**Tails - Feature #10951**

**Remove kernel .map files**

01/16/2016 03:06 PM - anonym

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
<th>Start date:</th>
<th>01/16/2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Target version:</td>
<td>Tails_2.4</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Feature Branch:</td>
<td>feature/10951-remove-kernel-map-files</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Type of work:</td>
<td>Code</td>
<td>Starter:</td>
<td></td>
</tr>
<tr>
<td>Blueprint:</td>
<td></td>
<td>Affected tool:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

We currently ship files like `/boot/System.map-${KERNEL_VERSION}-${ARCH}` which contains the memory addresses of kernel functions, making the life of malware a bit easier.

We should remove these, since their only purpose is kernel debugging (right?), and they increase the Tails image size (uncompressed the two we have are ~4.5 MiB).

**Associated revisions**

**Revision fa62a46e** - 02/16/2016 06:09 PM - anonym

Remove the kernel .map files.

These are only useful for kernel debugging (and slightly make things easier for malware, perhaps) and otherwise just occupy disk space.

**Will-fix:** #10951

**Revision 4adfe61** - 05/15/2016 11:15 AM - anonym

Document motivation for not exposing kernel memory addresses.

Refs: #10951

**Revision 98a18593** - 05/16/2016 07:26 PM - intrigeri

Merge remote-tracking branch 'origin/feature/10951-remove-kernel-map-files' into devel

Fix-committed: #10951

**History**

**#1** - 01/16/2016 03:07 PM - anonym

Actually, upstream Debian should perhaps split them into separate -dbg packages?

**#2** - 01/16/2016 03:33 PM - intrigeri

- Assignee set to anonym

I think the security argument is pretty weak, given 1. we use standard Debian kernels, whose symbols' address is public information; 2. there are tools like [https://github.com/jonoberheide/ksymhunter](https://github.com/jonoberheide/ksymhunter) around. Still, every small bit we do to make exploitation easier is welcome, I guess.

05/06/2020  1/5
I'm fine with removing them on the grounds that it'll save ISO & IUK size.

anonym: assigning to you since you set a target version; if you don't want it, then either drop the target version, or find a volunteer :)

---

intrigeri wrote:

I think the security argument is pretty weak, given 1. we use standard Debian kernels, whose symbols' address is public information;

Of course -- the original (private) bug report made the case that off the shelf malware might not have this info bundled, so this increases the size or complexity (if it has to identify what to fetch from some public source) a bit, which doesn't hurt. :)

2. there are tools like https://github.com/jonoberheide/ksymhunter around.

Yeah, yeah... :)

I'm fine with removing them on the grounds that it'll save ISO & IUK size.

Any decrease in the cruft we ship is a welcome change in itself, imho.

anonym: assigning to you since you set a target version; if you don't want it, then either drop the target version, or find a volunteer :)

I'll keep it.

---

#4 - 02/16/2016 06:08 PM - anonym
- % Done changed from 0 to 10
- Type of work changed from Research to Code
I do not really see why this needs any research. Implementation-wise we do `rm -f /boot/*.map /boot/*.map-*` in some appropriate hook.

#5 - 02/16/2016 06:10 PM - anonym
- Status changed from Confirmed to In Progress

Applied in changeset fa62a46eb21b509f28aabc4df0101e1a37a1cc9.

#6 - 02/16/2016 06:11 PM - anonym
- Assignee deleted (anonym)
- % Done changed from 10 to 40
- Feature Branch set to feature/10951-remove-kernel-map-files

Untested.

#7 - 02/16/2016 06:55 PM - anonym
- % Done changed from 40 to 50
- QA Check set to Ready for QA

Tested now. It works as it should, i.e. the .map files are removed, but the truncation we do on the resulting ISO image made the resulting image the same size as the previous devel one I built. :)

#8 - 02/24/2016 06:54 PM - anonym
- Assignee set to intrigeri

I doubt there's any one else that will take this one than you, intro. :S

Given the limited scope of this branch, I'd happily give it a freeze exception and get it into the final 2.2 image. Perhaps you can just throw it into the mix when automatically testing some other branch? Otherwise, unassign yourself and we deal with it later, possibly even for 2.3 (imho cleanups like this is more of a bugfix than a feature).

#9 - 02/26/2016 11:37 AM - intrigeri
- Assignee deleted (intrigeri)

anonym wrote:

I doubt there's any one else that will take this one than you, intro. :S

I won't have time to deal with this for 2.2, and not sure about 2.3, especially since I've taken over some more reviewing duties that were initially assigned to someone else. I suggest you ask other people to book some time for reviews for the next cycle (actually, until 2.5 included).

#10 - 03/15/2016 11:25 AM - anonym
- Target version changed from Tails_2.2 to Tails_2.3
If you remove these files, make sure the kernel.kptr_restrict sysctl is set to 2, otherwise it's almost completely pointless.

anonym, can you please give it a look?

intrigeri wrote:

If you remove these files, make sure the kernel.kptr_restrict sysctl is set to 2, otherwise it's almost completely pointless.

anonym, can you please give it a look?

I had a look, and it makes sense, see 5243e64 for details; personally I mostly wanted this for removing cruft, but if this change doesn't break anything for us, why not? :) Let's see how the automated test suite runs fare.

Now we just have to write a kernel module to hook into the uname() syscall so it doesn't leak the kernel we're using, so an attacker can't just download the corresponding .map files online. ;)

But in all seriousness, it does sound like a good idea. At least it increases the price of exploits, because out-of-the-box exploits are less likely to work, and many of the adversaries of Tails users will be using cheap exploits that simply check for /proc/kallsyms and /boot/System.map, and then bail if they don't exist.
cypherpunks wrote:

Now we just have to write a kernel module to hook into the uname() syscall so it doesn't leak the kernel we're using, so an attacker can't just download the corresponding .map files online. ;)

Even better: what if it **faked** the wrong version (e.g. randomly picked on boot) for **extra** confusion? :)

But in all seriousness, it does sound like a good idea. At least it increases the price of exploits, because out-of-the-box exploits are less likely to work, and many of the adversaries of Tails users will be using cheap exploits that simply check for /proc/kallsyms and /boot/System.map, and then bail if they don't exist.

Agreed. We'll merge this as long as there are no truly legit use cases for having these available in Tails, which seems highly likely.

This branch has passed the automated test suite on Jenkins, and I just pushed a corresponding design update. intrigeri, since you already has been involved on this ticket, can you do the review'n'merge?