Introduction to Intrusion Detection Systems and Implementation - Design Work and Testing

David Tran - A00801942 | Cole Rees - A00741578

COMP 6D

COMP 8006 - Final Project

British Columbia Institute of Technology

Aman Abdulla

Thursday, March 13 2014

# 

# 

Table of Contents

# 

# Design

## IDS - Pseudocode (/var/log/secure) w. Timed Banning

// check if the banlist exists; if not create one

// check crontab for existing jobs relevant to ours; if not, add them into the entry for 30 apart

// read in the last x amount of log lines, AWK it by snipping it up and look for tokens #13 or #11, and if it matches “sshd”, “Failed password for” and if it includes “invalid user”; if so, store into a hash and increment

// if the hash size is greater than our maximum for allowed failed password attempts, then we should ban it; if so fetch the ip that AWK got for us and begin our logic

// check if we already banned it, if so, do nothing; otherwise check if our ban length has been specified; if null, then it is banned indefinitely

// if the ban length is specified, then…

// check our ban list for that IP and grab its respective ban time and status

// if status is null, then it must be a new offense, ban it.

// get the current time and add the ban length to it

// log it

// set iptables to drop all incoming packets from that IP

// output the IP, the time up to unbanning, and BANNED status

// if status is not null then check what the status is and…

// see if the ban time is less than our current time

// if so, then grab the most recent offending time from the logs

// add the ban length to it

// and check to see if the ban length is greater than our current time

// if it is, then we need to block it

// get the current time and add the ban length to it

// log it

// set iptables to drop all incoming packets from that IP

// output the IP, the time up to unbanning, and BANNED status

// for every entry in the ban list, we should reach each line and fetch each line’s values

// check the current status and see if it is BANNED

// if so, check iptables to see if we have banned it

// if it is banned, check to see if the times have expired

// if it is expired, unblock it and log that we’ve unblocked it.

// change the list status for that IP from BANNED to UNBANNED

// if it isn’t a BANNED status…

// continue to the next line

## IDS - Pseudocode for Ban List (banned.txt)

The text file includes the following

<ip address> <time in seconds> <status>

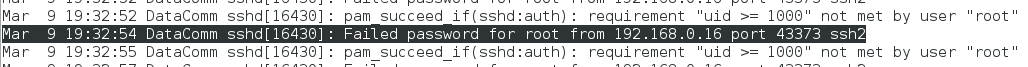
IP Address: The remote host that was attacking our machine, intentionally or not

Time in Seconds: This value is to represent the time where the host will be unbanned

Status: This is just to help us understand whether the host is either BANNED or UNBANNED

# Diagrams

## Sample Log Entry to Monitor (/var/log/secure)

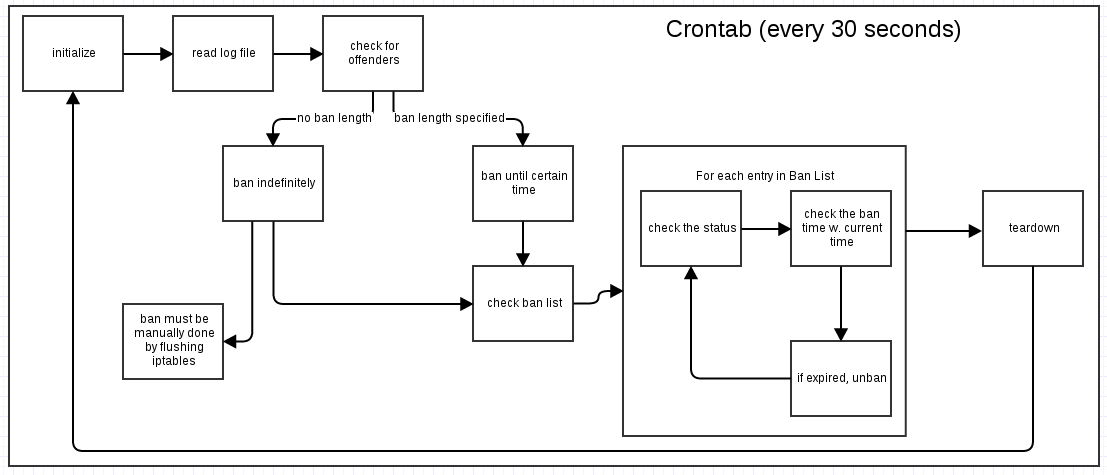


Note: “sshd[xxxx]”, “Failed password for” and “192.168.0.XX”

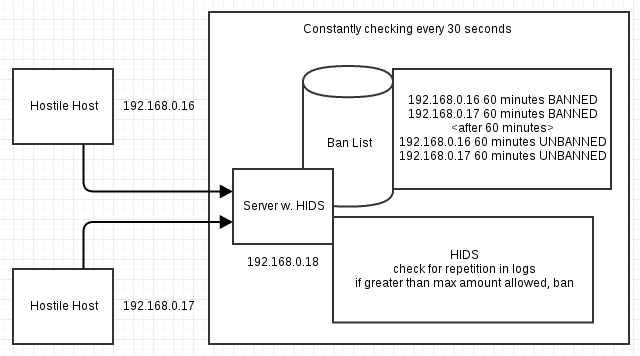
These are the values we will be checking in our HIDS logic

We will also note the time (ie. Mar 9 19:32:54)

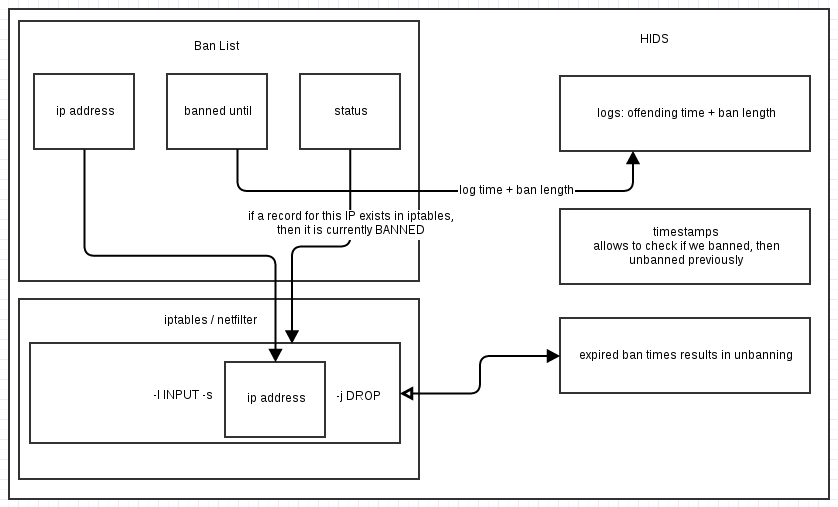
## State / Flow Chart Diagram



## Architecture Diagram

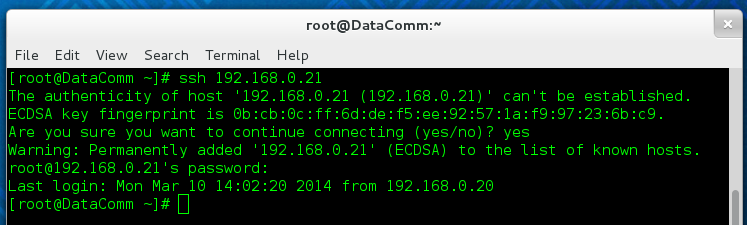


## HIDS Concept



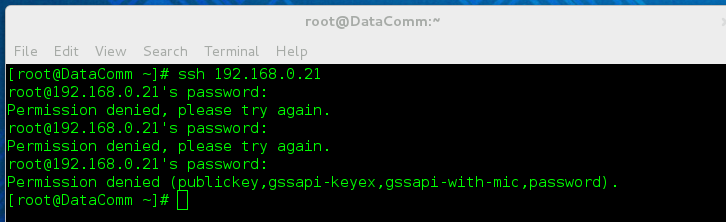
# Preliminary Testing

## Screenshot of SSH Attempt w. Correct PW



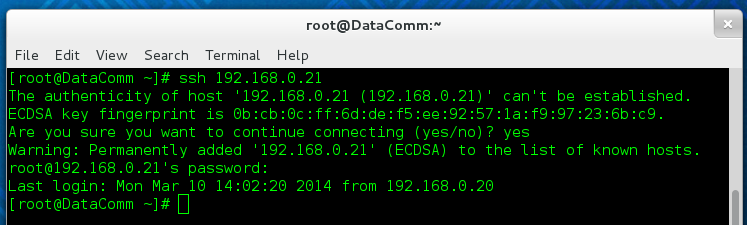
The user was able to access our terminal: 192.168.0.21 RESULT: PASSED

## Screenshot of SSH Attempt w. Incorrect PW



The user was unable to access our terminal: 192.168.0.21 after 3 tries RESULT: PASSED

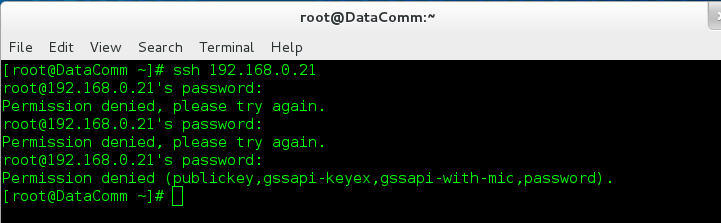
## Screenshot of SSH Attempt w. Correct PW on Different Host



The user was able to access our terminal: 192.168.0.21 from a different host

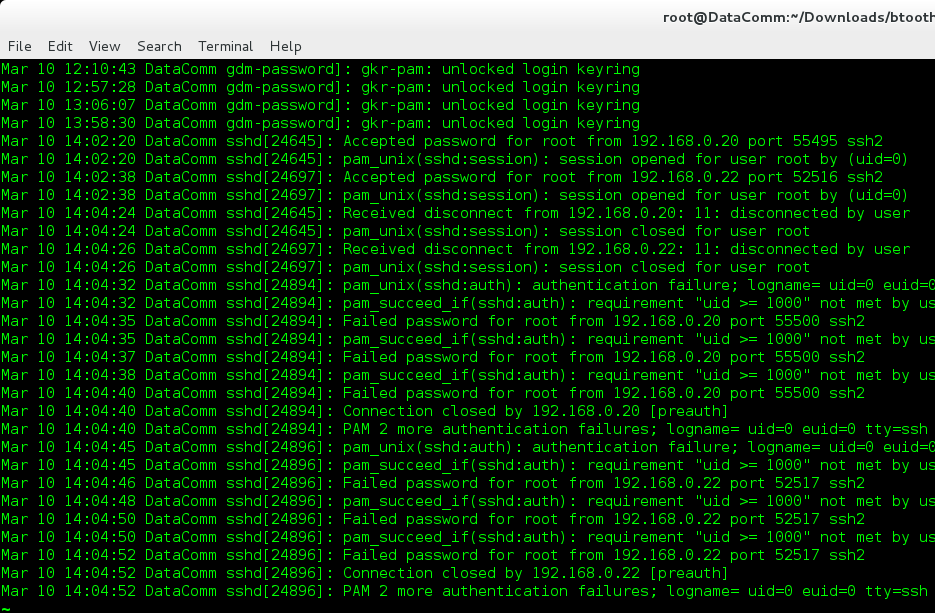
RESULT: PASSED

## Screenshot of SSH Attempt w. Incorrect PW on Different Host



The user was unable to access our terminal: 192.168.0.21 after 3 tries on different host RESULT: PASSED

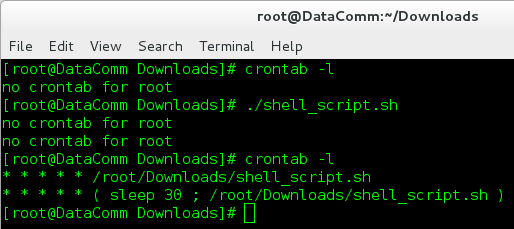
## Screenshot of /var/log/secure Logging Failed SSH Attempts



If you look closely, you will see attempts from hosts 192.168.0.20 and 192.168.0.22.

RESULT: PASSED

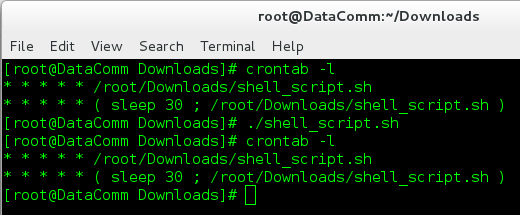
## Screenshot of Cron Job Insertion via Shell Script



There are no existing cron jobs on the server. After running the shell script and listing the crontab lists again, we see that there are now two new jobs specific to our script.

RESULT: PASSED

## Screenshot of Cron Job Insertion via Shell Script w. Existing Relevant Jobs



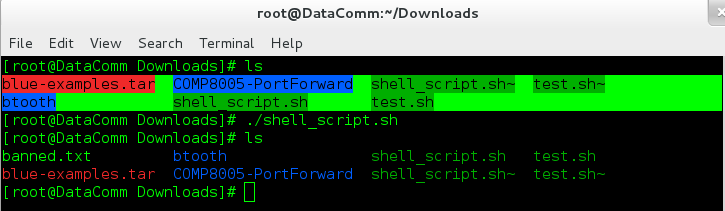
We see that our jobs are already listed. We ran the script again manually to see if it duplicates the job entries. It does not, as there are only 2 jobs still specific to our script.

RESULT: PASSED

## 

## 

## Screenshot of Ban List (banned.txt) Being Created

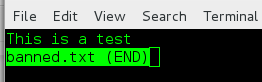
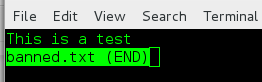
Firstly, we see that the Ban List does not exist until after we run our shell script.

RESULT: PASSED

## Screenshot of Ban List (banned.txt) Not Being Overwritten w. New Execution

## Screenshot from 2014-03-10 14:13:06.png

We simply add a line into our ban list. In our first instance of the less command (bottom left), we see that the insertion of a new line is there. After we run our shell script, in the second instance of the less command (bottom right), the file is still the same and not overwritten.



RESULT: PASSED

[End of Preliminary Testing]