UAF Cyber Security Club

Linux Logs

Arsh Chauhan
11/04/2017

Lecture

Loggins is something that exists in all Operating systems in various different formats. Logs are important since they keep a history of all the stuff that happened, this isn’t just super helpful while trying to find or respond to an intrusion but works wonders to diagnose why stuff isn’t working. Parsing through all the information amongst the different logs for every application is a herculean task and this doesn’t get any easier even if you’re focused on a single application since that application could have a lot of data and/or be in a weird format. (Fortunately?) we have tools that can help us. Before we go into tools, let’s look at how to access logs on Linux.

Linux Logs

All linux distributions store logs in ***/var/log***. Most common applications store their logs in subdirectories of ***/var/log***, but this directory has important logs related to “core” modules. Let’s look at some of these.

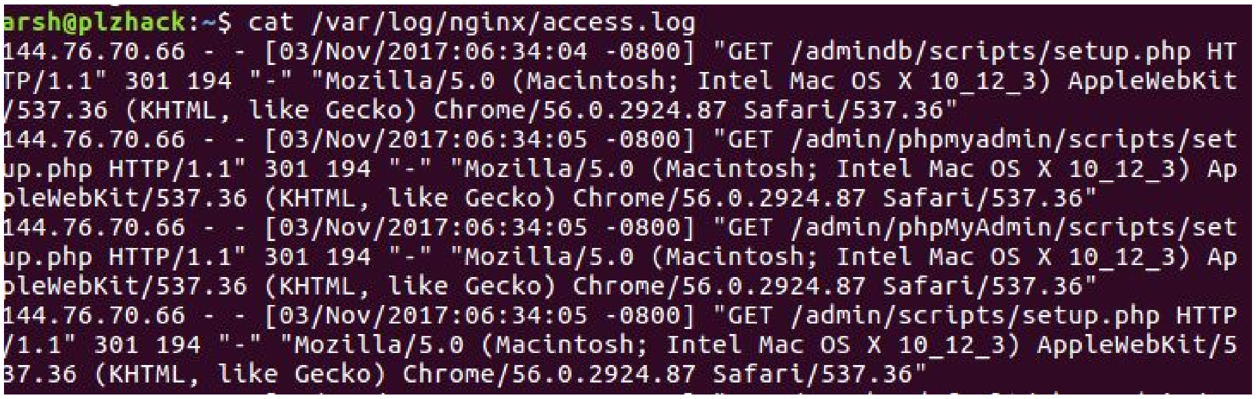
1) **auth.log**: Stores all authentication logs. Includes events generated by CRON,PAM, systemd and other utilities that handle anything to do with authentication (including gnome keyring and compiz). Basically, if it involves user authentication then the log is probably here. The big thing this file logs is SSH access attempts, we care about SSH logins since that is the big way people can access a Linux server remotely.
2) **lastlog**: Binary file. Used by the ***lastlog*** program to display the last time each user logged in.
3) **syslog**: Log for syslog, this basically logs everything. More on this later.

Programs generally store their logs in a subdirectory with the program name in **/var/log** (e.g. Nginx stores its logs in **/var/log/nginx**). If you don’t know if a program has logs, go check ***/var/log/programName*** and you'll probably find them.

Application Logs

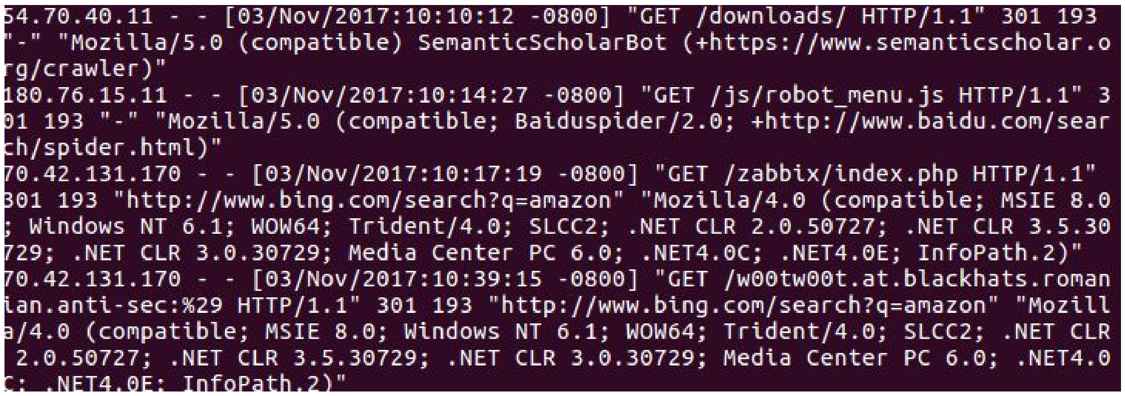
We mentioned that applications store their own logs which are often stored in a subdirectory under **/var/log/programName**. Last week we learned a little about Nginx so where do you think Nginx will store its logs? (***/var/log/nginx***).Nginx and Apache have two log files

1) access.log: Logs all the requests to the server along with the return code with a timestamp and the IP address the request came from. Any real web server will see many requests,depending on the traffic most of these will either be real people, bots or automated vulnerability scanning tools.
2) error.log: Shows errors that occurred during an access or configuration changes to the server. I think it logs errors for 404 since that causes a failure in the open() call



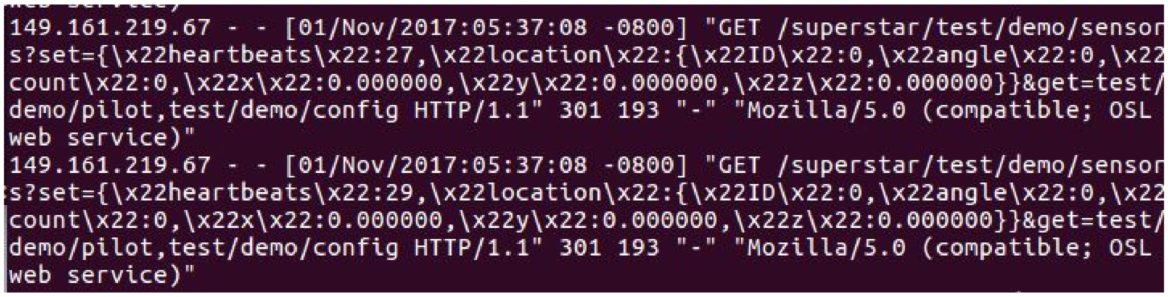
```
arsh@plzhack:~$ cat /var/log/nginx/access.log
144.76.70.66 - - [03/Nov/2017:06:34:04 -0800] "GET /admindb/scripts/setup.php HT
TP/1.1" 301 194 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit
/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
144.76.70.66 - - [03/Nov/2017:06:34:05 -0800] "GET /admin/phpmyadmin/scripts/set
up.php HTTP/1.1" 301 194 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) Ap
pleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
144.76.70.66 - - [03/Nov/2017:06:34:05 -0800] "GET /admin/phpMyAdmin/scripts/set
up.php HTTP/1.1" 301 194 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) Ap
pleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
144.76.70.66 - - [03/Nov/2017:06:34:05 -0800] "GET /admin/scripts/setup.php HTTP
/1.1" 301 194 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/5
37.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36"
```

access.log showing access to commonly scanned default directories for web applications



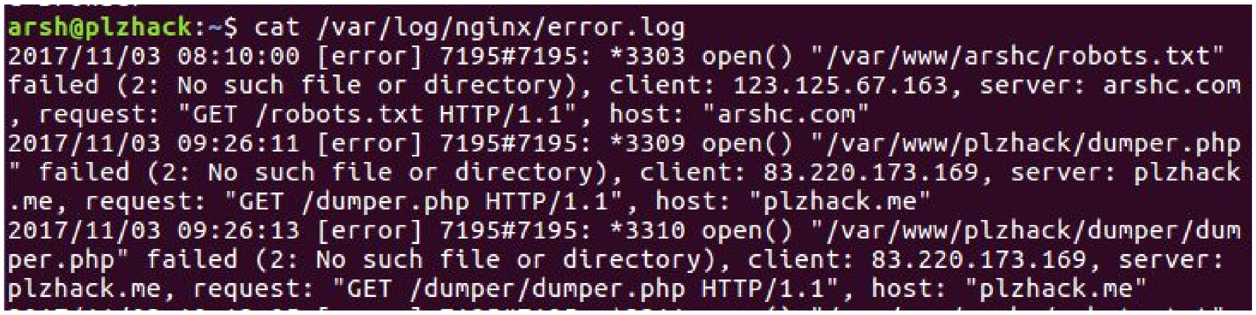
```
54.70.40.11 - - [03/Nov/2017:10:10:12 -0800] "GET /downloads/ HTTP/1.1" 301 193
"-" "Mozilla/5.0 (compatible) SemanticScholarBot (+https://www.semanticscholar.o
rg/crawler)"
180.76.15.11 - - [03/Nov/2017:10:14:27 -0800] "GET /js/robot_menu.js HTTP/1.1" 3
01 193 "-" "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/sear
ch/spider.html)"
70.42.131.170 - - [03/Nov/2017:10:17:19 -0800] "GET /zabbix/index.php HTTP/1.1"
301 193 "http://www.bing.com/search?q=amazon" "Mozilla/4.0 (compatible; MSIE 8.0
; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30
729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.2)"
70.42.131.170 - - [03/Nov/2017:10:39:15 -0800] "GET /w00tw00t.at.blackhats.roman
ian.anti-sec:%29 HTTP/1.1" 301 193 "http://www.bing.com/search?q=amazon" "Mozill
a/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR
 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0
C; .NET4.0E; InfoPath.2)"
```

access.log showing access by search bots



```
149.161.219.67 - - [01/Nov/2017:05:37:08 -0800] "GET /superstar/test/demo/sensor
s?set={\x22heartbeats\x22:27,\x22location\x22:{\x22ID\x22:0,\x22angle\x22:0,\x22
count\x22:0,\x22x\x22:0.000000,\x22y\x22:0.000000,\x22z\x22:0.000000}}&get=test/
demo/pilot,test/demo/config HTTP/1.1" 301 193 "-" "Mozilla/5.0 (compatible; OSL
web service)"
149.161.219.67 - - [01/Nov/2017:05:37:08 -0800] "GET /superstar/test/demo/sensor
s?set={\x22heartbeats\x22:29,\x22location\x22:{\x22ID\x22:0,\x22angle\x22:0,\x22
count\x22:0,\x22x\x22:0.000000,\x22y\x22:0.000000,\x22z\x22:0.000000}}&get=test/
demo/pilot,test/demo/config HTTP/1.1" 301 193 "-" "Mozilla/5.0 (compatible; OSL
web service)"
```

Access.log showing access to real resources with uri parameters



```
arsh@plzhack:~$ cat /var/log/nginx/error.log
2017/11/03 08:10:00 [error] 7195#7195: *3303 open() "/var/www/arshc/robots.txt"
failed (2: No such file or directory), client: 123.125.67.163, server: arshc.com
, request: "GET /robots.txt HTTP/1.1", host: "arshc.com"
2017/11/03 09:26:11 [error] 7195#7195: *3309 open() "/var/www/plzhack/dumper.php
" failed (2: No such file or directory), client: 83.220.173.169, server: plzhack
.me, request: "GET /dumper.php HTTP/1.1", host: "plzhack.me"
2017/11/03 09:26:13 [error] 7195#7195: *3310 open() "/var/www/plzhack/dumper/dum
per.php" failed (2: No such file or directory), client: 83.220.173.169, server:
plzhack.me, request: "GET /dumper/dumper.php HTTP/1.1", host: "plzhack.me"
```

error.log showing failed open() calls for file not found

SSH

SSH does not have its own subdirectory in ***/var/log*** because it uses PAM and its authentication logs go to **auth.log**. Again, you'll see a lot of failed (hopefully) brute force attempts on a real server. Here is how you filter ***auth.log*** for SSH events

```
cat auth.log | grep ssh
```

Filter for failed attempts

```
cat auth.log | grep -i ssh | grep -i failed
```

My approach for intrusion response when an SSH attack is suspected is to look for multiple failed login attempts and then find the one that succeeded.

UFW

Uncomplicated Firewall (UFW) is a wrapper around iptables. UFW has it pros and cons but one of the pros is it handles logging automatically so we'll be using it for this lecture due to it's simplicity. The UAF team ends up using UFW during CCDC anyways. Like SSH, UFW also does not have its own subdirectory but stores it's logs in the file ***ufw.log***. UFW logs are "easy"to understand,let's look at an example entry

```
Oct 29 06:26:45 plzhack kernel: [322517.222439] [UFW BLOCK] IN=eth0 OUT=
MAC=04:01:bc:1c:df:01:3c:8a:b0:0d:3f:f0:08:00 SRC=27.143.72.232 DST=159.203.237.114 LEN=40
TOS=0x00 PREC=0x00 TTL=48 ID=13767 PROTO=TCP SPT=33375 DPT=23 WINDOW=46191 RES=0x00 SYN URGP=0
```

This entry tells us that UFW blocked a packet on 10/29/2017 coming in on interface eth0 from IP 27.143.72.232 to TCP port 23 (Telnet).
PS: I haven't done it yet but a webpage that shows stats on your firewall (UFW) logs would be a cool project.
Again this is how you'd filter for all packets blocked by UFW

```
cat ufw.log | grep -i BLOCK
```

Find blocked connections from IP 220.181.12.15

```
cat ufw.log | grep -i BLOCK | grep "SRC=220.181.12.15"
```

Find all connections from IP 220.181.12.15

```
cat ufw.log | grep "SRC=220.181.12.15"
```

We'll probably do more on UFW in a later meeting.

Can logs be trusted?

Logs are stored on the machine,so can we really trust them as forensic evidence?
Yes and No, how much you can trust logs depends on who the "attacker" is and what level of access they had on the system. If your attacker is a disgruntled employee (or a script kiddie), then it's safe to assume that the logs can be trusted, but if the attacker is the government or an APT, it's highly likely that logs have been altered or deleted altogether.

In cases of an advanced attacker, it's wise to not trust anything installed on the machine since tools you use for forensics may have been altered by the attacker. A simple example of how an attacker may modify a commonly used command is to hide their files from not showing up in an *ls*

```
#!/bin/sh
/bin/ls "$@" | grep -v "ls" | grep -v "attacker*"
```

This script to hide any file starting with attacker and can be used system-wide if the attacker replaces the real */bin/ls* with this script and renames ls to something like *foo*. This is how the modified file would look

```
#!/bin/sh
/bin/real_ls "$@" | grep -v "ls" | grep -v "attacker*"
```

The $PATH variable has directories in the current user's home folder also which are checked before /bin, so an attacker without root privileges could still add their own script called *ls* and mess up forensics done using that user account. We may deal with detecting these changes in the future but Dr. Orion Lawlor (CS, UAF) talks about this in his lecture notes for his Computer Security class (2017).

Lab

Part 1: Intrusion Detection and Response

You're a security analyst at E-Corp. Tyrell Wellick suspects there has been an intrusion and has tasked you with figuring out if F-Society has managed to get into your web- server. From experience you know this webserver is running SSH, Nginx and UFW. Your task is to figure out if an intrusion has occurred or not. If yes, write a short report for Tyrell giving him:

1) Did an attack happen
2) Initial attack vector (how the attacker got in)
3) The attacker's IP
4) What access did the attacker gain
5) Was data exfiltrated
6) How can this attack be prevented in the future

Part 2: Linux remote log server

1) Client machine

   Setup the client to send logs to the server. The Client will be running SSH and have a bunch of failed login attempts and a successful login. There will also be stuff in bash_rc since that is sorta like a log. Maybe even have them send it to Splunk. This machine also has Nginx and firewall (UFW) logging so we have web access logs and nmap scan logs

2) Server

   The central syslog server that will receive the logs from the client.