This Simple String Blocks Your Web Scrapers

https://www.youtube.com/watch?v=L0gxQsCJ1hY

regardless of what cookies you're using what user agent you put in or what uh headers you use there is a method that antibot websites can figure out who you are and what your intentions probably are based on just a simple hash string that hash string is called the ja3 hash and it's made up of the TLs information that happens when you make the initial request so when we make an initial request to a server there's a TLS handshake a client hello a server hello and there's some information is shared what I've got on the left hand side of my screen here is the TLs report from my Chrome browser and on the right is from httpx within python now it's important to note that I use httpx here because it's http2 enabled and I'm going to talk about that in a little bit and why that's important too when we start to use techniques like TLS spoofing like we are here we want to make sure that our IP meets the quality standard necessary as this is another way for sites to decide to block our request that's where I use the sponsor of today's Video Proxy scrape proxy scrape gives us access to high quality secure fast and ethically sourced proxies that cover residential Data Center and mobile with rotating and sticky session options there's million plus proxies in the pool all to use all with unlimited concurrent sessions from countries all over the globe enabling us to scrape quickly and efficiently my go-to is either geot targeted residential proxies based on the location of the website or the mobile proxies as these are the best options for passing antibot proe on sites and with auto rotation or sticky sessions along with one of the Python packages I'm going to cover later in this video it's a great first step to avoid being blocked it's still only one line of code to add to your project and then we can let proxy scrape handle the rest and also any traffic you purchase is yours to use whenever you need as it doesn't ever expire so if this all sounds good to you go ahead and check out proxy scrape at the link in the description below okay let's check out the hash string and see what it means for our scrapers so we can clearly see that these are are different but let's Che let's take a look at the ja3 hash first for each one and they're obviously going to be different here now this little string is a hash representation of the full string report here and this is made up of various different bits of information for example is the TLs version used and then these represent the ciphers and then there's some other information tagged on the end what's important to know about this is that this information can be used to work out you know what sort of request you've made um if it's come from a browser if it's come from you know httpx in this case and it can be used to identify you and you can be blocked regardless of any other information so let's go back up to the top and let's check out the ciphers because this is the biggest difference really so straight away we can see the Chrome browser has much less ciphers enabled these are the ciphers the encryption that's decided upon and used when the TLs handshake takes place on the right with httpx there's lots more now clearly with this many more we're going to have much more information in that string which is going to lead to a very very unique or uh more identifiable hash of that string which we can then you will be used to block your request so this one has this grease here so let's have a look and see what that is so Greece is Google's way of uh was PA Google's way of trying to fix some incompatibility issues with a TLS uh within their

browser so this Cipher here is only ever found within Chrome browsers or Chrome based browsers so if you send a request that doesn't have this you're immediately going to stick out and you're going to get your request blocked for websites that have this enabled another thing that's worth looking at here is the actual way that the ja3 is implemented and this is the repo um from Salesforce we could look through all of this uh it tells you here's some like you know fingerprints for standard clients for malware etc etc uh and it can be used that way here's how it's constructed the SSL version the ciphers available and also all this information here and it leads you to this okay so then this is then turned into that md5 hash which can be then compared uh the j3s S is the same thing but on the server side um we're not really going to look at that at the moment so all this information is used to fingerprint you so if we go over to uh this one here we can see there's on the fingerprint.

comom website it talks a little bit about uh how it works how it came about uh what we just looked for again and the md5 talks about the benefits but it also also talks about the limitations and this is what we're going to focus on but in just a minute because I want to show you something else first and that's http2 fingerprinting now with http2 and I found this this report here from akami which is very interesting read um it's the new protocol so all your browsers are going to use this and again there's more information available that can be used now what uh this company has done is that they use the http2 fingerprint to go ahead and pull extra information this information here settings Frame Window update frame and the priority frame this is then turned into their own hash so websites that use this absolutely have to use HTTP because your http2 HTTP Point whatever version is 1.

will just straight up fail this test every time so this is worth me you know worth keeping an eye on and worth understanding and I think if I go back to my hashes here and scroll down I don't know if it had the AC my hash it does so you can see this one does have it and again this one has it too so this was worked out from the settings the uh this here's the frame um frame settings that we looked at frame Type window update and etc etc again this information is different on my HTTP X request and this will then be transferred into their own sort of their own way of checking their own database and you'll probably get stopped here too so how do we get round it well on this article again it talks about the limitations of ja3 which is obviously very interesting read so so when we're trying to get around this we need to understand not only how it works which is what we just looked at but also how that hash is then used to check things now obviously the more information that is given inside that hash inside that ja3 string that's taken from all those things that we looked at gives more away to what your uh what your request type is and what it's come from I python or go or whatever so the idea being is that we want to mimic the most um uh the most common browsers available so and give them as little information or just enough information so it looks like we're coming from this browser so it gives you a pass rather than you know blocking you straight away so this is worth the read uh if you want to have a look through and understand a little bit more uh it talks more about how it works here now it does mention somewhere down here curl impersonate so I've got the GitHub repo up for that here and it's quite popular repo it's a you know updated version or a patched version of curl for making Network requests and you can actually you know send the uh the the it will cover up the ciphers and etc etc and it will mic a proper request which gives you that kind of like extra chance of you beating that request and getting into the to the server and whatever you're trying to do from that uh and it's worth giving this a go and having a look and understanding how it all works talks about what versions it can mimic again you know

they're not all of them so we're like we're way up in like Point I think for Chrome now but still you know these are popular versions of these browsers they're not going to you know you don't you can't just block everybody that uses Chrome so it's not that big a deal or it hasn't been in my in my opinion in my use case but obviously we can't use this within python um what I want to show you is the go TLS client um this is kind of what a lot of the um a lot of the Python libraries are built on because this guy provides uh python bindings I believe um and it talks again about how it works and how to use it within go and you know but we're really interested in the python versions so the one that I've been using the most is called KL cffi and if you've been in my Discord you've probably seen me talking about it or mentioning it but of course there are other ones too um this is the one that I'm going to show you working now now this and other other like some of the more modern um python scraping libraries um like H requests also build upon this one from uh the bogin TLs client and if I come to the detailed documentation here and go down to Standalone or is it uh shared Library here we are so if we go to Python and we have examples you'll see um we can go to this part of the library uh where is it I want just there's one other bit that I wanted to see Community projects so you can see it talks so there's a few um Community projects for Python and these are worth checking out um I've used a couple of these to to good success but again the one that I use is K cffi so what I'm going to do is I'm going to come back here and I'm going to open up instead of uh my httpx one we're going to look at the C impersonate one and we're going to come down and we'll compare again so with this has come from the one on the right has come from curl impersonate or C cffi rather and we can see right away that we we were impersonating crime we have that grease TLS Cipher which is going to be important we have the good user agent http2 of course and you can see this matches much much closer and we could even scroll down and a lot of this is very very similar so what we're doing is we're basically just impersonating as much of the information on the left from the real Chrome browser as possible on the right hand side and if we come to we have like um let's let's have a look at the um yeah so we can see that we've even got mostly matching here settings this is for the akami http2 um fingerprinting these are all very very similar here look you can see and so this is a very very good way of doing it now now generally I don't don't use this until I need it but it's worth knowing about I always say web scraping is knowing how to do lots of different things and understanding where they're going to fit in and where they can help you and if you want to watch me actually using it to success and scraping sites with it you want to watch the last video I made which I'm going to link here which shows how I needed to use Curl cffi to um spoof my J my TLS fingerprint to get a good ja3 has to scrape our website