# The best write-ups 2018 brought us

BY INTIGRITI · DECEMBER 30, 2018 · LAST UPDATED ON MARCH 6, 2025


With 2019 just a few hours away, it is time to look back and appreciate the good stuff last year brought us. So in case you're stuck on a boring New Year's reception: now is the time to sneak out and take a moment and revisit the top ten best write-ups of 2018.


Link: https://medium.com/@jonathanbouman/local-file-inclusion-at-ikea-com-e695ed64d82f

Author: https://twitter.com/jonathanbouman

This extensive article provides a step-by-step dissemination of how the author discovered a critical vulnerability in a PDF parser. What is so interesting about this article is that it also discusses the whole responsible disclosure process that followed after initial discovery.

Link: https://samcurry.net/reading-asp-secrets-for-17000/

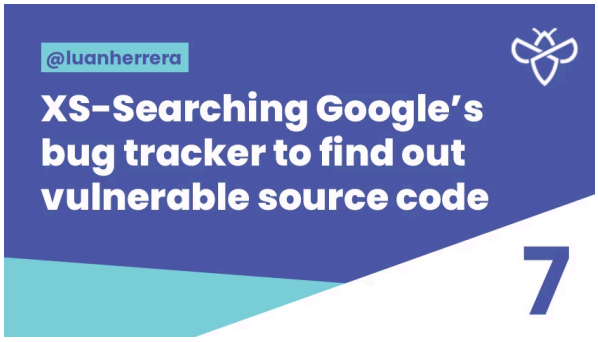
Author: https://twitter.com/samwcyo

Just in time for this article, Sam Curry (zlz) published a great story on how he got a $17.000 (!) dollar bonus for a path traversal vulnerability that allowed him to access ASP secrets. What is a recurring theme in this kind of write-ups is that the discovery of the vulnerability did not come without a struggle: while some researchers would stop when they noticed the traversal block, Sam did not give up and looked for possible bypasses. Persistence is the key to success.


Link: http://10degres.net/aws-takeover-ssrf-javascript/

Author: https://twitter.com/gwendallecoguic

The implementation of custom macro languages should always raise a red flag for bug bounty hunters, especially when the source code is up for grabs. The author of this blog article found himself in this situation, and 12 hours, 30 minutes and some StackOverflow questions later, he was able to collect  Amazon AWS credentials and a decent bounty. This write-up shows once again that official documentation often contains the keys to a bounty treasure chest.




Link: https://medium.com/@luanherrera/xs-searching-googles-bug-tracker-to-find-out-vulnerable-source-code-50d8135b7549

Author: https://medium.com/@luanherrera

Client-side timing attacks is a vulnerability type we don't see very often. Understanding them is not hard, but spotting them in realistic attack scenarios is a bigger deal. This is also a case where the programming and the hacking world collide: writing a proof-of-concept for these types of attacks is a challenging task and something the author of this blog article absolutely nailed.


Link: https://www.betterhacker.com/2018/12/rce-in-hubspot-with-el-injection-in-hubl.html

Author: https://twitter.com/fyoorer

Someone got an RCE in HubSpot's templating engine and this was the payload:

*{{'a'.getClass().forName('javax.script.ScriptEngineManager').newInstance().getEngineByName('JavaScript').eval(\"var x=new java.lang.ProcessBuilder; x.command(\\\"netstat\\\"); org.apache.commons.io.IOUtils.toString(x.start().getInputStream())\")}}*

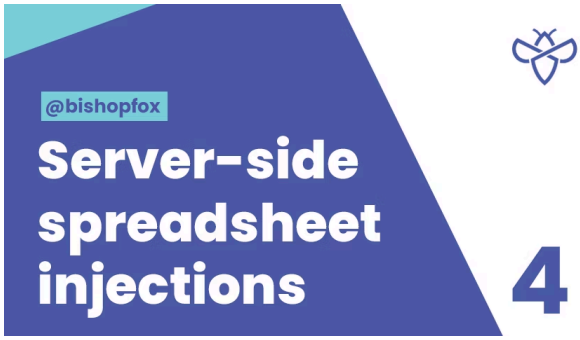
At first sight, this might look extremely complex and definitely not something the average bug bounty hunter could come up with. People tend to forget that payloads like these are constructed step-by-step with trial and error, and we love it how the author outlines every single step of his thought process, making it accessible and understandable for everyone.


Link: https://www.honoki.net/2018/12/from-blind-xxe-to-root-level-file-read-access/

Author: https://twitter.com/honoki

Some people stop when they discover a medium severity vulnerability.

And then you have people like @honoki, who try to leverage the severity as high as they can. We rarely see a blind XXE being escalated to root level file-read access, but @honoki tried to beat the odds and scored. Combining a blind XSS with a SSRF in an outdated Jira Instance to achieve root file-read access? Definitely one of our favorite bug chains this year.




Link: https://www.bishopfox.com/blog/2018/06/server-side-spreadsheet-injections/

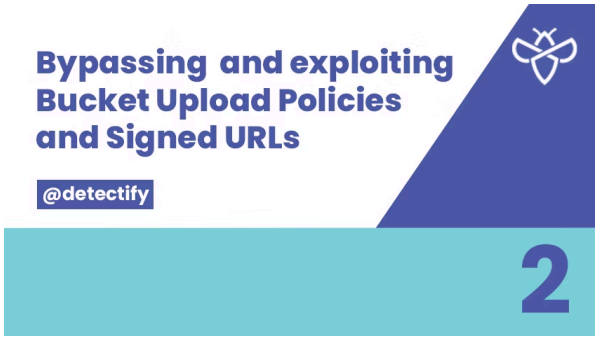
Author: https://twitter.com/bishopfox

Never judge a bug a bug by its vulnerability type. 2018 gave us an eavesdropping clickjacking, a content injection to RCE and in this case CSV injection to RCE. This great research by Bishopfox shows that we're definitely not prepared for client-side vulnerabilities that execute on a server.


Link: https://hackerone.com/reports/341876

Author: https://twitter.com/0xACB

André Baptista is someone who does not stop looking at a system until he achieves RCE and Shopify was no exception. This was hands-down one of the best publicly disclosed bugs on HackerOne in 2018, more than worth the $25.000 bounty!




Link: https://labs.detectify.com/2018/08/02/bypassing-exploiting-bucket-upload-policies-signed-urls/

Author: https://twitter.com/fransrosen

Frans Rosén is the living proof that Sweden is more than just IKEA. 'The swedish ninja' never fails to disappoint. Instead of hunting for known bugs and security misconfigurations, he and his Swedish companions do their own research and break the internet multiple times a year. This article on bucket upload policies is a must-read for every AWS user.  'Tack så mycket', mister Rosén!


Link: https://portswigger.net/blog/practical-web-cache-poisoning

Author: http://twitter.com/albinowax

Since *Titanic*, no ship was called *unsinkable*.

Since James Kettle's talk on Web Cache Poisoning, only few people dare to call a website *unhackable*. This vulnerability existed for years, and people knew about it, but did not want to think about it to figure out the possible consequences. Then James came along and nothing will ever be the same again. This was hands down one of best and most widespread configuration issues discovered in 2018. We look forward to what 2019 will bring!