and customize the user's environment. When the shell exits or terminates, additional shell scripts are executed to ensure the shell exits appropriately.

Adversaries may attempt to establish persistence by inserting commands into scripts automatically executed by shells. Using bash as an example, the default shell for most GNU/Linux systems, adversaries may add commands that launch malicious binaries into the `/etc/profile` and `/etc/profile.d` files.[1][2] These files typically require root permissions to modify and are executed each time any shell on a system launches. For user level permissions, adversaries can insert malicious commands into `~/.bash_profile` , `~/.bash_login` , or `~/.profile` which are sourced when a user opens a command-line interface or connects remotely.[3][4] Since the system only executes the first existing file in the listed order, adversaries have used `~/.bash_profile` to ensure execution. Adversaries have also leveraged the `~/.bashrc` file which is additionally executed if the connection is established remotely or an additional interactive shell is opened, such as a new tab in the command-line interface. [5][3][6][7] Some malware targets the termination of a program to trigger execution, adversaries can use the `~/.bash_logout` file to execute malicious commands at the end of a session.

For macOS, the functionality of this technique is similar but may leverage zsh, the default shell for macOS 10.15+. When the Terminal.app is opened, the application launches a zsh login shell and a zsh interactive shell. The login shell configures the system environment using `/etc/profile` , `/etc/zshenv` , `/etc/zprofile` , and `/etc/zlogin` .[8][9][10][11] The login shell then configures the user environment with `~/.zprofile` and `~/.zlogin` . The interactive shell uses the `~/.zshrc` to configure the user environment. Upon exiting, `/etc/zlogout` and `~/.zlogout` are executed. For legacy programs, macOS executes `/etc/bashrc` on startup.

Source: https://attack.mitre.org/techniques/T1546/004