# ECE 6747 Advanced Topics in Malware Analysis

Introduction Transcript

>> Hello everyone, and welcome to Advance Topics in Malware Analysis. I am Professor Brendan Saltaformaggio from the School of Electrical and Computer Engineering here at Georgia Tech. For this week, we're gonna cover an introduction to the course, what you can expect, and what everybody's in for analyzing some malware.

So the agenda for the course, for this module, is gonna be broken down into an introduction for the course. And then we're gonna go over x86 Assembly Language. And that's gonna be very important for understanding the malware that you're looking at and reverse engineering the malware samples and all of the labs.

We're gonna go over the different registers and how assembly code stores and handles data. The different executable files that you might encounter when you're reverse engineering malware. And then some subtle differences between malware disassemblies that focus on Intel versus AT&T syntax. And we're gonna get through all of that in this module.

But before we begin, just an introduction to me. My name is Professor Brendan Saltaformaggio, Salt-uh-for-mah-gee-oh. Yes, that's very long. No, you do not have to remember it. I tend to go by Professor Brendan, is just fine. My research interests are in cyber forensics and computer system security. I am the director of the cyber forensics innovation lab here at Georgia Tech.

And my lab really works on different types of binary analysis, memory forensics, and the vetting of untrusted software. So pretty much understanding what malware do and how they do it and how to defend against malware in large scale attacks. If you wanna get in touch with me, the best way is always to go through email, my email is right there on the slide.

And if you want to know any information about this course, or anything else about me, feel free to go to my website. So this course follows a kind of strict chain of command. First, if you have any questions, ask yourself, is this question private? Is this something that I don't want the rest of the class to hear about such as your cat getting sick, you want to kind of take an exam a week late.

If the answer to that is yes, feel free to email me or come to my office hours. I'm more than happy to help everybody out with anything that might come up during the semester. So feel free to get in touch with me. If the answer to is it private is no, like if you're curious about what happened on your last lab, why you lost so many points.

But it is a personal question, it's not something you necessarily want getting out to the whole class, we have a team of extremely skilled TAs who are gonna be absolutely happy to help you with that. I've built this course with the help of these great TAs and they know everything inside and out.

So any questions you have about your performance on the course, or technical questions, anything of that nature, feel free to go straight to the TAs. They're gonna be absolutely happy to help you. And then finally, what I want to stress is, if it's something that could benefit everyone in the course, please post on Piazza so that we can all discuss it and answer the questions collectively.

This course is really about cutting edge topics and malware analysis. And there's often gonna be not a single right answer to any question. There's gonna be different ways of approaching problems that change as we learn new techniques. So please kinda treat this course as an open discussion. I know there's a lot of people enrolled, but this course always works out best when everyone is involved in discussing these cutting edge topics.

So I'm always gonna be on Piazza. The TAs are always gonna be on Piazza, and so are your classmates. So please reach out to them, ask any questions you have on there. The best offerings of this course are when everyone is discussing things actively on Piazza. So an outline of this course, if I were to give you the whole course in just one slide.

We are gonna be learning advanced approaches for detecting vulnerabilities and malware within binary software. And each part of that sentence is very important. So I'm gonna break it down. When I say advanced approaches, I mean very, very cutting edge approaches. The field of software security is rapidly changing.

And there's no textbook out there that can keep up with such a fast paced field. Instead, what we're gonna do is we're gonna study published research papers from the recent top academic venues that look into these problems at the cutting edge. And they try to understand these problems and produce new approaches to solving these problems.

When I say detecting vulnerabilities and malware, this is actually built on a few fundamental principles of software analysis. And that's what these lectures are gonna be about. I am going to cover over the course of these lectures the building blocks of software analysis and, particularly, binary analysis. These building blocks individually aren't that difficult to get a handle on.

And, through the labs, you're gonna apply the building blocks, you're actually gonna make most of those building blocks yourself in the labs. So that you understand fundamentally how these principle techniques can be composed to accomplish very complex malware analysis tasks. And binary software here, this is also important.

I wanna stress that malware do not come with source code, malware authors do not want you to understand easily what they're trying to do. And most people, when they write programs, they write in some high level language like C or C++. And then a compiler works its magic and spits out a binary executable that just runs natively on your CPU.

That binary executable is what you find when you're doing malware triage. It's just the executable of that malware that's on that system. So that's what we as malware analysts are going to be looking at. We're gonna look at that binary code. The ones and zeros sometimes of actually what is executing raw on the processor.

And it's with that that I wanna kinda take a pause and give everyone a warning that this is not necessarily gonna be an easy, go by the books, kind of course. This is a very time consuming and challenging course. Primarily, because malware analysis is spy versus spy. You are a reverse engineer trying to understand what someone else's code is doing.

And even worse, the someone else here is a malware author. They're actively trying to hide from you and make it more difficult for you to understand what their malware is doing. So it's gonna require a significant amount of work to complete each assignment in the allotted time. Because you're actively going to be working against this malware author.

And in this course, we're going to be looking at real true malware samples that have been collected from the wild. And so nothing is watered down in this course. Everything is as real-world as I could possibly make it. So please start every assignment as soon as it opens.

Please don't wait till the last minute. Because if you're under a huge time crunch, the malware author is gonna get the better of you. And it's gonna be really difficult to complete each assignment. Malware analysis is going to become your new hobby. If you don't love malware analysis and software security, if you're not into that sort of low level assembly language C programming.

That's gonna make this course a little bit more challenging for you. If you don't already know assembly language, that's okay. We're gonna learn assembly language as we go. I know it's not something that people learn in just a general computer science curriculum. But you really do need to understand C and C++.

Those two programming languages are going to be essential for getting through this course. Assembly language we will learn, in fact, we'll learn it later on in this module. But you've got to at least be comfortable programming in C and C++ to get through this course. And with that warning, I wanna reassure you that getting through this course is a very rewarding experience.

Every time I've offered it, students have come back and told me things like, this class doesn't aim to mollycoddle you. And I appreciate that. It encouraged you to aspire for more and push your limits. Only in that extreme can one learn so much so well. And that's really what I shoot for in this class.

I don't want to make it any more difficult that it needs to be. But analyzing malware is definitely a challenging and time consuming thing. So you'll hear other things some students that I've taught before. One of the most rewarding and challenging courses I've taken at Georgia Tech. And this is a funny one, the amount of sleep I lost over this course was enormous.

But we were warned, and that's what I'm trying to get across here. This course was great. And, probably my favorite, professor Brendan is a boss. Again, I do everything I can to try to make this course as digestible as possible. But you're gonna be fighting against real malware.

And I hope at the end of this course you too would like to malware again.

## Week 1- Introduction p2

>> So the course here is gonna be divided into two halves. The first half of the course we're gonna cover those binary program analysis building blocks. Basically, what you're gonna hear about used in the research. You're gonna have to understand the different techniques what is a control flow graph?

What is a data dependence graph? How do you construct these things? This portion of the course is gonna be a traditional lecture format. And during my lectures you're also gonna complete six program analysis labs outside of class. 4 of those labs are gonna be static analysis using Ghidra.

Ghidra is a cutting edge tool developed by the United States government to do malware analysis, and we're gonna go over Ghidra a lot in the slides to come. After those, we're gonna do two labs that use dynamic analysis with a tool called pin. Pin is also a very mature binary analysis platform that's built by Intel.

So we're gonna talk more about these tools in the later videos. We're gonna use them to complete the labs and you're basically gonna be putting together those basic building blocks that we talk about in the lectures. Each lab, again, is gonna require very careful time allocation to complete.

I'm gonna give everybody one or two week deadlines for each of those labs, the harder ones, of course are gonna have two weeks. But that's because I really do hope that you'll spread out the effort over those two weeks, because it's going to take a lot of careful time allocation to complete them.

The second half of the course we're gonna focus on cutting edge research and like I said before, there's not really any textbook that can bring you to the cutting edge of software security and cyber forensics. So instead of, we're gonna be studying published research papers that, as you will see, leverage those building blocks that we talked about in class to put together very advanced malware analysis frameworks.

We're gonna learn about the limitations of these techniques, and we're gonna see how researchers are trying to solve the open problems that are there in malware analysis. And sometimes we're even gonna talk about how they're failing to solve those problems. When you read each of these research papers, I want you to pay close attention to the following.

Number one, what's the problem that that paper has identified that they're trying to solve? And then number two, why have the previous solutions or the previous published research failed to solve that problem? That's gonna have spurred the authors into number three, a novel solution that they're presenting in their paper.

What is it that makes their solution novel? How did they get above that bar to publish a brand new research paper in this cutting edge field? And number four, consider the limitations of their approach. What did they try do and since you yourself will have been working on these building blocks throughout the class, you'll know where they falter.

Where are their false positives where the corner cases that the authors might not have considered? And what are some future research opportunities that you may see in this field? I promise this is a very hot field of research and there's definitely a lot of room for future research.

So grading, there's no midterm, and there's no final exam for this course. And that's because I believe the best test of understanding how to combat malware is to actually combat the malware. And you're gonna be doing that in six labs. Each of those are 15% of your grade divided equally.

The grade is gonna be based on the results produced by your tool. There's a pretty clear right answer for each one of these the TA is all know how to build these tools and they've done it themselves in most cases. So you can go to them for help.

We're basically gonna be checking over was the results produced by your tool accurate to the malware sample that you're analyzing. For some of the labs, if there's any problem running your tools or understanding how they worked, me or the TA is might schedule some time to do a demo during office hours just to go over the results.

There's also 10% of the grade baked into class participation. And this goes back to what I was talking about with Piata where really this course is meant to be a discussion. A lot of times when you're implementing your tools for the different labs, there's gonna be open problems and something might not necessarily have a clear right answer.

If you discuss it on Piazza, it's all gonna become very clear, and I hope that that's where most people are gonna turn to help out themselves and others in the class. So 10% of the grade is there to kinda keep Piazza active and have everyone working on their class participation.

And I'm definitely gonna offer a few small extra credit assignments. Please pay attention to those, they'll absolutely help out your grade. They'll be mentioned in the slides, they'll also very likely be on Piazza before they open in the slides. And I definitely want to touch on cheating because each of these labs are done individually or in teams of two.

But in a lot of cases the line between discussing on Piazza, and sharing code, or even comments from your code, can be very thin. And so I wanna make it clear that I have a zero-tolerance policy for cheating. I do want everyone to discuss different ideas or different approaches that they may have to solving the problem.

But the line cannot be crossed. Please do not share any code or even comments from your code on Piazza. That's a no go. In order to make sure that everyone's code is unique, I'm gonna reserve the right to use MOSS which can detect substantial overlap in different code bases.

And this has been shown before, to identify cases of cheating and I do reserve the right to use that. Please review the Georgia Tech Honor Code, in case you have any questions. I have a zero-tolerance towards violating that. If you're caught cheating, I'm gonna give you a zero on that lab assignment.

You're gonna have a permanent one-letter grade drop in the class. And you'll be reported to the dean, which puts an academic warning on your file. So please just don't do it. So finally the goals for this semester. I hope everyone comes away from this course, learning how to apply the fundamental principles of dissecting malware.

Finding vulnerabilities and triaging cyber attacks. This is gonna come back to those basic building blocks of program analysis that we gonna be learning. I also want you to become aware of the limitations of these existing mechanisms and how to avoid them. If you do malware analysis as the career, you're going to come up with cases where the malware has outsmarted even the best of tools, but nothing can replace your brain.

And if you know to look for these limitations, you can find ways to outsmart even the best malware authors. We're also gonna be reading cutting edge research papers and engaging in critical discussion around these research topics. And I really hope that one thing you take away from this course is the ability to dive into the cutting edge in malware analysis.

Because, like I said, this field is very, very, hot and if you're able to keep up with the research, then you're able to keep up with the cutting edge of this field. And based on that we're all gonna be working together to propose solutions to these open-ended research problems.

And I just wanna touch on the programming requirements one more time for this course, because this course is going to require heavy programming. This is a three credit course, but by the end of it, you're gonna feel like it was a four or five credit course because of all the programming you are gonna need to do, to combat these malware samples.

You must be proficient and see, because when I talk about code and the slides or assembly language in the slides, I'm always gonna go back to see. As something that everybody in the course can understand. So you must be proficient with C to be successful in this course.

The course is going to be a little bit better if you know Python or Assembly Language already. But don't worry if you don't, it's okay if you don't know those languages just yet. We're gonna learn all of them in the course especially assembly language. In fact, that's the whole max slides set that we're gonna go over as an introduction to assembly language.

By the end of this course, everyone is gonna be a master in assembly language, and you're gonna be able to tackle even the most difficult malware samples that are written in any language because you're going to have seen them disassembled in assembly. So with that, I'll see everyone in the next video and we're on to Introduction to assembly language.