Security Now! #542 - 01-12-16

Q&A #227

This week on Security Now!

- TrendMicro drastically lowers the bar on "You're doing it wrong"
- Symantec issues banned SHA-1 Certs in 2016! Whoops!
- Meanwhile... Firefox backs off from disallowing newly issued SHA-1 certs in 2016.
- A sad day has finally arrived (today) for Windows XP Embedded SP3.
- How LastPass v4.0's new Emergency Access Feature can be TNO.
- Mark Russinovich's SigCheck v2.4
- A bunch of miscellany... a happy SpinRite story, and 10 questions & comments from our terrific listeners

Security News:

TrendMicro: "You're Doing It Wrong" hardly begins to describe it!

- via: Tavis Ormandy / Google
- https://code.google.com/p/google-security-research/issues/detail?id=693
- TrendMicro Antivirus for Windows, by default installs a Password Manager component which is automatically launched at 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.

- Tavis writes:
  It took about 30 seconds to spot one that permits arbitrary command execution, openUrlInDefaultBrowser, which eventually maps to 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 executed).

- Tavis also notes:
  TrendMicro helpfully adds a self-signed https certificate for localhost to the trust store, so you don't need to click through any security errors.

  Some back and forth ensues with screen shots. Then TrendMicro sends Tavis an updated version.

- Tavis writes:
  TrendMicro sent me a build to verify they had fixed the problem, it looks like they're no longer using ShellExecute, so it fixes the 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 tell them I'm not going to through them, but that they need to hire a professional security consultant to audit it urgently.

- More back and forth, Tavis sends to TM:
  Thanks Jean, I ran this on top of a TrendMicro Maximum Security 10 installation, and it looks like this fixes the most critical problem. Honestly, this thing still looks pretty fragile, I haven't looked through the dozens of other API's you're exposing - and 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";
  ```

  (These are 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.

- Tavis posts to the bug log:
  I happened to notice that the /api/showSB endpoint will spawn an ancient build of Chromium (version 41) with --disable-sandbox. To 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".

- Tavis sends to TrendMicro:
  I spent a few minutes trying to understand how the SB shell worked, and then realized they were just hiding the global objects. I 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 fact that you 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 an audit 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 passwords to 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.

- Trend Micro replies:
  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.

- Tavis then wrote and posts a working exploit:
  I noticed that there is a nice clean API for accessing passwords stored in the password manager, so anyone can just read all of the stored passwords:

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

  Users are prompted on installation to export their browser passwords, but that's optional. I think an attacker can force it with /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 an external consultancy to audit the code. In my experience dealing with security vendors, users are quite forgiving of mistakes if vendors act quickly to protect them once informed of a problem, I think the worst thing you can do is leave users exposed while you clean this thing up. The choice is yours, of course.

- TrendMicro thanks Tavis for pointing these things out…

- Tavis replies:
  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 exported all 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 password 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 an external consultancy to audit the code. In my experience dealing with security vendors, users are quite forgiving of mistakes if vendors act quickly to protect them once informed of a problem, I think the worst thing you can do is leave users exposed while you clean this thing up. The choice is yours, of course.

  Then the thread went public and there was a bunch of back and forth among

Symantec issues SHA-1 certs in 2016. Oops!

- https://cabforum.org/pipermail/public/2016-January/006519.html
- Summary:
  Symantec has identified a gap involving a limited use case on one of our platforms that allowed the issuance of five SHA-1 certificates after 31 December, 2015. We have released a patch that addresses this issue and are in the process of notifying customers and revoking the affected certificates.

  Details:
  Symantec maintains an enterprise portal in which customer administrators can approve or reject enrollments for certificates with domain and organization names that have already

been pre-vetted by Symantec, without requiring further Symantec review. We identified a gap in a specific use case where the code to block SHA-1 issuance after 31 December 2015 was not in place - specifically, where the certificate was enrolled in 2015 but approved in 2016.

Remediation:
We have updated our production code to perform the SHA-1 check for this use case. We have confirmed that the appropriate SHA-1 checks are in place on this platform and other platforms. Customers are being notified and the affected certificates are in the process of being revoked.

Meanwhile... filed under the category of "unintended consequences and side effects"

- Firefox back off from disallowing SHA-1 certs in 2016!... why??
- 43.0.3 --> 43.0.4
- https://bugzilla.mozilla.org/show_bug.cgi?id=1236975
  In Bug 942515, we configured Firefox to reject SHA-1 certificates with a notBefore date after 2016-01-01. That appears to be causing some users with MitM software installed to be unable to access *any* HTTPS sites.
  https://groups.google.com/d/topic/mozilla.dev.platform/ZNKxYgIk_Sg/discussion
  In order to enable measurement of the scope of this risk, we should (temporarily) change the default preference to accept all valid SHA-1 certificates, regardless of issance date.

- Google Group Commentary:
  Starting January 1st 2016 (a few days ago), Firefox rejects recently-issued SSL certs that use the (obsolete) SHA1 hash algorithm.[1]

  For users who unknowingly have a local SSL proxy on their machine from spyware/adware/antivirus (stuff like superfish), this may cause *all* HTTPS pages to fail in Firefox, if their spyware uses SHA1 in its autogenerated certificates. (Every cert that gets sent to Firefox will use SHA1 and will have an issued date of "just now", which is after January 1 2016; hence, the cert is untrusted, even if the spyware put its root in our root store.)

  I'm not sure what action we should (or can) take about this, but for now we should be on the lookout for this, and perhaps consider writing a support article about it if we haven't already. (Not sure there's much help we can offer, since removing spyware correctly/completely can be tricky and varies on a case by case basis.)

  (Context: I received a family-friend-Firefox-support phone call today, who this had this exact problem. Every HTTPS site was broken for her in Firefox, since January 1st. IE worked as expected (that is, it happily accepts the spyware's SHA1 certs, for now at least). I wasn't able to remotely figure out what the piece of spyware was or how to remove it -- but the rejected certs reported their issuer as being "Digital Marketing Research App" (instead of e.g. Digicert or Verisign). Googling didn't turn up anything useful, unfortunately; so I suspect this is "niche" spyware, or perhaps the name is dynamically generated.)

Anyway -- I have a feeling this will be somewhat-widespread problem, among users who have spyware (and perhaps crufty "secure browsing" antivirus tools) installed.

~Daniel

Hackers Install Free SSL Certs from Let's Encrypt On Malicious Web Sites

- http://thehackernews.com/2016/01/free-ssl-certificate-malware.html

Windows XP SP3 - Embedded Extended Support ends TODAY!
http://betanews.com/2016/01/11/windows-xp-embedded-service-pack-3-dies-tomorrow/

LastPass v4.0 Emergency Access -- How can it be TNO?

- All parties in the scenario have their own RSA key pair:
  - A public key which allows others to encrypt data which only they can decrypt.
  - A matching private key, which never leaves their control, which they can use to decrypt what was encrypted with their published public key.
- The master key used to encrypt and decrypt the user's Lastpass vault is then encrypted with the Emergency Access contact's public key... allowing it to ONLY be subsequently decrypted by their matching private key.
- Lastpass stores and holds each blob, each encrypted by each Emergency Access contact's public key.
- If the Emergency Access conditions are met, Lastpass sends each encrypted blobs to each Emergency Access party for decryption... and their subsequent access to the protected vault.

Mark Russinovich's SigCheck v2.4

- https://technet.microsoft.com/en-us/sysinternals/bb897441.aspx
- Sigcheck is a command-line utility that shows file version number, timestamp information, and digital signature details, including certificate chains. It also includes an option to check a file's status on VirusTotal, a site that performs automated file scanning against over 40 antivirus engines, and an option to upload a file for scanning.

- One way to use the tool is to check for unsigned files in your \Windows\System32 directories with this command:
  sigcheck -u -e c:\windows\system32
  You should investigate the purpose of any files that are not signed.

- -t[u][v] Dump contents of specified certificate store ('*' for all stores).
  Specify -tu to query the user store (machine store is the default).
  Append '-v' to have Sigcheck download the trusted Microsoft root certificate list and only output valid certificates not rooted to a certificate on that list. If Microsoft's online site is not accessible, authrootstl.cab or authroot.stl in the current directory are used instead, if present.

Errata:

Shawn (@oshae)

- You mentioned that WiFi uses CDMA. It's actually CSMA/CD. Ethernet uses collision detection, while WiFi uses collision avoidance. :)
- CSMA/CD: Collision Sense Multiple Access / Collision Detection.
- CSMA/CA: Collision Sense Multiple Access / Collision Avoidance. (WiFi transmitters cannot simultaneously listen!)

Miscellany:

- SyFy: "The Expanse" has turned out to be a huge win.
- SyFy: "The Magicians"
- Seen "Star Wars" twice.
- Showtime: Billions
- "The Big Short" - intensely disliked
- Bragi Dash ($199)

SpinRite

From: "Mark Clark"
Subject: SpinRite helped a retired vet and saved my wife's sanity
X-Location: Portland Oregon

Dear Steve and Leo blah blah blah,

My wife runs a nonprofit housing company and one of the tenants tends to visit the office when he has nothing better to do. His computer was not working so he was spending a lot of time in the office.

I happened to be in the office setting up a new computer and my wife asked me if there was a computer in the office she could give him. I asked him what was wrong and he said his screen was red and Windows would not boot.

So I told him to bring his computer to the office and I would look at it. I got out my trusted copy of SpinRite and let it go to work. I did notice the screen was not getting all the colors but I figured it was because I didn't have the video plug all the way in. SpinRite ran and had about 8,000,000 ecc errors and 600 seek errors, but otherwise it didn't complain.

Once SpinRite was finished Windows booted up just fine and let him know he could have his computer back, it was working again. He took it back to his apartment and when he tried to connect the video, the port fell completely off the motherboard. I told him to give it back to me and I would fix it. I went to the computer store and picked up an inexpensive video card. After installing the video card, I re-ran SpinRite on level four just to make sure. Now he has his computer back, which means he's not spending time in the office annoying the staff... and saving my wife's sanity. Thank you for a great product!

Mark Clark