How I earned $$$$ by finding confidential customer data including plain-text passwords!

Sushant Soni Follow
Oct 24 · 2 min read

How directory indexing and file path traversal led to confidential customer data in plain sight!

It was like any other Friday night while I was learning more about Web Application security, I remembered that I had forgotten to make arrangements for an upcoming family meeting. It required me to avail the services of a very popular Indian startup. And that's when it struck me "*why not spend some of my time to look for some security loopholes on the site which I use regularly?*"

. . .

1. **Finding Sub-Domains**

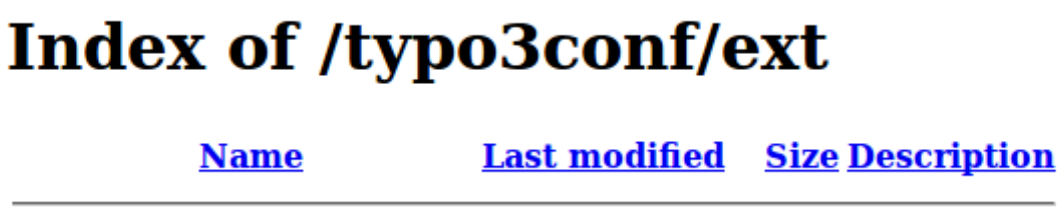
   I started my recon with enumerating the subdomains. Here I used @tomnomnom's **AssetFinder,** the output was then fed to an another great tool by @tomnomnom **httprobe**.

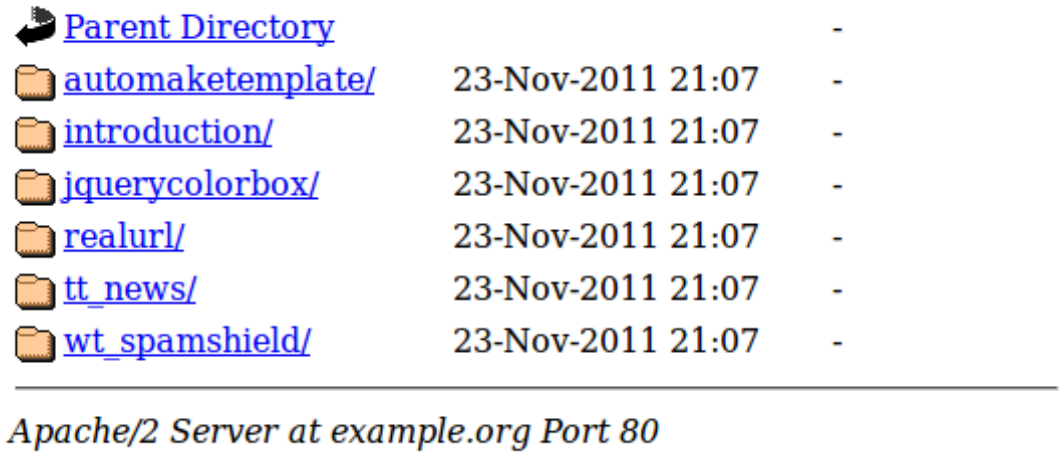
   One domain in particular looked important to me, it is something like "https://api.xxxx.com"

2. **Directory Searching**

   Second step I usually do is searching for directories/files. Here, I used **Dirsearch** with a custom wordlist from SecLists to discover content.

   While traversing through all the results, I browsed "https://api.xxxx.com/application/logs", to my surprise, the **directory was accessible and indexing was enabled**.





Source: https://docs.typo3.org/m/typo3/guide-security/8.7/en-us/GuidelinesAdministrators/DirectoryIndexing/Index.html

It was a log directory, some of the logs were old, dating back to 2018, so I tried to access the most recent log files, it was a php file "*log-09–09–2019.php*" and got an error "**No direct script access allowed**". Moving on I noticed there was a gunzipped/compressed version of the same log file "*log-09–09–2019.php.gz*".

The gunzip file was getting downloaded, I uncompressed it and opened the file in vim and "**VOILA!!**". It opened me to a completely different world, it was a stash of gold/customer data for any hacker out there.
In the file I found **customer's email address, phone no., credit card**

numbers (some digits masked), PLAIN-TEXT passwords (no way). There were **FB OAuth tokens**, basically all of the data which can lead to a data breach.

The issue was reported and I received a 4 digit bounty in $

)

Security Infosec Bug Bounty Hacking Passwords

312 claps




WRITTEN BY

Sushant Soni

QA | Security

Follow

See responses (2)

More From Medium

Related reads

The Client-Side Battle Against JavaScript Attacks Is Already Here

Ben Diamant in The Startup
Jun 12 · 12 min read ★

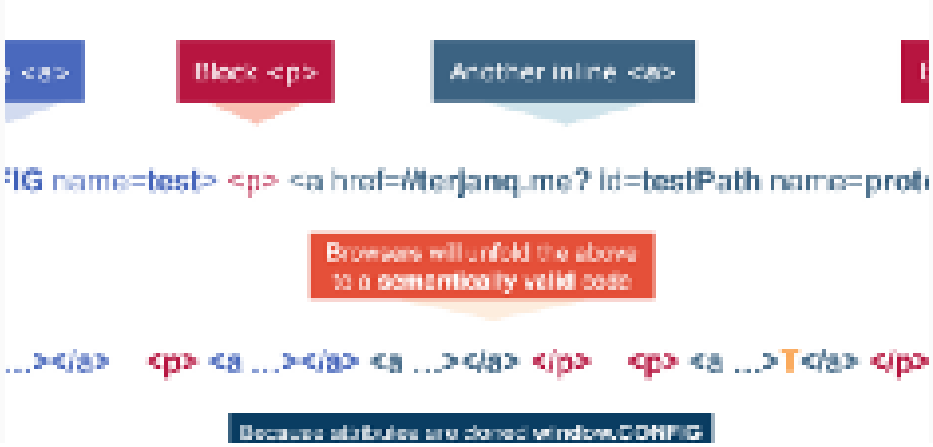
75

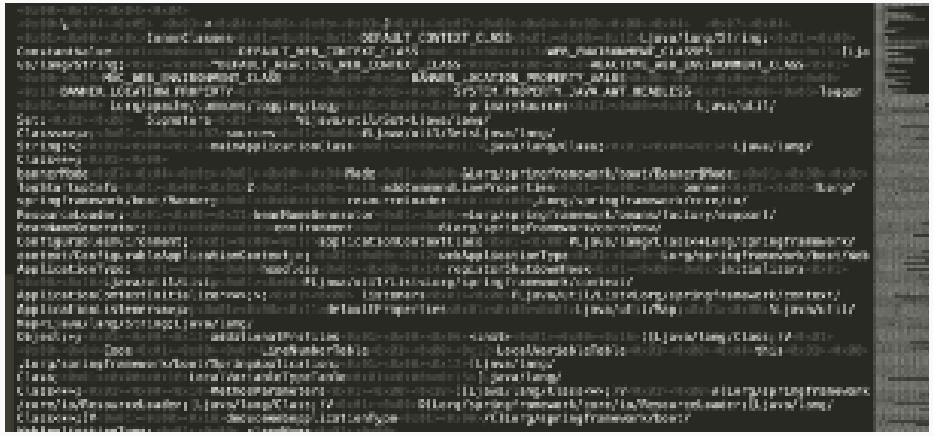
Related reads

Clobbering the clobbered — Advanced DOM Clobbering

terjanq
Sep 26 · 9 min read ★

116





Related reads

Demystifying Java JNDI attacks

Akshay 'Ax' Sharma
Mar 27 · 3 min read ★

21