

Forum:[Hack Forums](#)

Thread: [What knowledge is needed for coding exploits?](#)

KeepYourChinUp - March 23, 2014, 9:10 am

I guess this question has been asked a thousand times but for what it's worth I did use the search function but didn't get exactly what I was looking for.

Say you've been programming for about a year or two in C++ and you feel comfortable with most of the basic stuff, what would be the next step one would need to take to start coding their own trojans?

I'm not talking about a lame virus that just trashes someones PC, I have no desire to harm anyones PC it's pointless, but what I would like to do is learn how to code trojans that infect PC's and give me a backdoor into the system / server.

I've read about hack news where someone hacks into a system and has had access to that system for like 6 months or something outrageous, how does one create a backdoor that goes undetected for 6 months! lol.

I don't know much about this but sockets is something I know I would need to learn, what else?

Ideally it would be nice to find a book that taught how to code trojans.

Trisomy 21 - March 23, 2014, 9:27 am

First of all you don't "code" exploits, you find them. And second you are probably months away from achieving what you want to.

KeepYourChinUp - March 23, 2014, 9:40 am

(Mar 23, 2014 09:27 AM)zer?day Wrote: [\[CITE\]](#) First of all you don't "code" exploits, you find them. And second you are probably months away from achieving what you want to.

I'm years away, I have not even started programming yet. I'm simply asking just out of curiosity really.

Trisomy 21 - March 23, 2014, 9:44 am

(Mar 23, 2014 09:40 AM)KeepYourChinUp Wrote: [\[CITE\]](#) I'm years away, I have not even started programming yet. I'm simply asking just out of curiosity really.

Well I can't help you out much here, but to answer your question about backdoors that remain undetected for 6months+, read more about rootkits.

mariio - March 23, 2014, 10:56 am

Well I suggest to learn how to use remote administration tools first if you're really interested in malware.

g4143 - March 23, 2014, 11:14 am

Why don't you ask what it takes to become a competent system programmer or network programmer or programmer in general.

KeepYourChinUp - March 23, 2014, 11:34 am

(Mar 23, 2014 11:14 AM)g4143 Wrote: [\[CITE\]](#) Why don't you ask what it takes to become a

competent system programmer or network programmer or programmer in general.

Because I don't even know what a "system programmer" is. Basically I'm interesting in programming and I've been reading up about it. Most of the stuff I can find online but sometimes I just think of a question that isn't easily found on google... So I ask it here.

I do have a plan laid out for programming already and none of it involves programming malicious things, it was just something which sprung to mind.

First I'll learn python and C++. I'll study them both at the same time because I think if I study one language too much I'll get bored of it.

I'll study these two languages for at least a year before moving onto another. Then I'll look into something web based, I'm not sure yet, JavaScript and PHP most likely, and of course HTML.

My goal is to study these most common languages to give me a basic understanding of programming in different areas. Then I'll have a good foundation on which to build on. I don't really have an end goal, I guess the only reason I want to learn about programming is for a hobby. Remember I have not coded anything before so my view will probably change but breaking into things seems more interesting than fixing things.

Anyways I have an incredibly long way to go and it will most likely be a couple of years before I even peak into the world of "hacking." It really was just a question based on curiosity :)

.Zero - March 23, 2014, 11:53 am

Hi,

Exploit development, remains one of the most difficult programming sectors to be successful in keeping in mind a lot of the exploits present in applications which are widely used tend to be developed by the major technology firms such as Microsoft, Apple. That being said, they have large customer base which according to probability should hint many bugs are squashed in the early months\weeks of the release. The bugs tend to be resolved fairly quickly in such applications.

This creates problems both for pre-exploit development and exploit integration. As once the exploit has been discovered it gets patched fairly quickly making all machines "immune" to similar exploits in future. The next problem is if a exploit is developed the product is "majority" of time the latest product. This means not many computers have the software for the exploit to be used.

Furthermore, exploits are almost in ALL developed applications which are non-commercial and are programmed by learners. The only problem is that they could have the biggest exploit which can lead to complete system compromise but the only issue is it is not a viable option to exploit as it is not widely used.

This should imply, a exploit developer must have a good understanding of the popularity and the size of the industry leaders. That will guarantee you will not waste time on developing exploits for a software which is barely used in general public.

Next, exploit development involves understanding system internals and system specific details such as kernel design or bootloader design. This is to be able to have a rough idea of where a exploit may lie. Then there is a skill which almost all exploit\pentester developers should understand is assembly and low level programming languages (C, C++ etc.) as the exploits tend to obviously change the memory in specific addresses to exploit the application and only low-level languages seem to be a good option. Don't get me wrong but even VB.NET, C# and other .NET oriented languages also have capability to change memory and build memory editing tools suggesting they too can build exploits but C, C++ and other low level languages offer ease of such changes thus

preferred coding language. There are also multiple definitions of exploit development, it could mean website exploit discovery but I take it you mean general PC exploit development rather than web-based exploit jive.

The MOST important is debugging, it is the greatest tools a exploit developer can have as all exploits are debugged and crucified instruction by instruction to see if there are any security loopholes which could be used. The last and final need for a exploit developer is his or her PATIENCE and CONCENTRATION and CREATIVITY.

As for backdoor programming its pretty simple learn socket programming, detouring, injection and you can create a backdoor. Its not too difficult however as for general malware programming sources in terms of books, there are not widely accepted or even published books as malware programming is more or less a illegal activity to commit. It is near fraud and money laundering thus its a very sensitive topic to publish books about.

There are however books about designing rootkits (a major component in a malware software) which however requires a good deep understanding on C language and is platform specific to Windows. The book it called:

Rootkits: Subverting The Windows Kernel by Greg Hoglund, James Butler

There are few free PDF link websites, a notable and a good PDF is: [Rootkits: Subverting The Windows Kernel PDF](#)

There are much more books or source\material available on internet if one searches properly.

Don't expect to be able to learn exploit development and be able to exploit them exactly how you previously learnt them, in majority of cases the exploit developer need to be creative and apply multiple methods to make a exploit or even sometimes improvise on a specific technique to get through a exploit development. Next, a software especially large & notable softwares, which you will normally try and exploit will contain millions upon millions of assembly code to step through and each of these instructions can lead to a effective exploit being developed and if a single line of assembly instruction is skipped or ignored expect the exploit to never be discovered, so a exploit developer must have extreme concentration levels. Then there are cases of patience being just as important as skill.

Exploit development is a long-cumbersome process which that too is not guaranteed to have a rewards. I am not just taking about the companies handing out rewards but the reward of finding the exploit itself. In some cases, exploits may be there but not discovered thus making entire operation a failure due to a reason or other. This means you must wait to find a other applications to try discover exploit to or try again.

Let me leave you with few Albert Einsteins famous quotes:

"It's not that I'm so smart, it's just that I stay with problems longer." - PATIENCE & CONCENTRATION

"Creativity is intelligence having fun." - CREATIVITY

Hope I helped,
.Zero

SureBoy - April 6, 2014, 6:11 pm

thanks .zero for the information, really appreciated.

zenawenliang - April 7, 2014, 5:57 pm

(Mar 23, 2014 11:53 AM).Zero Wrote: [\[CITE\]](#) Hi,

Exploit development, remains one of the most difficult programming sectors to be successful in keeping in mind a lot of the exploits present in applications which are widely used tend to be developed by the major technology firms such as Microsoft, Apple. That being said, they have large customer base which according to probability should hint many bugs are squashed in the early months\weeks of the release. The bugs tend to be resolved fairly quickly in such applications.

This creates problems both for pre-exploit development and exploit integration. As once the exploit has been discovered it gets patched fairly quickly making all machines "immune" to similar exploits in future. The next problem is if a exploit is developed the product is "majority" of time the latest product. This means not many computers have the software for the exploit to be used.

Furthermore, exploits are almost in ALL developed applications which are non-commercial and are programmed by learners. The only problem is that they could have the biggest exploit which can lead to complete system compromise but the only issue is it is not a viable option to exploit as it is not widely used.

...

...

"It's not that I'm so smart, it's just that I stay with problems longer." - PATIENCE & CONCENTRATION

"Creativity is intelligence having fun." - CREATIVITY

Hope I helped,

.Zero

This is fucking great, do you know a good place for learning about sockets using C++, hopefully that is specifically for malware? Also, do you recommend learning the TCP/IP protocol beforehand or could I learn as I go along?

Also, you have helped me before, I'm surprised you don't post more often!

Xorus - April 7, 2014, 7:33 pm

(Apr 7, 2014 05:57 PM)zenawenliang Wrote: [\[CITE\]](#) This is fucking great, do you know a good place for learning about sockets using C++, hopefully that is specifically for malware? Also, do you recommend learning the TCP/IP protocol beforehand or could I learn as I go along?

Also, you have helped me before, I'm surprised you don't post more often!

Personally, I'd suggest <http://www.madwizard.org/programming/tutorials/netcpp/3> for learning about sockets in C++. As far as TCP/IP you should REALLY learn about that beforehand. Don't try and learn as you go along, it will only make it more difficult. If you want to see an implementation of WinSock I've been working on an API recently that uses it for networking.

https://github.com/MonkeyWrenchTeam/Monkey-Wrench/blob/master/Monkey-Wrench/Wrench_Net.h

https://github.com/MonkeyWrenchTeam/Monkey-Wrench/blob/master/Monkey-Wrench/Wrench_Net.cpp

Understandably its not the most beautiful code written by far but it may be able to help you if you run into any barriers along the way.

EmptyRedData - April 7, 2014, 8:17 pm

(Mar 23, 2014 09:27 AM)zer?day Wrote: [\[CITE\]](#) First of all you don't "code" exploits, you find them. And second you are probably months away from achieving what you want to. Actually you do code exploits. You find vulnerabilities.

I guess the logical progression towards learning to make exploits is to find vulnerabilities. Once you can find them, making exploits is another skill to itself.

topLevel - April 8, 2014, 4:04 am

(Mar 23, 2014 11:53 AM).Zero Wrote: [\[CITE\]](#) Hi,

Exploit development, remains one of the most difficult programming sectors to be successful in keeping in mind a lot of the exploits present in applications which are widely used tend to be developed by the major technology firms such as Microsoft, Apple. That being said, they have large customer base which according to probability should hint many bugs are squashed in the early months\weeks of the release. The bugs tend to be resolved fairly quickly in such applications.

This creates problems both for pre-exploit development and exploit integration. As once the exploit has been discovered it gets patched fairly quickly making all machines "immune" to similar exploits in future. The next problem is if a exploit is developed the product is "majority" of time the latest product. This means not many computers have the software for the exploit to be used.

Furthermore, exploits are almost in ALL developed applications which are non-commercial and are programmed by learners. The only problem is that they could have the biggest exploit which can lead to complete system compromise but the only issue is it is not a viable option to exploit as it is not widely used.

This should imply, a exploit developer must have a good understanding of the popularity and the size of the industry leaders. That will guarantee you will not waste time on developing exploits for a software which is barely used in general public.

Next, exploit development involves understanding system internals and system specific details such as kernel design or bootloader design. This is to be able to have a rough idea of where a exploit may lie. Then there is a skill which almost all exploit\pentester developers should understand is assembly and low level programming languages (C ,C++ etc.) as the exploits tend to obviously change the memory in specific addresses to exploit the application and only low-level languages seem to be a good option. Dont get me wrong but even VB.NET, C# and other .NET oriented languages also have capability to change memory and build memory editing tools suggesting they too can build exploits but C, C++ and other low level languages offer ease of such changes thus preferred coding language. There are also multiple definitions of exploit development, it could mean website exploit discovery but I take it you mean general PC exploit development rather than web-based exploit jive.

The MOST important is debugging, it is the greatest tools a exploit developer can have as all exploits are debugged and crucified instruction by instruction to see if there are any security loopholes which could be used. The last and final need for a exploit developer is his or her PATIENCE and CONCENTRATION and CREATIVITY.

As for backdoor programming its pretty simple learn socket programming, detouring, injection and you can create a backdoor. Its not too difficult however as for general malware programming sources in terms of books, there are not widely accepted or even published books as malware programming is more or less a illegal activity to commit. It is near fraud and money laundering thus its a very sensitive topic to publish books about.

There are however books about designing rootkits (a major component in a malware software) which however requires a good deep understanding on C language and is platform specific to Windows. The book is called:

Rootkits: Subverting The Windows Kernel by Greg Hoglund, James Butler

There are few free PDF link websites, a notable and a good PDF is: [Rootkits: Subverting The Windows Kernel PDF](#)

There are much more books or source material available on internet if one searches properly.

Don't expect to be able to learn exploit development and be able to exploit them exactly how you previously learnt them, in majority of cases the exploit developer need to be creative and apply multiple methods to make an exploit or even sometimes improvise on a specific technique to get through an exploit development. Next, a software especially large & notable softwares, which you will normally try and exploit will contain millions upon millions of assembly code to step through and each of these instructions can lead to an effective exploit being developed and if a single line of assembly instruction is skipped or ignored expect the exploit to never be discovered, so an exploit developer must have extreme concentration levels. Then there are cases of patience being just as important as skill.

Exploit development is a long-cumbersome process which that too is not guaranteed to have a reward. I am not just talking about the companies handing out rewards but the reward of finding the exploit itself. In some cases, exploits may be there but not discovered thus making entire operation a failure due to a reason or other. This means you must wait to find another application to try discover exploit to or try again.

Let me leave you with few Albert Einstein's famous quotes:

"It's not that I'm so smart, it's just that I stay with problems longer." - PATIENCE & CONCENTRATION

"Creativity is intelligence having fun." - CREATIVITY

Hope I helped,

.Zero

Thanks for taking your time on this post it's a good read, and thanks for the ebook I will be checking it out soon.

Torrída - April 12, 2014, 4:39 pm

Implementing exploits into code is easy. Finding the exploit is what is hard.

jammy47 - December 6, 2018, 10:40 am

You need cert collection baseline. They teach all exploit stuff.

you_can_never_hack_me - December 7, 2018, 9:31 pm

You follow the Rules.

Good luck!