Part1

1p

Hello, my name is Jeff Gennari and I am here to welcome you to the LG software security class. The first lecture which will be a gentle introduction to our class and what we can anticipate over the next five weeks also.

2p

So our classes of remote class, meaning that you guys are in Korea and we are in Pittsburgh Pennsylvania USA at Carnegie Mellon University. We're going to conduct this our go over the logistics of the class coming up. But more or less we're going to conduct this class with short lectures on various topics and software security engineering, software security analysis and defense. and then we will have lots and lots of exercises to cover concepts in a practical way. And we'll have a few meetings, a week virtual meetings on zoom to go over the materials of class do question and answer that kind of thing. So today in this first session we plan on doing an overview of the Program. An introduction to who we are, as your faculty myself and a few others who are going to help. We'll talk a little bit about Carnegie Mellon and why Carnegie Mellon is a good place to study software security talk. And then we'll talk more about the logistics of the program particular how the project is going to work. How you're going to complete this project and teams and then some ground rules with our course policies.

3p

So, my name is Jeffrey Gennari. You can call me Jeff. I've taught a couple different classes for LG over the last I don't know five or six years. I've been involved in the architecture program I ran a program for teaching software quality assurance called the debt training program. So I'm pretty familiar with LG and how they teach software engineering. I've been at Carnegie Mellon for approximately 16 years coming on 17 years all of that, within the CERT program, which is the program in a at Carnegie Mellon that focuses on software security I got my start doing software vulnerability analysis. I transitioned into doing more malware analysis and nowadays I build tools to do software security analysis and assurance. And then I also went to CMU, you in the software engineering program so I've been involved in many different parts of the university.

4p

Okay, so let's get into what we plan to cover as we embark on this course. So this course is going to be about software security and not just software security from one particular perspective. Because security encompasses a wide variety of concerns. Of course, you have things like vulnerabilities. These are fairly topical nowadays in the states with solarwinds and how that vulnerability. Solarwinds really called into question how vulnerabilities reporting is handled and management of patches might look at that, as a case study every now and then one of these a flaw a software vulnerability is released like specter. Or this the bleeding heart here is his heartbeat which was an openssl vulnerability that are significant events and you as a developer, have to be aware, for example in the case of Heartbleed. Are you using openssl in your product or you're using code that uses openssl in your product. Specter is one of those things, it was a low level of vulnerability, it took advantage of the design of CPUs. And solarwinds is another example of a piece of software that's installed everywhere. At least no large number of software network management systems, so the impact of the vulnerability was huge. Vulnerability analysis, vulnerability management and vulnerability prevention are going to be key parts of our class. UX engineers, software engineers, and developers at LG are really the forefront of trying to keep software vulnerabilities from becoming actual problems in software products that LG creates. So we'll see how that process. How that how it will see tools and techniques that you can use as developers in a forward thinking way to prevent software security issues from ever happening in the first place, at least vulnerabilities.

5p

More and more software is finding its way into all of our lives. So in this picture I have some physical systems. There's a road sign here to do same managed construction, it was obviously hacked. And then on the right, we have the interior of the Tesla if you ever been in the Tesla which is not your vehicle. It feels more like a computer than it does an automobile. With more and more autonomy being given to vehicles, some of the questions about software security when cars are online when more of the self-driving decisions are made by a computer and it's not connected to the Internet. It's becoming more and more critical that's obvious that security is going to be a significant factor in these types of systems and so we'll spend some time considering not just IT systems, but nontraditional systems safety critical systems. Cyber physical systems and how security fits into those worlds.

6p

Of course malware so you know, unfortunately, you know you don't suffer vulnerabilities and shoddy implementations are certainly a problem. But malware can take advantage of just the basic design of software. If something is vague, not necessarily a technical flaw or through clever social engineering, or perhaps just dumb luck malware can be installed on a system and us as developers and as analysts developers need to design systems that can resist certain types of exploitation, perhaps conducted by malware by design and then analysts need to understand what malware does. For a variety of reasons to understand what's been compromised understand how to stop it to understand where weaknesses are in there in your systems. We need to be vigilant in a variety of ways managers need to make investments in preventing malware from proliferating around the organization. So we're going to spend a little bit of time talking about malicious code how it works, how to detect it, how common tools detected to understand the limitations of those tools and understand what defenses are available to us to prevent malware from proliferating on our systems.

7p

A related a solarwinds and openssl Heartbleed are really examples of attacking the software supply chain. This is a problem that's becoming increasingly obvious in software engineering. We don't build systems from complete scratch, we use libraries off the shelf technologies and outsource different elements of our systems to consultancies and you know developers that don't have the same standards as our own organizations. And so, when we do that we introduce a level of risk into our system. So, for example with openssl is an open source library, you can fetch it. The source, you can go look at the source code for it, you can fetch it's used in a wide variety of systems so it's a challenge for developers when you went to buy as a component from an organization. That organization might use a vulnerable version of openssl something that's memorable to, for example, Heartbleed and now by virtue of acquiring that component. You know or your system becomes vulnerable by proxy, how do you manage that, how do you keep track of what elements are in your system. Or, in your software supply chain, this isn't as simple as just looking at what people use because more and more you get things like code clones and copy and pasting software from websites like stack overflow, github, MSDN all of these places, so you vulnerable code is posted, for example on github and everybody starts copying that vulnerable code, the vulnerability proliferates without being a specific implementation. So we're going to talk a bit about how software supply chain security and management and how processes can be put in place to do verification and not just of your systems, but up the systems that you use as developers so stay tuned for that.

8p

And then finally we're going to take a look. At the people aspects of software security, so you might not be managers per se, but it is good to understand kind of the human factors that come into software security why people either don't follow security protocols or become insider threats where somebody attacks an internal person and employee to company. For example, turns on that company neither conducts espionage or sabotage is the organization. So we need to pay attention to that human factors are part of security or an intrinsic part of security, we cannot separate them. So the point is that we're going to cover lots of different we're going to approach security from many different angles.

9p

We're going to take a more of a holistic view of security in the five weeks we have together. And a lot of the work we have. Planned touches all aspects of security in that way. So **software security, it really does impact all phases of software development** I'm not just thinking of for development I'm thinking of what happens after you release your software. What happens when you're maintaining software deployed in the field, certainly software engineers have a role to play. During forward development from inception through design, construction and transition out on deployment. Analysts dedicated software security analysts quality assurance engineers have a role to play in evaluating the security of a piece of software, not just for identified security requirements, but for things like intrinsic security implementation level. Security will be cold mobile code quality implementation quality so you might have specific security requirements that you elaborate hopefully you do. But there was also an intrinsic security of your system. Did you use secure coding practices, did you identify implementation flaws that could lead to software and vulnerabilities. These are concerns that span all the design and requirements with regard to security. And then you know management has a role to play to not just in making sure that standards are followed. Organizational standards were followed. But instituting processes and plans to deal with software once it's out in the field, so what happens when vulnerabilities are reported in your system, what happens during an incident what processes, do you have in place, how do you handle things like insider threats, how do you train your employees to follow good security practices. All of that comes into play. and we want to expose you to do, regardless of your role we want to expose you to all of these different aspects of software security because we think that you'll benefit from understanding, you know kind of the world around software security it'll make you a better engineer and a more appreciative of why certain things, why we're teaching you certain things.

10p

So software is one of the critical points of this class will be that we don't do anything in a vacuum. we have to deal with users, some of them know what they're doing some of them do not, we have to deal with, as I mentioned the **software supply chain**, so we really don't know under what conditions or decisions are made and components that we use. We have to deal with expectations, the change so a common story in security related issue. Or security systems is that we developed. We design and develop a system for one environment under a set of assumptions and requirements. System simply outlives that environment or it gets repurposed into a different environment. So you see this all the time, you know we make a big investment in a particular system. We establish a good strong user base, and then we want to extend it or we want to deploy new features. Or we want to retarget to work on different hardware. We don't scrap the original system. We modify it, we patch it, we update it, and we add functionality and in that functionality, we add complexity and we compound that we make it so that the original intention of the system becomes harder to discern and that lead that might be to decisions and design decisions or limitation decisions that are suboptimal when it comes to security. You see this all the time in control system software that was designed to work in a complete isolated air gapped environment but gets repurposed to be put on the Internet. So that you can easily interact with it or pull in data out of it, without having to send in specialized engineers. Unfortunately, once you put things on the Internet now you have to deal with a whole different threat environment and often sometimes those systems that were designed to be operated in isolation don't do so well when they're put on the Internet. We will be aware of that, as we go.

11p

Something that makes this difficult is that it's very hard to define security if I asked 100 engineers what security means I probably get a few dozen different answers. You know it's everything from your **system's ability to resist attack** to your system's ability, **to recover from an attack**, to your system's ability **to detect that it's under attack**. And some combination thereof, besides that other aspects of designing a system to handle security concerns would be, how do you assess what's happened to the system is there a way to do a **damage assessment** if a compromise does happen. How easy is that how easy, is it to take stock of what's happened what's changed for good or bad. There's an emerging. I don't want to say field, but an emerging idea that you can conduct **retribution**. Attack the attackers to get rid of them or evict them from your system. Some organizations don't do that. Some organizations just monitor what's going on. So it becomes kind of a cat and mouse game and then of course there's **compliance** so certainly here in the States compliance with security. Software security guidelines and regulations is becoming more and more critical. There are rules about what to do in the event of a compromise when people's personally identifiable information is compromised and we need to deal with that.

12p

So let's do a quick thought exercise. To really drive home this idea that security is a multidimensional concern. Let's think about a denial of service attack. Denial of service attack is where you have a coordinated attack on a particular networked entity, like a server to knock them off the Internet or take them down one way or another.

13p

It sounds like a relatively straightforward attack. But it's actually very hard to discern it from just a performance concern. So there is a phenomenon. Performance testing called the **Slashdot effect** there's a website called Slashdot.org effect. It's kind of like a hacker geek website and. If you are a researcher and you post an article or a paper or something, and it goes to Slashdot it gets put on posted on Slashdot. It means that it's garnering a lot of attention and it's not uncommon for all of a sudden. People will rush to your website once you get posted the Slashdot and your website crashes I'm going to look. Nearly indistinguishable from a denial of service, in fact, it is denial of service is just for legitimate purpose. And that's really the fundamental difference, even what we talk about dial of services and security problem, but we evaluated as a performance issue we the metrics we use, throughput traffic volume. All of those things are performance characteristics, **the big difference is intent.**

14p

We also make think about it, **not just this performance but as a reliability robustness concerned.** So this has to do with how we handle denial of service when we responded to a denial of service, we can take a few different approaches. We can absorb it, we can try to stop it. But we usually want to have our system survive it. Not go totally down. So we might frame our response to the low service in terms of robustness and or reliability so again it's really hard to kind of distinguish it from security.

15p

know you start considering intent, then you start that leads you to a question of who's conducting this attack. So **if you can actually decide that we're under you know malicious attack well.** That speaks to this idea that you can detect such a thing is happening and that's easier said than done. And then you know another way to respond to the attack might be to actually stop it by identifying them malicious actors and black holing them or cutting off their access to your site. It does this also raises the question of Is this something that we want to deal with internally. Do we want to build some capability to deal with denial of service or would we prefer to outsource this to some consultancy, of which there are many nowadays, who will defend you from denial of service attacks.

16p

**There's also some value in understanding the specific type of attack that's happening.** Because it helps you understand it might help you identify or stop the attack. So, is this a pure volume attack or we dealing with an entity or a set of entities that are overwhelming our resources. Such so that legitimate users can access the system is there a vulnerability and play here. like this, or are we being manipulated or some is some protocol or a flaw somewhere being exploited, you know server, for example, or web server. to conduct the attack and can we do something about that, and then in a related is there is this more like an application level attack so there's a very type of denial service called a slow loris attack that takes advantage of application level flaws in web style web like systems, the amount of traffic is less important than the specific type of traffic that sent. As I mentioned, intention matters not just you know, are we dealing with a denial of service or legitimate traffic spike. But why is this a ransom issue where people trying to extort money from us. Is this a political statement is it just unfortunate so understanding kind of that gets to the understanding the whole situation and why it happened that might lead to different decisions when you decide things like that we need to worry about this in the future, or. Re-architect our systems to be a bit more resilient or indeed contract with somebody to handle this type of issue.

17p

So the denial of service thought exercise is really meant to show you the breadth of the concerns. In software security. And so we're going to explore the that though each of those concerns throughout the class are the we're going to give you just enough theory on security to be dangerous, but we're really going to focus on practice to that end we're going to have lots of exercises that are simulate real world security phenomenon and scenarios. And this is all going to be centered around a capstone project for the class which we will get into in the next mini session.

18p

we're going to approach specific topics from different perspectives, from the perspective of the software engineer developer we're going to go through all the phases of software development design requirements coding implementation transition, devops, And then the role that developers have to play in quality assurance from the perspective of the security analyst we're going to look at how you can analyze artifacts to identify information relevant to your organization to defend or attribute various aspects of incident analysis artifact analysis and then from the management perspective we're going to look at how do you plan for security budget, how do you set up a reasonable incident response, capability, how do you manage your people what kind of training, do you do so, stay tuned and We will soon in the next session we're going to talk about our course logistics.

Part2

21p

So let's continue with our introduction to the course focusing on course logistics, starting with the other instructors who will be interacting with you during the class. Jonathan Woytek is an employee of the software engineers to hear Carnegie Melon and the CERT program with me. He's been focusing on software security primarily malware analysis and secure development operations for the better part of a decade at least. So he'll be focused more on the deployment aspects of computer security, development operations infrastructure system analysis those types of things.

22p

David Belasco is a malware analyst at Carnegie Mellon he's also an instructor who teaches a class in software reverse engineering key will be are primarily talk to you about incident response and analysis and malware analysis and some vulnerability analysis.

23p

Dan Plakosh is in charge of the project, so the project, he devised the project scenario and will be giving you a presentation about what the project means. Dan has been involved with LG for many years and in various capacities and he is incredibly adept technically when it comes to software security and secure coding, especially.

24p

Why come to Carnegie Mellon to study computer security or Carnegie Mellon has a rich history in computer science in general, and more specifically in computer security.

25p

We're here in Pittsburgh Pennsylvania if you're interested if you're curious in other times you would come and visit. Unfortunately, nowadays, with the way things are, we have to this remotely, but we still want to give you a great experience.

26p

Software security at CMU is also has a rich history. CMU. Years ago, decades ago actually. started the was made it a host of an institute federally funded research and development Center known as the software engineering institute the software engineering Institute was created to advise the US Government on software engineering related activities, mostly related to procurement an acquisition support. Now, when the Internet was just beginning it was relatively small entity, there were a couple of high profile incidents, one of them being the **Morris worm** which was one of the **original proof of concept Internet worms** that was actually designed to just map out the Internet kind of like a spider but ran out of control and ended up dragging Internet down. Really congesting the Internet ran out of control, and so the US Government in during that quote unquote incident recognize that it was they needed a place to go to get information about software security computer security incidents that was reliable. So they test the CMU and the software engineering to with create standing up such capability and that was the CERT division. One of the original centers for cyber security. In the United States. And so, with that Center with the CERT division placed at CMU computer security education has just emerged around it, as is really a force in industry so. That is why we feel confident that we will be able to confidently will be able to teach you about software security we've been studying it for literally decades.

27p

So our program is going to be focused on really three core areas. Software security engineering, **how do you build secure systems**, **how do you analyze systems to identify security issues** and then, **how do you manage all of this in the context of a larger organization**. Our programs designed to be intense, we only have five weeks and we want to focus on some specific things right.

28p

Our goals are to introduce you to a sought to a set of techniques to effectively evaluate software security. We want you to understand how security verification and analysis fits into the context of a larger project. We don't do this in a vacuum, we want to build a toolbox of security knowledge tools techniques that you can use on your job, and we want to really, I call it thinking like an attacker right security is a dynamic field it changes, very quickly, and so the things we teach you today, or will be relevant, but new threats will emerge new techniques will be discovered, and so you have to adapt. and so I find one way to do that is to train yourself to really think as though you were an adversary so approach the problem as an analyst or an adversary would.

29p

So each lecture will have be about 15 or 20 minutes you've already watched this will be your second one. will have two or three in person sessions those times have been hosted later we'll sit down as a group, virtually, of course, will resume and we will discuss will do an exercise, maybe a laboratory to bring your laptops or have access to your laptops. And we'll walk through an actual exercise will be relevant to the course into the project. The other details like the weekly examination and other evaluation activities will be discussed with you as the program proceeds.

30p

So our program goals and philosophies are a little different than a typical college course. We're going to focus mostly on practical instruction. So there'll be a bit of theory and described descriptive describing underlying principles. But we're primarily going to be a hands on class. We're going to the project is designed to give you a realistic experience, evaluating and developing secure systems. The exercises and laboratories are meant to walk you through real it real life tools, you can use on the job. And the structure of the program is meant to cover the core areas of software security. So, very little in terms of abstract theory in this class we're going to treat you as professional engineers.

31p

A couple things to keep in mind as we go through the class. we're going to go through a number of techniques, both from the perspective of the adversary, and the defender and we'll find that there are some common ways to think about different approaches to security and different defenses for security attacks, some are very sophisticated and require a bit of understanding complexities, some are very simple yet effective some have been known for a long time, some cutting edge the advantages, you get technical advantages, you get conducting an attack very and often you have to change things together. The way that we assess damage is also going to vary by attack, and so we will look at how is thinking about the consequences of. Thinking about the consequences of an attack factors into whether or not how we should deal with an attack and then finally. Not VIP few defenses are rarely 100% effective understanding the limitations of those defenses can be key in using them effectively so we'll take a look not only at the technical aspects of each Defense but, when and how effective they can be.

32p

There'll be assignments again more on this, as we get closer to the program the logistics will come out more or less will be a weekly assignment. We will read them, we will have a reading and then a simple quiz.

33p

The readings, are going to be over a broad range of topics there's no one specific textbook to offer security is a big field. And it's hard to find a textbook that covers everything that we want to talk about not just you could find textbooks that cover secure coding, security analysis, security testing, but it's rare to find a book that covers everything. So we're going to select papers and chapters and articles from various sources.

34p

The cornerstone of this class is a project, the project will be completed in teams, you should have been assigned a team, by the time you see this. We only have five weeks. So it's going to be kind of an expedited project and but we will complete this project. Nonetheless.

35p

The project is going to be broken into two phases, Dan will talk about this technical details of the project, the short summary is we're going to build we're going to develop retrofit an image analysis system, a face recognition system written in c++ and we're going to re-architect at some degree to prove security there's a new security requirements, the system is by is not complete so it's not really re architect thing it's more continuing initial development, so your team is going to do that, using the best software security engineering principle practices and principles, you can we will teach you those. And then, at about the midway point of the class so two or three weeks in you're going to swap projects with other teams and you're going to take on the role of an attacker and you're going to attempt to compromise their the complete system you're going to look for security relevant defects vulnerabilities you're in you're going to you're going to evaluate the security, both in terms of implementation and adherence to requirements. This is really where you'll learn to think like an attacker.

36p

Included into guide you along the way, you will be as each team will be assigned a mentor your mentors are professional security. Researchers at Carnegie Mellon or faculty members who have experienced working with student teams and industry experience in their role will be to more or less consult with you on the project to ask you questions, make sure you're applying the techniques, we learned in class as we go. And kind of guide you along the way they're not TEAM members, they don't do the work for you they're more like advisors who help you appreciate the meaning of the project.

37p

So they will give you hints provide advice they'll typically ask you lots of questions and challenge you. That may be a little off putting it beginning but it's all meant to stimulate discussion and thought about what you're learning they're going to push you to try new things right, one of the reason, one of the things we'd like you to try to do here in this program is to step outside of your comfort zone and embrace new. New approaches to software security and your mentor will be there to help with that process.

38p

Here's our cheating policy, you can read it yourself I don't expect to have to invoke it. We're going to work as honestly, as we can.

39p

Cooperation is another thing we encourage cooperation as much as possible, but your work must be your own What that means is you can discuss assignments, you can discuss approaches to the project or whatever you'd like, but please your when you write up your work to submit that must be done on your own.

40p

So with that get ready, you know this program is going to be fast we only have five weeks it's going to be intense, but it should be a lot of fun and very hopefully beneficial to you and your growth as a security software security professional. So I look forward to working with you and I know the other instructors are excited so welcome to Carnegie Mellon University.