Suspicious Account Creation

Key Takeaways:

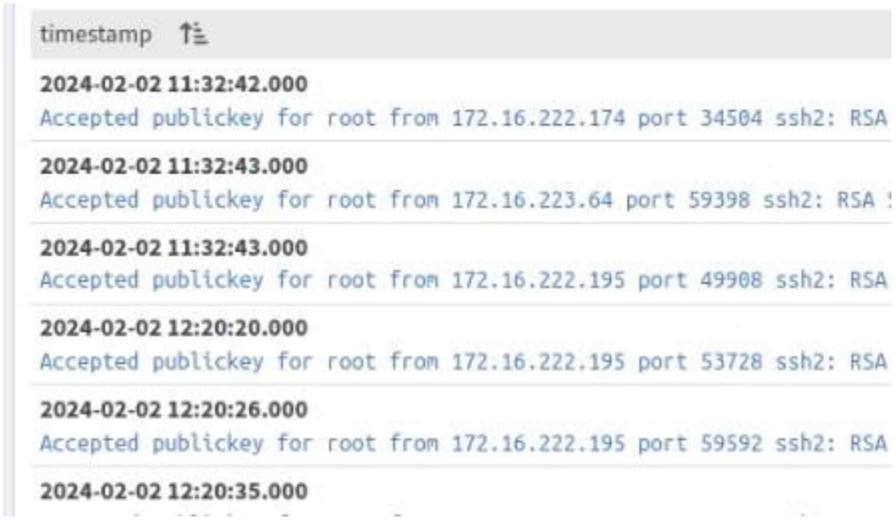
- February 2nd started a series of simultaneous attacks were made on our Linux based systems where a threat user had logged into the root account of these newly created systems using a SHA256 public key.
- Within 10 minutes of the attack, 8 new users were created on the system, with the account "webclient" being elevated sudo privileges.
- This attack was performed using a backdoor created by the previous IT employees to install persistence onto our system using legacy-ansible invoked commands to constantly copy files from the root directory.
- The actors behind this attack are the famous hacking group Kaiju

Initial Access:

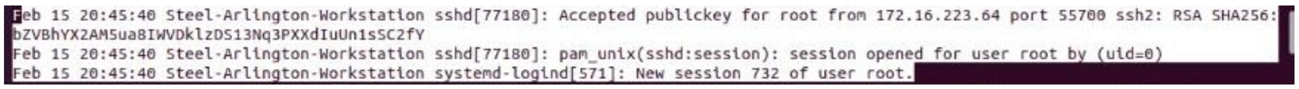
The attack initially began on our Dallas mining system. Initial access on all systems was facilitated through the root account. The IP's used were different depending on the system. A list of the IP's used are listed below:

- 172.16.222.174
- 172.16.223.64
- 172.16.222.195
- 172.16.222.191

Logs from auth.log show these connections were accepted using publickey for root allowing the threat user to bypass the password requirement, opening a new session as root.



```
timestamp
2024-02-02 11:32:42.000
Accepted publickey for root from 172.16.222.174 port 34504 ssh2: RSA
2024-02-02 11:32:43.000
Accepted publickey for root from 172.16.223.64 port 59398 ssh2: RSA
2024-02-02 11:32:43.000
Accepted publickey for root from 172.16.222.195 port 49908 ssh2: RSA
2024-02-02 12:20:20.000
Accepted publickey for root from 172.16.222.195 port 53728 ssh2: RSA
2024-02-02 12:20:26.000
Accepted publickey for root from 172.16.222.195 port 59592 ssh2: RSA
2024-02-02 12:20:35.000
```



```
Feb 15 20:45:40 Steel-Arlington-Workstation sshd[77180]: Accepted publickey for root from 172.16.223.64 port 55700 ssh2: RSA SHA256:
bZVBhYX2AM5uaBIWVDklzDS13Nq3PXXdIuUn1sSC2fY
Feb 15 20:45:40 Steel-Arlington-Workstation sshd[77180]: pam_unix(sshd:session): session opened for user root by (uid=0)
Feb 15 20:45:40 Steel-Arlington-Workstation systemd-logind[571]: New session 732 of user root.
```

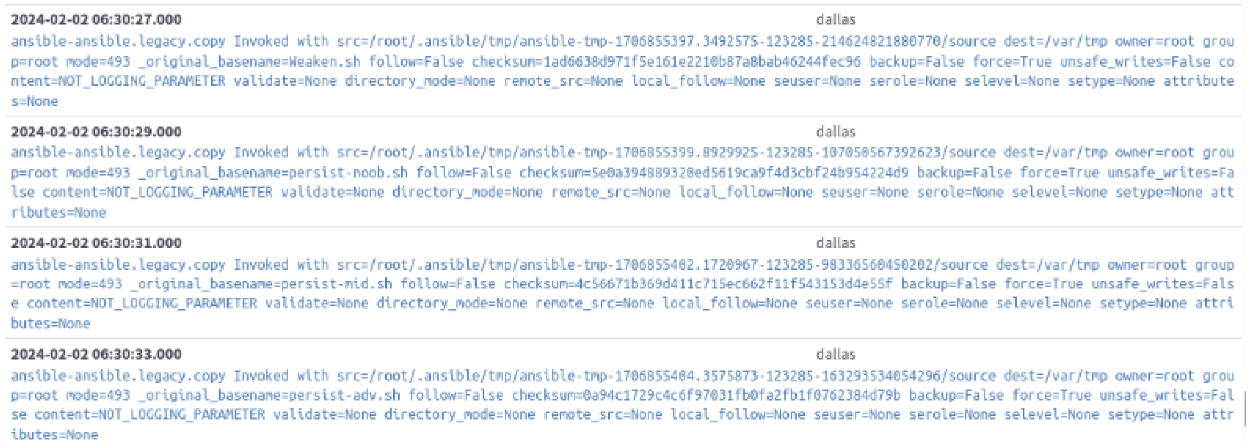
Since the attacks on the systems were identical, we will be using evidence from our dallas machine to outline the attack for consistency.

Execution:

This attack was executed using ansible-legacy to issue commands and copy system files to -and from their system. Specifically, there there were 3 commands that stood out:

- Ansible-legacy.file – Used for a multitude of actions, including creating/editing/deleting files from a given source
- Ansible-legacy.copy – Used to copy files to destinations on and outside the system
- Ansible-legacy.command – Used to issue commands on the system

The following listed are some examples of the commands being performed:



```
2024-02-02 06:30:27.000                                                        dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855397.3492575-123285-214624821880770/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=Weaken.sh follow=False checksum=1ad6638d971f5e161e2210b87a8bab46244fec96 backup=False force=True unsafe_writes=False co
ntent=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attribute
s=None
2024-02-02 06:30:29.000                                                        dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855399.8929925-123285-107050567392623/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=persist-noob.sh follow=False checksum=5e0a394889320ed5619ca9f4d3cbf24b954224d9 backup=False force=True unsafe_writes=Fa
lse content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None att
ributes=None
2024-02-02 06:30:31.000                                                        dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855402.1720967-123285-98336560450202/source dest=/var/tmp owner=root group
=root mode=493 _original_basename=persist-mid.sh follow=False checksum=4c56671b369d411c715ec662f11f543153d4e55f backup=False force=True unsafe_writes=Fals
e content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attri
butes=None
2024-02-02 06:30:33.000                                                        dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855404.3575873-123285-163293534054296/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=persist-adv.sh follow=False checksum=0a94c1729c4c6f97031fb0fa2fb1f0762384d79b backup=False force=True unsafe_writes=Fal
se content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attr
ibutes=None
```

While the use of ansible legacy blocks out most of the commands, it seems the threat actor was creating and copying files from the root directory (talked about more in the discovery section). After creating files, the user invokes commands to create elevated users for extra persistence within the system (talked about more in the persistence section). Finally, the users log out of the system as root and begin to perform what seems to be log in attempts, where they periodically used the root public key to copy our root directory files.

Persistence:

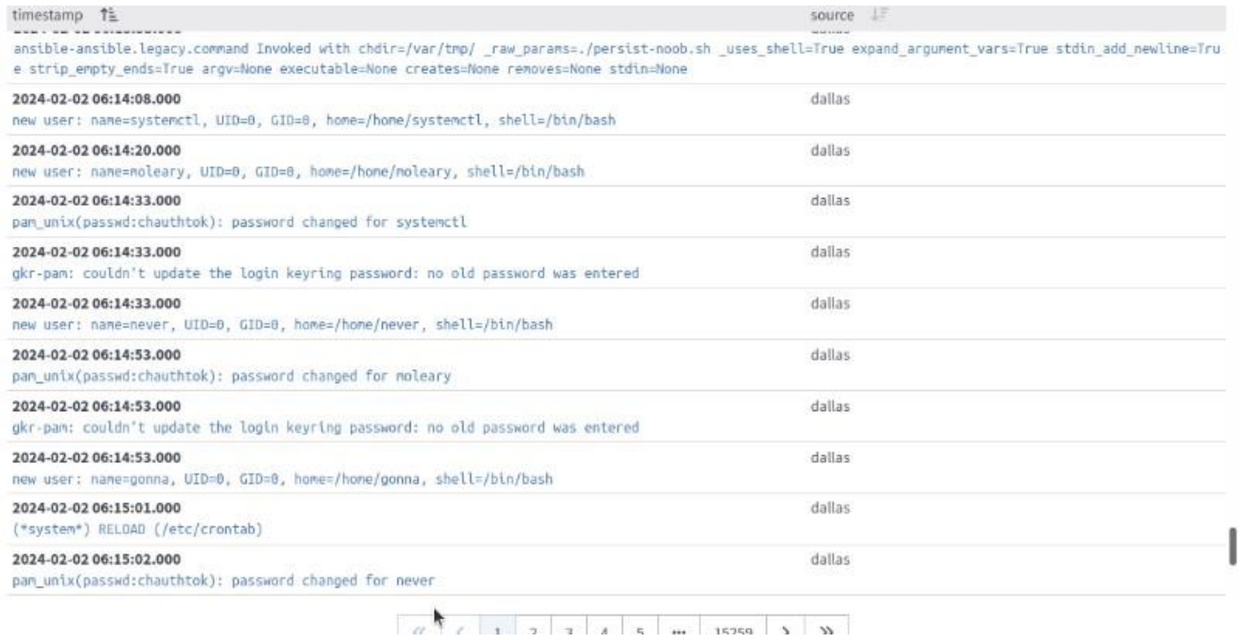
To maintain access within the system, the threat user used a stored public key found in roots authorized keys and created multiple accounts, with one stand out being "webclient".

Regarding the public key, this was both the initial access and persistence used to gain access within the system. Referencing the recent attack on our Arlington system, evidence shows that no previous attempts were made to access or modify the “authorized_keys” file for user root, suggesting this was a back door planted in advance before we had access to the systems.

The threat user also created several accounts using “ansible-ansible.legacy.command” with ansible legacy to issue the creation of the users listed below:

- Never
- Gonna
- Give
- You
- Uppp
- Systemctl
- Oleary

All accounts were set to have UID and GID of 0 (root) and have all their passwords changed.



```
timestamp                                                                       source
ansible-ansible.legacy.command Invoked with chdir=/var/tmp/ _raw_params=./persist-noob.sh _uses_shell=True expand_argument_vars=True stdin_add_newline=Tru
e strip_empty_ends=True argv=None executable=None creates=None removes=None stdin=None
2024-02-02 06:14:08.000                                                         dallas
new user: name=systemctl, UID=0, GID=0, home=/home/systemctl, shell=/bin/bash
2024-02-02 06:14:20.000                                                         dallas
new user: name=moleary, UID=0, GID=0, home=/home/moleary, shell=/bin/bash
2024-02-02 06:14:33.000                                                         dallas
pam_unix(passwd:chauthtok): password changed for systemctl
2024-02-02 06:14:33.000                                                         dallas
gkr-pam: couldn't update the login keyring password: no old password was entered
2024-02-02 06:14:33.000                                                         dallas
new user: name=never, UID=0, GID=0, home=/home/never, shell=/bin/bash
2024-02-02 06:14:53.000                                                         dallas
pam_unix(passwd:chauthtok): password changed for moleary
2024-02-02 06:14:53.000                                                         dallas
gkr-pam: couldn't update the login keyring password: no old password was entered
2024-02-02 06:14:53.000                                                         dallas
new user: name=gonna, UID=0, GID=0, home=/home/gonna, shell=/bin/bash
2024-02-02 06:15:01.000                                                         dallas
(*system*) RELOAD (/etc/crontab)
2024-02-02 06:15:02.000                                                         dallas
pam_unix(passwd:chauthtok): password changed for never
```

The threat user also created an account called “webclient”, that was given its own unique UID and added to a newly created group “webclient” (a group created by the threat user) and sudo, giving this account elevated privileges. To our knowledge searching the logs, these accounts were never access via SSH after their creation.

<u>Defense Evasion:</u>

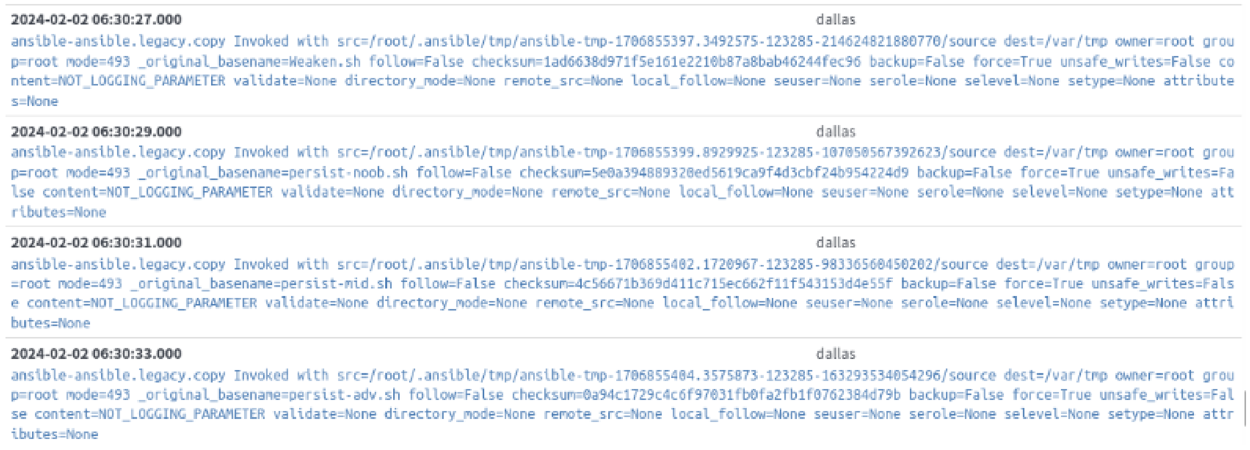
Two defenses were found to be used for evasion. In all incident cases, a four-hour window of logs were found to be missing on all local logs to hide traces of their execution. Most of the evidence brought forth has been shown from logs being forwarded to our graylog server live.

Using ansible-legacy, logs do not show what commands were run, having us to rely on the follow up executions in logs to get a better understanding of their attack. For example, if ansible was used to invoke a command, the command isn't labeled within the log, requiring the use of following logs to get a better understanding. However, some commands like "file" can create, edit, or delete a given file in a directory, making it hard to understand exactly what the threat actor was performing.

<u>Discovery:</u>

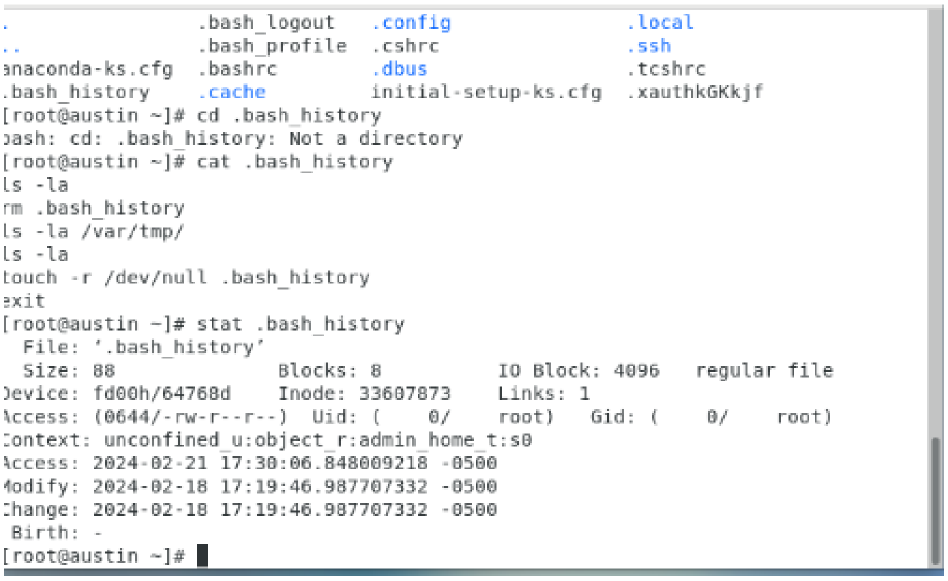
After logging in as root, the threat actor ran a command that is very common to follow up attacks, "whoami".

Following the initial attack, multiple cases of logging out and back into root can be found followed by the execution of multiple ansible-legacy commands, likely using an automated script on their end to ensure their persistence is still there and copy sensitive information off our system. This repeated behavior would use ansible to copy files from the root directory.



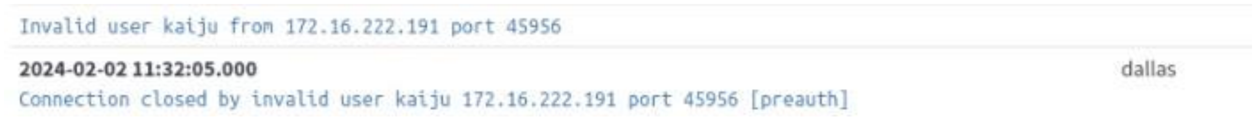
```
2024-02-02 06:30:27.000                                                    dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855397.3492575-123285-214624821880770/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=Weaken.sh follow=False checksum=1ad6638d971f5e161e2210b87a8bab46244fec96 backup=False force=True unsafe_writes=False co
ntent=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attribute
s=None
2024-02-02 06:30:29.000                                                    dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855399.8929925-123285-107050567392623/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=persist-noob.sh follow=False checksum=5e0a394889320ed5619ca9f4d3cbf24b954224d9 backup=False force=True unsafe_writes=Fa
lse content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None att
ributes=None
2024-02-02 06:30:31.000                                                    dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855402.1720967-123285-98336560450202/source dest=/var/tmp owner=root group
=root mode=493 _original_basename=persist-mid.sh follow=False checksum=4c56671b369d411c715ec662f11f543153d4e55f backup=False force=True unsafe_writes=Fals
e content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attri
butes=None
2024-02-02 06:30:33.000                                                    dallas
ansible-ansible.legacy.copy Invoked with src=/root/.ansible/tmp/ansible-tmp-1706855404.3575873-123285-163293534054296/source dest=/var/tmp owner=root grou
p=root mode=493 _original_basename=persist-adv.sh follow=False checksum=0a94c1729c4c6f97031fb0fa2fb1f0762384d79b backup=False force=True unsafe_writes=Fal
se content=NOT_LOGGING_PARAMETER validate=None directory_mode=None remote_src=None local_follow=None seuser=None serole=None selevel=None setype=None attr
ibutes=None
```

upon investigating the root directory, we can see suspicious files ".tcshrc" and ".xauthkGKkjf". .bash_history was also found to be manipulated to the time in which the file was modified was changed with the file /dev/null.



```
.                .bash_logout   .config               .local
..               .bash_profile  .cshrc                .ssh
anaconda-ks.cfg  .bashrc        .dbus                 .tcshrc
.bash_history    .cache         initial-setup-ks.cfg  .xauthkGKkjf
[root@austin ~]# cd .bash_history
bash: cd: .bash_history: Not a directory
[root@austin ~]# cat .bash_history
ls -la
rm .bash_history
ls -la /var/tmp/
ls -la
touch -r /dev/null .bash_history
exit
[root@austin ~]# stat .bash_history
  File: '.bash_history'
  Size: 88              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 33607873    Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:admin_home_t:s0
Access: 2024-02-21 17:30:06.848009218 -0500
Modify: 2024-02-18 17:19:46.987707332 -0500
Change: 2024-02-18 17:19:46.987707332 -0500
 Birth: -
[root@austin ~]#
```

Upon investigating logs before the initial attack, signs were shown of someone attempting to log in as the user "kaiju" from the same IP address used in the initial attack to log in as root. Kaiju is a famously known hacker group, and the culprit we believe to be behind this attack.



```
Invalid user kaiju from 172.16.222.191 port 45956
2024-02-02 11:32:05.000                                          dallas
Connection closed by invalid user kaiju 172.16.222.191 port 45956 [preauth]
```

When investigating the authorized_key files in root, we noticed the files had not been modified since 2022. With no evidence of elevated privileged users having logged in before the attack, we have concluded the threat actors' "Kaiju" used a back door put into place by the previously fired IT workers.

Command and Control:

After successfully completing the attack on our dallas machine, identical attacks can be seen performed simultaneously on the rest of our linux systems, with the same user accounts being found.

Due to our mining domain being composed of all Linux based systems, this has allowed the threat actor to successfully compromise our entire Mining Domain network.

Response:

Actions are being taken to find and remove all copies of this public key from our root accounts along with all other accounts created by the threat actor. SSH is also being configured to ensure root cannot be ssh’d onto the system. Files found within the root directory are being deleted and quarantined to investigate what data the attackers were attempting to steal.

We urge all companies to be weary upon seeing the public keys listed in the initial attack.