Don't break the door, the key is under the doormat

Gerard Fuguet (gerard@fuguet.cat)

Abstract

The multimedia content has an exponential increase. The final user feels the need to get the media content each time faster. One of the easiest way to get this content, is centralizing it in one place, and in most cases, making it available to the public, to the internet (almost all the things are connected to the network). This type of architecture is called “Media Server”, who is able to serve this type of content to many devices (Smartphones, computers, TV…). The processes that we focus on this white paper has relation with a software integration in a Media Server in order to get access through this intermediate element.

We will demonstrate how easy is get all content of a Media Server, in particular, a *Plex*, through a third party application without protection. This situation motivated me to write it.

The intention is take consciousness of these situations and let any user to know how easy get is any type of content of anyone if this is not well protected (We do not distortion with deep technical terms).

Table of Contents

1. Motivation

A provider invited me to a Cybersecurity event, this had the participation of a known Spain hacker called Amador Aparicio [1]. He showed two presentation about "*hacking browsers*" (*Hacking Web Technologies Old School* and *From hacking IT to Hacking OT: Jungla 4.0*) [2] using S*hodan* [3] as scanner-searcher looking for connected devices in the Internet.

Few days after, I decided "*take a walk*" into this trying to understand, as first glance, how to filter properly into S*hodan.io* for something interesting… A flash came to my head about the vulnerability I discovered on *360Fly 4k* [4] when it access to the webserver of the cam. In the URL can be seen: */resource/100_3FLY/* as content and is displayed not only on the search bar, also in the body.

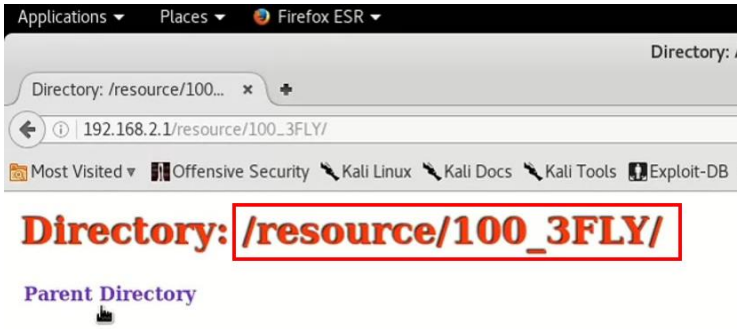

Figure 1: Part of URL to looking for on *Shodan*

I played a little with this string, put with space, only one word… and between the results, I clicked on this (acceded by port *8181*).

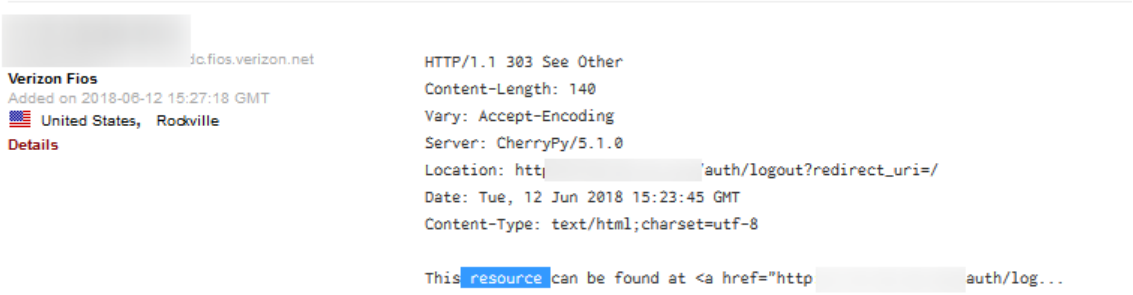

Figure 2: Results in *Shodan* matching the word "*resource*"

Is a HTTP Web based resource, the server type and version is *CherryPy/5.1.0* as figure 2 shows. We arrive well but the Login/Password screen stop us.

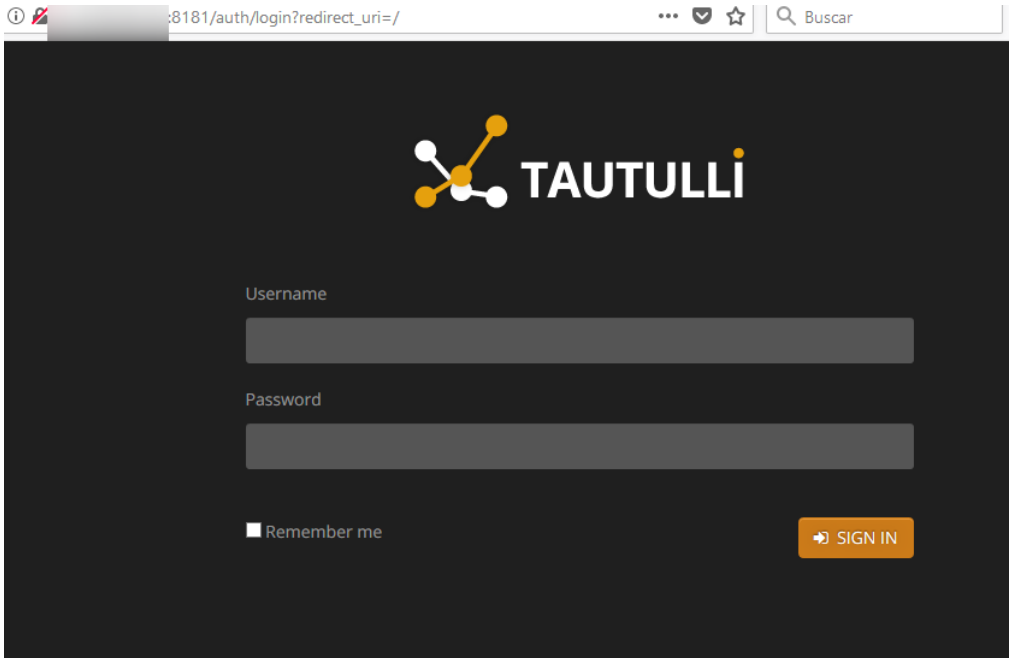

Figure 3: The service running in *CherryPy/5.1.0* Web server

We can read *Tautulli* at the top of the credential form (also, it can be observed that the URL is under unsecure HTTP, this is "-1" as not a best practice…). I had no idea about what is it. Is this service has a default user/password? The fast research in google concludes that no [5]. We found a FAQ about what to do if you lose your username and password, but of course, this implies to have full access to the server, so seems the story concluded here.

2. /taʊ'-tu'-liː/

Before ending this, I need knowing more about what *Tautulli* is. It has a Web Page [6] and the name is uncommon (or at least for me!). In page says that means "To watch or monitor", but my next question was, from what language it comes? I tried my luck playing with google translate but I can't see something similar, so "*surfing*" the network I discovered a brainstorming for choosing one new name (new name because the old was *PlexPy*) [7] And according to the reddit site, it comes from Inuktitut language [8].

2.1. Tautulli (for Plex)

Tautulli can't work alone, his dependency is *Plex* [9].

Tautulli borns to be an extension of *Plex* in order to track all actions of *Plex Media Server* (PMS) (nº of plays, time played for each user, user with more plays, the client platform most used…).

2.1.1. Get a "Real Demo"!

Exist the possibility (by default?) to setup the web environment without login needs as seen above in chapter 1, so let's see if S*hodan* can do our "free demo".

How can we refine the search? We can observe on figure 2 that the server header is *CherryPy/5.1.0* sure we will have more luck if we try searching this exclusive server and version. And yes, we have it.

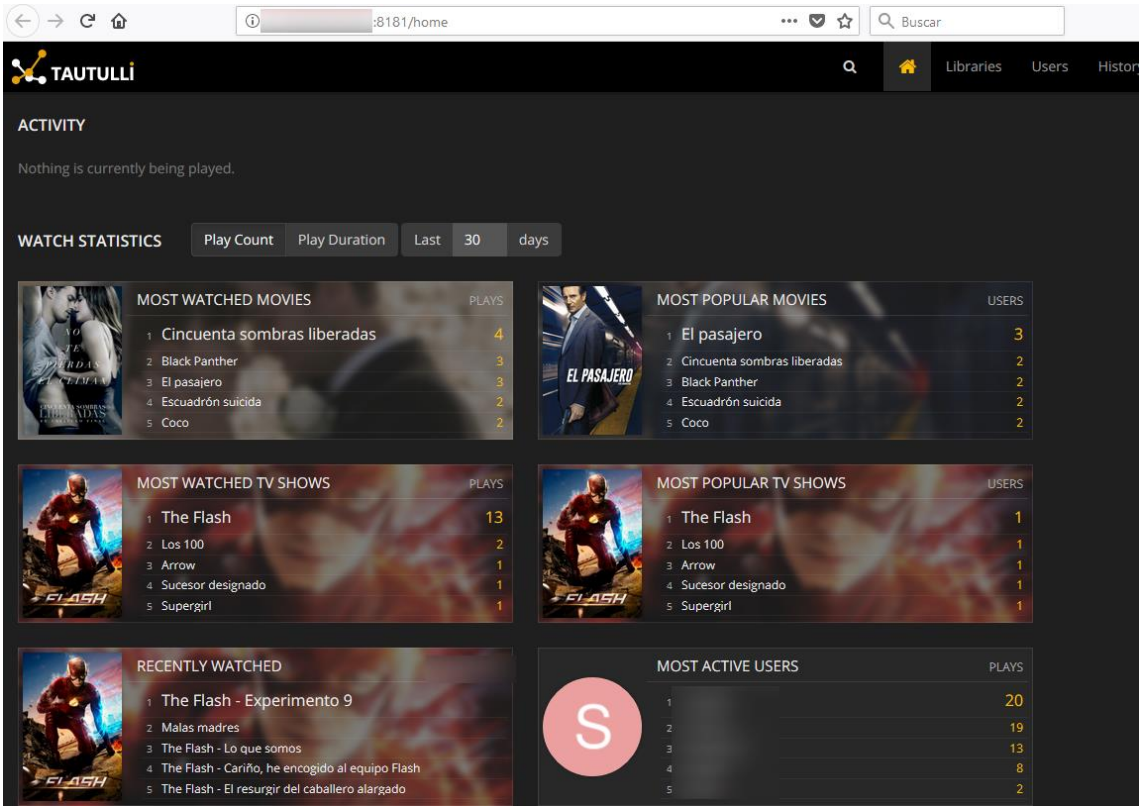

Figure 3: The *Tautulli's* splash screen with no login

Found it! We successfully enter inside *Tautulli's* web server of "somebody" easily. So seems it likes using *CherryPy* for the web frontend part.

This screen reveals beautiful information view:

1. Most watched Movies.
2. Most Popular Movies.
3. Most Watched TV Shows.
4. Most Popular TV Shows.
5. Recently Watched.
6. Most Active Users.
7. The Watch Statistics bar…
8. And… what is playing NOW!

Do you want more BigData my “Big Brother” ☺ ? Take a look onto this tabs! (Graphs, History…).

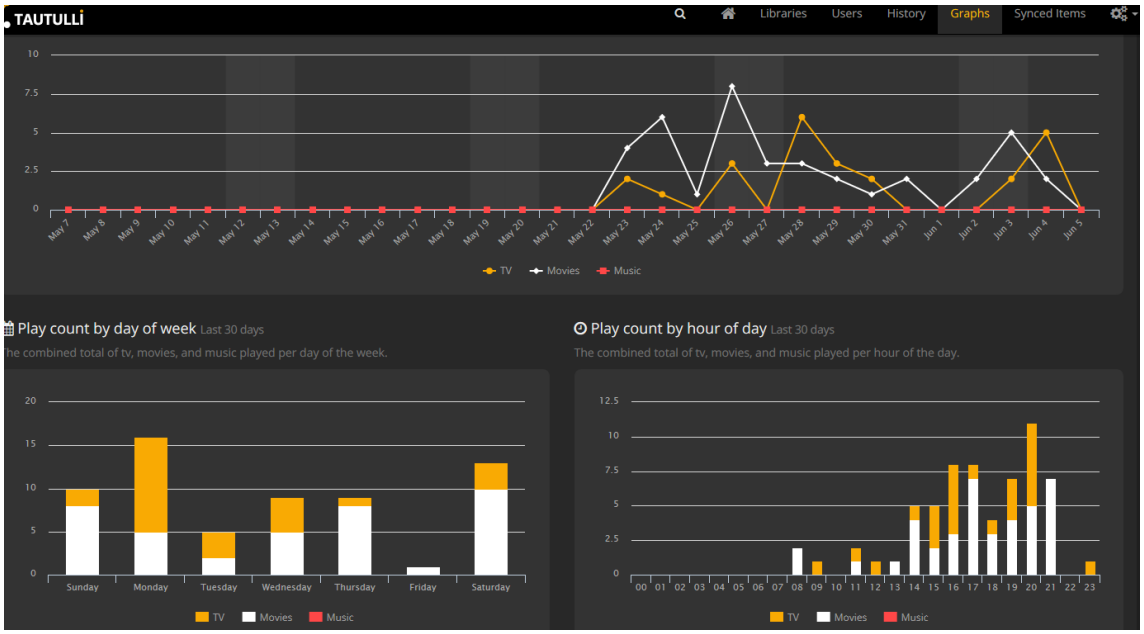

Figure 4: Graphics tab showing graphical statistics

OK, the data is displaying in a very clear mode, so “it’s fantastic”… Who had this brilliant idea?

2.2. Fingerprinting

Now is time to know the origin of all of this. Who is the creator? Is free, Open Source?

The software is on a *GitHub* repository [10] (By the way… Did you know that was acquired by Microsoft?) [11].

How old is this? Reviewing the Changelog [12] to determinate:

The software saw the light on 2015-08-11 under the name of *PlexPy* in his first release (*v1.0*) after some other releases, the name changed to *Tautulli* as described in chapter 2 under the version 2.0.0-beta (2017-12-18). Is well concluded in the releases section of his *GitHub* [13] or in r/PleX reddit [14].

2.2.1. The Owner

This software project involves two main contributors. The “father” was *drzoidberg33* [15] then appears *JonnyWong16* [16] as very active candidate at the beginning and finally seems is the owner of the *Tautulli’s* platform.

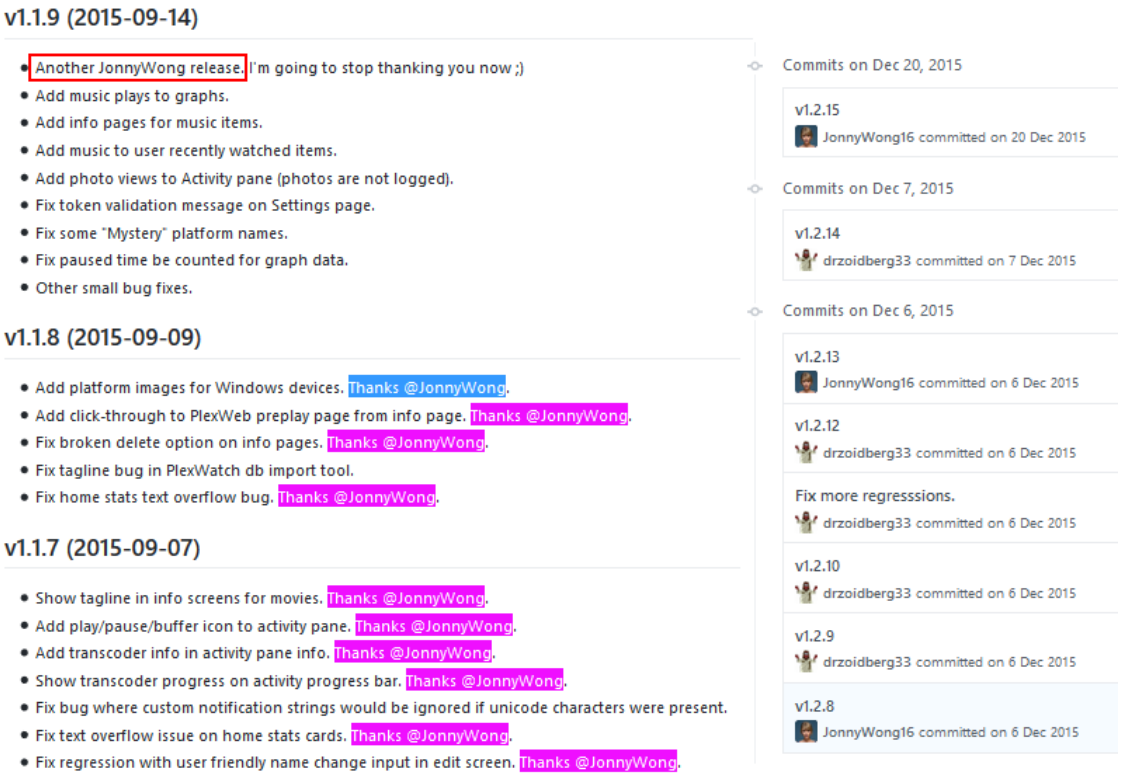

Figure 5: Changelog and history [12] [17] respectively

In figure 5, at left, almost in each line of the fixes there are many thanks to *JonnyWong16* from *drzoidberg33*.

At right, in part of the history of the changelog file, can be observe that some releases were delivered by *JonnyWong16* and since version *v1.2.15* all post releases were delivered by him until today.

2.2.2. Outside Face

"How can we identify you?" Well, let's put an example first. Imagine that in some long distance, there is someone but is far and you don't sure if is male or female but you know is a human, a person, because the shape and other characteristics, associates it with a person. When you are nearest in front on this person, is easy to know if male or female is, but is a person without any doubt, this was the first thing your eyes saw. So, what is the thing that can be seen at the beginning in case of *Tautulli*?

This is a web service, the first thing we see is the type of web server used, and it reveals on headers as described in figure 2 of chapter 1.

The Web server is *CherryPy* [18] a minimalist python web framework, seems is "easy going" with things developed under python programming language. The development maintains active, last version is *15.0.0* released in May 11 of this year (date according in the moment this paper was made) [19]. The next questions are, what is the actual version in *Tautulli*? And which is the most used? Through the version of web server (*CherryPy*) we will know possible range versions of the *Tautulli's* (or *PlexPy* if is too old) platform.

Navigating in the *GitHub's* project, we found the *CherryPy* component folder. The file that shows the current version is *wsgiserver2.py* and *wsgiserver3.py*, two python files, that we will use the "2" one because is used in all the history.

CherryPy Version	PlexPy/Tautulli Version	Range
5.1.0	v1.3.16 (2016-05-01)	v1.3.16 to v2.1.12
3.8.0	v1.0 (2015-08-11)	v1.0 to v1.3.15
3.6.0	-	-

Figure 6: History of changes of *wsgiserver2.py* file and comparison table

In the figure 6, exist three changes in the history in total until today. The first commit available was with the version *3.8.0* of *CherryPy*, the first commit in file was before the first release of the platform (*v1.0*) it means no release available of *PlexPy* (see chapter 2.2) with *CherryPy/3.6.0*.

PlexPy could "enjoy" of the *CherryPy/5.1.0* (or viceversa!) and this *CherryPy* is the winner version, which is still here and from long time ago.

3. Relationship: Tautulli & Plex

Plex can do "things" alone, but *Tautulli* needs the *Plex* engine to work. In some moment *Tautulli* communicates with *Plex*, let's see when.

In installation process, seems is mandatory authenticate against *Plex* according the *YouTube* video [20] (reached officially through the *GitHub* wiki).

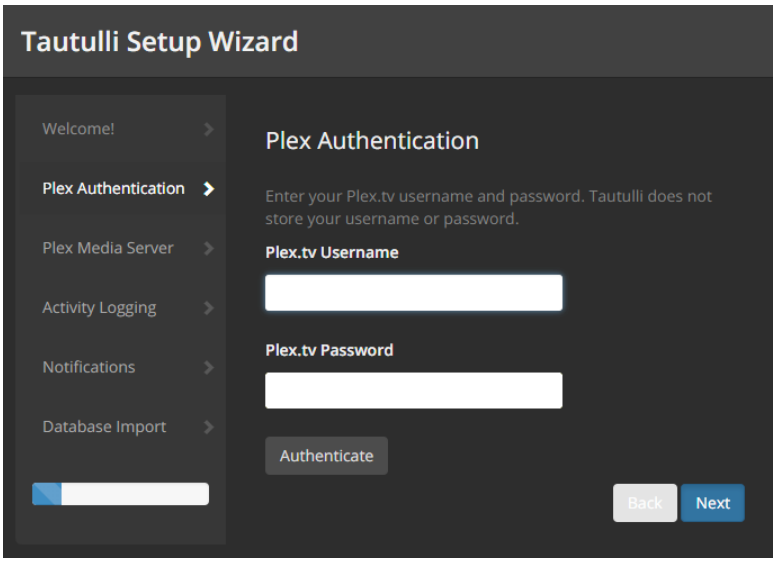

Figure 7: Authentication required for Setup

But… is really mandatory? Short answer is: "No" [21] But having a *Plex* account has more "benefits" and seems necessary for third parties apps to get rich of some interconnected functions. After authenticate with *Plex*, sure the relationship now is closer.

3.1. The Token: Get your Pass

How many times you get a pass or access that let you access to a place and you weren't buy it? Someone or a family member, access to a service with credentials and get the passes, you don't have any idea but what credentials has, but you get the pass, and with a pass you can enter to a place where only people can access with this "piece".

In *Tautulli*, under settings section there is an option: *Plex Media Server*.

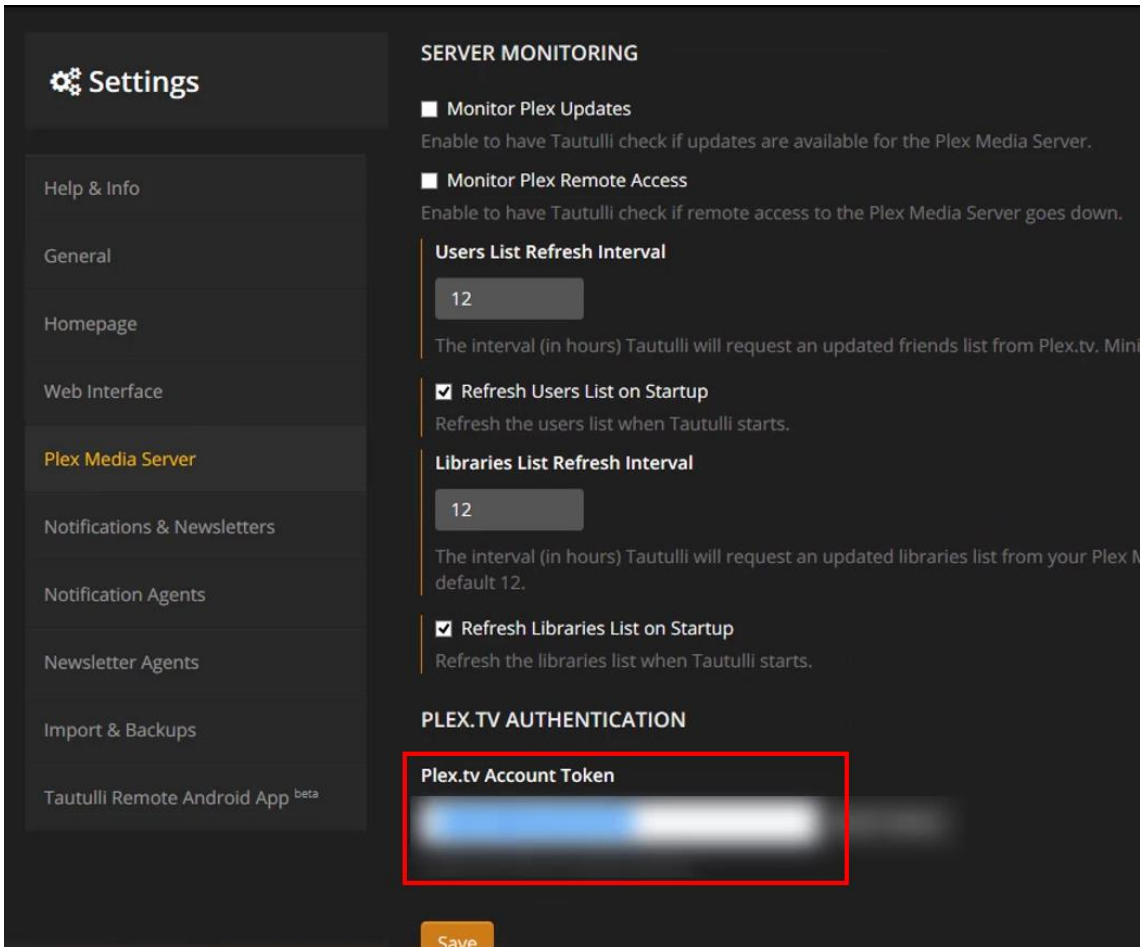

Figure 8: The *Plex* Account Token acquired in Settings

And the token is here, interesting… another question is: What privileges does it give me?

Well, lot of answers are in internet, google tells you who is "the correct" person for ask, and this person (*Plex* site, in this case) answer to you.

The token is acquired in three steps [22], you can "play" with it with the value *X-Plex-Token* if you pass it as a URL parameter to get requests under XML language.

4. Time to Hack

If you start reading this paper/document from here without reading the before pages, the perception will be like "read instructions before use". It's depends you are looking for but, please… take the following as a **disclaimer:**

A hacker is always a good person, who feel passion for computer science world and wants it to be as perfect as possible. A way to make it, is sharing this information to the public. And people who use these techniques to hurt, with bad manners, is not a hacker, this word is called cybercriminal (a bad person).

4.1. Panoramic View

Along the study, we have sufficient information for make a visual diagram helping with the hacking steps.

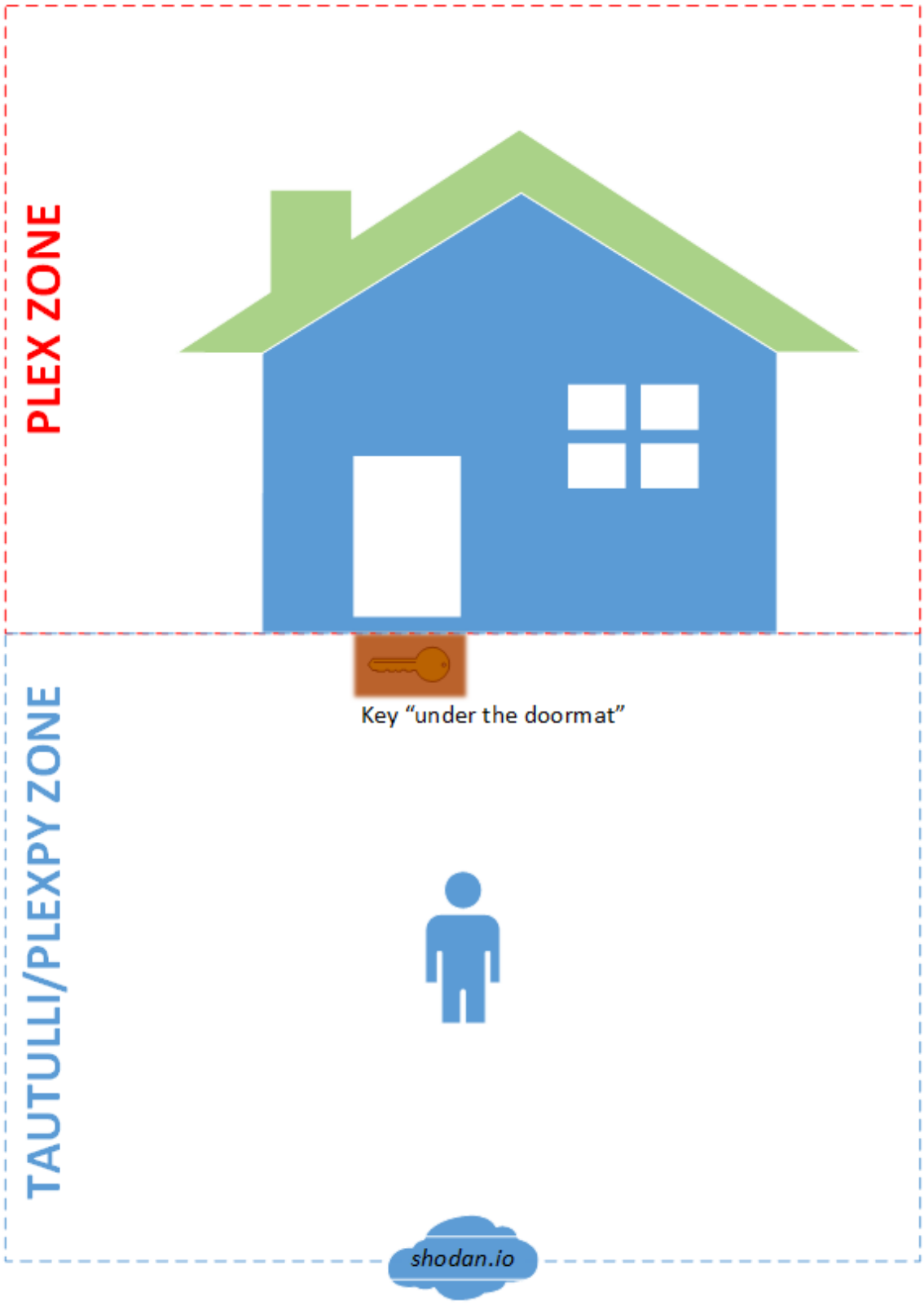

Figure 9: Visual Diagram to understand the situation better

The map of the figure 9, shows two clearly defined zones.

Blue zone is around us, our guide here is *Shodan* (shodan.io) an important piece of this puzzle. It can identify/filter "what houses are which has the key under the doormat". So *Shodan* can show us the unprotected (no user & password needed) *Tautulli/PlexPY* that are in the world.

Red zone is the perimeter that we can see once we get access through the key "under the doormat". We can't be here without first having been in the blue zone. And we arrived to the blue zone with the help of *Shodan*.

We can't jump directly to the red zone (unless there is another way!).

4.2. Partial or Normal View

No more theory, the practice demonstration is here so let's go!

Blue Zone:

Step 1: Tell to *Shodan* "the magic words".

1. Go to Shodan.io.

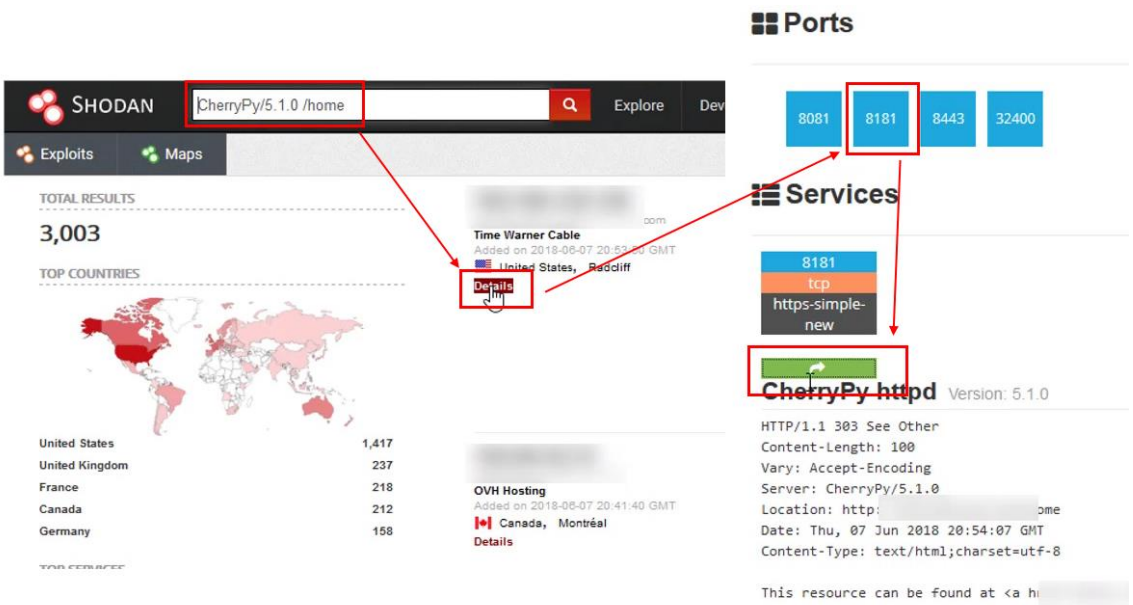

Figure 10: Searching & filtering it on Shodan

2. Put the two strings: "*CherryPy/5.1.0*" and "*/home*". *Enter*.

3. If you click in *Details* it will show info about the open ports.

4. Default *Tautulli's* port is *8181*, on click it goes to the HTTP headers.

5. Click on green button (is the same that typing *http://IP:8181*).

Step 2: “Get the key under the doormat”.

Here we will see how to play with the token.

1. You can find in… *settings – Plex Media Server – Plex.tv Account Token* as shown in Figure 8 of the chapter 3.1.

2. Use the direction (IP/Hostname) and the correct port (default for *Plex* is *32400* but remember this can be altered by end user). And insert the token at the end of the URL as a parameter: *http/s://IP:32400/?X-Plex-Token=YOURTOKENHERE*

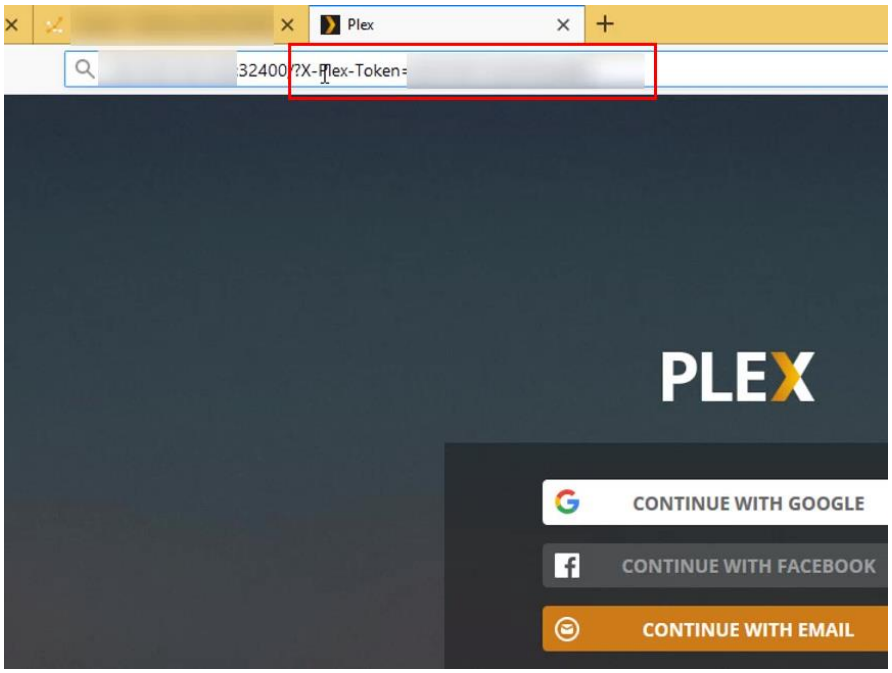

Figure 11: Putting the token; acquired in *Tautulli*

Red Zone:

Step 3: “Auto-Invitation”.

In this step, we shows how to get a media file as example.

1. After hitting *Enter* you are in, under XML language.

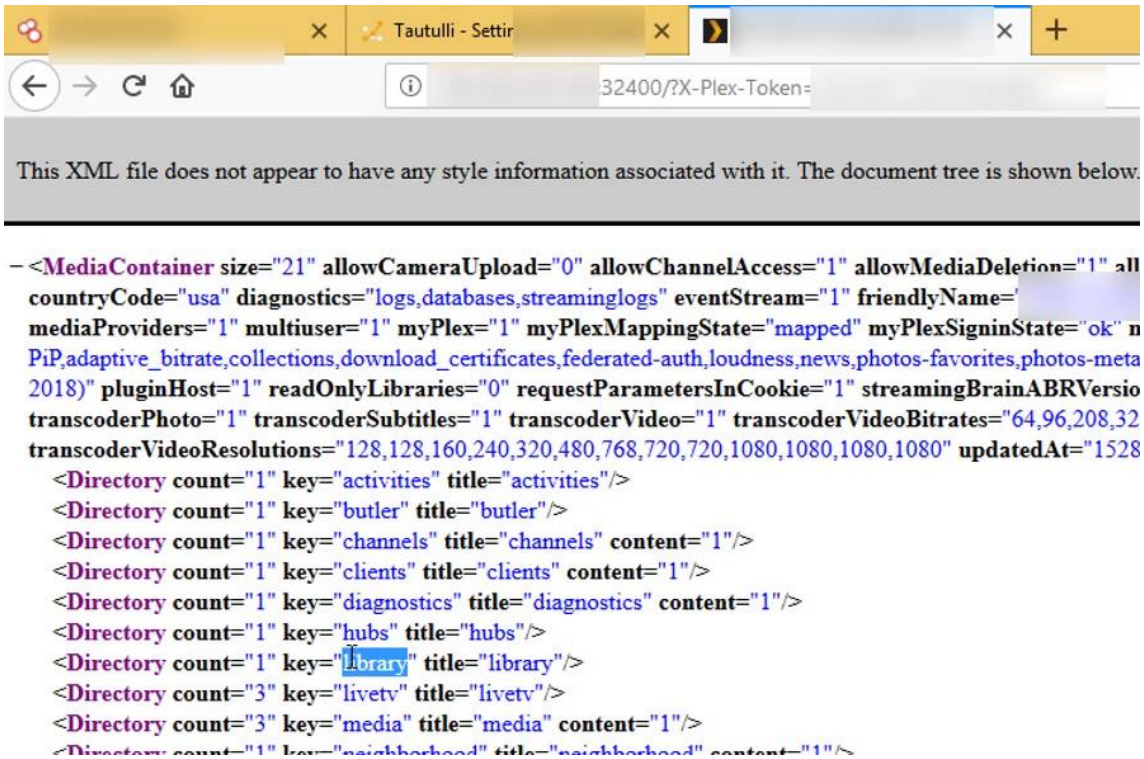

Figure 12: Inside *Plex* (PMS) under XML

2. Navigate like in a folder structure.

Figure 12 is root folder of PMS. To jump to another directory, the body content tells us the structure. The "folders" elements are identified with the name *Directory* and the attribute that URL "understands", is the value of *key* (the *title* attribute is describable info associated to the element), so if we want navigate inside *library* we put *library* (name/value of the *key*) between *IP:port* and the token as follows: *IP:32400/**library**?X-Plex-Token=YOURTOKENHERE*

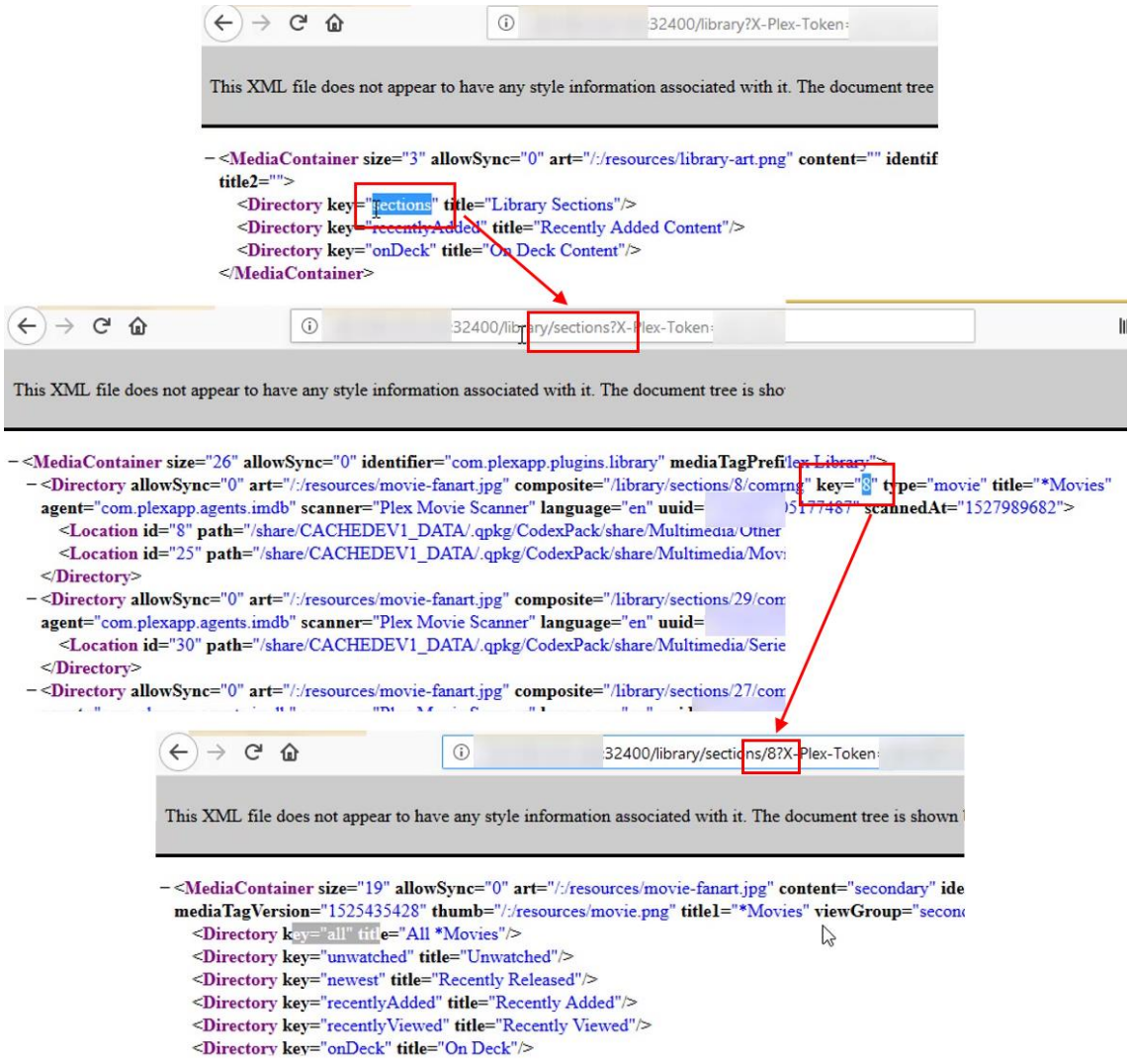

Figure 13: XML folder Navigation

Figure 13 shows the example of the folder tree like:

library

|__sections

|__8

|__all

URL representation:

http://IP:32400/library?X-Plex-Token=YOURTOKENHERE →
http://IP:32400/library/sections?X-Plex-Token=YOURTOKENHERE →
http://IP:32400/library/sections/8?X-Plex-Token=YOURTOKENHERE →
http://IP:32400/library/sections/8/all?X-Plex-Token=YOURTOKENHERE

3. And we arrived to the end of this step downloading a movie…

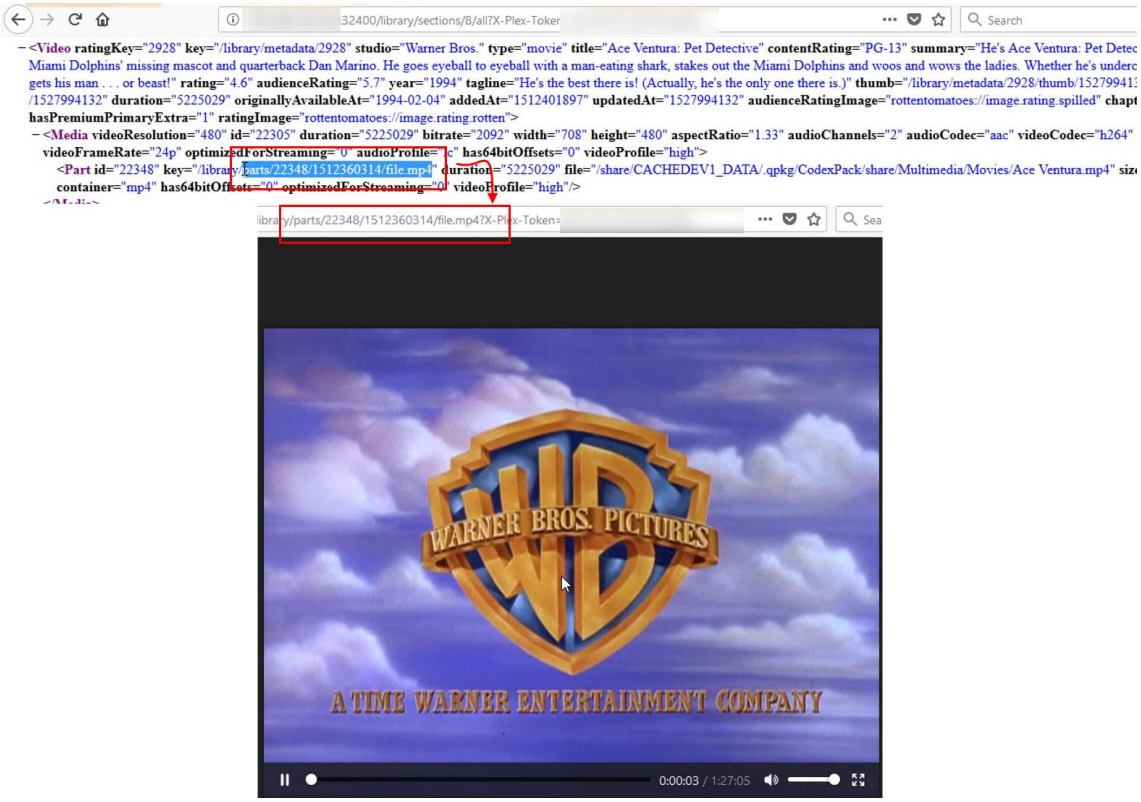

Figure 14: Downloading a movie from PMS

When we are in *.../library/sections/8/all* all the movies of the folder "*8*" are shown. The entire *key* of the *Part* element, will give us the movie, in this case is under mp4 format so *Firefox* browser can load embedding it.

Note: To know what movie is before the download, you can take a look on *title* attribute and there is suitable info in *file*.

If this is not clear enough or you want to see in action, no problem! Here you have a video:

https://youtu.be/e0UfKgBI5pI

5. Conclusions

Lot of things to think about it, right? Well, wait… little more here, in the "Bonus Track" (in conclusion, it's important to mention).

Tautulli (or *PlexPy* in his old version) offers a section to setup notifications and newsletters [6].

In the next figure 15, the form contains the username and the password, which seems protected but can be unmask easily using the developer tools in *Firefox* in that case, inspect the element and that's all!

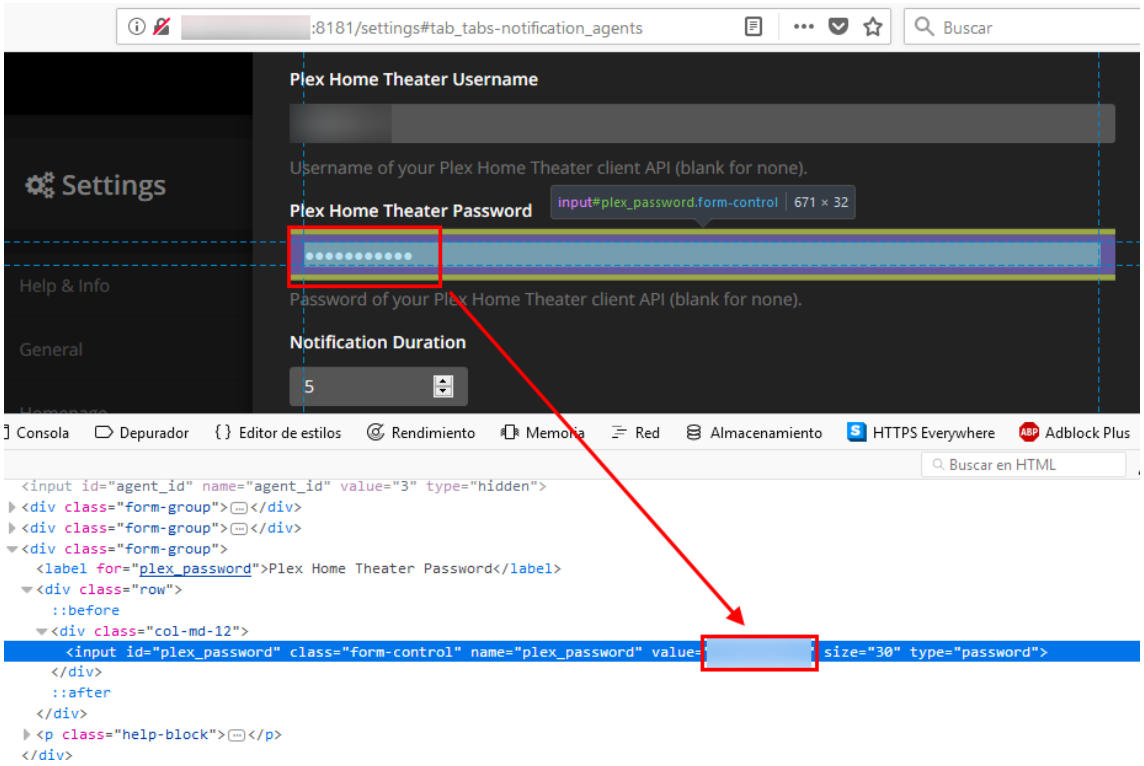

Figure 15: Unmasking the form password in Firefox

We have seen in chapter 4.2 in Step 3, how to access inside the Red Zone but what happens if it is not accessible from the outside (port not open or other security, things, rules…)? Don't worry, *Plex* save some of your info to "refresh" your mind :D [23].

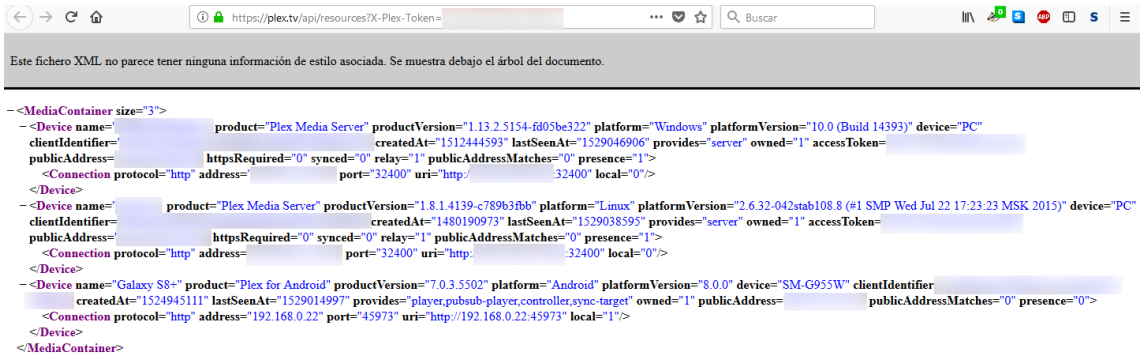

Figure 16: *Plex* resources information

Plex saves in his servers resources information of your PMS/s.

Above, in figure 16, we see lot of masked parts (for security reasons, of course) that "tells" about someone's profile, tells thinks like: name of device, client identifier, public address,… in this example, this user has two PMS and has the service in an Android Smartphone, he has a Samsung Galaxy S8, not bad… Therefore, if you can't connect directly to the local PMS, sure you will can see some (useful?) content publicly using the token (acquired through *Tautulli* as seen in figure 8 of the chapter 3.1).

OK, what a surprise… at least we have now more info to think about.

The situation is worrying, in my opinion, lot of (and I say "lot of" because I cannot imagine a number) multimedia content of the entire world is visible/accessible to anyone, not in a comfortable directly way (if you want to call it as that). You can say, "This type of things can only be broken by experts" or something like that, but, hey, the main goal is make it easy, in fact all things can be easy, if you understand something, can be divided in small "pieces" and sure at the end, you will get a big one working like a charm. "It's not rocket science" (but, is rocket science really difficult?!).

The most important we want to reflect in the document is not hack only for fun; we all need learn some valuable lesson. People need to know this is dangerous, you can't expose all your thoughts in public, you have a privacy, all we have it, and the digital content is not an exception, must be consider an extension, something that, is part of our live.

Through the "fingerprint" of our multimedia content, you can obtain very precise information, the data, amount of data, is useless if you don't filter (is like obtaining random noise without follow any pattern). If you control the BigData filtering well, and the source of this data is good, the possibilities are big, very "Big"!

Tautulli is the shadow of your multimedia content. In chapter 2.1.1, we told some of things of which he is able to track. If a malicious person (cybercriminal as synonym) can't access to your PMS, he will get other info (from *Tautulli*), important to do bad things.

Some users may have sensitive content (you can also store personal photos or videos) I mean… not only commercial videos/movies can be taken by a media server. Imagine a cybercriminal ask for a ransom for it (similar like a ransomware, but not necessary, in this case to encrypt the content) because they can divulge through social networks and/or other many networks that exist in internet. Would you like it? Absolutely NOT, other reason to "teach" this hack to the people, in order to take care and consciousness, with the possibility to reproduce the situation to see from another face and assessing the type of damage that this causes.

Little more above, we also have seen that if someone put extra info like user/password in using the notifications service (for example) this password in most of the cases is the same for lot of accounts of a person, imagine the damage in this situation, is worse than a virus, then you would become "the virus".

As usual, when a tragic accident happens to other, is common looking for help: calling the police, emergency number… Here is similar situation, this document pretends be helpful, pretend to be a help for alert of the outside dangers.

Shodan can become a double-edged sword, for evil o for good. *Tautulli* is able to track all your "media movements" but… also through XML & Token combination? No…

through this method you will be like a “ghost” appears and disappears, but no fingerprints on system!

“Oki Doki”, and now, what about the workarounds, how to protect?!

1. Use your common sense, like in a real life, if something seems not good, don't use the insecure option as first by default!

2. Don’t use HTTP when exist forms (user/password) on the website, use HTTPS (this will be mandatory) and, more important, put a user and a password that meets the minimum requirements.

3. Don’t active the UPnP (Universal Plug 'n' Play) [24] on router or, disable it if this is ON.

4. Is really necessary open the service to the outside? If the answer is yes, why not using a VPN? (Nowadays, almost all devices accept a client for it). One, as a good practice, could be open a single port for VPN, with a user to restricted access to your “first” perimeter and then, for the deepest perimeter, other type of credentials with more access…

5. If you are able to open a port, update all your digital world!!

6. Please assume; that the maximum and best security doesn’t exist, all things can be exposed, like a human, in all environments, so do not obsess ☺. Technology is imperfect because humans are not perfect.

As final words to add for closing; The actual owner, *JonnyWong16* is active and is aware of the bugs/problems of *Tautulli*, but maybe needs take care of the webserver part, of the *CherryPy* [18] because as you knows, is your “look in the street”. There are lot of versions above this, probably the version *5.1.0* (present version in *Tautulli*) has bugs and security flaws. Malicious people can attack using DDoS, thus collapsing the bandwidth and stopping services…

At hack process (chapter 4.2) we started using *Shodan* without login into this service. If you register an account, your filter can be more accurate (by countries…) and can see more (or all) results. This was made in a computer, a laptop but works well in a smartphone (the entire hacking process) or under low hardware requirements. Therefore, anyone can be a hacker using, practically, any device, making it, a largest range of damage scope (this is one of the goals of Amador Aparicio [1]).

Important to clarify; this is not against the products like *Tautulli* & *Plex*. This information was made and disclosed without any commercial intention and rivalry, totally free of charge and, again, done with to take consciousness of these type of situations.

Thanks for taking your time to read this paper, I assume, that it’s not perfect ☺ !

Remember…

Be Good, Be Hackers.

6. References

[1] Amador Aparicio's Twitter. https://twitter.com/amadapa

[2] Agenda of Cybersecurity IMPALA's event. https://twitter.com/amadapa/status/998881446331846656

[3] Shodan. https://www.shodan.io

[4] 360Fly 4k vulnerability [CVE-2017-8403]. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-8403

[5] Tautulli; Q: I forgot my username and/or password! https://github.com/Tautulli/Tautulli-Wiki/wiki/Frequently-Asked-Questions#general-q3

[6] Tautulli WebPage. https://tautulli.com

[7] PlexPy; choosing new name. https://www.reddit.com/r/PleX/comments/5zl1b4/plexpy_is_dead_rip

[8] Inuktitut language. https://en.wikipedia.org/wiki/Inuktitut

[9] Plex Media Server. https://www.plex.tv

[10] Tautulli on GitHub. https://github.com/Tautulli/Tautulli

[11] GitHub; acquired by Microsoft. https://www.theverge.com/2018/6/3/17422752/microsoft-github-acquisition-rumors

[12] Tautulli's changelog in GitHub. https://github.com/Tautulli/Tautulli/blob/master/CHANGELOG.md

[13] Tautulli releases. https://github.com/Tautulli/Tautulli/releases

[14] Birth of Tautulli in reddit. https://www.reddit.com/r/PleX/comments/7kmhl3/its_finally_here_tautulli_v2_beta_formerly_plexpy

[15] Creator of PlexPy on GitHub. https://github.com/drzoidberg33

[16] Actual owner of Tautulli on GitHub. https://github.com/JonnyWong16

[17] GitHub; changelog's history. https://github.com/Tautulli/Tautulli/commits/master/CHANGELOG.md

[18] CherryPy. https://cherrypy.org

[19] CherryPy releases. https://pypi.org/project/CherryPy/#history

[20] How to Install PlexPy on a Windows Plex Media Server. https://youtu.be/G2m5UJqHYRs

[21] Do I need a Plex account to stream locally? https://support.plex.tv/articles/207538527-do-i-need-a-plex-account-to-stream-locally

[22] Finding an authentication token / X-Plex-Token. https://support.plex.tv/articles/204059436-finding-an-authentication-token-x-plex-token

[23] Using plex.tv resources information to troubleshoot app connections. https://support.plex.tv/articles/206721658-using-plex-tv-resources-information-to-troubleshoot-app-connections

[24] Universal Plug and Play (UPnP). https://en.wikipedia.org/wiki/Universal_Plug_and_Play