

WWBN / AVideo Public

<> Code Issues 13 Pull requests Actions Projects Wiki Security a

Commit 8d8fc0c



Daniel Neto committed last week · ✓ 11 / 11

fix: Enhance SSRF protection by implementing DNS pinning in proxy requests

[GHSA-793q-xgj6-7frp](#)

master

1 parent [bf1c769](#) commit 8d8fc0c

2 files changed +114 -36 lines changed

↑ Top ⚙

Filter files...

- objects
 - functions.php
- plugin/LiveLinks
 - proxy.php

2 files changed +114 -36 lines changed

Search within code ⚙

objects/functions.php

```

@@ -4261,7 +4261,7 @@ function isSafeRedirectURL($url)
4261 4261     * @param string $url The URL to validate
4262 4262     * @return bool True if safe from SSRF, false otherwise
4263 4263     */
4264 - function isSSRFSafeURL($url)
+ function isSSRFSafeURL($url, &$resolvedIP = null)
4265 4265     {
4266 4266         global $global;
4267 4267         if (empty($url) || !is_string($url)) {
@@ -4317,17 +4317,19 @@ function isSSRFSafeURL($url)

```

			↑
4317	4317	return false;	
4318	4318	}	
4319	4319		
4320	-	// Resolve hostname to IP to check for DNS rebinding attacks	
4320	+	// Resolve hostname to IP to check for DNS rebinding attacks.	
4321	+	// \$resolvedIP (out-param) is set to the final validated IP so callers can	
4322	+	// pin the DNS resolution (e.g. via CURLOPT_RESOLVE) and eliminate TOCTOU races.	
4321	4323	\$ip = \$host;	
4322	4324	if (!filter_var(\$host, FILTER_VALIDATE_IP)) {	
4323	4325	// It's a hostname, resolve it	
4324	-	\$resolvedIP = gethostbyname(\$host);	
4325	-	if (\$resolvedIP === \$host) {	
4326	+	\$dnsResolved = gethostbyname(\$host);	
4327	+	if (\$dnsResolved === \$host) {	
4326	4328	// DNS resolution failed	
4327	4329	_error_log("isSSRFsafeURL: DNS resolution failed for: {\$host}");	
4328	4330	return false;	
4329	4331	}	
4330	-	\$ip = \$resolvedIP;	
4332	+	\$ip = \$dnsResolved;	
4331	4333	}	
4332	4334		
4333	4335	// Remove IPv6 brackets if present	
			↓
		@@ -4369,6 +4371,8 @@ function isSSRFsafeURL(\$url)	↑
4369	4371	}	
4370	4372	}	
4371	4373		
4374	+	// Expose the validated IP to the caller for DNS pinning.	
4375	+	\$resolvedIP = \$ip;	
4372	4376	return true;	
4373	4377	}	
4374	4378		
			↓

plugin/LiveLinks/proxy.php

...

↑

@@ -22,8 +22,11 @@

```

22 22     exit;
23 23 }
24 24
25 - // SSRF Protection: Block requests to internal/private networks
26 - if (!isSSRFSafeURL($_GET['livelink'])) {
27 + // SSRF Protection: Block requests to internal/private networks.
28 + // $resolvedIP receives the validated IP so we can pin it in the cURL call
    below,
29 + // eliminating the DNS TOCTOU race between validation and TCP connect.
30 + $resolvedIP = null;
31 + if (!isSSRFSafeURL($_GET['livelink'], $resolvedIP)) {
32     _error_log("LiveLinks proxy: SSRF protection blocked URL: " .
33     $_GET['livelink']);
34     echo "Access denied: URL targets restricted network";
35     exit;
36 @@ -32,43 +35,114 @@
37 header("Content-Type: video/vnd.mpegurl");
38 header("Content-Disposition: attachment;filename=playlist.m3u");
39
40 - $options = array(
41 -     'http' => array(
42 -         'user_agent' => 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',
43 -         "method" => "GET",
44 -         "header" => array("Referer: localhost\r\nAccept-language:
    en\r\nCookie: foo=bar\r\n"),
45 -         'follow_location' => 0,
46 -         'max_redirects' => 0,
47 -     )
48 - );
49 - $context = stream_context_create($options);
50
51 $_GET['livelink'] = addGlobalTokenIfSameDomain($_GET['livelink']);
52
53 - $headers = get_headers($_GET['livelink'], 1, $context);
54 - if (!empty($headers["Location"])) {
55 -     $redirectUrl = $headers["Location"];
56 -
57 -     // Validate the redirect target URL format and scheme before SSRF check

```

```

53     -     if (!filter_var($redirectUrl, FILTER_VALIDATE_URL) ||
54         -     !preg_match("/^https?:\\\/\\\/i", $redirectUrl)) {
55         -         _error_log("LiveLinks proxy: invalid redirect URL: " . $redirectUrl);
56         -         echo "Access denied: Invalid redirect URL";
57         -         exit;
58     }
59     + /**
60     +  * Fetch a URL through a DNS-pinned cURL request, following redirects manually
61     +  * so every hop is re-validated with isSSRFSafeURL() and re-pinned via
62     +  * CURLOPT_RESOLVE.
63     +  *
64     +  * This eliminates the DNS TOCTOU race: gethostbyname() is called once per hop
65     +  * (inside isSSRFSafeURL), and that same IP is forced into the TCP connection
66     +  * via
67     +  * CURLOPT_RESOLVE – no second DNS lookup can occur between validation and
68     +  * connect.
69     +  *
70     +  * @param string $url          The initial URL to fetch.
71     +  * @param string|null $pinnedIP The pre-validated IP returned by
72     +  * isSSRFSafeURL().
73     +  * @param int $maxRedirects    Maximum number of redirects to follow.
74     +  * @return array{content:string,finalUrl:string}|false
75     +  */
76     + function proxyDNSPinnedFetch($url, $pinnedIP, $maxRedirects = 5)
77     + {
78     +     if (!function_exists('curl_init')) {
79     +         // cURL unavailable: fall back to the non-pinned path and log the
80     +         // degradation.
81     +         // TOCTOU risk remains in this path; operators should ensure cURL is
82     +         // installed.
83     +         _error_log("LiveLinks proxy: cURL unavailable, DNS pinning disabled for
84     +         {$url}");
85     +         $content = fakeBrowser($url);
86     +         return $content !== false ? ['content' => $content, 'finalUrl' => $url]
87     +         : false;
88     +     }
89     + }
90     + }
91     + }
92     + }
93     + }
94     + }
95     + }
96     + }
97     + }
98     + }
99     + }
100    + }
101    + }
102    + }
103    + }
104    + }
105    + }
106    + }
107    + }
108    + }
109    + }
110    + }
111    + }
112    + }
113    + }
114    + }
115    + }
116    + }
117    + }
118    + }
119    + }
120    + }
121    + }
122    + }
123    + }
124    + }
125    + }
126    + }
127    + }
128    + }
129    + }
130    + }
131    + }
132    + }
133    + }
134    + }
135    + }
136    + }
137    + }
138    + }
139    + }
140    + }
141    + }
142    + }
143    + }
144    + }
145    + }
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   + }

```

```
61 -     _error_log("LiveLinks proxy: SSRF protection blocked redirect URL: " .
    $redirectUrl);
62 -     echo "Access denied: Redirect URL targets restricted network";
63 -     exit;

63 +     $currentUrl = $url;
64 +     $currentIP  = $pinnedIP;
65 +
66 +     for ($shop = 0; $shop <= $maxRedirects; $shop++) {
67 +         $host  = parse_url($currentUrl, PHP_URL_HOST);
68 +         $scheme = strtolower((string) parse_url($currentUrl, PHP_URL_SCHEME));
69 +         $port  = parse_url($currentUrl, PHP_URL_PORT) ?: ($scheme === 'https'
    ? 443 : 80);
70 +
71 +         $curlOpts = [
72 +             CURLOPT_URL           => $currentUrl,
73 +             CURLOPT_RETURNTRANSFER => true,
74 +             CURLOPT_FOLLOWLOCATION => false, // we handle each redirect
    ourselves
75 +             CURLOPT_SSL_VERIFYPEER => false,
76 +             CURLOPT_SSL_VERIFYHOST => 0,
77 +             CURLOPT_USERAGENT     => 'Mozilla/5.0 (Windows NT 10.0; Win64;
    x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169
    Safari/537.36',
78 +             CURLOPT_HTTPHEADER   => ['Referer: localhost', 'Accept-Language:
    en', 'Cookie: foo=bar'],
79 +             CURLOPT_HEADER       => true, // include response headers so
    we can read Location:
80 +         ];
81 +
82 +         // Pin the validated IP – CURLOPT_RESOLVE format: "hostname:port:ip"
83 +         // cURL uses this map instead of re-resolving the hostname, closing the
    TOCTOU window.
84 +         if (!empty($currentIP)) {
85 +             $curlOpts[CURLOPT_RESOLVE] = ["{$host}:{$port}:{$currentIP}"];
86 +         }
87 +
88 +         $ch = curl_init();
89 +         curl_setopt_array($ch, $curlOpts);
90 +         $response = curl_exec($ch);
91 +         $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
```

```

92 +     $headerSize = curl_getinfo($ch, CURLINFO_HEADER_SIZE);
93 +     curl_close($ch);
94 +
95 +     if ($response === false || $statusCode === 0) {
96 +         _error_log("LiveLinks proxy: CURL fetch failed for {$currentUrl}");
97 +         return false;
98 +     }
99 +
100 +     $responseHeaders = substr($response, 0, $headerSize);
101 +     $body            = substr($response, $headerSize);
102 +
103 +     if ($statusCode >= 300 && $statusCode < 400) {
104 +         // Parse the single Location header from the raw response headers.
105 +         if (!preg_match('/^Location:\s*(.+)$/im', $responseHeaders, $m)) {
106 +             _error_log("LiveLinks proxy: 3xx with no Location header at
107 + {$currentUrl}");
108 +             return false;
109 +         }
110 +         $redirectUrl = trim($m[1]);
111 +
112 +         // Validate redirect target and get its pinned IP for the next hop.
113 +         $nextIP = null;
114 +         if (!isSSRFSafeURL($redirectUrl, $nextIP)) {
115 +             _error_log("LiveLinks proxy: SSRF protection blocked redirect
116 + to: {$redirectUrl}");
117 +             return false;
118 +         }
119 +         $currentUrl = $redirectUrl;
120 +         $currentIP  = $nextIP;
121 +         continue;
122 +     }
123 +     return ['content' => $body, 'finalUrl' => $currentUrl];
64 124     }
65 125
66 -     $_GET['livelink'] = $redirectUrl;
67 -     $urlinfo = parse_url($_GET['livelink']);
68 -     $content = fakeBrowser($_GET['livelink']);
126 +     _error_log("LiveLinks proxy: too many redirects for {$url}");

```

```
127 +     return false;
128 + }
129 +
130 + $fetchResult = proxyDNSPinnedFetch($_GET['livelink'], $resolvedIP);
131 + if ($fetchResult === false) {
132 +     echo "Access denied or fetch failed";
133 +     exit;
134 + }
135 +
136 + $content = $fetchResult['content'];
137 + $finalUrl = $fetchResult['finalUrl'];
138 +
139 + // Preserve the original base-URL logic for relative path resolution in m3u8
    playlists.
140 + if ($finalUrl !== $_GET['livelink']) {
141 +     // URL was redirected – use scheme://host:port as the base (no trailing
    path).
142 +     $urlinfo = parse_url($finalUrl);
69 143     $_GET['livelink'] = "{$urlinfo["scheme"]}://{$urlinfo["host"]}:"
    {$urlinfo["port"]}";
144 +     unset($pathinfo);
70 145 } else {
71 -     $content = fakeBrowser($_GET['livelink']);
72 146     $pathinfo = pathinfo($_GET['livelink']);
73 147 }
74 148 if($content === "Empty Token"){
    ↓
```

Comments 0



Please [sign in](#) to comment.