New issue

Jump to bottom

Server-Side Request Forgery <iframe src='file:///etc/passwd'> (in version 6.3.0) #261

Closed **etsms** opened this issue on Nov 24, 2020 · 8 comments · Fixed by #319

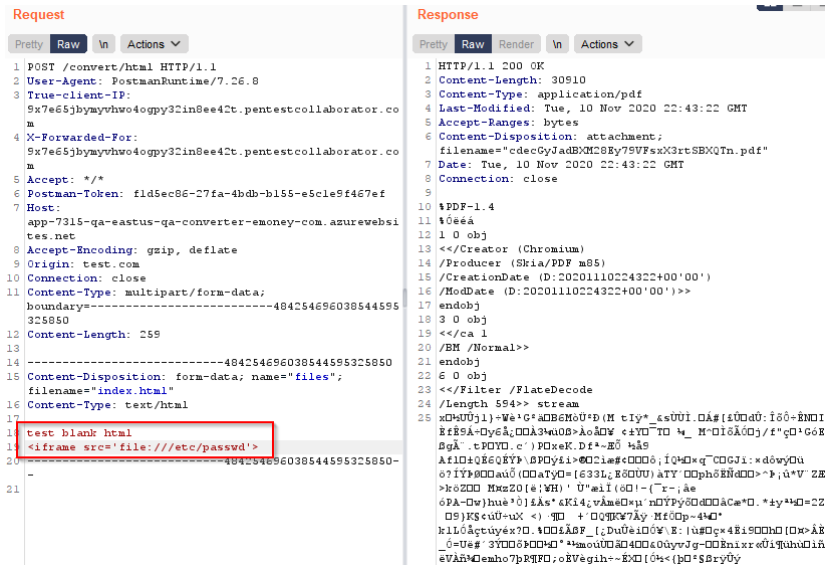
etsms commented on Nov 24, 2020

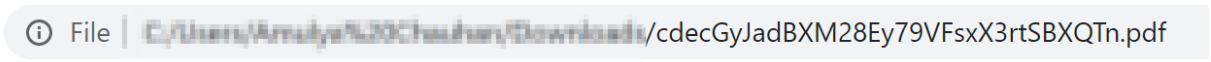
Expected Behavior

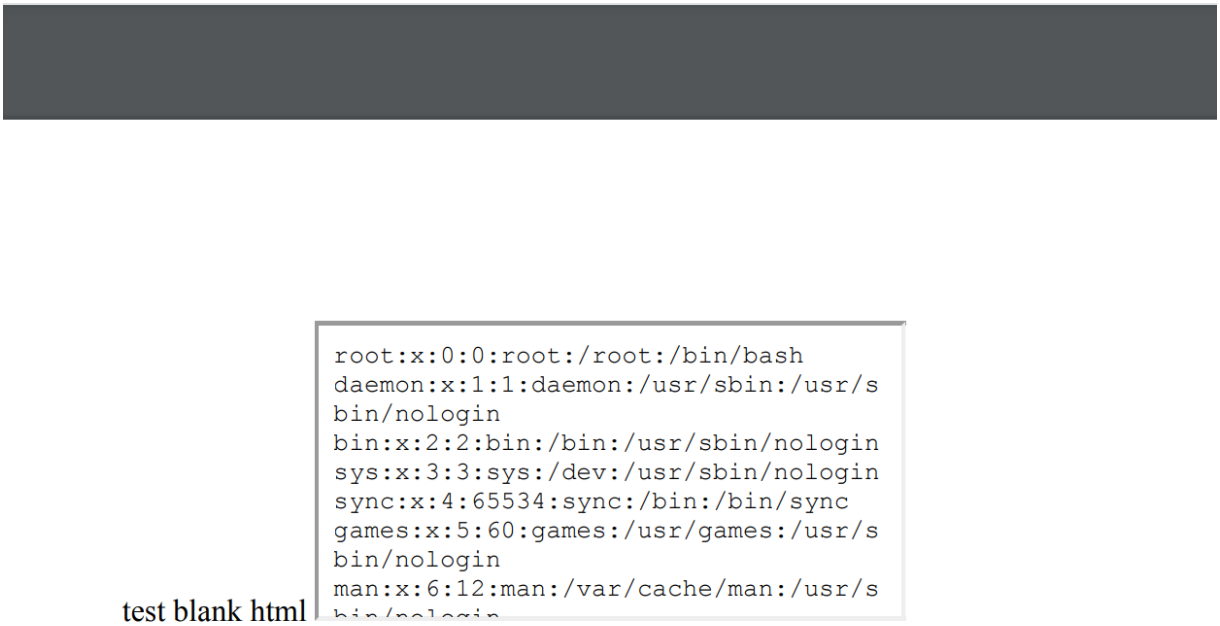
- When attempting to convert html into a PDF using the /html endpoint, gotenberg should throw an exception (respond with a bad request) whenever there is an attempt to convert html with a source (src) reference to an internal system file.
- [HTTP POST] /convert/html
- [BODY] file: index.html >> test blank html <iframe src='file:///etc/passwd'>
- [RESPONSE] 400 Bad Request (request was improperly formed - or something similar)

Current Behavior

- [HTTP POST] /convert/html
- [BODY] file: index.html >> test blank html <iframe src='file:///etc/passwd'>
- [RESPONSE] (contents of the etc/password directory







Possible Solution

- Respond with a bad request. Or something that indicates the requested operation is not allowed or prohibited.
- Check the contents of the HTML file being uploaded to determine if similar file:// protocols are referenced

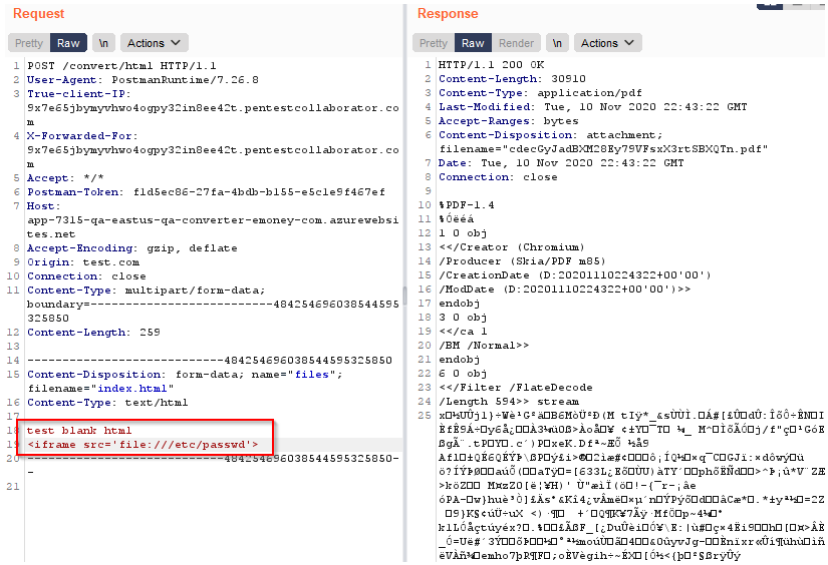
The use of server-side requests must be limited to specific scenarios required by the application and servers must be hardened to only allow those requests. SSRF can be remediated using a variety of standard web application protections, such as:

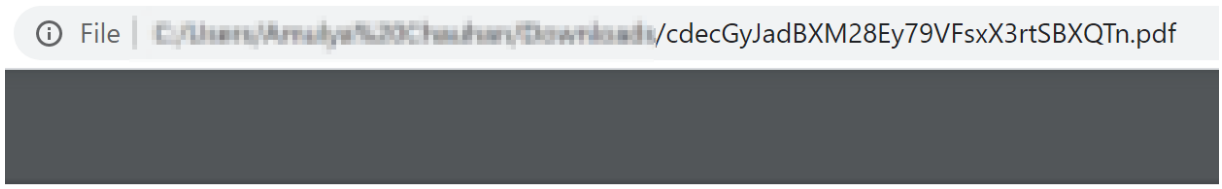
1. Limit service types via input validation on user input: Analyze if the application must use URI/ URL in requests and if not, disallow them. When a use case requires URIs and URLs in requests, perform input validation on URI/URLs using whitelists or robust blacklists. For example, only allow URL schemes such as HTTP/HTTPS which are often required for accessing applications. Disable all other URL schemas such as file://, dict://, gopher://, ftp://. Alternatively, create a token based forwarding for SSRs which may eliminate the use of URIs altogether.
2. Limit service interactions: Carefully analyze if the server requires interactions with external servers. Based on the use case, prevent unintended behavior in one of the following ways: External service interactions required: If the server is to perform interactions with external services, the server must be hardened to safely eliminate calls to services available on the local loopback adapter and prevent any calls from the server to other internal systems.
3. External service interactions not required: If the server is to only interact with local services, configure outbound firewall rules to check any calls being made from the application. The rules must limit the web server's interaction with other hosts by routing traffic only to specified resources on the local network; i.e., web servers must disallow connections to any host not included in the whitelist.

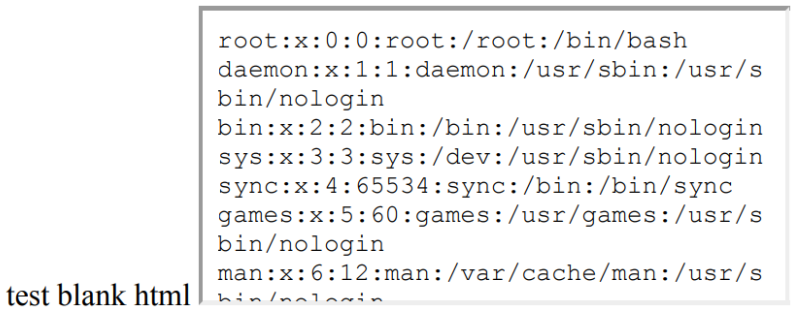
Steps to Reproduce (for bugs)

1. Configure Postman to use a proxy tool such as Burp Suite.
2. Create an HTML file using the below data: test blank html <iframe src='file:///etc/passwd'>
3. Toggle Burp Intercept on.
4. Upload this HTML file and make a request to the "Converter" endpoint.
5. Forward the intercepted request to Repeater and hit send.
6. In Repeater > response section, right click and select "show response in browser".

7. Open the response in a browser.
8. Note that the PDF file contains contents of /etc/passwd file.







Context

- We'd like to hide information about the server and not allow attackers to steal sensitive data

Your Environment

- Azure
- Linux Container
- OS version: Unix 4.15.0.112
- 64 bit system: True
- 64 bit process: True
- Processor count: 1

gulien commented on Nov 25, 2020 — Collaborator

Hello **@etsms**,

Thanks for this well-written issue!

It's already on my TODO list 👍 AFAIK, it would be possible to check each requests done by Google Chrome and prevent loading "unsafe" ones.

🎉 1

etsms commented on Nov 25, 2020 — Author

@gulien - great to hear! Thanks for the update. Looking forward to testing again.

etsms commented on Dec 1, 2020 — Author

@gulien - Question for ya... Our security team is wondering about a timeline for this patch. Do you or the gotenburg team have a timeline for this?

gulien commented on Dec 2, 2020 — Collaborator

No because I'm mostly working on Gotenberg on my free time so..

Anyway, Gotenberg should always be called from a trusted source 🤔

👍 2

serge-melis commented on Jan 13, 2021

Is this not mitigated by the fact that, that the process runs inside of the docker container? Which has no access to anything outside of the container?

Zettersten commented on Jan 13, 2021

@serge-melis yes exactly. I had to prove to our security teams that it lives in a container and is short-lived. I was able to reduce the criticality of the finding by communicating that much. However, it'll stay a vulnerability until the software is patched (or updated) with a feature to disable local file navigation in the chrome driver (eg. no access to the etc/passwords path). So that's good news. As **@gulien** said earlier, its not (should not) publicly accessible and the client/requesters should be known and authorized.

So the risk is pretty low at this point.

BUT - a savvy hacker may find away around those other controls and ultimately pull/read sensitive information from the container. Hypothetical.... but this is the way security folks think. I'm certainly not a hacker pro.

👍 2

NoRelect mentioned this issue on Mar 1, 2021

local file inclusion via /convert/url remoteURL endpoint #283

Closed

abergmann commented on Mar 2, 2021

CVE-2021-23345 was assigned to this issue.

yumauri commented on Mar 11, 2021 • edited

Wow, that's an interesting one :)
If Chrome doesn't support such filtering out of the box, I think it could be done using some sort of proxy for Chrome. Like Privoxy, for example. But I don't know, does Chrome use proxy settings for `file` protocol?

gulien mentioned this issue on Aug 22, 2021

7.0.0 #319

Merged

gulien closed this as completed in #319 on Aug 22, 2021

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

7.0.0
gotenberg/gotenberg

6 participants