

containers / **aardvark-dns** Public

<> **Code** Issues 15 Pull requests 4 Discussions Actions Projects

Commit 3b49ea7

Luap99 committed 5 days ago Verified

fix handling of incorrect tcp packets

Ensure we correctly abort the tcp stream on errors and do not keep retrying.

Signed-off-by: Paul Holzinger <pholzinger@redhat.com>

main

1 parent [6486a7d](#) commit 3b49ea7

2 files changed +46 -15 lines changed

[↑ Top](#)

src/dns

coredns.rs

test

100-basic-name-resolution.bats

2 files changed +46 -15 lines changed

src/dns/coredns.rs

@@ -17,7 +17,6 @@ use hickory_proto::{

17 17 DnsStreamHandle,

18 18 };

19 19 use log::{debug, error, trace, warn};

20 - use std::io::Error;

21 20 use std::net::{IpAddr, SocketAddr};

22 21 use std::sync::Arc;

23 22 use std::sync::Mutex;

| | ↓ | ↑ | |
|-----|-----|---|--|
| | | | @@ -83,15 +82,21 @@ impl CoreDns { |
| 83 | 82 | | break; |
| 84 | 83 | | }, |
| 85 | 84 | | v = receiver.next() => { |
| 86 | - | | let msg_received = match v { |
| 87 | - | | Some(value) => value, |
| | 85 | + | let msg = match v { |
| | 86 | + | Some(value) => match value { |
| | 87 | + | Ok(msg) => msg, |
| | 88 | + | Err(e) => { |
| | 89 | + | debug!("Error parsing dns message {e:?}"); |
| | 90 | + | continue; |
| | 91 | + | }, |
| | 92 | + | }, |
| 88 | 93 | | None => { |
| 89 | 94 | | // None received, nothing to process so continue |
| 90 | 95 | | debug!("None received from stream, continue the |
| | | | loop"); |
| 91 | 96 | | continue; |
| 92 | 97 | | } |
| 93 | 98 | | }; |
| 94 | - | | Self::process_message(&self.inner, msg_received, |
| | | | &sender_original, Protocol::Udp).await; |
| | 99 | + | Self::process_message(&self.inner, msg, &sender_original, |
| | | | Protocol::Udp).await; |
| 95 | 100 | | }, |
| 96 | 101 | | res = tcp_listener.accept() => { |
| 97 | 102 | | match res { |
| | | | @@ -122,9 +127,16 @@ impl CoreDns { |
| | | | // we do not want this so add a 3s timeout then we close the |
| | | | socket. |
| 123 | 128 | | match tokio::time::timeout(Duration::from_secs(3), |
| | | | hickory_stream.next()).await { |
| 124 | 129 | | Ok(message) => match message { |
| 125 | - | | Some(msg) => { |
| 126 | - | | Self::process_message(&data, msg, &sender_original, |
| | | | Protocol::Tcp).await |
| 127 | - | | } |

```

130 +             Some(msg_result) => match msg_result {
131 +                 Ok(msg) => {
132 +                     Self::process_message(&data, msg, &sender_original,
133 + Protocol::Tcp).await
134 +                 }
135 +                 Err(e) => {
136 +                     debug!("Error parsing dns message {e:?}");
137 +                     // error on that stream, abort so we do not try
138 +                     // reusing this one for more messages.
139 +                     break;
140 +                 }
141 +             },
142 +         },
143 +     },
144 +     // end of stream
145 +     None => break,
146 + },
147 + },
148 + },
149 + },
150 + },
151 + },
152 + },
153 + },
154 + },
155 + },
156 + },
157 + },
158 + },
159 + },
160 + },
161 + },
162 + },
163 + },
164 + },
165 + },
166 + },
167 + },
168 + },
169 + },
170 + },
171 + },
172 + },
173 + },
174 + },
175 + },
176 + },
177 + },
178 + },
179 + },
180 + },
181 + },
182 + },
183 + },
184 + },
185 + },
186 + },
187 + },
188 + },
189 + },
190 + },
191 + },
192 + },
193 + },
194 + },
195 + },
196 + },
197 + },
198 + },
199 + },
200 + },
201 + },
202 + },
203 + },
204 + },
205 + },
206 + },
207 + },
208 + },
209 + },
210 + },
211 + },
212 + },
213 + },
214 + },
215 + },
216 + },
217 + },
218 + },
219 + },
220 + },
221 + },
222 + },
223 + },
224 + },
225 + },
226 + },
227 + },
228 + },
229 + },
230 + },
231 + },
232 + },
233 + },
234 + },
235 + },
236 + },
237 + },
238 + },
239 + },
240 + },
241 + },
242 + },
243 + },
244 + },
245 + },
246 + },
247 + },
248 + },
249 + },
250 + },
251 + },
252 + },
253 + },
254 + },
255 + },
256 + },
257 + },
258 + },
259 + },
260 + },
261 + },
262 + },
263 + },
264 + },
265 + },
266 + },
267 + },
268 + },
269 + },
270 + },
271 + },
272 + },
273 + },
274 + },
275 + },
276 + },
277 + },
278 + },
279 + },
280 + },
281 + },
282 + },
283 + },
284 + },
285 + },
286 + },
287 + },
288 + },
289 + },
290 + },
291 + },
292 + },
293 + },
294 + },
295 + },
296 + },
297 + },
298 + },
299 + },
300 + },
301 + },
302 + },
303 + },
304 + },
305 + },
306 + },
307 + },
308 + },
309 + },
310 + },
311 + },
312 + },
313 + },
314 + },
315 + },
316 + },
317 + },
318 + },
319 + },
320 + },
321 + },
322 + },
323 + },
324 + },
325 + },
326 + },
327 + },
328 + },
329 + },
330 + },
331 + },
332 + },
333 + },
334 + },
335 + },
336 + },
337 + },
338 + },
339 + },
340 + },
341 + },
342 + },
343 + },
344 + },
345 + },
346 + },
347 + },
348 + },
349 + },
350 + },
351 + },
352 + },
353 + },
354 + },
355 + },
356 + },
357 + },
358 + },
359 + },
360 + },
361 + },
362 + },
363 + },
364 + },
365 + },
366 + },
367 + },
368 + },
369 + },
370 + },
371 + },
372 + },
373 + },
374 + },
375 + },
376 + },
377 + },
378 + },
379 + },
380 + },
381 + },
382 + },
383 + },
384 + },
385 + },
386 + },
387 + },
388 + },
389 + },
390 + },
391 + },
392 + },
393 + },
394 + },
395 + },
396 + },
397 + },
398 + },
399 + },
400 + },
401 + },
402 + },
403 + },
404 + },
405 + },
406 + },
407 + },
408 + },
409 + },
410 + },
411 + },
412 + },
413 + },
414 + },
415 + },
416 + },
417 + },
418 + },
419 + },
420 + },
421 + },
422 + },
423 + },
424 + },
425 + },
426 + },
427 + },
428 + },
429 + },
430 + },
431 + },
432 + },
433 + },
434 + },
435 + },
436 + },
437 + },
438 + },
439 + },
440 + },
441 + },
442 + },
443 + },
444 + },
445 + },
446 + },
447 + },
448 + },
449 + },
450 + },
451 + },
452 + },
453 + },
454 + },
455 + },
456 + },
457 + },
458 + },
459 + },
460 + },
461 + },
462 + },
463 + },
464 + },
465 + },
466 + },
467 + },
468 + },
469 + },
470 + },
471 + },
472 + },
473 + },
474 + },
475 + },
476 + },
477 + },
478 + },
479 + },
480 + },
481 + },
482 + },
483 + },
484 + },
485 + },
486 + },
487 + },
488 + },
489 + },
490 + },
491 + },
492 + },
493 + },
494 + },
495 + },
496 + },
497 + },
498 + },
499 + },
500 + },
501 + },
502 + },
503 + },
504 + },
505 + },
506 + },
507 + },
508 + },
509 + },
510 + },
511 + },
512 + },
513 + },
514 + },
515 + },
516 + },
517 + },
518 + },
519 + },
520 + },
521 + },
522 + },
523 + },
524 + },
525 + },
526 + },
527 + },
528 + },
529 + },
530 + },
531 + },
532 + },
533 + },
534 + },
535 + },
536 + },
537 + },
538 + },
539 + },
540 + },
541 + },
542 + },
543 + },
544 + },
545 + },
546 + },
547 + },
548 + },
549 + },
550 + },
551 + },
552 + },
553 + },
554 + },
555 + },
556 + },
557 + },
558 + },
559 + },
560 + },
561 + },
562 + },
563 + },
564 + },
565 + },
566 + },
567 + },
568 + },
569 + },
570 + },
571 + },
572 + },
573 + },
574 + },
575 + },
576 + },
577 + },
578 + },
579 + },
580 + },
581 + },
582 + },
583 + },
584 + },
585 + },
586 + },
587 + },
588 + },
589 + },
590 + },
591 + },
592 + },
593 + },
594 + },
595 + },
596 + },
597 + },
598 + },
599 + },
600 + },
601 + },
602 + },
603 + },
604 + },
605 + },
606 + },
607 + },
608 + },
609 + },
610 + },
611 + },
612 + },
613 + },
614 + },
615 + },
616 + },
617 + },
618 + },
619 + },
620 + },
621 + },
622 + },
623 + },
624 + },
625 + },
626 + },
627 + },
628 + },
629 + },
630 + },
631 + },
632 + },
633 + },
634 + },
635 + },
636 + },
637 + },
638 + },
639 + },
640 + },
641 + },
642 + },
643 + },
644 + },
645 + },
646 + },
647 + },
648 + },
649 + },
650 + },
651 + },
652 + },
653 + },
654 + },
655 + },
656 + },
657 + },
658 + },
659 + },
660 + },
661 + },
662 + },
663 + },
664 + },
665 + },
666 + },
667 + },
668 + },
669 + },
670 + },
671 + },
672 + },
673 + },
674 + },
675 + },
676 + },
677 + },
678 + },
679 + },
680 + },
681 + },
682 + },
683 + },
684 + },
685 + },
686 + },
687 + },
688 + },
689 + },
690 + },
691 + },
692 + },
693 + },
694 + },
695 + },
696 + },
697 + },
698 + },
699 + },
700 + },
701 + },
702 + },
703 + },
704 + },
705 + },
706 + },
707 + },
708 + },
709 + },
710 + },
711 + },
712 + },
713 + },
714 + },
715 + },
716 + },
717 + },
718 + },
719 + },
720 + },
721 + },
722 + },
723 + },
724 + },
725 + },
726 + },
727 + },
728 + },
729 + },
730 + },
731 + },
732 + },
733 + },
734 + },
735 + },
736 + },
737 + },
738 + },
739 + },
740 + },
741 + },
742 + },
743 + },
744 + },
745 + },
746 + },
747 + },
748 + },
749 + },
750 + },
751 + },
752 + },
753 + },
754 + },
755 + },
756 + },
757 + },
758 + },
759 + },
760 + },
761 + },
762 + },
763 + },
764 + },
765 + },
766 + },
767 + },
768 + },
769 + },
770 + },
771 + },
772 + },
773 + },
774 + },
775 + },
776 + },
777 + },
778 + },
779 + },
780 + },
781 + },
782 + },
783 + },
784 + },
785 + },
786 + },
787 + },
788 + },
789 + },
790 + },
791 + },
792 + },
793 + },
794 + },
795 + },
796 + },
797 + },
798 + },
799 + },
800 + },
801 + },
802 + },
803 + },
804 + },
805 + },
806 + },
807 + },
808 + },
809 + },
810 + },
811 + },
812 + },
813 + },
814 + },
815 + },
816 + },
817 + },
818 + },
819 + },
820 + },
821 + },
822 + },
823 + },
824 + },
825 + },
826 + },
827 + },
828 + },
829 + },
830 + },
831 + },
832 + },
833 + },
834 + },
835 + },
836 + },
837 + },
838 + },
839 + },
840 + },
841 + },
842 + },
843 + },
844 + },
845 + },
846 + },
847 + },
848 + },
849 + },
850 + },
851 + },
852 + },
853 + },
854 + },
855 + },
856 + },
857 + },
858 + },
859 + },
860 + },
861 + },
862 + },
863 + },
864 + },
865 + },
866 + },
867 + },
868 + },
869 + },
870 + },
871 + },
872 + },
873 + },
874 + },
875 + },
876 + },
877 + },
878 + },
879 + },
880 + },
881 + },
882 + },
883 + },
884 + },
885 + },
886 + },
887 + },
888 + },
889 + },
890 + },
891 + },
892 + },
893 + },
894 + },
895 + },
896 + },
897 + },
898 + },
899 + },
900 + },
901 + },
902 + },
903 + },
904 + },
905 + },
906 + },
907 + },
908 + },
909 + },
910 + },
911 + },
912 + },
913 + },
914 + },
915 + },
916 + },
917 + },
918 + },
919 + },
920 + },
921 + },
922 + },
923 + },
924 + },
925 + },
926 + },
927 + },
928 + },
929 + },
930 + },
931 + },
932 + },
933 + },
934 + },
935 + },
936 + },
937 + },
938 + },
939 + },
940 + },
941 + },
942 + },
943 + },
944 + },
945 + },
946 + },
947 + },
948 + },
949 + },
950 + },
951 + },
952 + },
953 + },
954 + },
955 + },
956 + },
957 + },
958 + },
959 + },
960 + },
961 + },
962 + },
963 + },
964 + },
965 + },
966 + },
967 + },
968 + },
969 + },
970 + },
971 + },
972 + },
973 + },
974 + },
975 + },
976 + },
977 + },
978 + },
979 + },
980 + },
981 + },
982 + },
983 + },
984 + },
985 + },
986 + },
987 + },
988 + },
989 + },
990 + },
991 + },
992 + },
993 + },
994 + },
995 + },
996 + },
997 + },
998 + },
999 + },
1000 + },

```

```

test/100-basic-name-resolution.bats
@@ -371,3 +371,29 @@ function teardown() {
371 371     run_in_container_netns "$a1_pid" "dig" "+short" "second-server.test" "@$gw"

```

```
372 372     assert "$output" == "192.168.100.2" "should resolve using second DNS server
      after resolv.conf change"
373 373 }
374 +
375 + @test "check for incorrect tcp packet" {
376 +     setup_dnsmasq
377 +
378 +     subnet_a=$(random_subnet 5)
379 +     create_config network_name="podman1" container_id=$(random_string 64)
      container_name="aone" subnet="$subnet_a"
380 +     config_a1=$config
381 +     ip_a1=$(echo "$config_a1" | jq -r .networks.podman1.static_ips[0])
382 +     gw=$(echo "$config_a1" | jq -r .network_info.podman1.subnets[0].gateway)
383 +     create_container "$config_a1"
384 +     a1_pid=$CONTAINER_NS_PID
385 +
386 +     # send custom crafted package, first two bytes mean package length 60 but
      we never send more and close instead
387 +     run_in_container_netns "$a1_pid" socat - TCP4:$gw:53 <<<'\x00\x3c'
388 +
389 +     # wait a second to meaningful check cpu usage
390 +     sleep 1
391 +
392 +     av_cpu=$(ps -o c --no-headers -p $(<$AARDVARK_TMPDIR/aardvark-
      dns/aardvark.pid))
393 +     echo $av_cpu --
394 +     assert "$av_cpu" -lt 5 "aardvark-dns used to much cpu"
395 +
396 +     # ensure dns via tcp still works
397 +     run_in_container_netns "$a1_pid" "dig" +tcp "+short" "aone" "@$gw"
398 +     assert "$ip_a1"
399 + }
```

Comments 0



Please [sign in](#) to comment.