Introduction to Intrusion Detection Systems and Implementation

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

# 

# Background

The next level of security introduces Intrusion Detection and Prevention Systems known as IDS and IPS respectively. For the purpose of this assignment, we will focus on the IDS side, especially concerning Host-based IDS’s or HIDS (the latter consists of Network-Based IDS’s or NIDS).

At the host-level intrusion detection system, the concept is to be able to monitor resources that can help tell a story. These come in the form of log files. Once we are able to monitor these log files, the HIDS will identify each log entry and look for a signature. For the purpose of this assignment, we will utilize SSH protocol and look for a signature pertaining to failed password attempts (ie. “Failed password for” in the log file entry).

To trigger the action to ban an offending host, we must set a threshold and this is user-definable in our shell script implementation. This threshold essentially states how many times the user can have failed password attempts before being considered a hostile or attacking host. Once the attacker meets the threshold, the system will take appropriate actions.

“Appropriate actions” could be to retaliate to the offending host. However, as discussed in class, this is not an effective mode of action for legal and psychological reasons. Instead, we will utilize Netfilter implementation and ban the user’s IP address from communicating with us again. We can take it one step further to ban the network (or even to the extreme by banning the entire country) but for the sake of simplicity and understanding, we will just ban the host.

In our implementation, we will have a timed ban that will be stored in a tertiary ban list external of our shell script. This will help us keep track of when to ban and unban hosts. Like the threshold for failed SSH password attempts, the length of the ban is also user-definable. When not defined or left null, any offenders are banned indefinitely.

# Tools & Equipment

## Hardware

|  |  |  |
| --- | --- | --- |
| * 8GB RAM | * Intel i5 Quad Core | * 500GB HDD |
| * 2x Hostile Hosts | * 1 Host w. HIDS |  |

## Software

|  |  |  |
| --- | --- | --- |
| * Wireshark | * hping3 | * shell scripting |
| * ssh | * /var/log/secure log file | * iptables / Netfilter |

# Testing Procedure

To begin, we will set up our shell script with the following user-defined variables:

|  |  |
| --- | --- |
| **IP** | **Role** |
| 192.168.0.15 | Attacker |
| 192.168.0.14 | Attacker |
| 192.168.0.16 | Host w. HIDS |

|  |  |  |
| --- | --- | --- |
| **User-Defined Variable** | **Type** | **Value** |
| MAX\_FAILED\_ATTEMPTS | integer | 2 |
| BAN\_LENGTH | string | “30 seconds” |

After setting these variables, we will execute our shell script. Then we will start with the testing procedures.

## Test Cases Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case #** | **Test Case** | **Tools Used** | **Expected Outcome** | **Results** |
| 1 | SSH with proper credentials | ssh, Wireshark, log file | No banning, log files reflect this | PASSED. See details |
| 2 | SSH with improper credentials x1 | ssh, log file | No banning, log files reflect this | PASSED. See details |
| 3 | SSH with improper credentials x3 | ssh, log file, iptables | Threshold reached, banning, log files reflect this, iptables has a record | PASSED. See details |
| 4 | Ban List has a record of the IP Address | banned.txt | The attacking host will have their IP, the time they are banned until, and their current status as BANNED | PASSED. See details |
| 5 | Attacker cannot connect to Host with HIDS | hping3, iptables | No connection allowed, iptables drops all packets | PASSED. See details |
| 6 | After banning, host will be unbanned after an amount of time | ssh, iptables, Wireshark | Connection allowed, iptables drops the banning rule | PASSED. See details |
| 7 | Ban List updates the correct record with respect to the attacking host’s IP | banned.txt | Their record will update from BANNED status to UNBANNED status | PASSED. See details |
| 8 | SSH with proper credentials from another host | ssh, iptables, Wireshark | No banning, log files reflect this | PASSED. See details |
| 9 | SSH with improper credentials x1 from another host | ssh, iptables, log file | No banning, log files reflect this | PASSED. See details |
| 10 | SSH with improper credentials x3 from another host | ssh, iptables, log file | Threshold reached, banning, log files reflect this, iptables has a record | PASSED. See details |
| 11 | Ban List has a record of the IP Address from new host | banned.txt | The attacking host will have their IP, the time they are banned until, and their current status as BANNED | PASSED. See details |
| 12 | Attacker (new host) cannot connect to Host with HIDS | hping3, iptables | No connection allowed, iptables drops all packets | PASSED. See details |
| 13 | After banning, new host will be unbanned after an amount of time | ssh, iptables | Connection allowed, iptables drops the banning rule | PASSED. See details |
| 14 | Ban List updates the correct record with respect to the attacking host’s IP | banned.txt | Their record will update from BANNED status to UNBANNED status | PASSED. See details |
| 15 | Indefinite banning (no ban length specified) | banned.txt, iptables | iptables will have the correct record; ban list will not have any records since there is no ban length specified | PASSED. See details |

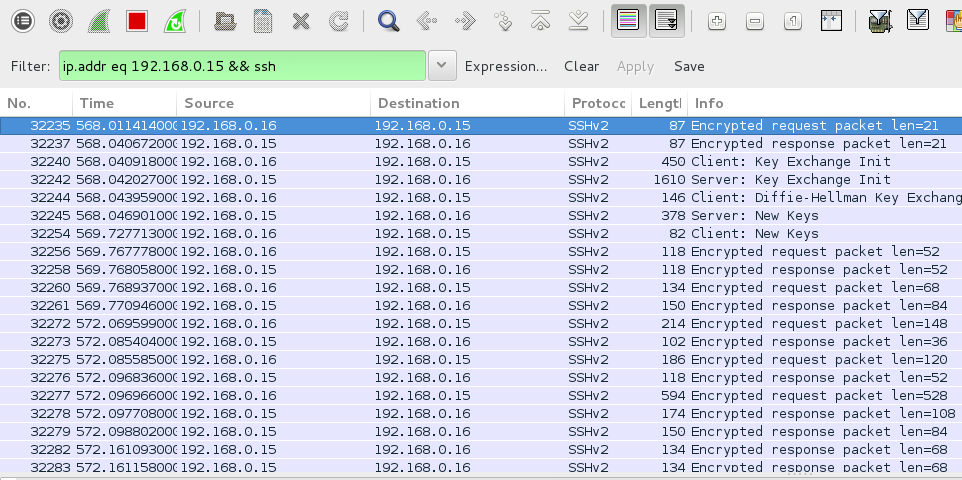
## 

## 

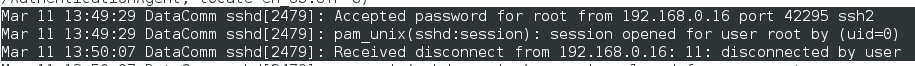
## Test Case Details and Evidence

### 1. SSH w. Proper Credentials

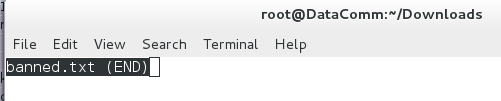
Wireshark Capture:



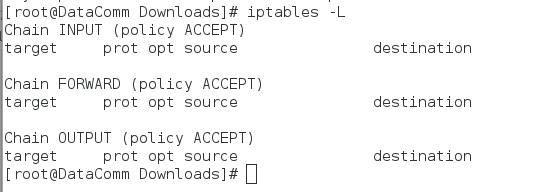
Log File Stream:



Ban List Contents:



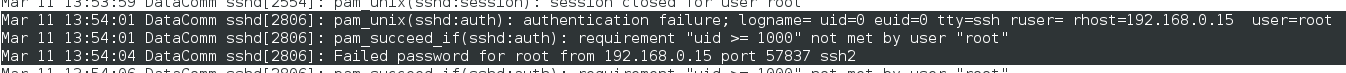
IPtables Policies:



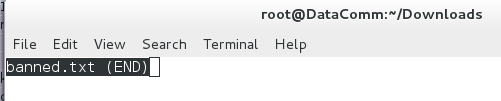
### 

### 2. SSH w. Improper Credentials x2

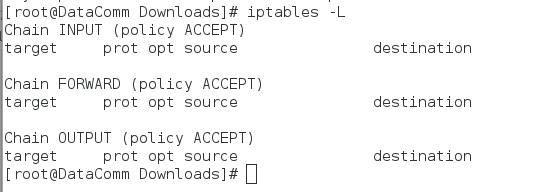
Log file stream:



Ban list contents:

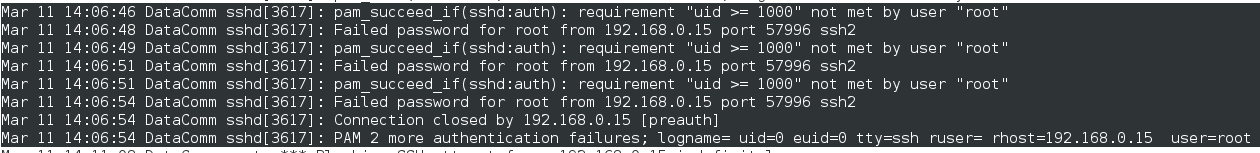


IPtables Policies:



### 3. SSH w. Improper Credentials x3

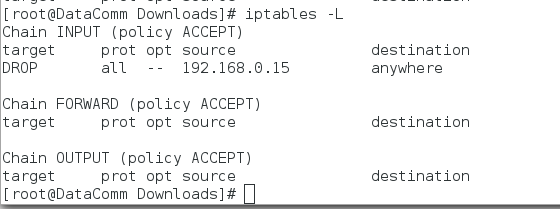
Log file stream:



Note these logs from our shell script:

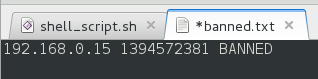


IPtables Policies (note the listings):



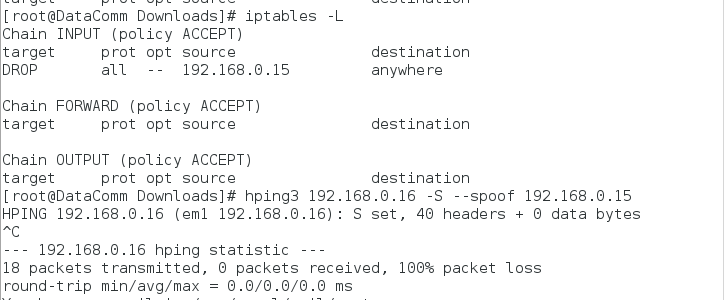
### 4. Ban List has a record of the IP Address

Ban List contents after 3 failed tries:



### 5. Attacker cannot connect to Host with HIDS

HPing Spoof on our host with banned IP:

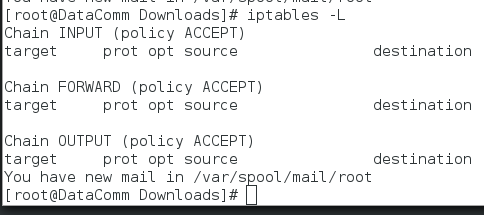


### 

### 

### 6. After banning, host will be unbanned after an amount of time

IPtables Policies:



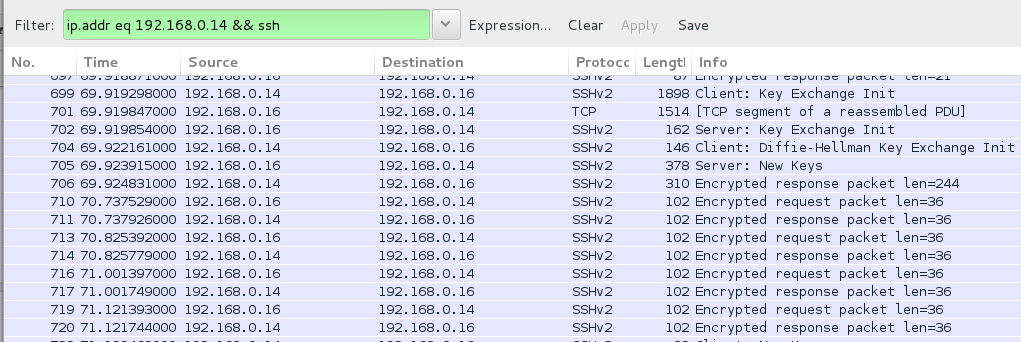
### 7. Ban list updates the correct record with respect to attacker’s IP

Ban list contents:

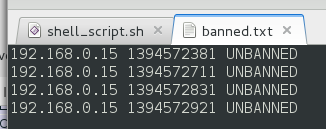


### 8. SSH w. Proper Credentials from another host

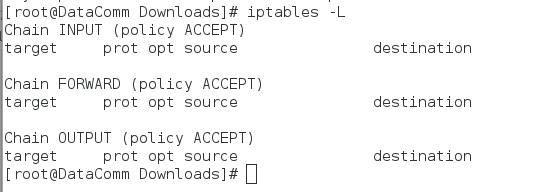
Wireshark Capture:



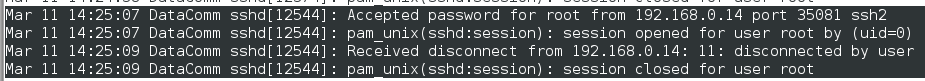
Banned List Contents:



IPtables Policies:

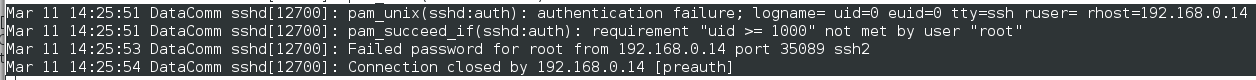


Log file stream:

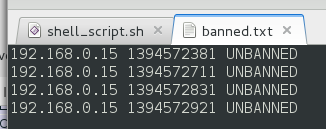


### 9. SSH w. Improper Credentials from another host x1

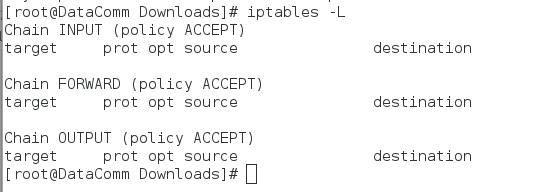
Log file stream:



Ban list contents:



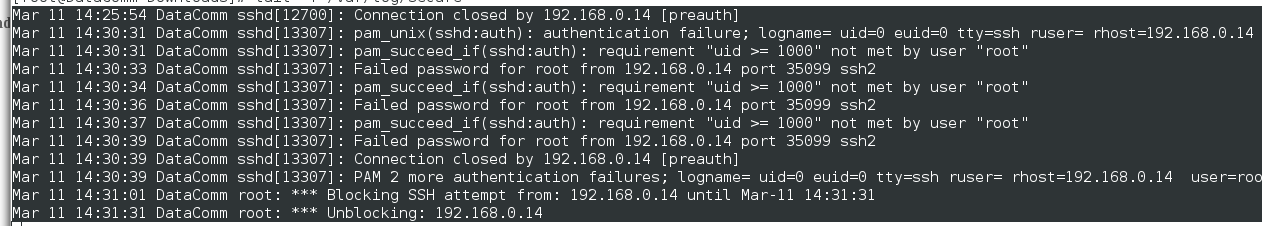
IPtables Policies:



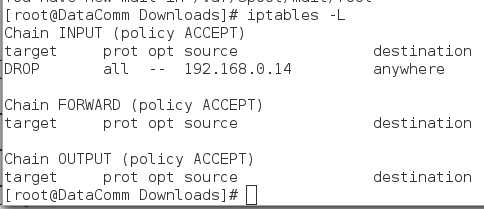
### 

### 10. SSH w. Improper Credentials from another host x3

Log file stream:

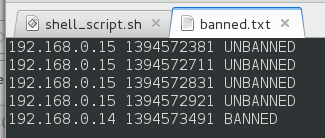


IPtables Policies (note the listings):



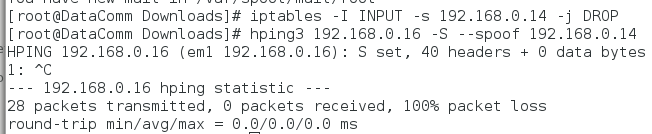
### 11. Ban List has a record of the IP Address from new host

Ban list contents (note the status of the last entry):



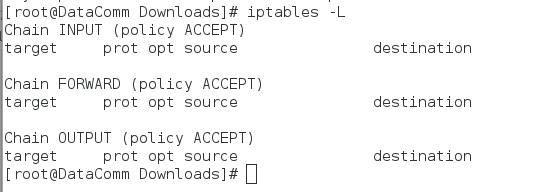
### 12. Attacker (new host) cannot connect to Host with HIDS

HPing Spoof on our host with banned IP:



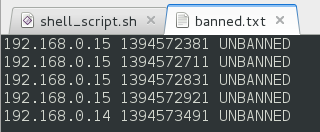
### 13. After banning, new host will be unbanned after an amount of time

IPTables Policies:



### 14. Ban List updates the correct record with respect to the attacking host’s IP

Ban list contents (note the change in status for the last entry):

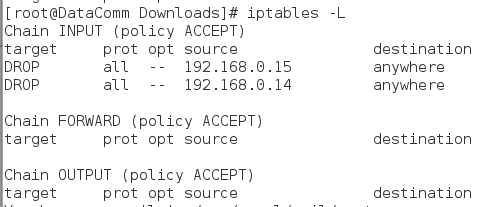


### 15. Indefinite banning (no ban length specified)

Log file stream:



IPTables Policies (note the two entries):



## 

## 

## Scenario Test Cases

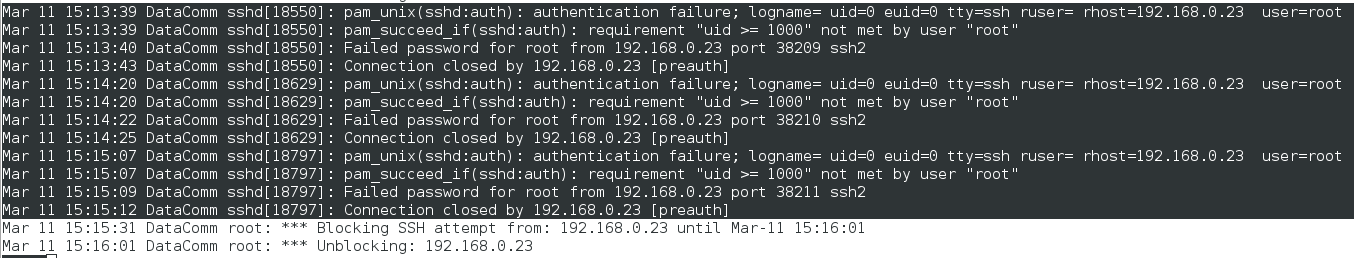
Let’s consider some hypothetical scenarios to test our IDS:

|  |  |  |
| --- | --- | --- |
| **User-Defined Variable** | **Type** | **Value** |
| MAX\_FAILED\_ATTEMPTS | integer | 2 |
| BAN\_LENGTH | string | “30 seconds” |
| READ\_LAST | integer | 100 |

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Tools Used** | **Expected Outcome** |
| Attacker with a slow scan | ssh, log file | Ban the attacker |
| Removing Entries in the Ban List | banned.txt, iptables, log file | Ban all previous attackers up to the last x lines of our log specified by our user |

### Attacker with a Slow Scan

The idea of a slow scan is the attacker sends password guessing attempts over a period of time instead of one big burst during one session. The attack is still malicious and thus, we need to ban the offending host.

Log file stream:

We see here that the attacker has spread out their guessing by close to 1 minute per attempt. Regardless, we are still able to see the threshold being met, so we ban the attacker, and then unblock him after 30 seconds.

RESULTS: PASSED

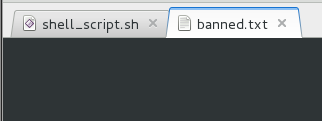
### 

### 

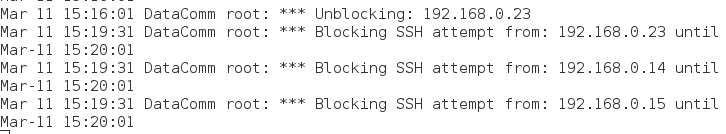
### Removing Entries in the Ban List

The removal of entries in the ban list can be malicious or unintentional. Whatever the case may be, it is better for us to suspect that perhaps one of the banned hosts is linked to the tampering of the ban list. So ban all the attackers in our log file again up to the last number of lines specified by our user to read.

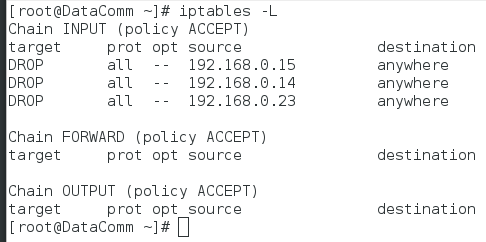
Ban List Contents:



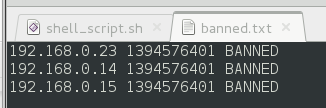
Log File Stream:



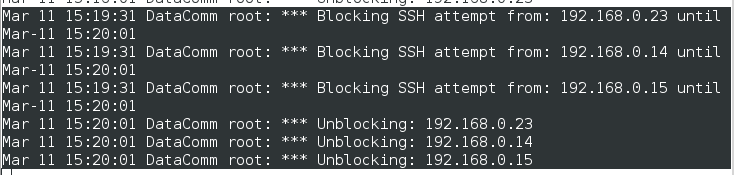
IPtables Policy:



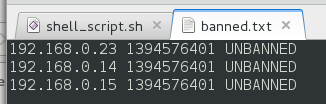
Ban List Contents after Iptables policy update:



Log File Stream (notice the last three entries):



Ban List Contents after Unblocking (note their statuses):



Being unrelenting as we are, we ban all suspects and then unban them after their ban has been expired.

RESULTS: PASSED

# 

# Conclusion

Through the extensive test cases of this assignment, it is sufficient to say that our host-based Intrusion Detection System has covered the relevant criteria of Assignment 3. They are the following:

* monitoring the /var/log/secure file for failed SSH password attempts
* implementing user-specified parameters including:
  + the number of attempts before blocking the IP
  + time limit for blocking the IP and if left null, to block indefinitely
* continuous monitoring of the log file is activated through crontab
* using Netfilter and iptables to ban offending hosts
* adding a rule when an offending host is detected after crossing the threshold
* deleting the rule when the expiry date on the ban has passed

This assignment and code are by all means NOT a catch-all intrusion detection system. Rather, it is a taste of what is to come in the future and an introductory experience to understand and comprehend what the IDS is capable of.

# 

# 

# 

# Appendix

These are the files you will find located on-disk:

* Introduction to Intrusion Detection Systems and Implementation (.pdf)
* Introduction to Intrusion Detection Systems and Implementation - Design Work & Testing (.pdf)
* shell\_script.sh (which produces)
  + banned.txt (ban list)
* README.txt