While AC-Hunter generally performs well, there are circumstances where it's not able to keep up with the flow of incoming data. If you're seeing any of the following in your AC-Hunter system, we encourage you to read on to find ways to speed it up:

- The timestamps next to the rolling databases are more than a few hours old, and as AC-Hunter stays up longer, they get more and more out of date.
- The Web user interface is sluggish; when you click on something it takes 10s of seconds to perhaps minutes to switch to the new display.
- The system load (visible on the AC-Hunter system command line with "uptime") numbers are all beyond 200 and stay there.

The suggestions below are things to check and things to adjust to help AC-Hunter run faster. They are largely independent of each other, so you may end up using more than one approach to resolve the performance issue.

As always, if you need help with any of these, please feel free to contact support at https://portal.activecountermeasures.com/support/ . We welcome additions and corrections.

Zeek/Capture system

1. Implement a BPF at the Zeek sensor to strip out high volume traffic that you will not inspect.
    a. Is this needed? If you know you have high volumes of packets and/or bytes between pairs of (trusted) systems, you can instruct Zeek to totally ignore that traffic by using a filter called a BPF. The following blog covers the concepts and how to implement it. Note that any traffic you filter out will be permanently removed from the Zeek logs (as opposed to safelisting which temporarily removes it from view and is reversible).
    b. How to: See https://www.activecountermeasures.com/filtering-out-high-volume-traffic/
    c. If you're using a traffic aggregator to provide the packets to the Zeek sensor you may be able to instruct it to remove certain types of traffic so those packets don't even cross the wire to the Zeek sensor's capture port. Please see your traffic aggregator's documentation to see if this is possible.

2. Use Ethtool to discard packets coming in on an interface. This is similar to using a BPF, but ethtool discards the traffic entirely; BPF discards the traffic for just the sniffer that uses it.
    a. Is this needed? You'd choose this approach if you have multiple sniffers and wish to discard traffic to all of them unconditionally.
    b. How to: Here's an example command that discards all traffic coming into 1.2.3.0/21 with a destination port of 3128/tcp. Note that the value after "m" is the subnet mask *in Cisco wildcard notation*. To turn a subnet mask like 255.255.248.0 into a Cisco wildcard, subtract each number from 255 to get 0.0.7.255:

```
ethtool -N eth1 flow-type tcp4 dst-ip 1.2.3.0 m 0.0.7.255 dst-port 3128 action -1
ethtool -N eth1 flow-type tcp4 src-ip 1.2.3.0 m 0.0.7.255 src-port 3128 action -1
```

3. Implement a BPF at the Zeek sensor to strip out all internal->internal traffic that AC-Hunter does not need.
    a. Is this needed? This removes all traffic between two systems with internal/reserved/rfc1918 IP addresses as AC-Hunter doesn't need to inspect it. The following blog covers the concepts and how to implement it. Note that any traffic you filter out will be permanently removed from the Zeek logs (as opposed to safelisting which temporarily removes it from view and is reversible). Make sure you continue to inspect any traffic to internal DNS servers and/or proxies; see the references to "port 53" in the blog to see how.
    b. How to: See https://www.activecountermeasures.com/filtering-out-high-volume-traffic/

4. If none of the above is possible, strip out unneeded traffic types from the Zeek logs.
    a. Is this needed? Justin Azoff from Corelight covers a wide range of ways to remove or reroute entire lines (or even individual fields) from the Zeek logs in https://www.youtube.com/watch?v=6d-yDDnxibM . It's good background material for what you may choose to do; worth watching even if you don't end up using Zeek scripts.
    b. How to, using a Zeek plugin to drop connections: Go to https://github.com/JustinAzoff/zeek-log-filtering for the Zeek scripts used in Justin's talk.
    c. How to, using pare-zeek.sh: pare-zeek.sh is simpler in that it will only *remove* lines from the Zeek logs based on criteria you specify. It pulls the raw logs in from one tree of files and places the (filtered) Zeek logs in a second directory tree. It's far less flexible and powerful than the Zeek scripts above but is simpler to set up. Contact support@activecountermeasures.com if you'd like to give this a try.

5. Make sure the network subnet list matches what is on your internal systems. Please check "InternalSubnets" in /etc/AC-Hunter/rita.yaml .

   a. Is this needed? If you only use "reserved" addresses, also called "internal" or "rfc1918" addresses for your internal systems, you're all set - Zeek is set up by default to treat these as your "internal" systems. If you use some public IP addresses (something other than address that start with 10, 192.168, 172.16-172.31, or fe80:) inside your network, you should update the list of addresses stored in rita.yaml
   b. How to: See https://portal.activecountermeasures.com/support/faq/?Display_FAQ=852

AC-Hunter system hardware

6. If your system uses (rotating media) hard disks, replace them with SSDs.
   a. Is this needed? MongoDB makes a huge amount of disk reads and writes. On a hard drive, the seek times needed to move the heads drastically slow the system. SSDs do not suffer from this.
   b. How to: The most straightforward way to do this is to rebuild the system using SSDs for all storage.
   c. Note: this is not solely a recommendation for AC-Hunter hosted on your own hardware. Cloud providers have to decide whether to use rotating media hard drives or SSDs as well; check the cloud provider's documentation to see which they use, and migrate to a cloud instance that uses SSDs if you're currently on hard drives.

7. If available ram is low, add memory. This is especially important if the system is swapping heavily. To check, install and run atop and look at the si/so fields; these should be small, not constantly at high levels. As a second check, run top and look at combined buff/cache and available memory - these should be at least 5GB; add ram if not. Consider moving up to 64GB or 128GB if you're monitoring high hundreds or thousands of systems.
   a. Is this needed? It's a little tricky to tell if your system is low on memory as the Linux kernel allocates everything that's not in active use as disk cache. While running the "top" program, look to the right of the top 5 lines for a number to the left of "buff/cache"; if this number is less than 3000 (less than 3GB available for caching), you should add memory to the system.
   b. How to: On a physical system, add more physical ram chips. On a virtual machine/cloud server, shut the system down gracefully, resize to a model with more memory, and restart it.

8. Add processor cores. 8, 16 or more would be worth trying.
   a. Is this needed? If your load average (the last three numbers in the output of "uptime") are all over 200, it's worth considering adding more processors.

b. How to: On a physical system, add more processor cores if your motherboard supports doing so. On a virtual machine/cloud server, shut the system down gracefully, resize to a model with more processors/cores, and restart it.

9. If you're running AC-Hunter on a virtual machine or cloud server, make sure it has *dedicated disks* - ones that aren't shared with other users. Alternatively, make sure they have guaranteed IOPS (IOs per second).
    a. Is this needed? In both cloud and virtual machine environments, there's always a risk that your system will be on a host with other systems that hog processor time, memory, disk bandwidth, and/or network bandwidth. With AC-Hunter, the competition over processor time and disk bandwidth are most likely to cause slowdowns for the Mongo database software.
    b. How to: If you use a virtual machine package, see if it's possible to place the full AC-Hunter system (or just the filesystem containing /var/lib/docker) on a separate drive that's not used by any other virtual machine. In a cloud environment, see your provider's documentation or their tech support to see if it's possible to get dedicated disks for AC-Hunter.

10. Consider using a raid array for mongo's databases (located under /var/lib/docker/).
    a. Is this needed? Mongo is heavily limited by disk access; giving mongo storage on a raid array may reduce one slowdown.
    b. How to: This will likely require a reinstall as switching from a single SSD to a raid array is not generally available on a live system.
    c. Note: RAID levels 0, 1, 10, 0+1, 1+0 are likely to provide a performance boost. RAID 0 does not include any redundancy, so RAID 10 would be a safer choice as it includes redundancy. RAID 2-6 do include redundancy and may provide a higher percentage of usable space out of the pool of disks, but have a write penalty in that a single write to the array can lead to far more reads, writes, and seeks. These are discouraged in a write-intensive database environment. See http://www.stearns.org/slartibartfast/uml-coop.current.html#lessonslearned , and contact support@activecountermeasures.com if you'd like to go over the choices for raid. While a UPS with the ability to gracefully shut down when power gets low is always a good idea, it becomes even more important when a raid array is in use.

11. If on a cloud server, check that it's a CPU optimized instance if your cloud provider offers it.
    a. Is this needed? Some cloud providers offer higher end processors for a small premium in cost/hour. Since AC-Hunter can be constrained by CPU (along with memory and disk), running it on a system with faster processors should speed it up.
    b. How to: See if your cloud provider offers CPU optimized instances, and take a look at the difference in cost per hour. If available and the price is reasonable, consider migrating over to one of those.

AC-Hunter operating system

12. Move other CPU/Memory/Disk/Network intensive tasks off to other systems.
    a. Is this needed? AC-Hunter (in particular, the MongoDB database package that's used for storing your databases) uses a substantial amount of memory and processor time. Depending on the size of your network, it may also use a substantial amount of disk bandwidth. Removing any resource hogs that are not part of AC-Hunter itself may allow AC-Hunter to run more quickly.
    b. How to: see https://www.activecountermeasures.com/why-is-my-program-running-slowly/ .

13. Use the XFS filesystem on the filesystem under /var/lib/docker/ . Consider putting it on a raid array.
    a. Is this needed? While the other components of AC-Hunter are filesystem agnostic, the Mongo database can get a performance boost if you use the XFS filesystem instead of the default ext4 for the directory tree under /var/lib/docker/.
    b. How to: It's not practical to change filesystems after the initial install. This will very likely mean reinstalling the Linux operating system (perhaps on a duplicate server) and selecting XFS as the filesystem. Note that /var/lib/docker/ may not be on its own filesystem, so you would select XFS for the root filesystem if your installer allows it.

14. Disable selinux. Selinux is not a Linux distribution, but rather a kernel technology and set of profiles that provide additional security to a Linux system. Because it has the ability to block applications from certain actions, AC-Hunter is not set up to work with it. Having it enabled can lead to errors during the install, during an upgrade, or while using AC-Hunter.
    a. Is this needed? Yes, selinux needs to be disabled on an AC-Hunter system. Even if AC-Hunter operates correctly with it in permissive mode, selinux' checks and exception logging will still slow down the system.
    b. How to: run the command:

```
setenforce permissive
```

to disable it on this boot.

To make the change permanent, edit /etc/selinux/config with

```
sudo nano /etc/selinux/config
```

and change the SELINUX= line to

```
SELINUX=disabled
```

15. Disable strace, gdb, and other application profiling tools.
    a. Is this needed? These tools are not likely to be in use, but if they are, they will use up part of your processing time.

   b. How to: Revert the steps you used to enable these.

16. Use ionice and renice to raise disk and CPU priority.
    a. Is this needed? Ideally AC-Hunter is running on a system by itself, so there aren't other tasks that need lots of CPU, memory, disk bandwidth, or network bandwidth. If there are additional tasks like this on the system, you can use the ionice and renice commands to adjust the disk priorities and processor priorities respectively.
    b. How to: See "Using ionice to prioritize disk traffic" in https://www.activecountermeasures.com/why-is-my-program-running-slowly/ , which covers both "nice" and "ionice". See also "IO Priority" in https://www.activecountermeasures.com/improving-packet-capture-performance-3-of-3/ .

17. Set up swap. You should have *at least* 4GB of swap, perhaps up to 8-16GB if space allows. The goal is to give the operating system some space to push out infrequently used code and data, freeing up system ram for AC-Hunter.
    a. Is this needed? Swap space on disk allows Linux to push infrequently used blocks of program data out to disk, freeing up memory for more urgent needs and improved disk caching. On a system with far too little memory for the tasks being performed the use of swap is associated with system *slowdowns* as Linux needs to spend more and more time swapping things out, and then back in again (though swapping isn't the problem, having too little ram is the problem.) However, if you do have enough ram for the database and disk caching tasks, swap space can be beneficial as it frees up a little more ram for caching.
    b. How to: See "Creating swap space" in https://www.activecountermeasures.com/why-is-my-program-running-slowly/ . Note that you do *not* need to repartition your disks to provide swap; the above instructions show how to do this by creating a file.

AC-Hunter software and settings

18. Check that your databases have all the correct indexes and add any that are missing.
    a. Is this needed? We have a tool called "check_indexes.sh" that looks at all your AC-Hunter databases and reports back on any that are missing any indexes (database components that speed up database performance). Please see indexes.howto.txt for instructions on how to use both tools. Please contact support@activecountermeasures.com for the two scripts and the instruction file.
    b. How to: If check_indexes reports any missing databases, please run add_indexes.sh (again, see indexes.howto.txt for how to use both).

19. If safelisting (specifically, adding a new safelist entry or removing an old one) is taking too long remove old database snapshots that are no longer needed for your retention policy.
    a. Is this needed? The processing time needed to apply safelist entries goes up in proportion to the number of databases. If safelist processing is taking too long, you might consider removing old database snapshots (after checking that your retention policy allows it).
    b. How to: You can either remove the databases manually inside the gear icon in the web interface or use "rita delete" from the command line. See https://portal.activecountermeasures.com/support/faq/?Display_FAQ=4970 . For a more permanent and hands-off solution, we recommend using https://github.com/activecm/zeek-log-clean , which can keep disk usage below a high water mark by removing older databases and Zeek log files. (Note that this last tool frees up space but does not have a way to follow your retention requirements.)

20. Upgrade to AC-Hunter 6.4.0 or higher.
    a. Is this needed? AC-Hunter includes a number of performance improvements over previous versions. We encourage all customers to upgrade to it when time allows. (If your AC-Hunter system runs Ubuntu 16.04 note that you'll need to upgrade that first as AC-Hunter 6.4.0 no longer supports Ubuntu 16.04)
    b. How to: Please see the Install Guide in https://portal.activecountermeasures.com/support/product-documentation/ and/or Tutorial 02 in https://portal.activecountermeasures.com/tutorials/ .

21. Replace AC-Hunter 6.4.0 with AC-Hunter 6.4.0-2311.
    a. Is this needed? This minor patch to 6.4.0 affects how wildcard domain safelists are applied. If one chooses to safelist, say, *.microsoft.com, plain 6.4.0 will add all IP addresses under that domain one at a time. 6.4.0-2311 batches up some IP addresses and safelists them as a group. Functionally there's no difference, but this change noticeably speeds up safelisting wildcard domains.
    b. How to: If you're finding that safelisting is slow, contact support@activecountermeasures.com for a download link and apply it like any other upgrade. This minor release has not gone through a full QA cycle, but we've had only good feedback that it helps safelisting performance.

22. If using AC-Hunter <6.4.0, disable processing on External->Internal connections.
    a. Is this needed? (If you're running AC-Hunter 6.4.0 or higher, this is the default so you can skip this one.) For AC-Hunter below 6.4.0, you may wish to set FilterExternalToInternal to True . This tells AC-Hunter to only look for Threat activity in *outgoing* connections from your network, where the majority are found.
    b. How to: edit /etc/AC-Hunter/rita.yaml with

```
sudo nano /etc/AC-Hunter/rita.yaml
```

, scroll down to the "Filtering" section, then scroll down to the line with:

```
  #FilterExternalToInternal: true
```

or

```
  #FilterExternalToInternal: false
```

Edit that line, making sure that it ends in "true", does **not** have a "#" at the beginning, and contains exactly the same number of spaces to its left as when you opened the file (it should be two spaces, never tabs). Save and exit, then run

```
hunt down ; hunt up -d –force-recreate
```

23. Make sure the network subnet list matches what is on your systems.
    a. Is this needed? If you only use "reserved" addresses, also called "internal" or "rfc1918" addresses for your internal systems, you're all set - Zeek is set up by default to treat these as your "internal" systems. If you use some public IP addresses (something *other than* addresses that start with 10, 192.168, 172.16-172.31, or fe80:) inside your network, you should update the list of addresses stored in networks.cfg .
    b. How to: See https://portal.activecountermeasures.com/support/faq/?Display_FAQ=3132

24. If on a cloud/virtual machine instance, move to one that has dedicated CPUs and disks.
    a. Is this needed? Virtual machines (which include cloud instances in this discussion) have a lot of advantages. One potential disadvantage is the fact that multiple virtual machines share the processors and disk on their physical host. A virtual machine that has worked fine for months might start to slow down because another virtual machine on the same physical box starts to use huge amounts of processing power or disk bandwidth.
    b. How to: If possible, move the storage for the AC-Hunter cloud instance or virtual machine to a disk that's dedicated to only AC-Hunter. You could also consider dedicating processors to AC-Hunter, though the benefit would be somewhat less.

25. If you've enabled the COMBINED__0000 database, make sure you've disabled processing the individual sensors.
    a. Is this needed? When you have enabled the COMBINED__0000 database, AC-Hunter needs to import all log files *twice* - once for the individual sensors and once for COMBINED__0000. This doubles the processing load and the amount of time needed to import all log files.
    b. How to: Follow the instructions in the AC-Hunter User Guide in the section "Only View the Combined Database, Not the Individual Sensors".

26. Look to see if there are long query times in Mongo.
    a. Is this needed? There are circumstances where the code that interacts with the database exceeds one of many limits imposed by MongoDB. The errors for these are logged, but we need to run the following command to expose and review them.
    b. How to: Run:

```
hunt logs db >db.log
```

Please send this file to support@activecountermeasures.com . If the file is too large to send, create a trimmed down version with the following command instead:

```
hunt logs db | tail -n 5000 >db.log
```

The entries in this file will show cases where a database query exceeded a limit or had some other kind of error.

27. Manually set up Mongodb in cluster mode.
    a. Is this needed? AC-Hunter sets up the Mongo database software as a single storage node; this is the most straightforward way to implement it. As the database load increases with more bandwidth, connections, and systems, Mongo may get to the point where a single storage node may not be able to keep up. Migrating to a database cluster should provide a significant performance boost.
    b. How to: This is a complex task, unfortunately, and doesn't lend itself to step-by-step instructions. We'd only recommend this if you have someone on your staff that's already familiar with setting up a Mongo cluster.

28. If you have more than one sensor feeding AC-Hunter, set up a second AC-Hunter system and split the sensors between them.
    a. Is this needed? When you have more than one sensor, their log files are imported one after the other, not in parallel. By setting up a second system and splitting up the sensors between them, the import work can be done in parallel. Note that the safelists in the two systems are independent - after adding new safelist entries you'll need to copy that safelist to the other AC-Hunter, or set up safelist synchronization with https://www.activecountermeasures.com/safelist-synchronization/ .
    b. How to: Build a second AC-Hunter system. For the sensors that will feed the second system, follow the steps in https://portal.activecountermeasures.com/support/faq/?Display_FAQ=863 to send your logs to the new system. Finally, on each sensor that is now feeding logs to the new system, edit /etc/cron.d/zeek_log_transport and remove the line that feeds the old AC-Hunter system.

29. If your network has a significant amount of high volume beacon traffic whose volume is not high enough to classify it as strobes (which are processed more efficiently), consider lowering the threshold for the number of connections.
    a. Is this needed? If you're using AC-Hunter 6.4.0 or higher, the default is already 86400, so this hint is for versions before 6.4.0.
    b. How to: To change the strobe limit in the rita configuration, edit

```
/etc/AC-Hunter/rita.yaml
```

and change the following line:

```
  ConnectionLimit: 250000
```

to

```
  ConnectionLimit: 86400
```

(preserving the spaces you found in that file to the left of "ConnectionLimit'). This filters incoming data to one connection per second (per chunk).

Once you've made this change, please run

```
sudo hunt up -d --force-recreate
```

to load the new setting.

30. Disable transparent hugepage defrag
    a. Is this needed? This is a recommendation from the Mongo database package.
    b. How to:

    If /sys/kernel/mm/transparent_hugepage/defrag is 'always', set it to 'never'

```
echo 'never' | sudo dd of=/sys/kernel/mm/transparent_hugepage/defrag
```

To make the change permanent, create

```
/etc/systemd/system/disable-transparent-huge-pages.service
```

with the following content:

```
[Unit]
Description=Disable Transparent Huge Pages (THP)
DefaultDependencies=no
After=sysinit.target local-fs.target
Before=mongod.service

[Service]
Type=oneshot
ExecStart=/bin/sh -c 'echo never | tee
/sys/kernel/mm/transparent_hugepage/enabled > /dev/null ; echo never
| tee /sys/kernel/mm/redhat_transparent_hugepage/enabled > /dev/null'

[Install]
WantedBy=basic.target
```

Then run:

```
sudo systemctl daemon-reload
sudo systemctl start disable-transparent-huge-pages
sudo systemctl enable disable-transparent-huge-pages
```

Note; see https://docs.mongodb.com/manual/tutorial/transparent-huge-pages/ for more notes on setting up tuned/ktune.

31. Disable the AC-Hunter module that is the heaviest user of import time/processor/memory. In AC-Hunter 6.x, this is likely to be BeaconsWeb.
    a. Is this needed? This should be considered last, as it disables one of the most powerful tools in the Threat Hunting arsenal. If the previous approaches cannot

get you to a working AC-Hunter system you may wish to disable BeaconsWeb to see if you can at least get the other AC-Hunter modules working.

b. How to: edit /etc/AC-Hunter/rita.yaml with

```
sudo nano /etc/AC-Hunter/rita.yaml
```

, scroll down to the "BeaconSNI" section, then scroll down to the line with:

```
  #Enabled: true
```

Edit that line, making sure that it ends in "false", does **not** have a "#" at the beginning, and contains exactly the same number of spaces to its left as when you opened the file (it should be two spaces, never tabs). Save and exit, then run

```
hunt down ; hunt up -d –force-recreate
```

References

https://www.activecountermeasures.com/why-is-my-program-running-slowly/
https://www.activecountermeasures.com/improving-packet-capture-performance-1-of-3/
https://www.activecountermeasures.com/improving-packet-capture-performance-2-of-3/
https://www.activecountermeasures.com/improving-packet-capture-performance-3-of-3/
https://www.activecountermeasures.com/?s=BPF
https://www.activecountermeasures.com/packet-loss-or-why-is-my-sniffer-dropping-packets/
https://www.activecountermeasures.com/filtering-out-high-volume-traffic/
https://www.activecountermeasures.com/building-a-global-ignore-filter/

Thanks

Many thanks to Keith and Ethan for their excellent help in reviewing and improving this document.

This is version 202309201332 .