Comp 8006 Asst #3

Intrusion Detection

By: Andrew H.
& John Warren

For: Aman Abdulla

Due: March 8, 2017

Design Work

Due:

March 8, 2018 – 1300 hrs. You may work in groups of two.

Objective:

To design, implement and test a simple monitor application that will detect password guessing attempts against a service and block that IP using Netfilter.

Requirements

Design, implement and test an application that will monitor the /var/log/secure file and detect password guessing attempts and then use iptables to block that IP.

Your application will get user specified parameters (see constraints) and then continuously monitor the log file specified.

As soon as the monitor detects that the number of attempts from a particular IP has gone over a user-specified threshold, it will generate a rule to block that IP.

If the user has specified a time limit for a block, your application will flush the rule from Firewall rule set upon expiration of the block time limit.

Design a test procedure that will test your application under a variety of conditions. For example, how will you handle a situation when an attacker sends a slow scan of your system, meaning several password guessing attempts, but spaced far enough apart in time so that your application will miss the attack.

Constraints

The application will be implemented using any scripting or programming language of your choice.

The Firewall rules will be implemented using Netfilter.

Your application will obtain user input for the following parameters:
- The number of attempts before blocking the IP
- The time limit for blocking the IP. The default setting will be block indefinitely.
- Monitor a log file of user's choice (Optional - bonus). Keep in mind that different log files have different formats.

Your application will be activated through the crontab.

Submission

Hand in complete and well-documented design work and the firewall script.

You are also required to demonstrate your working programs during the lab the day the assignment is due.

A formal and detailed test plan as well as the test results for each test case.

Provide your test data and code and all supporting documentation. Include a set of instructions on how to use your application. Essentially a small "HOW-TO".

Submit a zip file containing all the code and documents as described below in the sharein folder for this course under “Assignment #3”.

Evaluation

(1). Design/Documentation: / 5
(2). Functionality: / 30
(3). Testing: / 15
Total: / 50

Design

monitor /var/log/secure file
detect password guessing
use IP tables to block IP
allow user specified parameters (#attempts, blocked-time[, additional-log-files)
over-threshold=>generate, activate rule
if time limit; flush after time limit exceeded
design test procedures
tests include missed slow scan
crontab activation
design work, firewall script, test plan, test results, instructions (HOW-TO)

generally;
monitor 'log' file for outside activity
- what to look for
- where to look

add rule to block IP
-
add cron job to remove rule after N secs/minutes/hours

macro level
- set parameters
  - times before lockout
  - duration of lockout
- add log and condition to monitor

Components

tool to add rule to iptables rules blocking IP
job to remove rule after N time
track number of instances of IP address incursions
method of modifying number of attempts
method of defining time limit
using system values?
add item to iptables

Pseudo Code

```
iptables -A INPUT -s $(block-ip) -j DROP; service iptables save
iptables -D INPUT -s $(block-ip) -j DROP; service iptables save

activate()
    set IPINT=3; export IPINT   // attack interval
    set IPRM=5; export IPRM     // remove after period
    ipblock () { iptables -A INPUT -s "$1" -j DROP; iptables -L | grep "$1"; `iptables
-D INPUT -s "$1" -j DROP | at now + "$IPRM" minutes`; } // command line function to
add rule and drop after defined period

duration()
```

```
    set IPRM="$1"; export IPRM

status()
    at -l
    iptables -L | grep "$1"

monitor()
    add entry to watchlist
    time, IP[, service]
```

Test Plan

Crontab

Insert
Frequency

Flexibilty

Ability to monitor other logs

Behaviour

test multiple entries
test with slow scan
test single instances