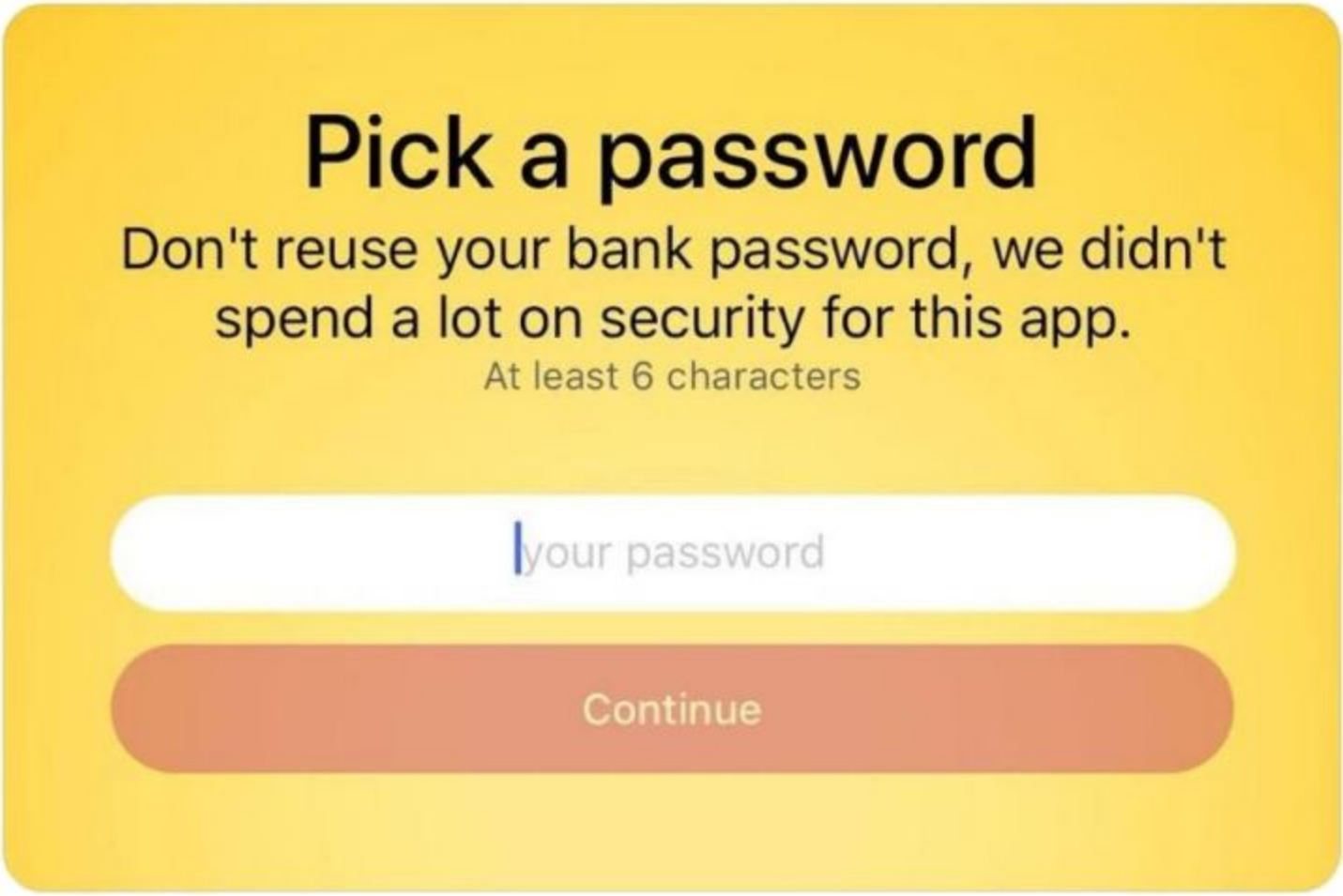
Security Now! #686 - 10-23-18

Libssh's Big Whoopsie!

This week on Security Now!

This week a widely used embedded OS (FreeRTOS) is in the doghouse, as are at least eight D-Link routers which have serious problems most of which D-Link has stated will never be patched. We look at five new problems in Drupal 7 and 8, two of which are rated critical, trouble with Live Networks RTSP streaming server, still more trouble with the now-infamous Windows 10 Build 1809 feature update, and a long standing 0-day in the widely used and most popular plugin for jQuery. We then look at what can only be described as an embarrassing mistake in the open source libssh library, and we conclude by examining a fun recent hack and pose its solution to our audience as our Security Now! puzzler of the week!



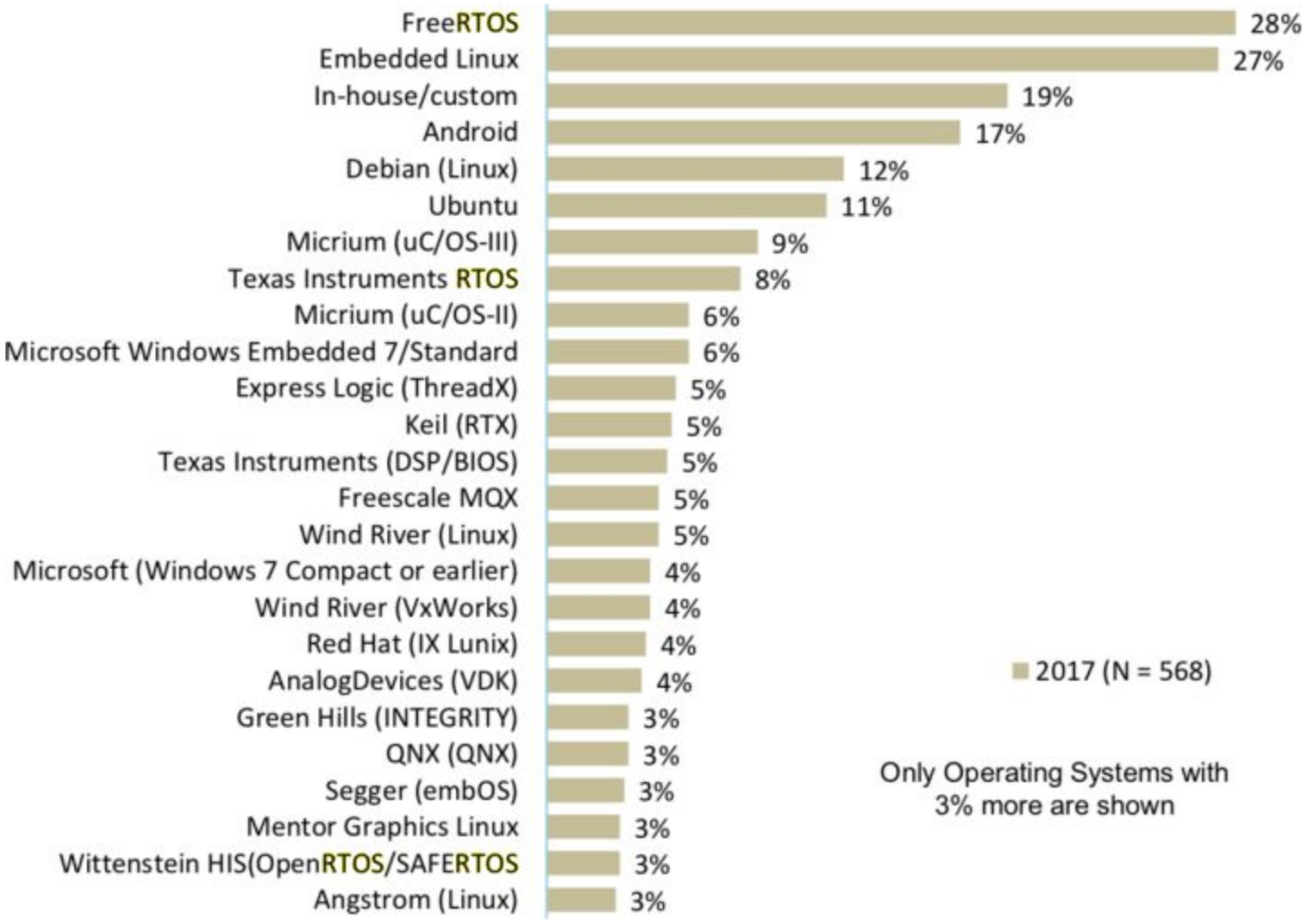
Security News

FreeRTOS in the doghouse
Last Thursday's blog posting by Zimperium Labs was titled: "FreeRTOS TCP/IP Stack Vulnerabilities Put A Wide Range of Devices at Risk of Compromise: From Smart Homes to Critical Infrastructure Systems"
https://blog.zimperium.com/freertos-tcpip-stack-vulnerabilities-put-wide-range-devices-risk-compromise-smart-homes-critical-infrastructure-systems/

Affected Operating Systems:
- FreeRTOS up to V10.0.1 (with FreeRTOS+TCP),
- AWS FreeRTOS up to V1.3.1,
- Wittenstein High Integrity Systems OpenRTOS and SafeRTOS (With TCP/IP components) .

Zimperium sets the stage with some background on FreeRTOS:

As a part of our ongoing IoT platform research, zLabs recently analyzed some of the leading operating systems in the IoT market, including FreeRTOS. FreeRTOS is a market leader in the IoT and embedded platforms market, being ported to over 40 hardware platforms over the last 14 years. In November 2017, Amazon Web Services (AWS) took stewardship for the FreeRTOS kernel and its components.



Previous page: Answers to questionaire about future plans for embedded OS deployment.

AWS FreeRTOS aims to provide a fully enabled IoT platform for microcontrollers, by bundling the FreeRTOS kernel together with the FreeRTOS TCP/IP stack, modules for secure connectivity, over the air updates, code signing, AWS cloud support, and more.

With the infrastructure that AWS provides, and the AWS FreeRTOS platform, developers can focus solely on innovation, thus reducing development time and costs.

There is also a commercial version of FreeRTOS, named OpenRTOS and maintained by WITTENSTEIN high integrity systems. Wittenstein also offers a safety-oriented RTOS named SafeRTOS, that is based on the functional model of FreeRTOS, and is certified for use in safety critical systems.

FreeRTOS and SafeRTOS have been used in a wide variety of industries: IoT, Aerospace, Medical, Automotive, and more. Due to the high risk nature of devices in some of these industries, zLabs decided to take a look at the connectivity components that are paired with these OS's. Clearly, devices that have connectivity to the outside world are at a higher degree of risk of being attacked.

During our research, we discovered multiple vulnerabilities within FreeRTOS's TCP/IP stack and in the AWS secure connectivity modules. The same vulnerabilities are present in WHIS Connect TCP/IP component for OpenRTOS\SafeRTOS.

These vulnerabilities allow an attacker to crash the device, leak information from the device's memory, and remotely execute code on it, thus completely compromising it.

We disclosed these vulnerabilities to Amazon, and collaborated (and continue to do so) with them to produce patches to the vulnerabilities we detected.

The patches were deployed for AWS FreeRTOS versions 1.3.2 and onwards. We also received confirmation from Wittenstein that they were exposed to the same vulnerabilities, and those were patched together with Amazon.

Since this is an open source project, we will wait 30 days before publishing technical details about our findings, to allow smaller vendors to patch the vulnerabilities.

Zimperium's Ori Karliner, who conducted the research discovered FOUR =critical= RCE (remote code execution) vulnerabilities, one denial of service, seven information leaks and one other which was unspecified.

Amazon FreeRTOS consists of the following components:

- A microcontroller operating system based on the FreeRTOS kernel
- Amazon FreeRTOS libraries for connectivity, security, and over-the-air (OTA) updates.
- A configuration wizard that allows you to download a zip file that contains everything you need to get started with Amazon FreeRTOS.
- Over-the-air (OTA) Updates.

The FreeRTOS Kernel
The FreeRTOS kernel is a real-time operating system kernel that supports numerous architectures and is ideal for building embedded microcontroller applications.

The kernel provides:
- A multitasking scheduler.
- Multiple memory allocation options (including the ability to create statically allocated systems).
- Inter-task coordination primitives, including task notifications, message queues, multiple types of semaphores, and stream and message buffers.

A lightweight RTOS is not really what we think of as a modern Operating System. It is designed for lightweight embedded applications where memory is measured in "Kilos" rather than "Megas" or "Gigas." As such an RTOS can be thought of as a "thread of execution" coordinator. It schedules the execution of a device's multiple threads of execution. It doles out memory to requesting threads as needed. It manages inter-thread communication queues. It provides mechanisms for threads to obtain exclusive access to sharable resources... and that's about all.

If you're going to have code running in a lightbulb, it's going to be running on an RTOS.

Amazon FreeRTOS Over-the-Air Updates:
Over-the-air (OTA) updates allow you to deploy files to one or more devices in your fleet. Although OTA updates were designed to be used to update device firmware, you can use them to send any files to one or more devices registered with AWS IoT. When you send files over the air, it is a best practice to digitally sign them so that the devices that receive the files can verify they have not been tampered with en route. You can use Code Signing for Amazon FreeRTOS to sign and encrypt your files or you can sign your files with your own code-signing tools.

When you create an OTA update, the OTA Update Manager Service creates an AWS IoT job to notify your devices that an update is available. The OTA demo application runs on your device and creates an Amazon FreeRTOS task that subscribes to notification topics for AWS IoT jobs and listens for update messages. When an update is available, the OTA agent publishes requests to AWS IoT streaming topics and receives file blocks using the MQTT protocol. It reassembles the blocks into files and checks the digital signature of the downloaded files. If the files are valid, it installs the firmware update. If you are not using the Amazon FreeRTOS OTA Update demo application, you must integrate the OTA Agent Library into your own application to get the firmware update capability.

Amazon FreeRTOS over-the-air updates make it possible for you to:
- *Digitally sign and encrypt firmware before deployment.*
- *Deploy new firmware images to a single device, a group of devices, or your entire fleet.*
- *Deploy firmware to devices as they are added to groups, reset, or reprovisioned.*
- *Verify the authenticity and integrity of new firmware after it's deployed to devices.*
- *Monitor the progress of a deployment.*
- *Debug a failed deployment.*

Amazon is a case study in how to do modern IoT deployment correctly.

In a unfortunate example of how **NOT** to do IoT deployment correctly… we have D-Link:

The D-Link Trio: Multiple D-Link routers are open to a complete takeover by a simple combined attack.

A Polish researcher at the Silesian University of Technology, in Poland, discovered and reported a trio of vulnerabilities in eight widely different D-Link routers last May. However, D-Link reportedly informed him that only two of the eight routers (the DWR-116 and 111) would ever be patched because the rest have reached end-of-life and will no longer be supported.

https://seclists.org/fulldisclosure/2018/Oct/36

- DWR-116 through v1.06,
- DIR-140L through v1.02,
- DIR-640L through v1.02,
- DWR-512 through v2.02,
- DWR-712 through v2.02,
- DWR-912 through v2.02,
- DWR-921 through v2.02,
- DWR-111 through v1.01,
- ... and VERY LIKELY others with the same type of firmware.

THREE vulnerabilities which can be combined for full remote device compromise:

- *Vulnerability #1*: Allows remote attackers to read arbitrary files via a /.. or // after "GET /uir" in an HTTP request. NOTE: this vulnerability exists because of an incorrect fix for CVE-2017-6190.

PoC:
```
+----
| $ curl http://{routerip}/uir//etc/passwd
+----
```

The vulnerability can be used retrieve administrative password using the other disclosed vulnerability - CVE-2018-10824. This vulnerability was reported previously by Patryk Bogdan in CVE-2017-6190 but he reported it as fixed in a specific release but unfortunately it is still present in newer releases. The vulnerability is also present in other D-Link routers and can be exploited not only (as the original author stated) by double dot but also absolutely using double slash.

- *Vulnerability #2*: Password stored in plaintext in several series of D-Link routers

NOTE: I have redacted the filename in the description to XXX because the vendor leaves some EOL routers unpatched and the attack is too simple. (In other words, this is SO AWFUL that even the researcher was unwilling to disclose it fully.)

The administrative password is stored in plaintext in the "/tmp/XXX/0" file. An attacker having a directory traversal can easily obtain full router access.

PoC using the directory traversal vulnerability disclosed at the same time - CVE-2018-10822

```
+----
¦ $ curl http://routerip/uir//tmp/XXX/0
+----
```

This command returns a binary config file which contains admin username and password as well as many other router configuration settings. By using the directory traversal vulnerability it is possible to read the file without authentication.

- *Vulnerability #3*: Shell command injection in httpd server of a several series of D-Link routers

An authenticated attacker may execute arbitrary code by injecting the shell command into the chkisg.htm page Sip parameter. This allows for full control over the device internals.

PoC:
1. Login to the router.
2. Request the following URL after login:
3. $ curl http://routerip/chkisg.htm%3FSip%3D1.1.1.1%20%7C%20cat%20%2Fetc%2Fpasswd
4. See the passwd file contents in the response.

Exploiting all three together:

Taking all the three together, it is easy to gain full router control including arbitrary code execution.

Description with video: http://sploit.tech/2018/10/12/D-Link.html

Disclosure Timeline:
- 09.05.2018 - vendor notified
- 06.06.2018 - asked vendor about the status because of long vendor response
- 22.06.2018 - received a reply that a patch will be released for DWR-116 and DWR-111, for the other devices which are EOL an announcement will be released
- 09.09.2018 - still no reply from vendor about the patches or announcement, I have warned the vendor that if I will not get a reply in a month I will publish the disclosure
- 12.10.2018 - disclosing the vulnerabilities

Latest firmware for DWR-111 is still v1.01 -- *still vulnerable*
Latest firmware for DWR-116 is still v1.06 -- *still vulnerable*

Time to update Drupal again! ... and DO NOT WAIT!
https://www.drupal.org/sa-core-2018-006

Drupal Core - Multiple Vulnerabilities - SA-CORE-2018-006

Last Wednesday the 17th, the Drupal Security Team released an advisory detailing two =critical= and three "moderately critical" vulnerabilities affecting all v7.x and v8.x Drupal installations.

US-CERT (the United States Computer Emergency Readiness Team) announcement said: "A remote attacker could exploit some of these vulnerabilities to take control of an affected system."

One of the two critical bugs is an injection vulnerability in the default Drupal mail backend, which uses PHP's mail function [DefaultMailSystem::mail()] in Drupal 7 and 8. When using this default mail system to send emails, some variables were not being sanitized for shell arguments. As is common, when untrusted input is not sanitized correctly remote code execution may result.

The second of the two remote code execution bugs exists in Drupal 8's Contextual Links module. In Drupal, these modules supply contextual links that allow privileged users to more easily perform tasks related to regions of the page without having to navigate to the Admin Dashboard.

However, the Contextual Links module doesn't sufficiently validate the requested contextual links. This allows an attacker to launch a remote code execution attack in these links.

In addition to these two baddies, the Drupal Security Team also acknowledged three other "moderately critical" bugs in its advisory.

The bottom line here, as always, is to immediately upgrade to the most recent version of Drupal 7 or 8 core.

- Users of 7.x, should move to at least Drupal 7.60.
- Users of 8.6.x, should move to at least Drupal 8.6.2.
- Users of 8.5.x or earlier should move to Drupal 8.5.8.

Drupal notes that: Minor versions of Drupal 8 prior to 8.5.x are not supported and do not receive security coverage, so sites running older versions should update to the above 8.5.x release immediately. 8.5.x will receive security coverage until May 2019.

Although Joomla and Drupal both lag far behind Wordpress' nearly 60% domination of the content management system (CMS) market -- having 6.6% and 4.6% usage respectively -- having even 4.6% of CMS-driven sites vulnerable to remote code execution attack is no laughing matter.

Live Networks "LIVE555" RTSP Streaming Media Server...
... contains a critical remote execution bug affecting versions prior to last Wednesday's release of v0.93.

(Version 0.93, released October 17 2018)

The "LIVE555 Media Server" is a complete RTSP server application. It can stream several kinds of media file (which must be stored in the current working directory - i.e., the directory from which you launch the application - or a subdirectory.):

A MPEG Transport Stream file (with file name suffix ".ts")
A Matroska or WebM file (with filename suffix ".mkv" or ".webm")
An Ogg file (with filename suffix ".ogg", "ogv", or ".opus")
A MPEG-1 or 2 Program Stream file (with file name suffix ".mpg")
A MPEG-4 Video Elementary Stream file (with file name suffix ".m4e")
A H.264 Video Elementary Stream file (with file name suffix ".264")
A H.265 Video Elementary Stream file (with file name suffix ".265")
A VOB video+audio file (with file name suffix ".vob")
A DV video file (with file name suffix ".dv")
A MPEG-1 or 2 (including layer III - i.e., 'MP3') audio file (with file name suffix ".mp3")
A WAV (PCM) audio file (with file name suffix ".wav")
An AMR audio file (with file name suffix ".amr")
An AC-3 audio file (with file name suffix ".ac3")
An AAC (ADTS format) audio file (with file name suffix ".aac")

If you know that you're using Live Networks LIVE555 as a publicly-exposed streaming service you'll want to update to last Wednesday's release.

Windows 10 October 2018 Update, the infamous Build 1809 has another problem...
As we know, when the content of a ZIP file extraction would cause the overwrite of an existing same-named file, the user should be prompted about the pending overwrite and asked whether to replace or skip the extraction for the colliding file.

But, it turns out that Build 1809 -- still working to bring us that new era of enhanced productivity -- is reportedly (and reproducibly) either overwriting existing ZIP file content without notification, or silently failing and doing nothing.

The good news is this problem, too, has been caught before the full formal re-release of Build 1809 into the general population. A recent tweet by an IT Staff Engineer at Microsoft on the Windows Insider Program Team indicated that this problem has been resolved back on October 6th with the Windows Insider Preview Build 18234 (19H1).

And just to kick this dead horse one last time, in some updated reporting, Microsoft's forensic analysis revealed that as many as 1500 of Build 1809's pre-release testers had their files deleted and complained... without Microsoft noticing.

Also... I updated the Win10 Home, which came pre-installed on the little Win10 box I use for this podcast, to Win10 Pro since I decided that I definitely wanted to begin hanging back from each month's security updates and also each bi-annual feature update. There's nothing I need that much each month that's worth being bit by.

An extremely popular jQuery File Upload Plugin has been vulnerable for 10 years
https://blogs.akamai.com/sitr/2018/10/having-the-security-rug-pulled-out-from-under-you.html

The Messaging Malware Mobile Anti-Abuse Working Group met in Brooklyn, New York two weeks ago, Monday through Thursday. Attending that meeting was Akamai's Larry Cashdollar. He failed to bring a raincoat, and it rained throughout that week. So Larry was hotel bound. Having, therefore, nothing else to do, Larry decided to poke around at the various add-on packages available for Node.js at https://www.npmjs.com/.

Skipping the details of his investigation, two days later Larry posted an entry on PacketStorm titled: "jQuery-File-Upload 9.22.0 Arbitrary File Upload" with the description: "jQuery-File-Upload versions 9.22.0 and below suffer from an unauthenticated arbitrary file upload vulnerability that allows for remote command execution."

First of all, accepting uploaded files to a server is EXTREMELY fraught with danger. This is not to say that it's not possible to do it safely. But few things should install more fear in the heart of the responsible web designer. How many times have we covered buffer overrun expoits in image handlers? So even carefully restricting uploads to non-executable images does not and cannot provide a guarantee of non-exploitability.

And what we have here is much much worse: Due to a presumably well-meaning change that the Apache group made back in 2010 -- at and since Apache version 2.3.9 -- the Apache maintainers deliberately disabled support for the .htaccess file as a performance improvement so that the server would not need to check for this file everytime it accesses a directory. But this change back then has subsequently left some developers and their projects open to attack, especially if they relied on .htaccess as a security function.

Starting with this version, the Apache HTTPD server offered an option that would allow server admins to ignore custom security settings made to individual folders via .htaccess files. This setting was made for security reasons, was enabled by default, and remained so for all subsequent Apache HTTPD server releases. But in the process, jQuery File Upload's assumption that it could protect its file uploads using a local .htaccess file was rendered invalid.

Title: jQuery-File-Upload <= v9.22.0 unauthenticated arbitrary file upload vulnerability
Author: Larry W. Cashdollar, @_larry0
Date: 2018-10-09
CVE-ID:[TBD]
Download Site: https://github.com/blueimp/jQuery-File-Upload/releases
Vendor: https://github.com/blueimp
Vendor Notified: 2018-10-09
Advisory: http://www.vapidlabs.com/advisory.php?v=204

Description: File Upload widget with multiple file selection, drag&drop support, progress bar, validation and preview images, audio and video for jQuery. Supports cross-domain, chunked and resumable file uploads. Works with any server-side platform (Google App Engine, PHP, Python, Ruby on Rails, Java, etc.) that supports standard HTML form file uploads.

Vulnerability: The code in
https://github.com/blueimp/jQuery-File-Upload/blob/master/server/php/UploadHandler.php
doesn't require any validation to upload files to the server. It also doesn't exclude file types. This allows for remote code execution.

As Larry wrote in his description of this discovery:
I started looking through the package's source and found myself peering at two PHP files under the directory server/php. The files are named upload.php and UploadHandler.php. The upload.php file calls the main file UploadHandler.php where all of the file upload code resides. I also saw that all files were uploaded to the files/ directory in the web server's root path. I wrote a quick command line test with curl and a simple PHP shell file confirmed that I could upload a web shell and run commands on the server.

$ curl -F "files=@shell.php"
http://example.com/jQuery-File-Upload-9.22.0/server/php/index.php

Where shell.php is:

<?php $cmd=$_GET['cmd']; system($cmd);?>

A browser connection to the test web server with cmd=id returned the user id of the web server's running process. I suspected this vulnerability hadn't gone unnoticed and a quick Google search confirmed that other projects that used this code or possibly code derived from it were vulnerable. There are a few Youtube videos demonstrating the attack for similar software packages.

This is of extra concern because this jQuery File Upload plug-in is not some obscure widget. It is an extremely capable and this extremely popular add-on, having been forked on Github 7,828 times to create descendant projects of that base package which are spread widely throughout the industry, deployed on websites far and wide.

Since discovering this critical vulnerability, Larry has examined 1000 out of the 7,828 forks of the plugin.. and every one of them was exploitable.

And... still worse, it turns out that at least some in the underground hacker community have been aware of this widespread backdoor for years...

As ZDNet explains in their coverage under the title: *"Zero-day in popular jQuery plugin actively exploited for at least three years" --> "A fix is out but the plugin is used in hundreds, if not thousands, of projects. Patching will take ages!"*

For at least three years, hackers have abused a zero-day in one of the most popular jQuery plugins to plant web shells and take over vulnerable web servers, ZDNet has learned.

The plugin is the second most starred jQuery project on GitHub, after the jQuery framework itself. It is immensely popular, has been forked over 7,800 times, and has been integrated into hundreds, if not thousands, of other projects, such as CMSs, CRMs, Intranet solutions, WordPress plugins, Drupal add-ons, Joomla components, and so on.

A vulnerability in this plugin would be devastating, as it could open gaping security holes in a lot of platforms installed in a lot of sensitive places.

This worse case scenario is exactly what happened. Earlier this year, Larry Cashdollar, a security researcher for Akamai's SIRT (Security Intelligence Response Team), has discovered a vulnerability in the plugin's source code that handles file uploads to PHP servers.

Cashdollar says that attackers can abuse this vulnerability to upload malicious files on servers, such as backdoors and web shells.

The Akamai researcher says the vulnerability has been exploited in the wild. "I've seen stuff as far back as 2016," the researcher told ZDNet in an interview.

The vulnerability was one of the worst kept secrets of the hacker scene and appears to have been actively exploited, even before 2016.

Cashdollar found several YouTube videos containing tutorials on how one could exploit the jQuery File Upload plugin vulnerability to take over servers. One of three YouTube videos Cashdollar shared with ZDNet is dated August 2015.

Note that it is for exactly this reason that GRC's forthcoming SQRL public web forums are running on their own physically separate and network-isolated machine. When a solution is assembled from hundreds of separate pieces, their interactions are too many to secure perfectly.

Abandoned Tweet Counter Hijacked With Malicious Script

We've talked before about the dangers of lapsed domains.

Since domain ownership is valuable, we have systems in place to rigorously protect that ownership. So over time, trust is created since domains are rarely successfully hijacked. But what about when a domain that's in use for some purpose is deliberately allowed to lapse? =That= is something we do see all the time, since the Internet is a constant churn with domains being abandoned. We sometimes find that a link we haven't visited in a long time now takes us to a weird search engine of marketing page. Advertisers long ago figured out that lapsed domains would see a non-zero level of traffic, so they began snatching up any that lapsed to camp their own nonsense there.

This happened to me with a domain I once had "grcmail.com" which I decided I would no longer need and allowed to go free. Now it hosts some marketing advertising crap. I even referred to it back in episode #44 of this podcast, explaining how I had grabbed it as an alternative domain since I wanted to keep any high-volume eMail away from GRC.

But what happens when a supplier of active content -- like embedded web page scripting -- decides to throw in the towel and to no longer host something they had been providing for years?

The Securi blog tells the story very nicely…

When Twitter announced their new design for “Tweet” and “Follow” buttons back in October 2015, marketers across the web developed a mild anxiety—the new design came with a decision to nuke their beloved Tweet count feature.

Social signals can be a huge credibility indicator for visitors and site content. Who doesn’t think there’s a psychological relationship between the number of social shares and the credibility of a content piece? It’s social validation, plain and simple.

Naturally, bloggers and website owners with an aversion to change started looking for alternative solutions that offered the same feature. Marketers breathed a sigh of relief when easy-to-use services started popping up to offer Twitter share counts, and "New Share Counts" quickly gained traction. It even integrated with other existing social share plugins like SumoMe, AddThis, and Shareaholic.

Setting up New Share Counts on a site was simple:

- Navigate to newsharecounts[.]com (this site no longer works)
- Link your Twitter account and website
- Add the following code snippet to the bottom of every page you wanted to track shares from:

```
<script type="text/javascript"
src="//newsharecounts.s3-us-west-2.amazonaws.com/nsc.js"></script>
<script type="text/javascript">window.newShareCountsAuto="smart";</script>
```

Then, this summer in July 2018, "NewShareCounts.com" was abandoned and its service discontinued. The service's provider was about as responsible as one might hope. He posted a notice on his site and referred visitors to "opensharecount.com" or "twitcount.com" (Sorry Leo.)

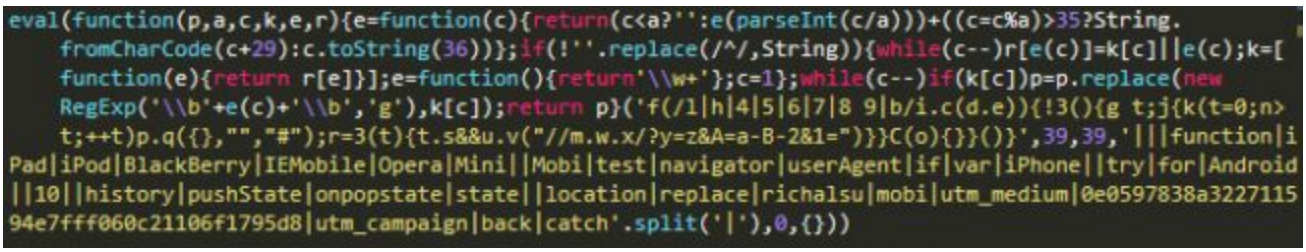
However, Securi found that more than 800 websites did not get the message. They continued to embed the discontinued script which, as we can see, pulls a script titled "nsc.js" from an AWS S3 bucket. That nsc.js script originally loaded another script from newsharecounts.com... which, after the domain was discontinued, stopped functioning.

On October 3rd of this month, three weeks ago tomorrow, the original AWS S3 bucket was cancelled.

Someone clever and nefarious had been waiting for just that to happen since the next day a very clever bad guy registered a new AWS S3 bucket under the same name and uploaded a malicious version of the nsc.js script.

Securi wrote...
During a recent remediation investigation, our Remediation Team Lead Ben Martin noticed malicious redirects being served by websites using the New Share Counts service.

The original Amazon AWS script (hxxp://newsharecounts.s3-us-west-2.amazonaws[.]com/nsc.js) still worked, but now served the following code:



Once decoded, this script contains absolutely no reference to Twitter or New Share Count. Instead, this snippet of code adds 10 fake browser history entries for the same page. An interesting feature of these history entries is that it prevents a user from choosing the previous page from the back button.

When the user loads the page, a malicious event handler is added. This handler waits until the user taps the “Back” button on their device or tries to navigate to a previous page. It then fires an event, which causes the browser to open to the following destination:

m.richalsu[.]mobi/?utm_medium=0e0597838a322711594e7fff060c21106f1795d8&utm_campaign=a-back-2&1= It then directs the user to a scam page, instead of returning to the previous page.

This behavior is only seen on Android, iOS, and mobile devices—and only under the condition that the user has tapped the “Back” button in their browser. Users on other devices won’t notice anything suspicious, except for maybe the lack of Twitter share counts that would exist if the original New Share Count script actually functioned.

At the time of writing, 800+ sites use this script. That’s over 800 sites (some of which are fairly popular) who simply wanted to show off their social signals, and are now instead maliciously redirecting their mobile traffic.

Loading third-party scripts and elements on your website always opens up the risk of unwanted content being served on your site without consent, especially when they come from an expired or unmaintained service.

SpinRite

Saturday, 8:03am
Stephan Budach @budachst

SpinRite giving a blast from the past

Hi Steve,

I have been a long-time listener to SecurityNow and each week, someone gets to tell his Spinrite success story. I have been owning a copy of Spinrite for years and occasionally tried it on hard drives, when one of my numerous RAID boxes would drop and SpinRite would perform its magic.

Two weeks ago on a Sunday, our Active Directory master server dropped of the network and, when I came in on Monday, I immediately checked it. It had crashed and would no longer boot. The AD server was using some older SAS (SCSI) drives, in RAID 0, on a controller without BIOS support, so I needed to add the old DOS ASPI8XX.SYS and SYMDISK.SYS drivers to SpinRite's boot config. That allowed SpinRite to see the drive. Running a Level-2 SpinRite scan on the drive found and repaired 2 errors on the drive: one which the drive fixed on its own and one where SpinRite's Dynastat recovery kicked in to repair that sector.

Since Dynastat took a while, I left it running over night and when I came back the next morning, Spinrite showed me a green screen, stating that it had successfully completed all it's tasks. Just to be double-sure, I re-ran the level-2 scan and Spinrite zipped through the drive in 90 minutes. I then re-installed the drive into its original server. The SAS controller recognized the drive and re-assembled the raid0 and virtual drive, Windows 2012 Server booted right up and worked again without issues.

Thanks for a terrific product and for letting me have my very own Spinrite-Story.

Kind regards,
Stephan Budach (Hamburg, Germany)

Libssh’s Big Whoopsie!

Surprising flaw found in the open source libssh SSH library

What is SSH? / Telnet -to-> SSH. (Telnet on FreeBSD Unix at GRC)

So what happened?
SSH has a "state machine" which determines what's going on and changes state based upon events which affect the SSH server.

Peter Winter-Smith of NCC Group discovered what can only be described as a deeply embarassing flaw in the libssh implementation of its state machine.

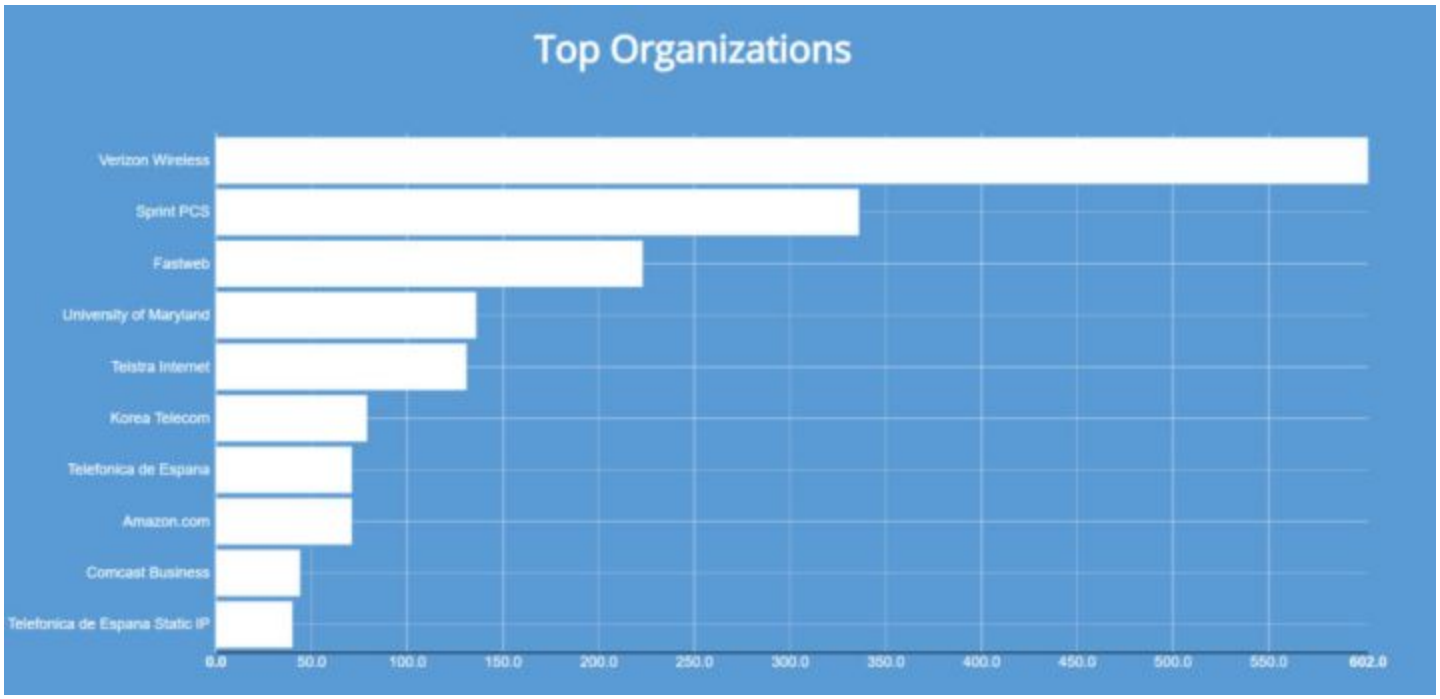
Introduced four years ago back in 2014, libssh versions 0.6 and above have an authentication bypass vulnerability in their SSH service. During normal SSH protocol, the client requests a user authentication challenge by sending the server an "SSH2_MSG_USERAUTH_REQUEST" message. If all goes well, the authenticating code will later emit an "SSH2_MSG_USERAUTH_SUCCESS" message to indicate that the authentication succeeded and the user is now authenticated.

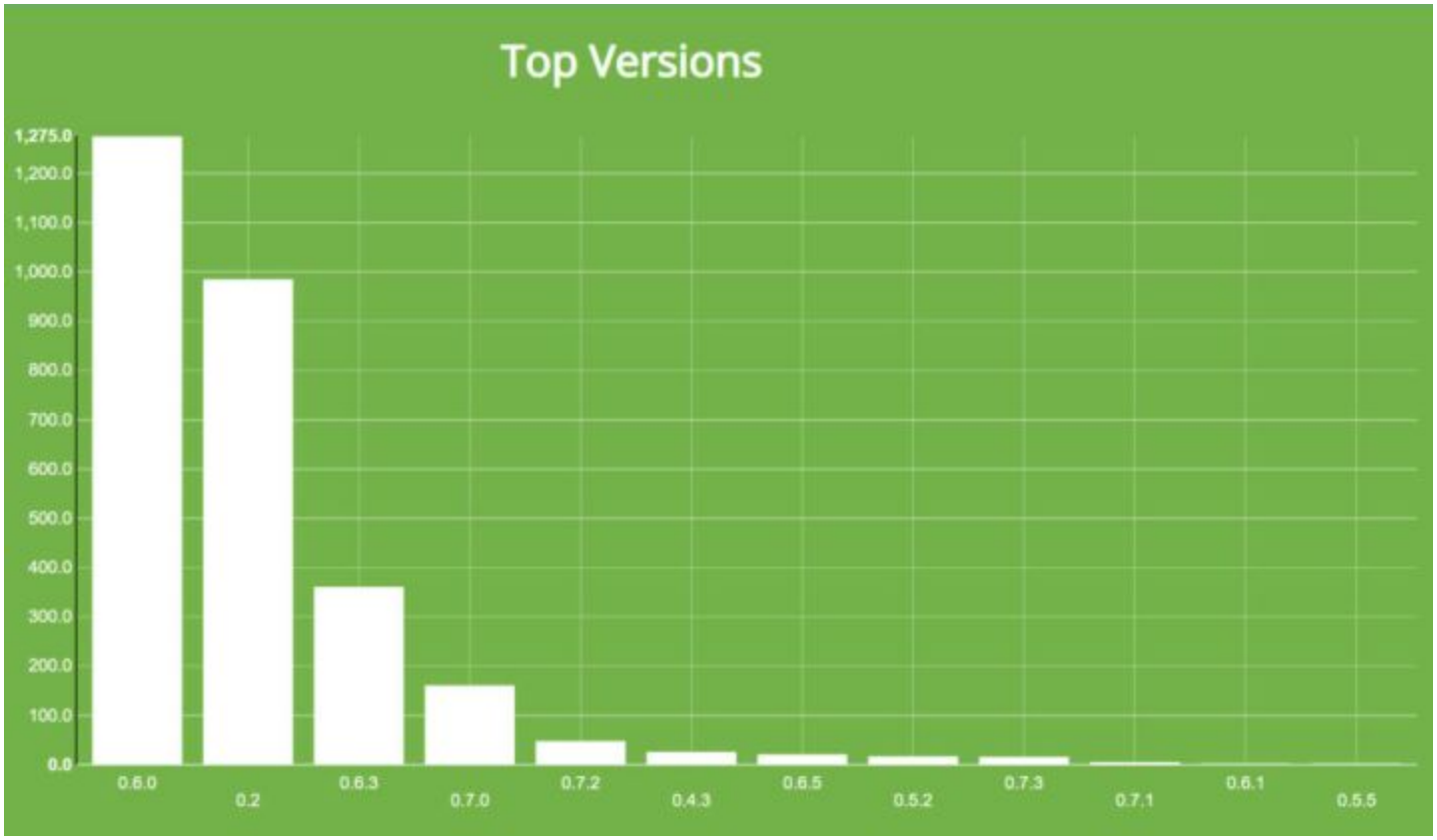
But, believe it or not, the buggy libssh state machine does not discriminate between the source of the message, and the NCC Group's Peter Winter-Smith discovered that a remote connecting client can skip all of that hassle and simply send the "SSH2_MSG_USERAUTH_SUCCESS" message itself... to be immediately authenticated to the server and given a full SSH remote shell on the target system. In other words, ANYONE may authenticate successfully without any credentials.

How widespread are publicly accessible libssh servers?
A simple SHODAN scan identifies 6,000 candidate libssh-based servers.
Closer probing [ahem] confirms that there are approximately 3,000 servers connected to the Internet that use the library, and roughly 1,800-1,900 of them are currently using a vulnerable version of the libssh library.

Who uses libssh?

- RedHat indicated that Red Hat Enterprise Linux 7 Extras.
- F5 Networks BIG-IP load balancers
- KDE uses libssh for the sftp file transfers
- Cisco is examining their many devices since they also chose libssh
- GitHub implemented their git ssh server with libssh,
- ... but the way Github use it protected them from this exploit.
- X2Go is a Remote Desktop solution for Linux
- csync a bidirectional file synchronizer
- Remmina the GTK+/Gnome Remote Desktop Client
- XMBC a media player and entertainment hub for digital media
- GNU Gatekeeper a full featured H.323 gatekeepe





Since the announcement at least four proof-of-concept (PoC) scripts have been uploaded to GitHub:
https://github.com/SoledaD208/CVE-2018-10933/blob/master/CVE-2018-10933.py
https://github.com/blacknbunny/libSSH-Authentication-Bypass/
https://github.com/hackerhouse-opensource/cve-2018-10933
https://github.com/vulhub/vulhub/tree/master/libssh/CVE-2018-10933

And Leap Security has posted a Python-based scanner to search for vulnerable versions of libssh:
https://www.leapsecurity.io/blog/cve-2018-10933-libssh-authentication-bypass-tool/

Leap's scanner has two modes: passive mode which determines if systems are vulnerable by banner grabbing, and active mode which attempts to actually leverage the vulnerability to bypass the server's authentication to confirm a system's vulnerability.

This week's Security Now! Puzzler

Vending Machine App Hacked for Unlimited Credit
Vending machines made by Argenta, a successful Italy-based purveyor of coffee services, are widely used to dispense all sorts of products from drinks to dry goods and cigarettes. Some of these machines were located on a University campus where they came to the attention of Matteo Pisani, an Italian hacker and CTO of Remoria VR.

Matteo was made curious by the fact that these high-end vending machines support both Bluetooth Low Energy (BLE) and Near Field Communications (NFC) technologies to link to a companion "Argenta Wallet" smartphone app. (You can see where this is headed, right?)

The app was poorly designed, easily reverse engineered and exploited. The "Wallet" maintained a balance which the vending machines naturally assumed was valid. Presumably, unlike last week's picture of the week with the coffee machine offline and updating itself, these high-tech vending machines lacked a connection to the Internet to enabled independent verification of the user's balance. This lack of verification allowed Matteo to easily manipulate the Wallet's database to give himself 1000 EU of credit... which the machines readily accepted. He did report and responsibly disclose this to the parent company who presumably worked out a solution.

But what solution?
We have often talked about the impossibility of encrypting a DVD which a consumer's DVD player must decrypt right there in the living room. The keys must be present in the accessible player for it to work.

But the security model here is subtly different, and using **ONLY** the concepts we have often discussed here on this podcast, there IS a wonderfully simple, robust and utterly uncrackable solution for the design of this system. So I want everyone to think about this for the next week.

Next week I'll describe how I would solve this problem and everyone can check to see whether they came up with the same simple, elegant, and uncrackable solution… or perhaps something even better?

~30~