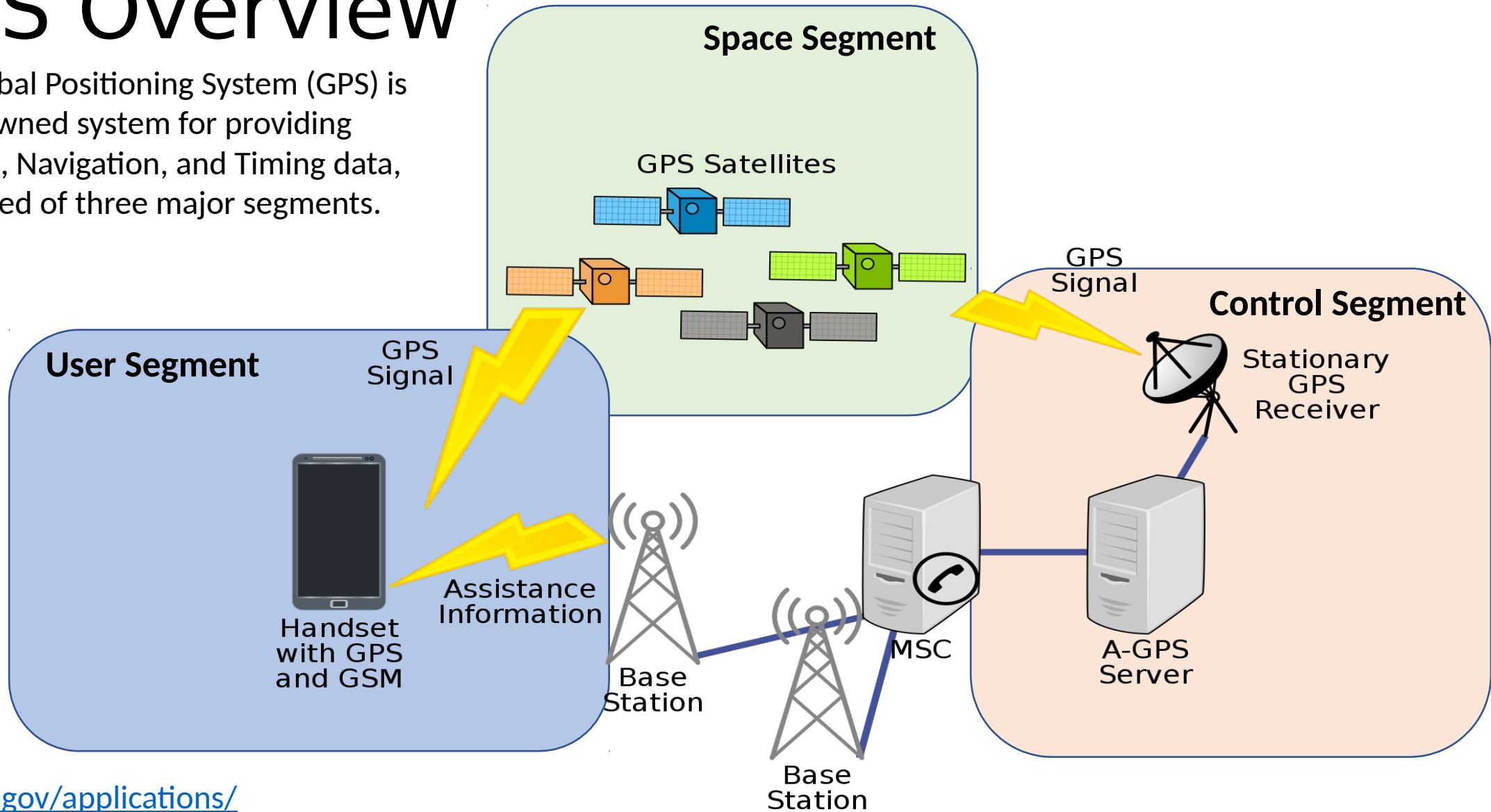


Agenda

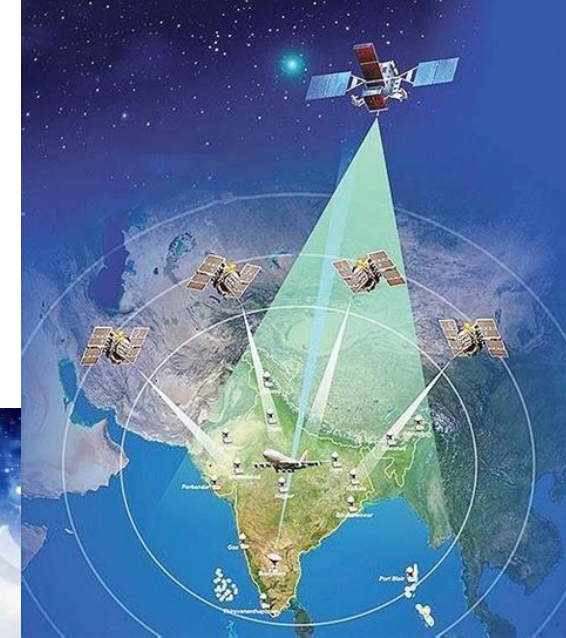
- GPS Overview
- STRIDE Threat Model
 - Spoofing
 - Tampering
 - Repudiation
 - Information Disclosure
 - Denial Of Service
 - Escalation of Privilege

GPS Overview

The Global Positioning System (GPS) is a U.S. owned system for providing Position, Navigation, and Timing data, composed of three major segments.



Applications



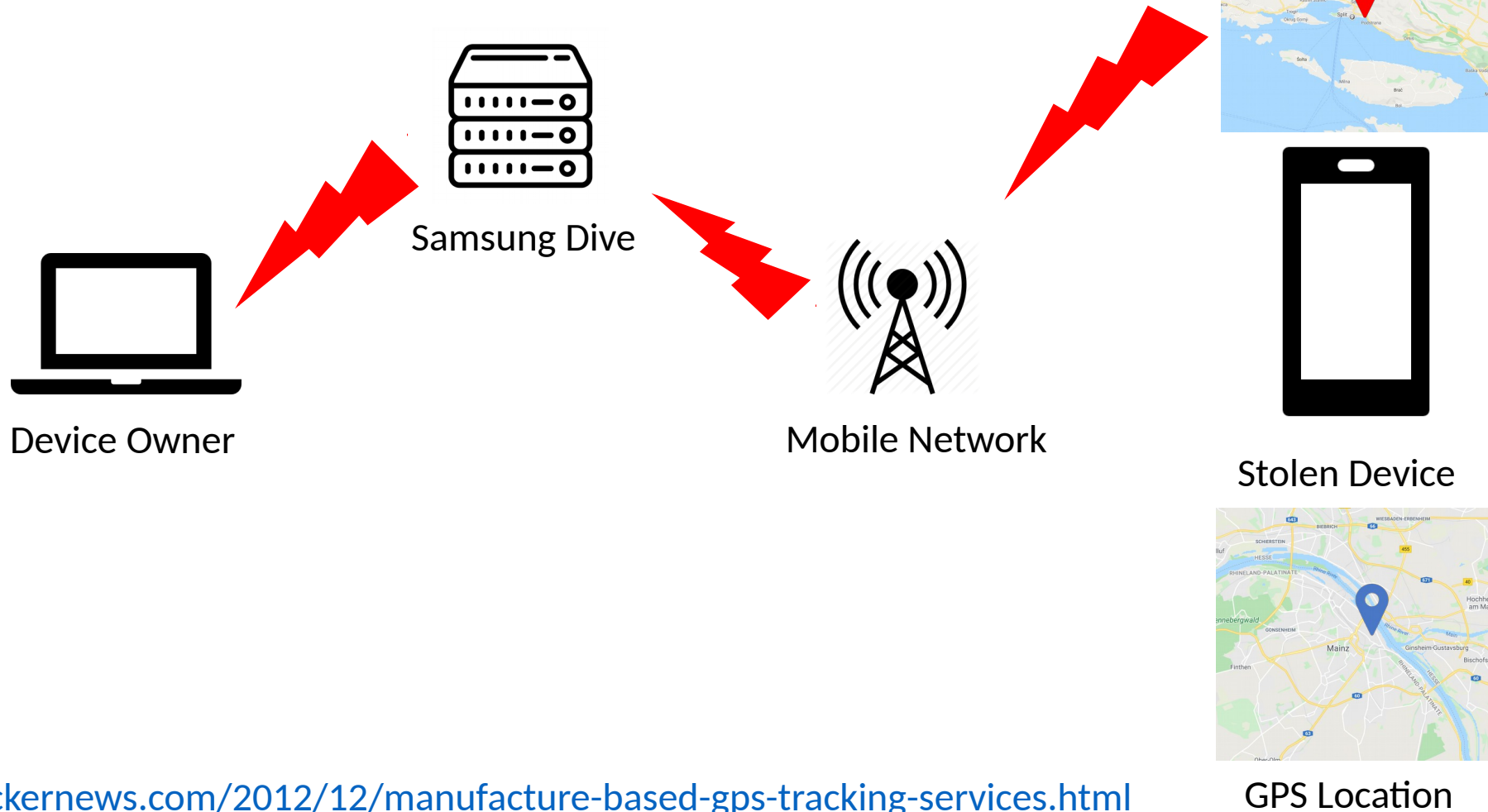
GPS is used in many industries, such as agriculture, aviation, geology and surveying, and military operations.

It is also used in day-to-day activities such as navigating on a road trip or tracking fitness activities like running or cycling.

STRIDE Assessment

Spoofing

CVE-2012-6336



- <https://thehackernews.com/2012/12/manufacture-based-gps-tracking-services.html>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-6336>

Spoofing

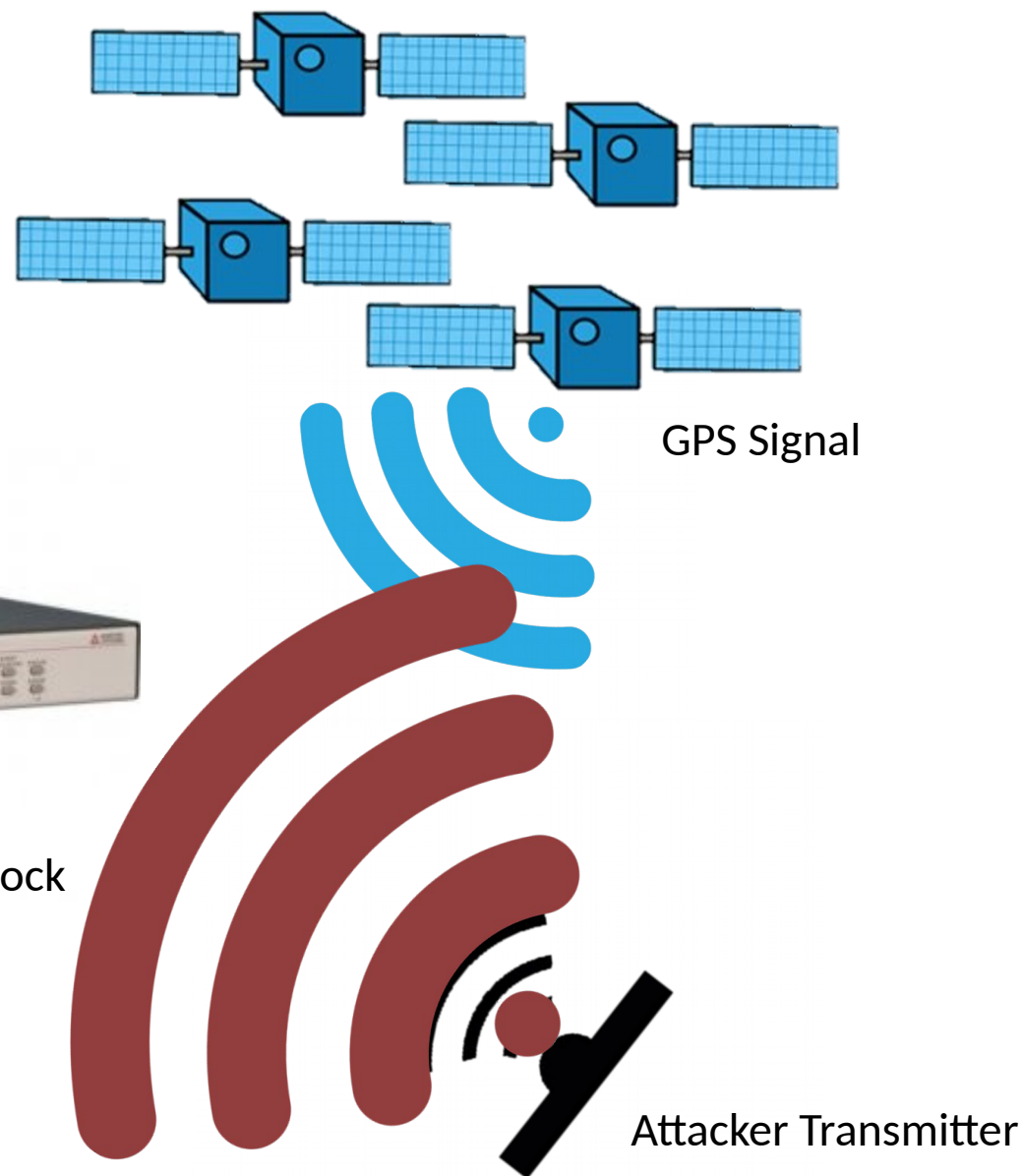
CVE-2014-9194

Arbiter Systems has identified a GPS clock spoofing vulnerability in its 1094B clock.

An attacker with specialized radio equipment and knowledge could transmit signals that can disrupt the clock.



Arbiter Model 1094B GPS Substation Clock
Used in electric utilities substations
Depends on GPS for precision timing



Tampering

SCOUT GPS LINK Vulnerability – Vulnerability in Scout GPS App provides access to multimedia screen in Toyota and Lexus Vehicles



$hash(h1, salt1, round1) \rightarrow h2.$

For each incoming message enclosed by the connectivity protocol *MQTT*, the broker verifies the access code *AC* firstly.

$hash(AC, salt2, round2) \rightarrow h3;$

$hash(h3, salt1, round1) \rightarrow h4.$

By comparing *h2* and *h4*, the *Scout GPS Link* decides to accept or deny the message



An attacker can move the comparison process from *h2* and *h4* to *h1* and *h3* to bypass this security mechanism -- *h1* and $hash(AC, salt2, round2)$. All these values are encoded in the binary code.

Since the app accepts any connections, attacker can remotely find the *Scout GPS Link* users by scanning port 7050.

Using MQTT protocol, messages can be published or subscribed

```
mqttc.publish("uma/jsonrpc/mobile", "{\"jsonrpc\":\"2.0\", \"method\":\"DrivingRestriction\", \"params\": {\"level\":\"50\"}}\", 1, True)
```

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-14951>

<https://sites.google.com/site/iosappnss/more-vulnerable-apps-and-libraries>

Repudiation & Information Disclosure

CVE-2017-5237

Given knowledge of the EV-07S's registered phone number, the EV-07S device can be reset to factory level setting by sending "RESET!" as a command in an SMS message to the device

Manual fuzzing test via SMS revealed that the command "REBOOT!" will cause the device to respond with the message "Format error!". Due to providing this negative response, a malicious actor could use this command to enumerate all devices by trying all likely phone numbers, commonly known as a [war dialing](#) operation, using SMS messages containing the "REBOOT!" command.

An unauthenticated attacker can poison the realtime tracking data by injecting device data to the server at www.smart-tracking.com over TCP port 5050. The attacker can do this only if they know a device's IMEI number, but that data is learnable through an additional vulnerability: R7-2016-28.5

An authenticated user can gain access to others users configuration and device GPS data if they know or guess a valid userId, device IMEI or TrackerID

EV-07S GPS Tracker



The EV-07S is a personal GPS tracker device used for personal safety and security

Information Disclosure

The web application used for realtime tracking web application, hosted at <http://www.smart-tracking.com>, did not utilize SSL/TLS encryption for HTTP services. Also the EV-07S device passed IMEI and GPS data to this website over the Internet on TCP port 5050 without any encryption

An authenticated user can gain access to others users configuration and device GPS data if they know or guess a valid userId, device IMEI or TrackerID.

GPS online tracking s... x Modify User - GPS online... x

www.smart-tracking.com/edituser.php?userId=5

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng

Modify User

User ID: Deral *
Password: *
Country: us *
Time Zone: (GMT-08:00) *
User real name: *
Created By: eview1 *
Notes: *

Gender: ☒ Male ☐ Female
User Role: Administrator
Email Address: deral_heiland@rapid7.com *
Company name: *
Address: *
Contact number: *
Creation date: 2016-11-08
Expiry date: 2017-11-08

Modify Delete

An authenticated user can gain access to user configuration data on another users account if the know or guess userId= number

Request

Raw Params Headers Hex

POST /get_position.jsp HTTP/1.1
Host: www.smart-tracking.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://www.smart-tracking.com/main.jsp
Cookie: language=en; isRemember=false; JSESSIONID=0C0FF22E16797D5C9673900226A30906; DWRSESSIONID=ymTNS9IHpAQNB5*6A0v6F4m\$axl
Content-Length: 50
Connection: close

Response

Raw Headers Hex

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=utf-8
Content-Length: 259
Date: Sat, 12 Nov 2016 05:29:33 GMT

An authenticated user can gain access to any device GPS data if the IMEI number is known.

Timestamp must also be reasonably accurate

IMEI

GPS data

Denial of Service

CVE-2016-5348

Android devices can be crashed remotely forcing a halt and then a soft reboot by a MITM attacker manipulating assisted GPS/GNSS data provided by Qualcomm. The Android issue was fixed by in the October 2016 Android bulletin.

The device makes periodic calls to the Qualcomm servers to retrieve gpsOneXtra assistance files. These requests were performed almost every time the device connected to a WiFi network at the following URLs:

<http://xtra1.gpsonextra.net/xtra.bin>

<http://xtra2.gpsonextra.net/xtra.bin>

<http://xtra3.gpsonextra.net/xtra.bin>

<http://xtrapath1.izatcloud.net/xtra2.bin>

<http://xtrapath2.izatcloud.net/xtra2.bin>

<http://xtrapath3.izatcloud.net/xtra2.bin>

Both the Java and the C++ code do not check how large the data file actually is. If a file is served that is larger than the memory available on the device, this results in all memory being exhausted and the phone halting and then soft rebooting. The soft reboot was sufficient to recover from the crash and no data was lost.

To attack, an man-in-the-middle attacker located anywhere on the network between the phone being attacked and Qualcomm's servers can initiate this attack by intercepting the legitimate requests from the phone, and substituting their own, larger files. Because the default Chrome browser on Android reveals the model and build of the phone, it would be possible to derive the maximum memory size from that information and deliver the appropriately sized attack file. Possible attackers can be hostile hotspots, hacked routers, or anywhere along the backbone.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-5348>

<https://www.exploit-db.com/exploits/40502>

Escalation of Privileges

CVE-2020-5246



Traccar GPS Tracking System before version 4.9 has a LDAP injection vulnerability. It occurs when user input is being used in LDAP search filter. By providing specially crafted input, an attacker can modify the logic of the LDAP query and get admin privileges. The issue only impacts instances with LDAP configuration and where users can craft their own names. This has been patched in version 4.9.

```
81 81      if (this.adminFilter != null) {
82 82          try {
83 83              InitialDirContext context = initContext();
84 84              String searchString = adminFilter.replace(":login", accountName);
85 85              String searchString = adminFilter.replace(":login", encodeForLdap(accountName));
86 86              SearchControls searchControls = new SearchControls();
87 87              searchControls.setSearchScope(SearchControls.SUBTREE_SCOPE);
88 87              NamingEnumeration<SearchResult> results = context.search(searchBase, searchString, searchControls);
89 87
90 87      @@ -107,7 +107,7 @@ public InitialDirContext initContext() throws NamingException {
91 87
92 87      private SearchResult lookupUser(String accountName) throws NamingException {
93 87          InitialDirContext context = initContext();
94 87
95 87
96 87
97 87
98 87
99 87
100 87
101 87
102 87
103 87
104 87
105 87
106 87
107 87
108 87
109 87
110 87
111 87
112 87
113 87
114 87
115 87
116 87
117 87
118 87
119 87
120 87
121 87
122 87
123 87
124 87
125 87
126 87
127 87
128 87
129 87
130 87
131 87
132 87
133 87
134 87
135 87
136 87
137 87
138 87
139 87
140 87
141 87
142 87
143 87
144 87
145 87
146 87
147 87
148 87
149 87
150 87
151 87
152 87
153 87
154 87
155 87
156 87
157 87
158 87
159 87
160 87
161 87
162 87
163 87
164 87
165 87
166 87
167 87
168 87
169 87
170 87
171 87
172 87
173 87
174 87
175 87
176 87
177 87
178 87
179 87
180 87
181 87
182 87
183 87
184 87
185 87
186 87
187 87
188 87
189 87
190 87
191 87
192 87
193 87
194 87
195 87
196 87
197 87
198 87
199 87
200 87
201 87
202 87
203 87
204 87
205 87
206 87
207 87
208 87
209 87
210 87
211 87
212 87
213 87
214 87
215 87
216 87
217 87
218 87
219 87
220 87
221 87
222 87
223 87
224 87
225 87
226 87
227 87
228 87
229 87
230 87
231 87
232 87
233 87
234 87
235 87
236 87
237 87
238 87
239 87
240 87
241 87
242 87
243 87
244 87
245 87
246 87
247 87
248 87
249 87
250 87
251 87
252 87
253 87
254 87
255 87
256 87
257 87
258 87
259 87
260 87
261 87
262 87
263 87
264 87
265 87
266 87
267 87
268 87
269 87
270 87
271 87
272 87
273 87
274 87
275 87
276 87
277 87
278 87
279 87
280 87
281 87
282 87
283 87
284 87
285 87
286 87
287 87
288 87
289 87
290 87
291 87
292 87
293 87
294 87
295 87
296 87
297 87
298 87
299 87
300 87
301 87
302 87
303 87
304 87
305 87
306 87
307 87
308 87
309 87
310 87
311 87
312 87
313 87
314 87
315 87
316 87
317 87
318 87
319 87
320 87
321 87
322 87
323 87
324 87
325 87
326 87
327 87
328 87
329 87
330 87
331 87
332 87
333 87
334 87
335 87
336 87
337 87
338 87
339 87
340 87
341 87
342 87
343 87
344 87
345 87
346 87
347 87
348 87
349 87
350 87
351 87
352 87
353 87
354 87
355 87
356 87
357 87
358 87
359 87
360 87
361 87
362 87
363 87
364 87
365 87
366 87
367 87
368 87
369 87
370 87
371 87
372 87
373 87
374 87
375 87
376 87
377 87
378 87
379 87
380 87
381 87
382 87
383 87
384 87
385 87
386 87
387 87
388 87
389 87
390 87
391 87
392 87
393 87
394 87
395 87
396 87
397 87
398 87
399 87
400 87
401 87
402 87
403 87
404 87
405 87
406 87
407 87
408 87
409 87
410 87
411 87
412 87
413 87
414 87
415 87
416 87
417 87
418 87
419 87
420 87
421 87
422 87
423 87
424 87
425 87
426 87
427 87
428 87
429 87
430 87
431 87
432 87
433 87
434 87
435 87
436 87
437 87
438 87
439 87
440 87
441 87
442 87
443 87
444 87
445 87
446 87
447 87
448 87
449 87
450 87
451 87
452 87
453 87
454 87
455 87
456 87
457 87
458 87
459 87
460 87
461 87
462 87
463 87
464 87
465 87
466 87
467 87
468 87
469 87
470 87
471 87
472 87
473 87
474 87
475 87
476 87
477 87
478 87
479 87
480 87
481 87
482 87
483 87
484 87
485 87
486 87
487 87
488 87
489 87
490 87
491 87
492 87
493 87
494 87
495 87
496 87
497 87
498 87
499 87
500 87
501 87
502 87
503 87
504 87
505 87
506 87
507 87
508 87
509 87
510 87
511 87
512 87
513 87
514 87
515 87
516 87
517 87
518 87
519 87
520 87
521 87
522 87
523 87
524 87
525 87
526 87
527 87
528 87
529 87
530 87
531 87
532 87
533 87
534 87
535 87
536 87
537 87
538 87
539 87
540 87
541 87
542 87
543 87
544 87
545 87
546 87
547 87
548 87
549 87
550 87
551 87
552 87
553 87
554 87
555 87
556 87
557 87
558 87
559 87
560 87
561 87
562 87
563 87
564 87
565 87
566 87
567 87
568 87
569 87
570 87
571 87
572 87
573 87
574 87
575 87
576 87
577 87
578 87
579 87
580 87
581 87
582 87
583 87
584 87
585 87
586 87
587 87
588 87
589 87
590 87
591 87
592 87
593 87
594 87
595 87
596 87
597 87
598 87
599 87
600 87
601 87
602 87
603 87
604 87
605 87
606 87
607 87
608 87
609 87
610 87
611 87
612 87
613 87
614 87
615 87
616 87
617 87
618 87
619 87
620 87
621 87
622 87
623 87
624 87
625 87
626 87
627 87
628 87
629 87
630 87
631 87
632 87
633 87
634 87
635 87
636 87
637 87
638 87
639 87
640 87
641 87
642 87
643 87
644 87
645 87
646 87
647 87
648 87
649 87
650 87
651 87
652 87
653 87
654 87
655 87
656 87
657 87
658 87
659 87
660 87
661 87
662 87
663 87
664 87
665 87
666 87
667 87
668 87
669 87
670 87
671 87
672 87
673 87
674 87
675 87
676 87
677 87
678 87
679 87
680 87
681 87
682 87
683 87
684 87
685 87
686 87
687 87
688 87
689 87
690 87
691 87
692 87
693 87
694 87
695 87
696 87
697 87
698 87
699 87
700 87
701 87
702 87
703 87
704 87
705 87
706 87
707 87
708 87
709 87
710 87
711 87
712 87
713 87
714 87
715 87
716 87
717 87
718 87
719 87
720 87
721 87
722 87
723 87
724 87
725 87
726 87
727 87
728 87
729 87
730 87
731 87
732 87
733 87
734 87
735 87
736 87
737 87
738 87
739 87
740 87
741 87
742 87
743 87
744 87
745 87
746 87
747 87
748 87
749 87
750 87
751 87
752 87
753 87
754 87
755 87
756 87
757 87
758 87
759 87
760 87
761 87
762 87
763 87
764 87
765 87
766 87
767 87
768 87
769 87
770 87
771 87
772 87
773 87
774 87
775 87
776 87
777 87
778 87
779 87
780 87
781 87
782 87
783 87
784 87
785 87
786 87
787 87
788 87
789 87
790 87
791 87
792 87
793 87
794 87
795 87
796 87
797 87
798 87
799 87
800 87
801 87
802 87
803 87
804 87
805 87
806 87
807 87
808 87
809 87
810 87
811 87
812 87
813 87
814 87
815 87
816 87
817 87
818 87
819 87
820 87
821 87
822 87
823 87
824 87
825 87
826 87
827 87
828 87
829 87
830 87
831 87
832 87
833 87
834 87
835 87
836 87
837 87
838 87
839 87
840 87
841 87
842 87
843 87
844 87
845 87
846 87
847 87
848 87
849 87
850 87
851 87
852 87
853 87
854 87
855 87
856 87
857 87
858 87
859 87
860 87
861 87
862 87
863 87
864 87
865 87
866 87
867 87
868 87
869 87
870 87
871 87
872 87
873 87
874 87
875 87
876 87
877 87
878 87
879 87
880 87
881 87
882 87
883 87
884 87
885 87
886 87
887 87
888 87
889 87
890 87
891 87
892 87
893 87
894 87
895 87
896 87
897 87
898 87
899 87
900 87
901 87
902 87
903 87
904 87
905 87
906 87
907 87
908 87
909 87
910 87
911 87
912 87
913 87
914 87
915 87
916 87
917 87
918 87
919 87
920 87
921 87
922 87
923 87
924 87
925 87
926 87
927 87
928 87
929 87
930 87
931 87
932 87
933 87
934 87
935 87
936 87
937 87
938 87
939 87
940 87
941 87
942 87
943 87
944 87
945 87
946 87
947 87
948 87
949 87
950 87
951 87
952 87
953 87
954 87
955 87
956 87
957 87
958 87
959 87
960 87
961 87
962 87
963 87
964 87
965 87
966 87
967 87
968 87
969 87
970 87
971 87
972 87
973 87
974 87
975 87
976 87
977 87
978 87
979 87
980 87
981 87
982 87
983 87
984 87
985 87
986 87
987 87
988 87
989 87
990 87
991 87
992 87
993 87
994 87
995 87
996 87
997 87
998 87
999 87
1000 87
```

In the below example a query is constructed to validate a user's credentials for the purpose of logging in.

String filter = "(&(USER = " + user_name + ") (PASSWORD = " + user_password + "))";

An attacker can enter a crafted input for the variable user_name such as johnDoe(&) and any value for password the finished query will become (&(USER = johnDoe(&))(PASSWORD = pass)).

Only the first portion of this query is processed by the LDAP server (&(USER = johnDoe(&)), which always evaluates to true allowing the attacker to gain access to the system without needing to provide valid user credentials.

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5246>

<https://github.com/traccar/traccar/commit/e4f6e74e57ab743b65d49ae00f6624a20ca0291e>

https://en.wikipedia.org/wiki/LDAP_injection