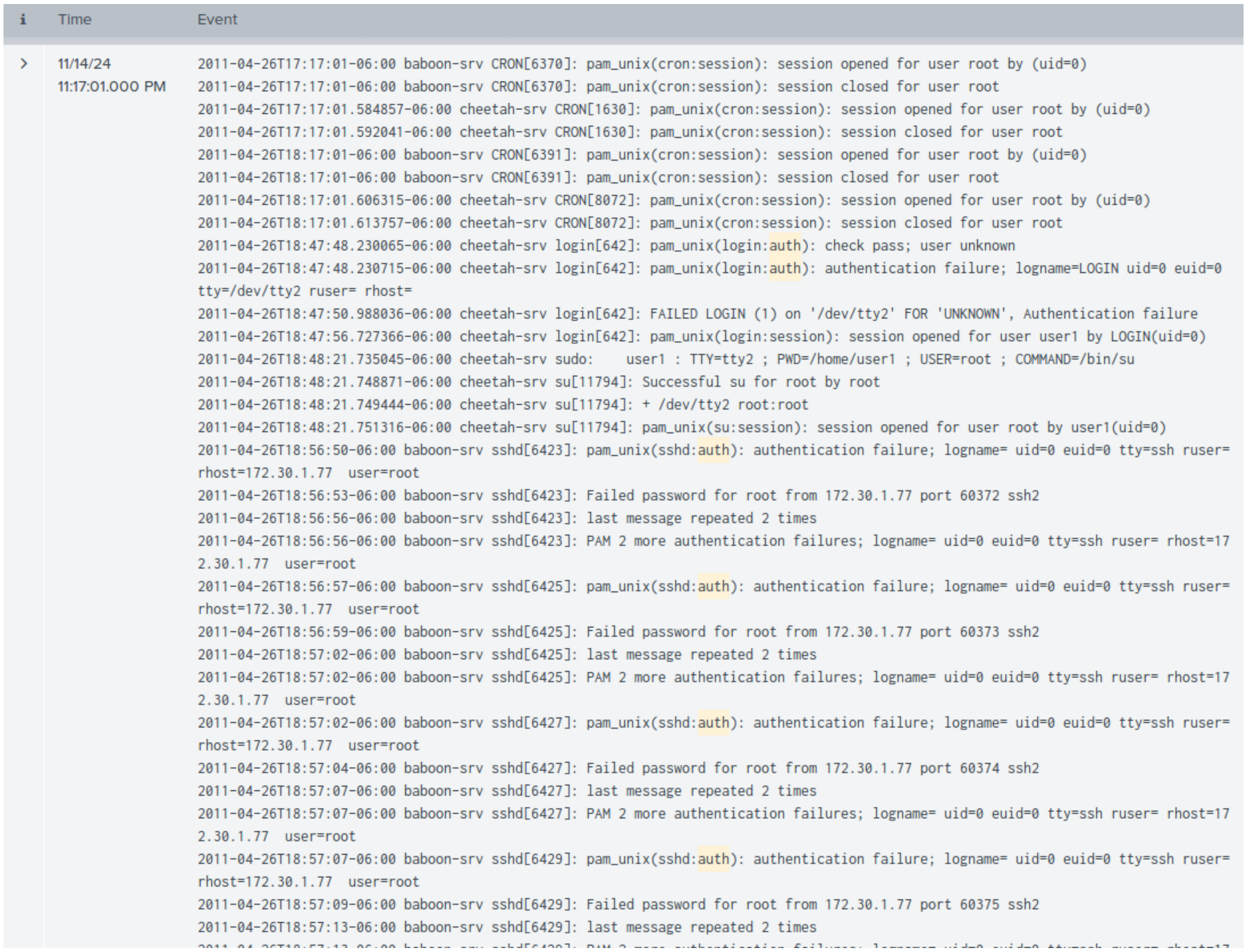
LOG AND MALWARE ANALYSIS

Log Analysis

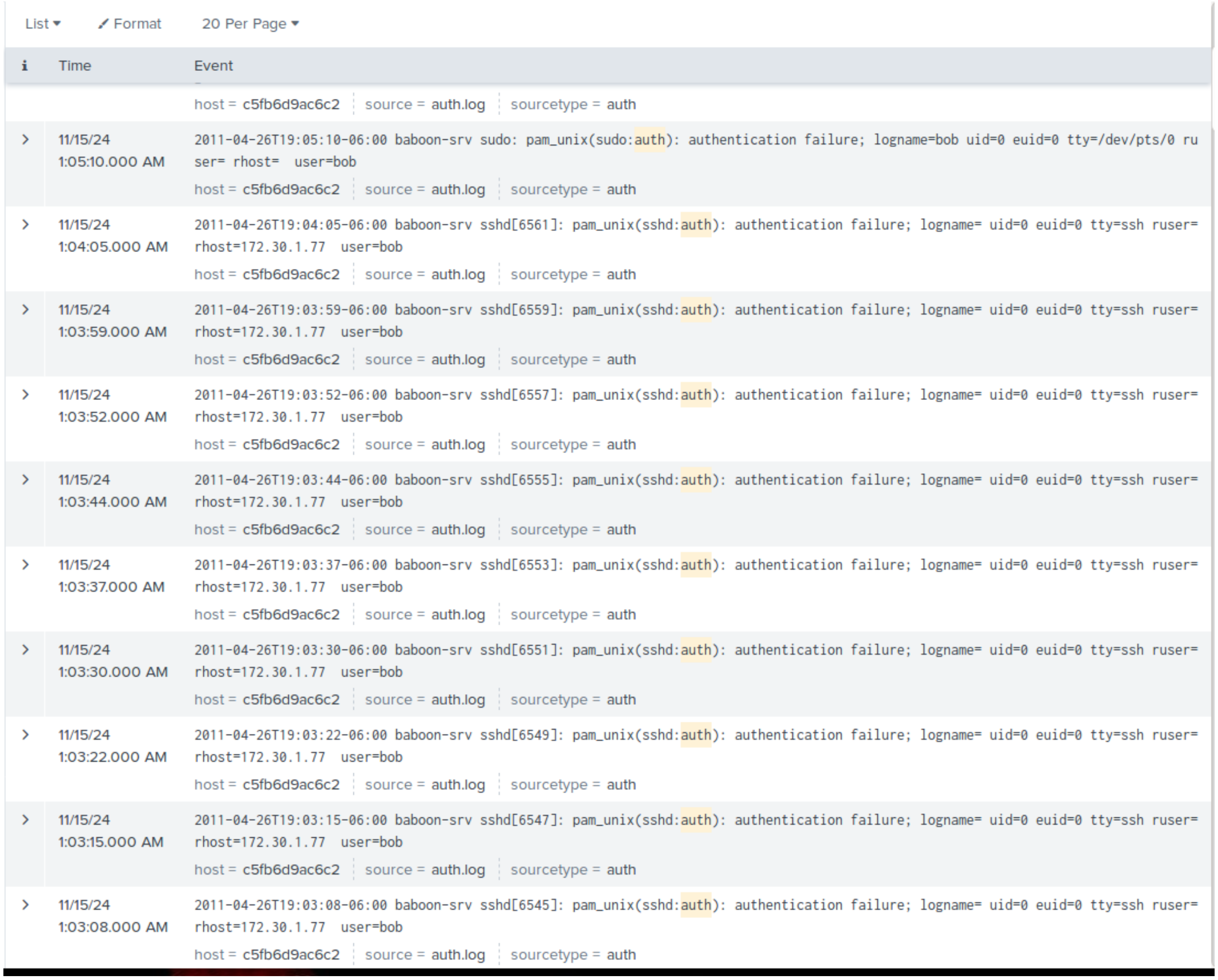
The failed login attempts clearly indicated a deliberate attack since the intervals of the logins are so short with the difference of only seconds. This shows an overwhelming login attempts within a brief time by a host 172:30:1:77 which is from the internet targeting the root and bob user. The screenshots below show the logs saved from this malicious activity.



11/14/24
11:17:01.000 PM

```
2011-04-26T17:17:01-06:00 baboon-srv CRON[6370]: pam_unix(cron:session): session opened for user root by (uid=0)
2011-04-26T17:17:01-06:00 baboon-srv CRON[6370]: pam_unix(cron:session): session closed for user root
2011-04-26T17:17:01.584857-06:00 cheetah-srv CRON[1630]: pam_unix(cron:session): session opened for user root by (uid=0)
2011-04-26T17:17:01.592041-06:00 cheetah-srv CRON[1630]: pam_unix(cron:session): session closed for user root
2011-04-26T18:17:01-06:00 baboon-srv CRON[6391]: pam_unix(cron:session): session opened for user root by (uid=0)
2011-04-26T18:17:01-06:00 baboon-srv CRON[6391]: pam_unix(cron:session): session closed for user root
2011-04-26T18:17:01.606315-06:00 cheetah-srv CRON[8072]: pam_unix(cron:session): session opened for user root by (uid=0)
2011-04-26T18:17:01.613757-06:00 cheetah-srv CRON[8072]: pam_unix(cron:session): session closed for user root
2011-04-26T18:47:48.230065-06:00 cheetah-srv login[642]: pam_unix(login:auth): check pass; user unknown
2011-04-26T18:47:48.230715-06:00 cheetah-srv login[642]: pam_unix(login:auth): authentication failure; logname=LOGIN uid=0 euid=0
tty=/dev/tty2 ruser= rhost=
2011-04-26T18:47:50.988036-06:00 cheetah-srv login[642]: FAILED LOGIN (1) on '/dev/tty2' FOR 'UNKNOWN', Authentication failure
2011-04-26T18:47:56.727366-06:00 cheetah-srv login[642]: pam_unix(login:session): session opened for user user1 by LOGIN(uid=0)
2011-04-26T18:48:21.735045-06:00 cheetah-srv sudo:    user1 : TTY=tty2 ; PWD=/home/user1 ; USER=root ; COMMAND=/bin/su
2011-04-26T18:48:21.748871-06:00 cheetah-srv su[11794]: Successful su for root by root
2011-04-26T18:48:21.749444-06:00 cheetah-srv su[11794]: + /dev/tty2 root:root
2011-04-26T18:48:21.751316-06:00 cheetah-srv su[11794]: pam_unix(su:session): session opened for user root by user1(uid=0)
2011-04-26T18:56:50-06:00 baboon-srv sshd[6423]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=172.30.1.77  user=root
2011-04-26T18:56:53-06:00 baboon-srv sshd[6423]: Failed password for root from 172.30.1.77 port 60372 ssh2
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: last message repeated 2 times
2011-04-26T18:56:56-06:00 baboon-srv sshd[6423]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=17
2.30.1.77  user=root
2011-04-26T18:56:57-06:00 baboon-srv sshd[6425]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=172.30.1.77  user=root
2011-04-26T18:56:59-06:00 baboon-srv sshd[6425]: Failed password for root from 172.30.1.77 port 60373 ssh2
2011-04-26T18:57:02-06:00 baboon-srv sshd[6425]: last message repeated 2 times
2011-04-26T18:57:02-06:00 baboon-srv sshd[6425]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=17
2.30.1.77  user=root
2011-04-26T18:57:02-06:00 baboon-srv sshd[6427]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=172.30.1.77  user=root
2011-04-26T18:57:04-06:00 baboon-srv sshd[6427]: Failed password for root from 172.30.1.77 port 60374 ssh2
2011-04-26T18:57:07-06:00 baboon-srv sshd[6427]: last message repeated 2 times
2011-04-26T18:57:07-06:00 baboon-srv sshd[6427]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=17
2.30.1.77  user=root
2011-04-26T18:57:07-06:00 baboon-srv sshd[6429]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=172.30.1.77  user=root
2011-04-26T18:57:09-06:00 baboon-srv sshd[6429]: Failed password for root from 172.30.1.77 port 60375 ssh2
2011-04-26T18:57:13-06:00 baboon-srv sshd[6429]: last message repeated 2 times
2011-04-26T18:57:13-06:00 baboon-srv sshd[6429]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=17
```

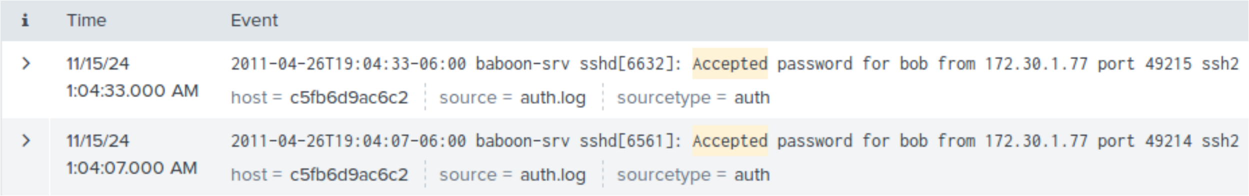
Figure 1/Root.



```
i  Time                    Event
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:05:10-06:00 baboon-srv sudo: pam_unix(sudo:auth): authentication failure; logname=bob uid=0 euid=0 tty=/dev/pts/0 ru
   1:05:10.000 AM          ser= rhost=  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:04:05-06:00 baboon-srv sshd[6561]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:04:05.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:59-06:00 baboon-srv sshd[6559]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:59.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:52-06:00 baboon-srv sshd[6557]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:52.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:44-06:00 baboon-srv sshd[6555]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:44.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:37-06:00 baboon-srv sshd[6553]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:37.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:30-06:00 baboon-srv sshd[6551]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:30.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:22-06:00 baboon-srv sshd[6549]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:22.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:15-06:00 baboon-srv sshd[6547]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:15.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:03:08-06:00 baboon-srv sshd[6545]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
   1:03:08.000 AM          rhost=172.30.1.77  user=bob
                           host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
```

Figure 2/Bob.

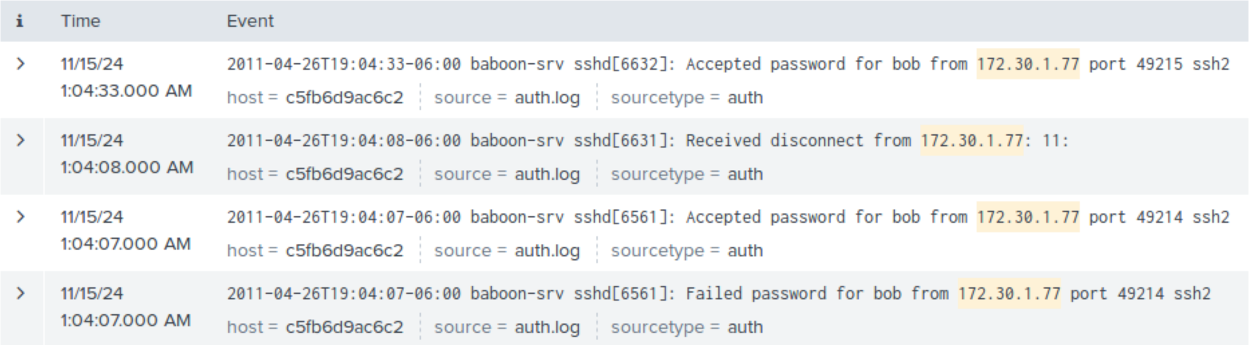
After the login attempts bob user system was compromised as shown in the figure below.



```
i  Time                    Event
>  11/15/24                2011-04-26T19:04:33-06:00 baboon-srv sshd[6632]: Accepted password for bob from 172.30.1.77 port 49215 ssh2
   1:04:33.000 AM          host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
>  11/15/24                2011-04-26T19:04:07-06:00 baboon-srv sshd[6561]: Accepted password for bob from 172.30.1.77 port 49214 ssh2
   1:04:07.000 AM          host = c5fb6d9ac6c2 | source = auth.log | sourcetype = auth
```

Figure 3/Accepted.

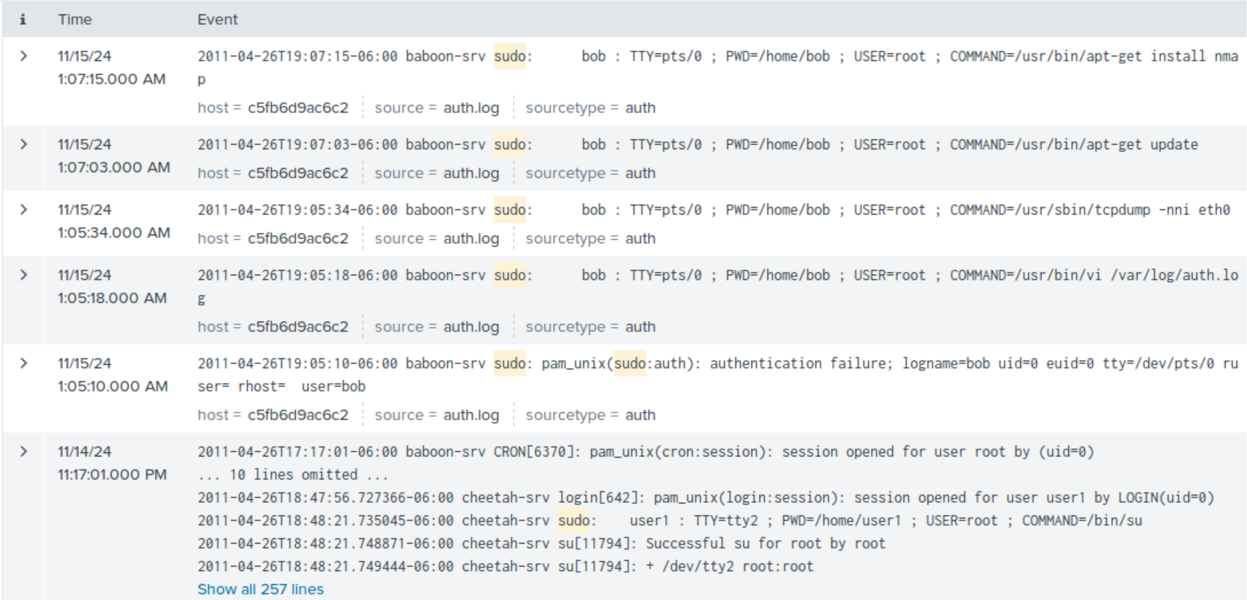
The compromise didn’t last for long before the system was disconnected. The figure below shows the activities relating to the compromise.



i	Time	Event
>	11/15/24 1:04:33.000 AM	2011-04-26T19:04:33-06:00 baboon-srv sshd[6632]: Accepted password for bob from 172.30.1.77 port 49215 ssh2 host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:04:08.000 AM	2011-04-26T19:04:08-06:00 baboon-srv sshd[6631]: Received disconnect from 172.30.1.77: 11: host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:04:07.000 AM	2011-04-26T19:04:07-06:00 baboon-srv sshd[6561]: Accepted password for bob from 172.30.1.77 port 49214 ssh2 host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:04:07.000 AM	2011-04-26T19:04:07-06:00 baboon-srv sshd[6561]: Failed password for bob from 172.30.1.77 port 49214 ssh2 host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth

Figure 4/Extent.

During the compromise period the attacker did run commands as a root user from the compromised bob user as shown in the figure below. The sudo command grants the user super user privileges and can run commands as a root user (Brian, 2004).



i	Time	Event
>	11/15/24 1:07:15.000 AM	2011-04-26T19:07:15-06:00 baboon-srv sudo: bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ; COMMAND=/usr/bin/apt-get install nma p host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:07:03.000 AM	2011-04-26T19:07:03-06:00 baboon-srv sudo: bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ; COMMAND=/usr/bin/apt-get update host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:05:34.000 AM	2011-04-26T19:05:34-06:00 baboon-srv sudo: bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ; COMMAND=/usr/sbin/tcpdump -nni eth0 host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth
>	11/15/24 1:05:18.000 AM	2011-04-26T19:05:18-06:00 baboon-srv sudo: bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ; COMMAND=/usr/bin/vi /var/log/auth.lo g host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth

>	11/15/24
	1:05:10.000 AM
		2011-04-26T19:05:10-06:00 baboon-srv sudo: pam_unix(sudo:auth): authentication failure; logname=bob uid=0 euid=0 tty=/dev/pts/0 ru
		ser= rhost= user=bob
		host = c5fb6d9ac6c2 ¦ source = auth.log ¦ sourcetype = auth

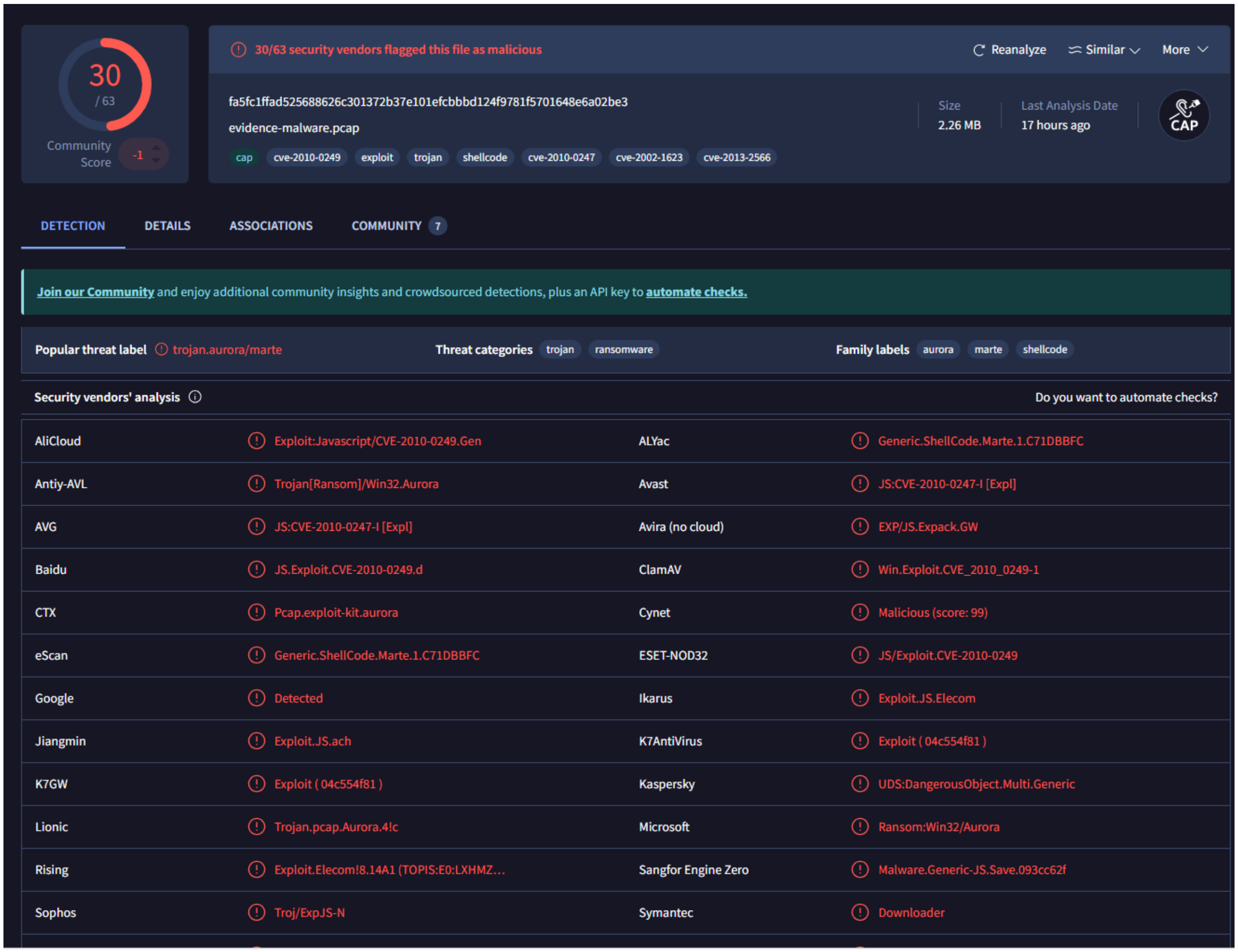
>	11/14/24 11:17:01.000 PM	2011-04-26T17:17:01-06:00 baboon-srv CRON[6370]: pam_unix(cron:session): session opened for user root by (uid=0) ... 10 lines omitted ... 2011-04-26T18:47:56.727366-06:00 cheetah-srv login[642]: pam_unix(login:session): session opened for user user1 by LOGIN(uid=0) 2011-04-26T18:48:21.735045-06:00 cheetah-srv sudo: user1 : TTY=tty2 ; PWD=/home/user1 ; USER=root ; COMMAND=/bin/su 2011-04-26T18:48:21.748871-06:00 cheetah-srv su[11794]: Successful su for root by root 2011-04-26T18:48:21.749444-06:00 cheetah-srv su[11794]: + /dev/tty2 root:root Show all 257 lines

PWD – print working directory shows the current directory that was used during when the compromise happened and the directory on which the commands were ran. Apt-get install nmap - this command was used to install nmap (network mapper) from the package database that holds the records available in a distribution. Apt-get is a package management tool used in Linux distributions. Nmap is a command line tool that is used for network exploration, security auditing and penetration testing. It is used to discover hosts and services on a network (Christopher, 2013).

Tcpdump command is used to capture network traffic on a Linux system. The above Tcpdump -nni eth0 meant -n- not to resolve hostnames and only display ip addresses. -n this prompt to show only raw port numbers and – I to capture network on the eth0 interface specified which is the ethernet network interface (Korbin, 2021).

Dynamic Malware Analysis

This section we will be investigating a packet capture containing the network traffic from a target's home network. We uploaded the file on www.virustotal.com an online tool that is used to analyze suspicious files, URL, domains and IPs detecting malware and other malicious activities and log it to the security community. The file returned the following results as shown in the screenshot below.



30/63 security vendors flagged this file as malicious

fa5fc1ffad525688626c301372b37e101efcbbbd124f9781f5701648e6a02be3

evidence-malware.pcap

cap cve-2010-0249 exploit trojan shellcode cve-2010-0247 cve-2002-1623 cve-2013-2566

Size 2.26 MB

Last Analysis Date 17 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.aurora/marte

Threat categories trojan ransomware

Family labels aurora marte shellcode

Security vendors' analysis

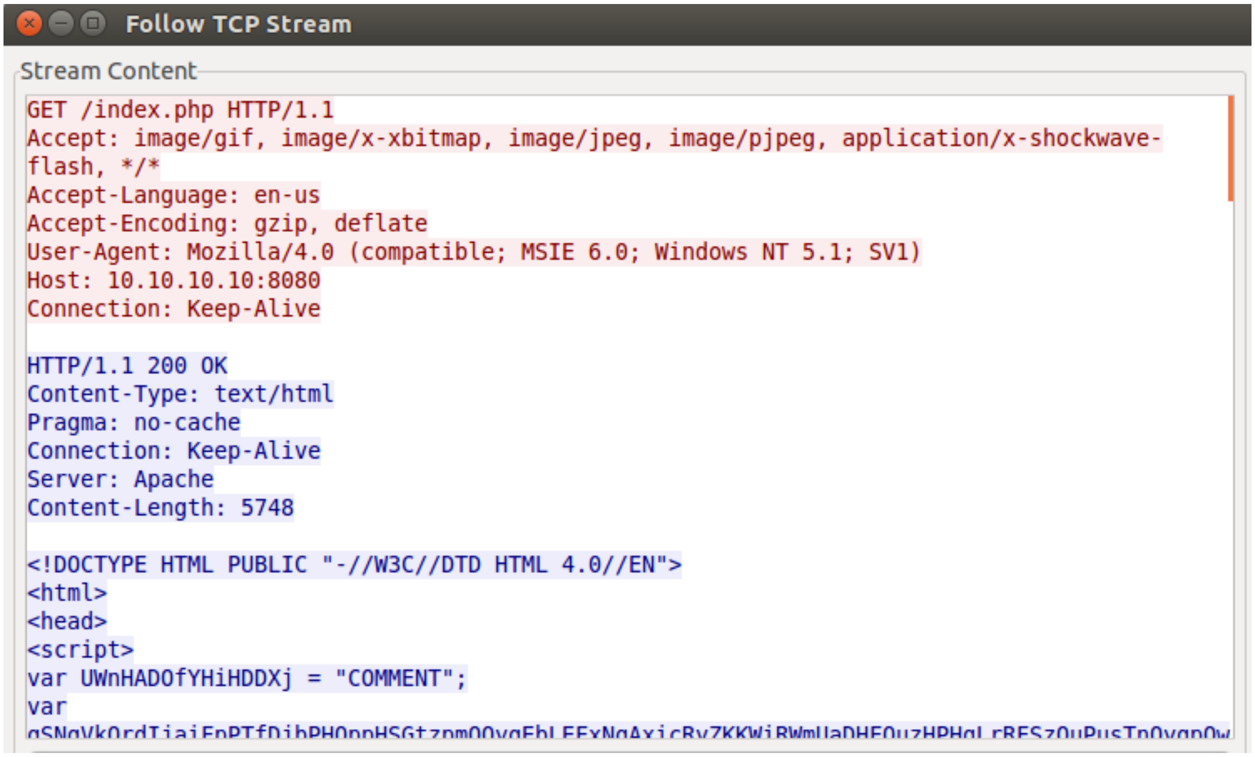
Do you want to automate checks?

AliCloud	Exploit:Javascript/CVE-2010-0249.Gen	ALYac	Generic.ShellCode.Marte.1.C71DBBFC
Antiy-AVL	Trojan[Ransom]/Win32.Aurora	Avast	JS:CVE-2010-0247-I [Expl]
AVG	JS:CVE-2010-0247-I [Expl]	Avira (no cloud)	EXP/JS.Expack.GW
Baidu	JS.Exploit.CVE-2010-0249.d	ClamAV	Win.Exploit.CVE_2010_0249-1
CTX	Pcap.exploit-kit.aurora	Cynet	Malicious (score: 99)
eScan	Generic.ShellCode.Marte.1.C71DBBFC	ESET-NOD32	JS/Exploit.CVE-2010-0249
Google	Detected	Ikarus	Exploit.JS.Elecom
Jiangmin	Exploit.JS.ach	K7AntiVirus	Exploit (04c554f81)
K7GW	Exploit (04c554f81)	Kaspersky	UDS:DangerousObject.Multi.Generic
Lionic	Trojan.pcap.Aurora.4!c	Microsoft	Ransom:Win32/Aurora
Rising	Exploit.Elecom!8.14A1 (TOPIS:E0:LXHMZ…	Sangfor Engine Zero	Malware.Generic-JS.Save.093cc62f
Sophos	Troj/ExpJS-N	Symantec	Downloader

Figure 5/Virustotal.

The file was flagged by 30 out of a total of 63 participating communities as a malicious file where Trojan aurora was found to be the popular threat label. Trojan Aurora is a type of trojan that is leveraged to steal sensitive information from the targeted computer such as login credentials, financial information and other personal data (Stelian, 2022).

We continued with the investigation that showed that the compromise originated from the server 10.10.10.10 through a GET request as shown in the figure below.



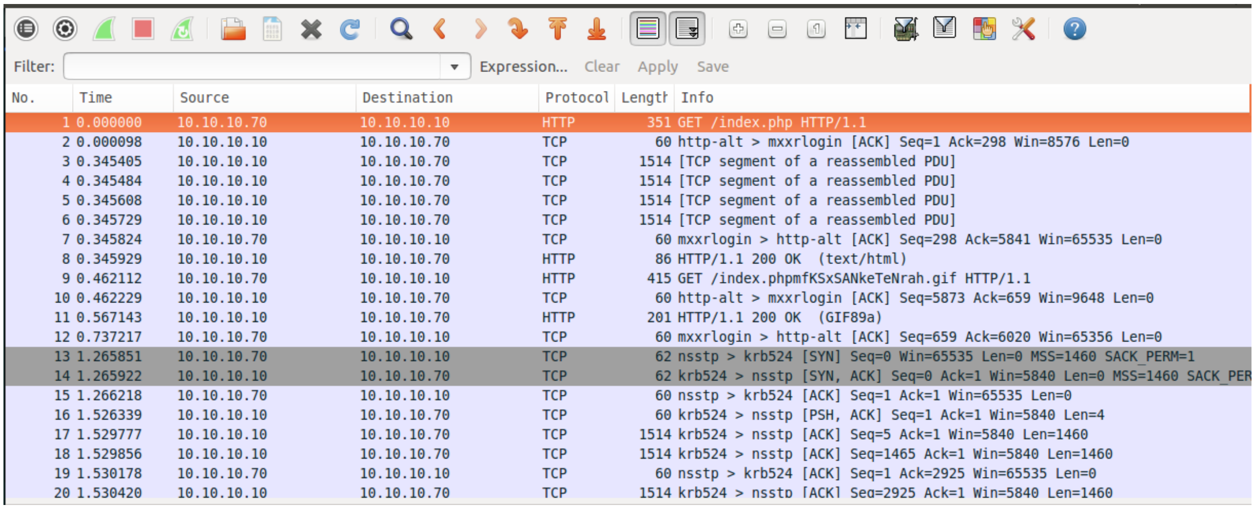
```
GET /index.php HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-
flash, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 10.10.10.10:8080
Connection: Keep-Alive

HTTP/1.1 200 OK
Content-Type: text/html
Pragma: no-cache
Connection: Keep-Alive
Server: Apache
Content-Length: 5748

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<script>
var UWnHADOfYHiHDDXj = "COMMENT";
var
qSNqVkOrdIiajFnPTfDihPHOnnHSGtznmOOvqFbLFFxNqAxicRvZKKWiRWmUaDHFOuzHPHqLrRFSzOuPusTnOvqnOw
```

Figure 6/TCP Stream

This compromise used the GET request where the user 10.10.10.70 sent a GET request to http://10.10.10.10:8080/index.php From the Wireshark screenshot below , after the response another GET request was sent for a gif file /index.phpmfKSxSANkeTeNrah.gif which after it was received a three-way TCP hand shake happened.

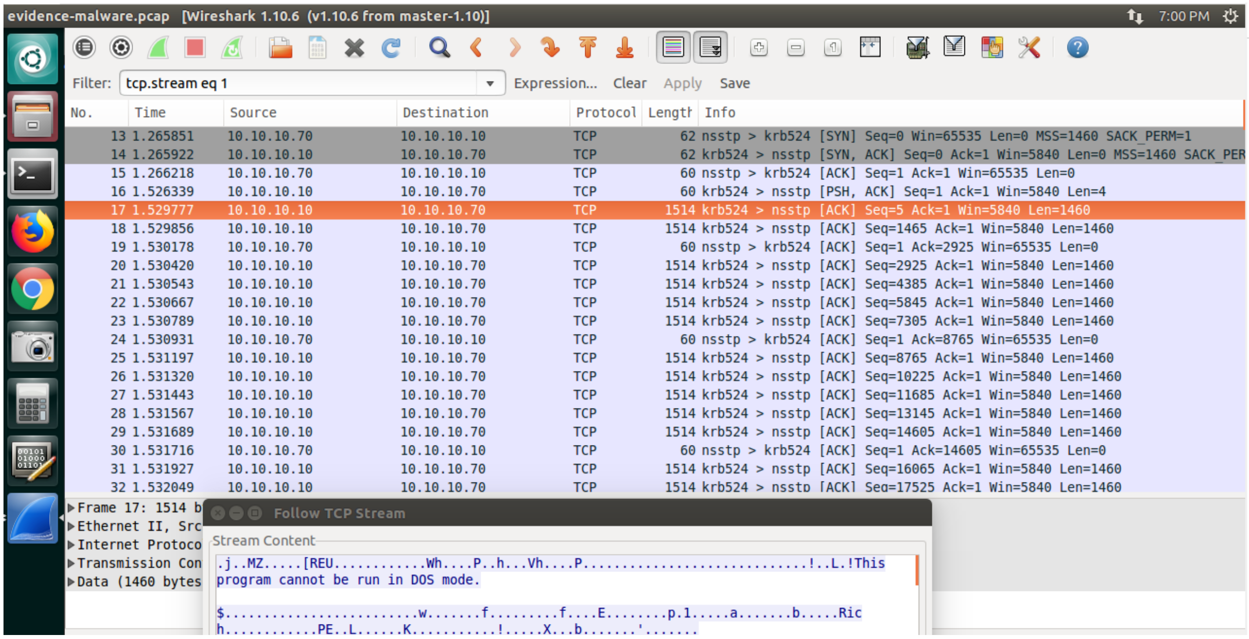


```
No.  Time       Source        Destination   Protocol Lengtł Info
   1 0.000000   10.10.10.70   10.10.10.10   HTTP      351 GET /index.php HTTP/1.1
   2 0.000098   10.10.10.10   10.10.10.70   TCP        60 http-alt > mxxrlogin [ACK] Seq=1 Ack=298 Win=8576 Len=0
   3 0.345405   10.10.10.10   10.10.10.70   TCP      1514 [TCP segment of a reassembled PDU]
   4 0.345484   10.10.10.10   10.10.10.70   TCP      1514 [TCP segment of a reassembled PDU]
   5 0.345608   10.10.10.10   10.10.10.70   TCP      1514 [TCP segment of a reassembled PDU]
   6 0.345729   10.10.10.10   10.10.10.70   TCP      1514 [TCP segment of a reassembled PDU]
   7 0.345824   10.10.10.70   10.10.10.10   TCP        60 mxxrlogin > http-alt [ACK] Seq=298 Ack=5841 Win=65535 Len=0
   8 0.345929   10.10.10.10   10.10.10.70   HTTP       86 HTTP/1.1 200 OK  (text/html)
   9 0.462112   10.10.10.70   10.10.10.10   HTTP      415 GET /index.phpmfKSxSANkeTeNrah.gif HTTP/1.1
  10 0.462229   10.10.10.10   10.10.10.70   TCP        60 http-alt > mxxrlogin [ACK] Seq=5873 Ack=659 Win=9648 Len=0
  11 0.567143   10.10.10.10   10.10.10.70   HTTP      201 HTTP/1.1 200 OK  (GIF89a)
  12 0.737217   10.10.10.70   10.10.10.10   TCP        60 mxxrlogin > http-alt [ACK] Seq=659 Ack=6020 Win=65356 Len=0
  13 1.265851   10.10.10.70   10.10.10.10   TCP        62 nsstp > krb524 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
  14 1.265922   10.10.10.10   10.10.10.70   TCP        62 krb524 > nsstp [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PER
  15 1.266218   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=1 Win=65535 Len=0
  16 1.526339   10.10.10.10   10.10.10.70   TCP        60 krb524 > nsstp [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4
  17 1.529777   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=5 Ack=1 Win=5840 Len=1460
  18 1.529856   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=1465 Ack=1 Win=5840 Len=1460
  19 1.530178   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0
  20 1.530420   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=2925 Ack=1 Win=5840 Len=1460
```

Figure 7/TCP.

The three-way handshake is a process in the TCP/IP network used to establish connection between the server and the client. This process requires the server and client to exchange synchronization and acknowledgement packets before the data communication commences. This process encompasses SYN(synchronization) which is used to initiate and establish a connection between the devices, ACK(acknowledgement) which is used to confirm that it has received the SYN, SYN-ACK which has the SYN from the local device and the ACK of the earlier packet and FIN which is used to terminate the connection (Bryce, 2024).

The next section was to recover the malware and do further analysis. We selected the ACK(Acknowledgement) and followed the TCP stream as shown in the figure below.



evidence-malware.pcap [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
7:00 PM
Filter: tcp.stream eq 1
Expression... Clear Apply Save

```
No.   Time       Source        Destination   Protocol Length Info
   13 1.265851   10.10.10.70   10.10.10.10   TCP        62 nsstp > krb524 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
   14 1.265922   10.10.10.10   10.10.10.70   TCP        62 krb524 > nsstp [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PER
   15 1.266218   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=1 Win=65535 Len=0
   16 1.526339   10.10.10.10   10.10.10.70   TCP        60 krb524 > nsstp [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4
   17 1.529777   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=5 Ack=1 Win=5840 Len=1460
   18 1.529856   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=1465 Ack=1 Win=5840 Len=1460
   19 1.530178   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=2925 Win=65535 Len=0
   20 1.530420   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=2925 Ack=1 Win=5840 Len=1460
   21 1.530543   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=4385 Ack=1 Win=5840 Len=1460
   22 1.530667   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=5845 Ack=1 Win=5840 Len=1460
   23 1.530789   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=7305 Ack=1 Win=5840 Len=1460
   24 1.530931   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=8765 Win=65535 Len=0
   25 1.531197   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=8765 Ack=1 Win=5840 Len=1460
   26 1.531320   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=10225 Ack=1 Win=5840 Len=1460
   27 1.531443   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=11685 Ack=1 Win=5840 Len=1460
   28 1.531567   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=13145 Ack=1 Win=5840 Len=1460
   29 1.531689   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=14605 Ack=1 Win=5840 Len=1460
   30 1.531716   10.10.10.70   10.10.10.10   TCP        60 nsstp > krb524 [ACK] Seq=1 Ack=14605 Win=65535 Len=0
   31 1.531927   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=16065 Ack=1 Win=5840 Len=1460
   32 1.532049   10.10.10.10   10.10.10.70   TCP      1514 krb524 > nsstp [ACK] Seq=17525 Ack=1 Win=5840 Len=1460
```

```
▶Frame 17: 1514 b
▶Ethernet II, Src
▶Internet Protoco
▶Transmission Con
▶Data (1460 bytes
```

Follow TCP Stream

Stream Content

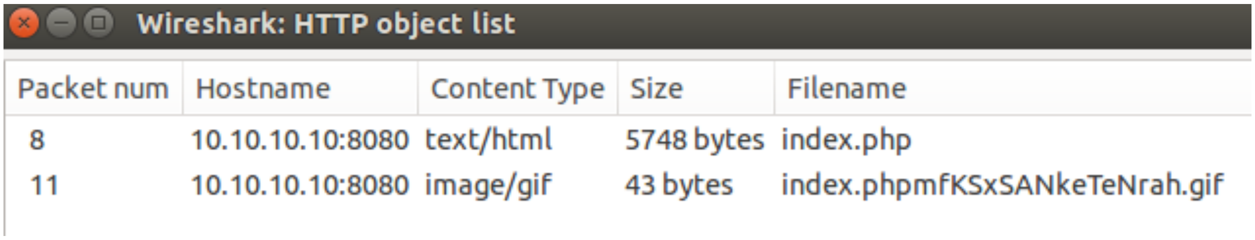
```
.j..MZ.....[REU...........Wh....P..h...Vh....P............................!..L.!This
program cannot be run in DOS mode.

$.......................w.......f.........f....E........p.1.....a.......b.....Ric
h...........PE..L......K..........!.....X...b.......'.......
```

Figure 8/Stream.

The MZ header stands for the initials of Microsoft engineer Mark Zbikowski and this file format was designed as a relocatable executable running on real mode. This executable can be used so that a single executable can provide 2 ports of the same application (Mediawiki, 2024).

We then extracted the two files that engaged in the prior steps which were index.php and index.phpmfKSxSANkeTeNrah.gif.

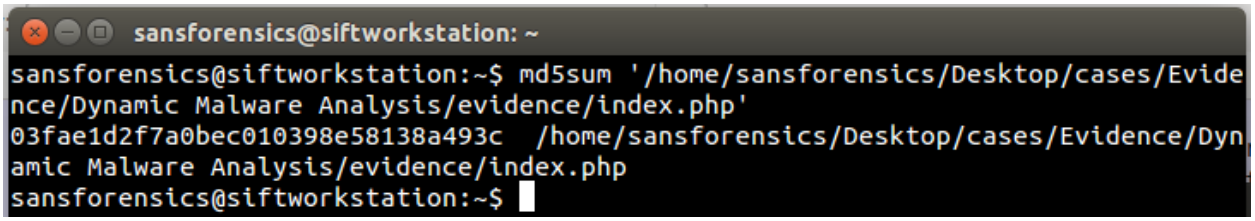


Wireshark: HTTP object list

Packet num	Hostname	Content Type	Size	Filename
8	10.10.10.10:8080	text/html	5748 bytes	index.php
11	10.10.10.10:8080	image/gif	43 bytes	index.phpmfKSxSANkeTeNrah.gif

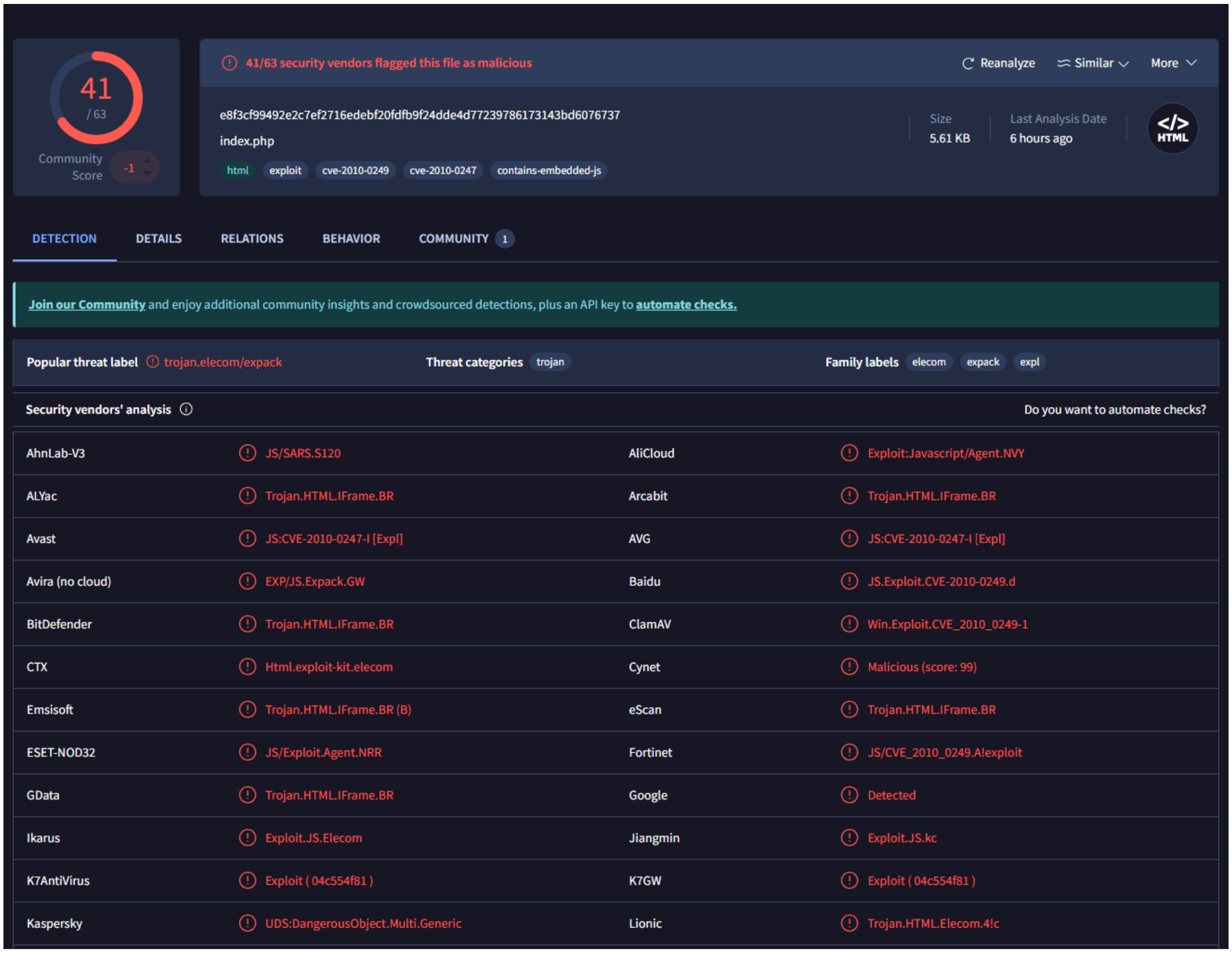
Figure 9/Files.

After extracting we ran a md5sum on the files to get the md5 values and then uploaded the file to virustotal as shown and discussed below.



```
sansforensics@siftworkstation:~$ md5sum '/home/sansforensics/Desktop/cases/Evide
nce/Dynamic Malware Analysis/evidence/index.php'
03fae1d2f7a0bec010398e58138a493c  /home/sansforensics/Desktop/cases/Evidence/Dyn
amic Malware Analysis/evidence/index.php
sansforensics@siftworkstation:~$ 
```

Figure 10/md51.



41/63 security vendors flagged this file as malicious

e8f3cf99492e2c7ef2716edebf20fdfb9f24dde4d77239786173143bd6076737

index.php

html exploit cve-2010-0249 cve-2010-0247 contains-embedded-js

Size 5.61 KB

Last Analysis Date 6 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.elecom/expack

Threat categories trojan

Family labels elecom expack expl

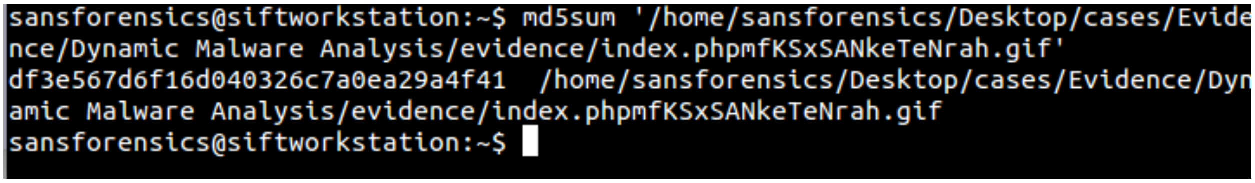
Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	JS/SARS.S120	AliCloud	Exploit:Javascript/Agent.NVY
ALYac	Trojan.HTML.IFrame.BR	Arcabit	Trojan.HTML.IFrame.BR
Avast	JS:CVE-2010-0247-I [Expl]	AVG	JS:CVE-2010-0247-I [Expl]
Avira (no cloud)	EXP/JS.Expack.GW	Baidu	JS.Exploit.CVE-2010-0249.d
BitDefender	Trojan.HTML.IFrame.BR	ClamAV	Win.Exploit.CVE_2010_0249-1
CTX	Html.exploit-kit.elecom	Cynet	Malicious (score: 99)
Emsisoft	Trojan.HTML.IFrame.BR (B)	eScan	Trojan.HTML.IFrame.BR
ESET-NOD32	JS/Exploit.Agent.NRR	Fortinet	JS/CVE_2010_0249.A!exploit
GData	Trojan.HTML.IFrame.BR	Google	Detected
Ikarus	Exploit.JS.Elecom	Jiangmin	Exploit.JS.kc
K7AntiVirus	Exploit (04c554f81)	K7GW	Exploit (04c554f81)
Kaspersky	UDS:DangerousObject.Multi.Generic	Lionic	Trojan.HTML.Elecom.4!c

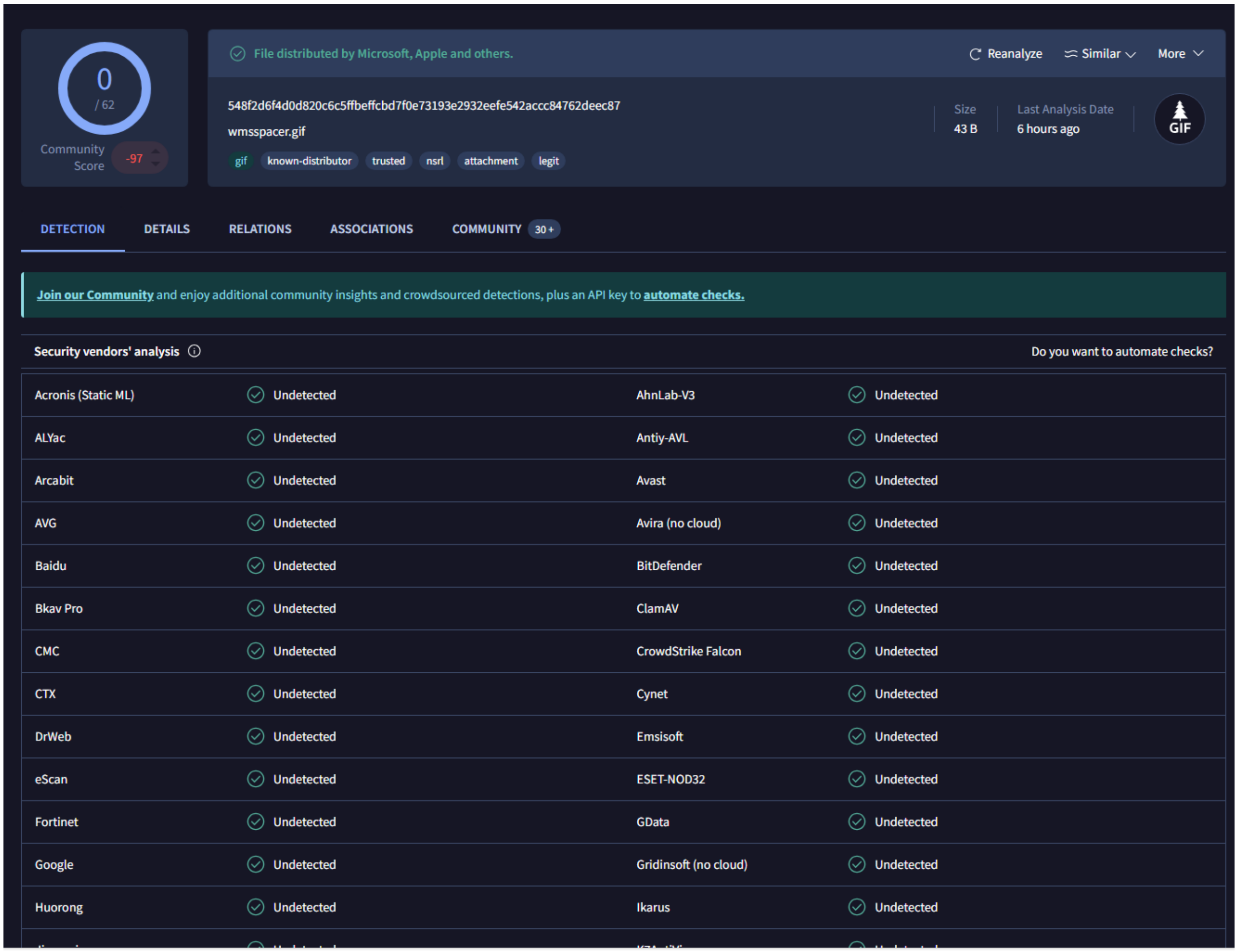
Figure 11/php1.

The index.php returned a malicious file according to virustotal being flagged by 41 community participants with the popular threat being trojan.elecom/exepack. This trojan type was related to elecom software that exploited the camera assistant and quickfiledealer software that allowed adversaries to gain elevated privileges on affected systems. The exepack section defines a program that is used to compress and decompress executables, hence a format for self-extracting executables (clouddefense, 2023).



```
sansforensics@siftworkstation:~$ md5sum '/home/sansforensics/Desktop/cases/Evide
nce/Dynamic Malware Analysis/evidence/index.phpmfKSxSANkeTeNrah.gif'
df3e567d6f16d040326c7a0ea29a4f41  /home/sansforensics/Desktop/cases/Evidence/Dyn
amic Malware Analysis/evidence/index.phpmfKSxSANkeTeNrah.gif
sansforensics@siftworkstation:~$ 
```

Figure 12/md5sum2.



File distributed by Microsoft, Apple and others.

548f2d6f4d0d820c6c5ffbeffcbd7f0e73193e2932eefe542accc84762deec87

wmsspacer.gif

gif known-distributor trusted nsrl attachment legit

Size 43 B

Last Analysis Date 6 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	Undetected	AhnLab-V3	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
Arcabit	Undetected	Avast	Undetected
AVG	Undetected	Avira (no cloud)	Undetected
Baidu	Undetected	BitDefender	Undetected
Bkav Pro	Undetected	ClamAV	Undetected
CMC	Undetected	CrowdStrike Falcon	Undetected
CTX	Undetected	Cynet	Undetected
DrWeb	Undetected	Emsisoft	Undetected
eScan	Undetected	ESET-NOD32	Undetected
Fortinet	Undetected	GData	Undetected
Google	Undetected	Gridinsoft (no cloud)	Undetected
Huorong	Undetected	Ikarus	Undetected

Figure 13/giffile.

This file was not flagged by any community sources therefore it didn't have any malicious programs.

References

Brian, W. (2004). *How Linux Works : What every superuser should know.* San Francisco: no starch press Inc.

Bryce, L. (2024, June 27). *TCP 3-Way Handshake (SYN, SYN-ACK,ACK)*. Retrieved from guru99: https://www.guru99.com/tcp-3-way-handshake.html

Christopher, N. (2013). *Ubuntu Linux Toolbox: 1000+ commands for power Users.* Indianapolis: John Wiley & Sons Inc.

clouddefense. (2023, October 10). *CVE-2023-22368 : Security Advisory and Response*. Retrieved from Clouddefense.ai: https://www.clouddefense.ai/cve/2023/CVE-2023-22368

Korbin, B. (2021, March 30). *How to use tcpdump command on linux*. Retrieved from Linuxconfig: https://linuxconfig.org/how-to-use-tcpdump-command-on-linux

Mediawiki. (2024, February 28). *MZ*. Retrieved from Osdev.org: https://wiki.osdev.org/MZ#MZ_File_Structure

Stelian, P. (2022, December 26). *How To Remove Trojan Aurora [Virus Removal Guide]*. Retrieved from Malware Tips: https://malwaretips.com/blogs/remove-trojan-aurora/