

Forum:[Hack Forums](#)

Thread: [My backdoors are getting detected](#)

evilai - September 16, 2018, 3:39 pm

Hi all,

The backdoors that I am generating with veil-evasion, fatrat and empire. Do you and tips and tricks to make them undetectable with these tools or other tools.

And what antivirus scanner do you use? spyscanner or something like that?

CyberStrike - September 16, 2018, 4:50 pm

(Sep 16, 2018 03:39 PM)evilai Wrote: [\[CITE\]](#) Hi all,

The backdoors that I am generating with veil-evasion, fatrat and empire. Do you and tips and tricks to make them undetectable with these tools or other tools.

And what antivirus scanner do you use? spyscanner or something like that?

Well all of these are all well known I personally don't like to just compile whatever I work from source code then compile. Something else in Veil you probably are using pyinstaller to generate that exe generally pyinstaller tends to get flagged. Are you signing these executable's btw?

Edit: As for AV I do all testing on a VM disconnected from the internet with multiple snapshots each with their own AV product, depending on what a client is using.

sh4moon - September 16, 2018, 7:00 pm

Veil evasion is kind of made for evading av. If you can't manage with those, just make your own or mod the backdoors.

evilai - September 17, 2018, 1:44 am

(Sep 16, 2018 04:50 PM)SilentWatch Wrote: [\[CITE\]](#) Well all of these are all well known I personally don't like to just compile whatever I work from source code then compile. Something else in Veil you probably are using pyinstaller to generate that exe generally pyinstaller tends to get flagged. Are you signing these executable's btw?

Edit: As for AV I do all testing on a VM disconnected from the internet with multiple snapshots each with their own AV product, depending on what a client is using.

Thank you a lot for your input

I am using pyinstaller hmm... and no I am not signing it - would that have an impact and in what way? I am thinking of modifying the hex values later - maybe that will improve the results.

What is your workflow when you create backdoors? What tools do you use? How do you make them undetectable? Do you make your own scripts?

I look forward to reading your input

(Sep 16, 2018 07:00 PM)sh4moon Wrote: [\[CITE\]](#) Veil evasion is kind of made for evading av. If you can't manage with those, just make your own or mod the backdoors.

Thank you a lot for your answer.

If you have success with making them undetectable - what specifications do you use to make them undetectable?

CyberStrike - September 17, 2018, 3:00 am

(Sep 17, 2018 01:44 AM)evilai Wrote: [\[CITE\]](#) Thank you a lot for your input

I am using pyinstaller hmm... and no I am not signing it - would that have an impact and in what way? I am thinking of modifying the hex values later - maybe that will improve the results.

What is your workflow when you create backdoors? What tools do you use? How do you make them undetectable? Do you make your own scripts?

I look forward to reading your input

Thank you a lot for your answer.

If you have success with making them undetectable - what specifications do you use to make them undetectable?

I've described my processes in rather in depth details .. But I tend to start out from source code, from there depending on what language is being used maybe I need to call up different API's or say instead of using virtualalloc maybe I use something else, but I do make sure that the threads and process and memory is mapped properly. Once I have something that looks alright I move over to a VM I have multiple snapshots with various AV's which a client has, I will make sure I have pretty close if not the exact update and version of the AV. From there I really tend to take OPSEC seriously, so often I will use objcopy and just use sections of the file see where I get detections. If you read "Taking Back Netcat" that will help with scantime, but the idea is we find the signatures and then we break it up. There is a project on github AVSignSeek I think that's the name but it does pretty close to the manual method I use. The real trick is just having good opsec you want to protect your payload, because say that the whole file is burned accidentally the VM is not offline and uploads your file, that's not good total waste of time. I don't scan things on online scanners only time is for a demonstration or testing. Between ghex/objcopy

Runtime really is a whole set of things you will see people setting up timers all this jazz, I have found that having a loop where it does something for a certain amount of time until it's not being actively monitored by the AV, also during this time I will keep the memory permissions on just RW, maybe having one section (the payload) later on becomes X. Also signing is important AV's will give you more leeway with signed files, legit certs of course.

You have to remember that Veil payloads have been around wow for a long while now, and most all use pyinstaller when generating python payloads, so AV's tend to call EVERYTHING py2exe/pyinstaller/cxfreeze as malware. You will find that now a days AV makers just lump certain things as malicious and call it a day. Look at Empire their powershell payloads are really struggling. I hate to say it and it's not meant to be a put down but AV companies sometimes just don't know what to do look at how Microsoft Defender/Defender Cloud/Defender SmartScreen now operate. Smartscreen with block just says "Not signed it's malicious" GateKeeper for OSX does the same crap.

Example of backdoor in GO and scan not yet messed with runtime.

<https://hackforums.net/showthread.php?tid=5885565&pid=57607862#pid57607862>

AV Evasion (Included sources so others may use them for reference talks about runtime) :

<https://hackforums.net/showthread.php?tid=5808303>

sh4moon - September 17, 2018, 5:41 am

silent is completely right. personally I just use things that will already be on the system. like python, ruby, bash. I made a python meterpreter and just encoded it with 50 iterations of shikata ga nai as an experiment. surprisingly it had 0% detection rate. live off the land. use what's on the system if you can.

evilai - September 17, 2018, 11:48 am

(Sep 17, 2018 03:00 AM)SilentWatch Wrote: [\[CITE\]](#) I've described my processes in rather in depth details .. But I tend to start out from source code, from there depending on what language is being used maybe I need to call up different API's or say instead of using virtualalloc maybe I use something else, but I do make sure that the threads and process and memory is mapped properly. Once I have something that looks alright I move over to a VM I have multiple snapshots with various AV's which a client has, I will make sure I have pretty close if not the exact update and version of the AV. From there I really tend to take OPSEC seriously, so often I will use objcopy and just use sections of the file see where I get detections. If you read "Taking Back Netcat" that will help with scantime, but the idea is we find the signatures and then we break it up. There is a project on github AVSignSeek I think that's the name but it does pretty close to the manual method I use. The real trick is just having good opsec you want to protect your payload, because say that the whole file is burned accidentally the VM is not offline and uploads your file, that's not good total waste of time. I don't scan things on online scanners only time is for a demonstration or testing. Between ghex/objcopy

Runtime really is a whole set of things you will see people setting up timers all this jazz, I have found that having a loop where it does something for a certain amount of time until it's not being actively monitored by the AV, also during this time I will keep the memory permissions on just RW, maybe having one section (the payload) later on becomes X. Also signing is important AV's will give you more leeway with signed files, legit certs of course.

You have to remember that Veil payloads have been around wow for a long while now, and most all use pyinstaller when generating python payloads, so AV's tend to call EVERYTHING py2exe/pyinstaller/cxfreeze as malware. You will find that now a days AV makers just lump certain things as malicious and call it a day. Look at Empire their powershell payloads are really struggling. I hate to say it and it's not meant to be a put down but AV companies sometimes just don't know what to do look at how Microsoft Defender/Defender Cloud/Defender SmartScreen now operate. Smartscreen with block just says "Not signed it's malicious" GateKeeper for OSX does the same crap.

Example of backdoor in GO and scan not yet messed with runtime.

AV Evasion (Included sources so others may use them for reference talks about runtime) :

Thanks a lot you guys!! I value your input immensely and I am so happy that you take the time to give me advice. I come from a background in machine learning and data engineering and have

decided to change and focus a lot more on security to try to make the region that I live in a little bit safer.

If you have any other practical resources that you really like on malware generation ex backdoors and credential harvesters (both for phones and computers) and network security (offensive side preferably) then that would be very appreciated.