

Forum:[Hack Forums](#)

Thread: [Eavesdropper - HTTP/S Traffic Sniffer](#)

AgedPeanuts - June 29, 2015, 5:50 pm

I've been working on this proxy for almost a week now, and with the benefit of sharing something, I'd like some feedback on it too. With that said, it's not 100% perfect, and you will notice some hiccups at times.

Quote:Fucking bs, a week? Copy pasting shithead

Well I had most of the code written already, but last time I touched it was around 4-5 months ago, all I did was re-factor it(which basically forced me to re-write it all).

I'm not that great at creating topics, but here is a screenshot of the demo running: Spoiler ([Click to View](#))

You may notice a lack of documentation in the source, but I assure you, I will go back and write it out. For now you'll have to stick to your gut and assume what a method does, I'm sorry people.

Unfortunately, this library was written in VS15 using the latest C#(6) version, so most of you might not be able to compile it out of the box.

The library comes with two event arguments under the names of:EavesdropperRequestEventArgs

EavesdropperResponseEventArgs

Both of which inherit from CancelEventArgs, yes that means either the request, or response can be canceled. Each of these event arguments also carry a property by the name of Payload(byte[]) which depending on the event, either carries the POST data, or the response data(which you can also alter). They also carry their respective HttpRequest/HttpResponse instances, which can also be changed...

Keep 'makecert.exe' in the same location as Eavesdropper.dll, it is in charge of creating the certificates.

[Eavesdropper on GitHub](#)
[Download\(Source, Demo\)](#)

_Raw - June 30, 2015, 4:56 am

I downloaded the compiled file, but they dont work.

cleerboy1 - June 30, 2015, 1:37 pm

You had better grades if you had done it in (.Net 4.0).

There are not many who have the new version yet.

AgedPeanuts - June 30, 2015, 3:32 pm

(Jun 30, 2015 01:37 PM)cleerboy1 Wrote: [\[CITE\]](#) You had better grades if you had done it in (.Net 4.0).

There are not many who have the new version yet. It primarily was using .NET 4.0, but I ported it over to .NET 4.5+ since I really wanted to play around with the async/await keywords.

I actually kinda want to port it back to .NET 4.0 now, but I have other projects that have my greater focus. The source is there, so if anyone desires to do so, go right ahead.

cleerboy1 - July 1, 2015, 4:56 am

(Jun 30, 2015 03:32 PM)AgedPeanuts Wrote: [\[CITE\]](#) It primarily was using .NET 4.0, but I ported it over to .NET 4.5+ since I really wanted to play around with the async/await keywords.

I actually kinda want to port it back to .NET 4.0 now, but I have other projects that have my greater focus. The source is there, so if anyone desires to do so, go right ahead.

What a .NET version on GitHub?

AgedPeanuts - July 1, 2015, 2:37 pm

(Jul 1, 2015 04:56 AM)cleerboy1 Wrote: [\[CITE\]](#) What a .NET version on GitHub? I changed the target framework back to 4.0, it's actually a lot more simpler to understand without all those 'await' keywords.

The interception of HTTPS traffic is 'buggy', so if you want to disable it and only listen for HTTP, you can set: `Eavesdrop.IsSSLSupported = false;`

I also split the code up a bit more, since it was sort of a cluster fuck.

cleerboy1 - July 1, 2015, 3:33 pm

Am I doing something wrong here?

<https://i.imgur.com/5eoqAKg.png>

Corios - July 1, 2015, 6:57 pm

(Jul 1, 2015 02:37 PM)AgedPeanuts Wrote: [\[CITE\]](#) I changed the target framework back to 4.0, it's actually a lot more simpler to understand without all those 'await' keywords.

The interception of HTTPS traffic is 'buggy', so if you want to disable it and only listen for HTTP, you can set: `Eavesdrop.IsSSLSupported = false;`

I also split the code up a bit more, since it was sort of a cluster fuck.

You should only be using await when you need to, and how did removing them make it simpler? It just returns execution to the calling thread when you no longer need it.

AgedPeanuts - July 1, 2015, 7:35 pm

(Jul 1, 2015 06:57 PM)Corios Wrote: [\[CITE\]](#) You should only be using await when you need to, and how did removing them make it simpler? It just returns execution to the calling thread when you no longer need it.

Yea, when I meant it was easier to understand, I meant it was better to read something like this:

```
Code byte[] requestPayload = GetRequestPayload(requestCommand);
```

Rather than:

```
Code byte[] requestPayload = await
```

```
GetRequestPayloadAsync(requestCommand).ConfigureAwait(false);
```

You're right about only using await when needed, I was just overcomplicating a lot of things that already could have worked really well without it.

qqppqq - August 19, 2015, 2:56 am

Hi AgedPeanuts!

I really need your help with some flash in Visual Studio?

AgedPeanuts - August 19, 2015, 7:12 pm

(Aug 19, 2015 02:56 AM)qqppqq Wrote: [\[CITE\]](#) Hi AgedPeanuts!

I really need your help with some flash in Visual Studio?! I'm not so sure what you mean by this man, although if it's not related to this particular project in any way and has something to do with FlashInspect?, then create an issue on that repo: <https://github.com/ArachisH/FlashInspect>

This repository was renamed quite a while ago, so the link in the main post won't work:

<https://github.com/ArachisH/Eavesdrop>

DragonHunter - August 21, 2015, 11:59 am

If all u do is grabbing links, why not use UrlSnarf instead, maybe bashes the purpose of coding it yourself but does the same thing really

Or if you were like "yes we can capture now all the traffic !" you could have used ARP Spoofer and use BDF Proxy just for the heck of infecting all .exe's the poor victim downloads ?

AgedPeanuts - August 21, 2015, 12:30 pm

(Aug 21, 2015 11:59 AM)DragonHunter Wrote: [\[CITE\]](#) If all u do is grabbing links, why not use UrlSnarf instead, maybe bashes the purpose of coding it yourself but does the same thing really

Or if you were like "yes we can capture now all the traffic !" you could have used ARP Spoofer and use BDF Proxy just for the heck of infecting all .exe's the poor victim downloads ? I really felt like making my own at the time, also because I didn't really like using FiddlerCore. I made this since I was working on a packet sniffer that needed to extract the host/port of the game's client loading page, and replace a variable in the response so the game connects to localhost.

Then when the actual flash game payload has been intercepted, it gets "parsed" so that I'm able to extract/replace the public RSA keys, the response is then replaced with the modified client and we do some handshake stuff and tada.

SIM-2944284 - October 18, 2017, 2:07 pm

Fix your links noob. Top tier shit anyways.

Washi - October 18, 2017, 2:41 pm

(Oct 18, 2017 02:07 PM)SpeAnon Wrote: [\[CITE\]](#) Fix your links noob. Top tier shit anyways.

Fix your eyes noob and check the date of the post. Top tier gravedigger.