Practical Cyber Security for Cyber Security Practitioners

Prof. Sandeep Kumar Shukla

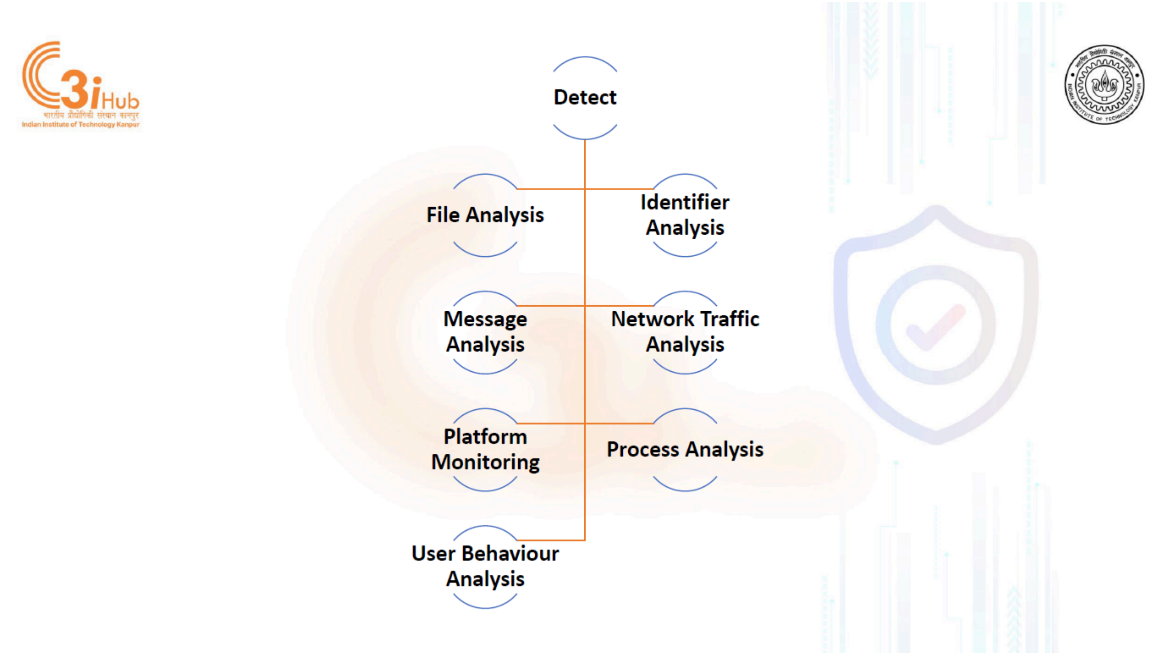
Department of Computer Science and Engineering

Indian Institute of Technology, Kanpur

Lecture 17

Deep dive into MITRE DEF3ND framework-I

Hello there. So this is in continuation of our discussion on the MITRE defend model. So we are recording this portion of the discussion on different model so that we can complete the discussion and move on to the next topic. So let me show you the slides first. We'll go from there. OK, so we were discussing.



Detect

File Analysis

Identifier Analysis

Message Analysis

Network Traffic Analysis

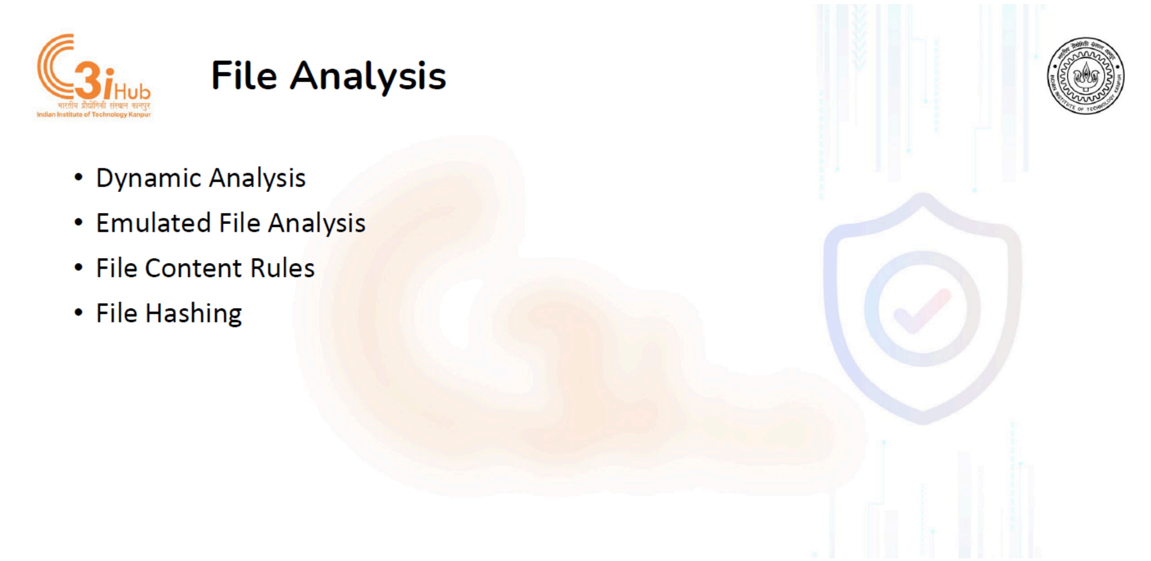
Platform Monitoring

Process Analysis

User Behaviour Analysis

So first tactic in the different model was to actually harden. So we talked about hardening the application, hardening credentials, hardening messaging, and then hardening the platform. Then we also discussed that hardening gives you some relief from attacks and also reduces the probability of an attack being successful. Doesn't mean that the attackers will not be able to circumvent the attack, circumvent the attacks and sorry, the defenses and especially hardening based defenses.

So we have to always be on our toes. And we have to always think in terms of detecting whether some attack is going on. And in this context, we have the detect tactic. And within the detect tactic, there are several sub tactics like file analysis, identifier analysis, message analysis, network traffic analysis, platform monitoring, process analysis, and user behavior analysis. So let's see what they are.



File Analysis

- Dynamic Analysis
- Emulated File Analysis
- File Content Rules
- File Hashing

So file analysis, you are mostly familiar with file analysis. Whenever a file is downloaded or delivered in some fashion to your system, to an endpoint, it is important that that file is analyzed using some kind of an anti-malware, anti-virus system so that you know that at least the file is not malicious or does not contain any malicious payload. As far as known payloads are concerned, known malware, known virus, known malicious content is concerned, doesn't mean that there is no zero-day vulnerability, no zero-day malware which has not been seen before, whose signature is not known by signature-based analysis. Sometimes we also apply machine learning-based file analysis techniques. In such cases, sometimes even zero-day malware or zero-day malicious content may be detected or at least certain probability may be assigned to the file that it might be malicious.

And based on your risk appetite, you might decide to actually discard the file or you may actually accept to use the file for whatever reason the file has been delivered. But you may actually decide based on your risk appetite. But when you do file analysis, there are many different ways to do file analysis. For example, dynamic analysis. So dynamic analysis involves executing the binary or executing the script that has been delivered, but not in your own environment but you actually use a virtual machine or some kind of a container in which to actually execute the file and observe its behavior, collect information from its various activities that this execution entails.

For example, what kind of IP addresses or URLs it is trying to connect to, what kind of registry changes it is trying to make, or if it is creating any other files, what kind of system calls it is making, what kind of libraries it is loading, et cetera, et cetera. So this kind of information can give you a good indication of whether the file when executed has any kind of malicious activity involved. So that's dynamic analysis. We often call it also detonation. So we detonate the file in a sandbox environment so that we can observe it and if the observation is satisfactory, you are more or less sure that this file is not that harmful, you may decide to actually accept the file, otherwise you can quarantine it or remove it based on your policy.

Emulated file analysis could be another one where you use an emulator instead of a virtual sandbox. So you can emulate the execution. File content rules. So you may actually have rules like ERA rules. So many of you probably know about ERA rules or sigma rules that are used for actually encoding signatures for malicious files so that whenever a new file is downloaded into the system the error rules are applied or sigma rules are applied and and they if they actually trigger then you know that at least some signatures are matching with the new file and you could actually then decide what to do with that file.

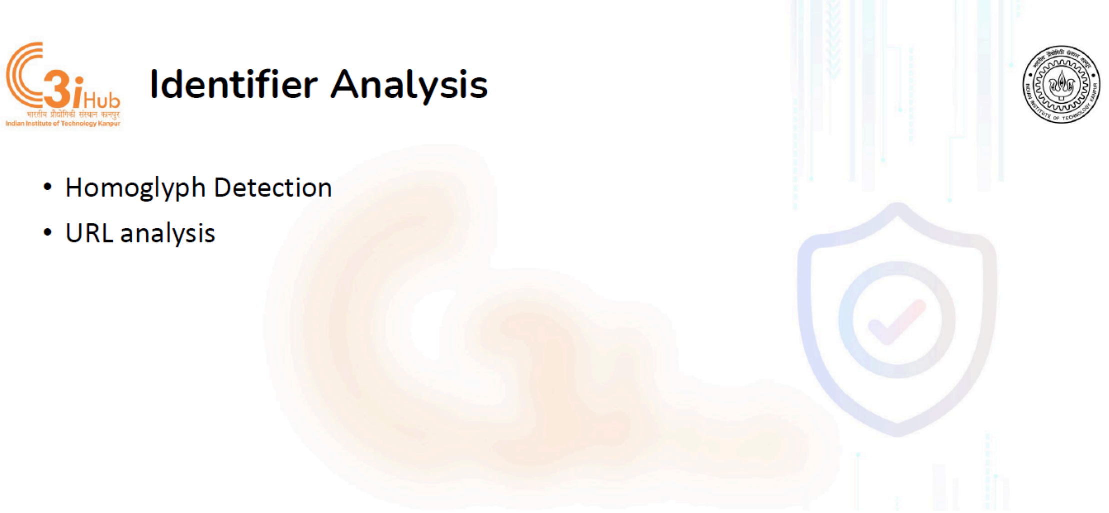
Quarantine it or remove it. File hashing, of course, there are certain malicious files whose hashes are well known. You can actually maintain a list of hashes of malicious files. So every time you get a file, you hash it first and match it against hash and the hash hashes are basically some somewhat like a signature of the of the file. So if you get a list of hashes of malicious files, and if one of your files match matches with that, then you know that you do not want to use that file.

So this is one way to detect that you have a malicious payload or malicious file inserted into your system. Doesn't mean this is 100%. You can actually have very, attackers may use very clever techniques to avoid any kind of signature-based or ERA based or even emulation or dynamic analysis-based file analysis. And it can evade all kinds of detection. But you have to do your best to reduce the possibilities.

And unless the attacker is very, very resourceful, and these kind of attacks are difficult to do unless to which will avoid all kinds of detection mechanisms that are currently available. Of course, you have to have the detection mechanisms in your system. Many people do not even use antivirus in their laptops, desktops, or on their phone. And then, attacking such systems are obviously easy for any attacker. It doesn't have to be resourceful.

any kind of identifier that you notice in your system. For example, if you notice that when you are clicking on an URL and that URL looks similar to a well-known URL, a safe and well-known URL. But actually, there is a slight difference. For example, the letter A is replaced by alpha, or there are various other ways to actually fool the eyes of the person to make him think that he or she is going to the right website where they're not. This is called a homoglyph.

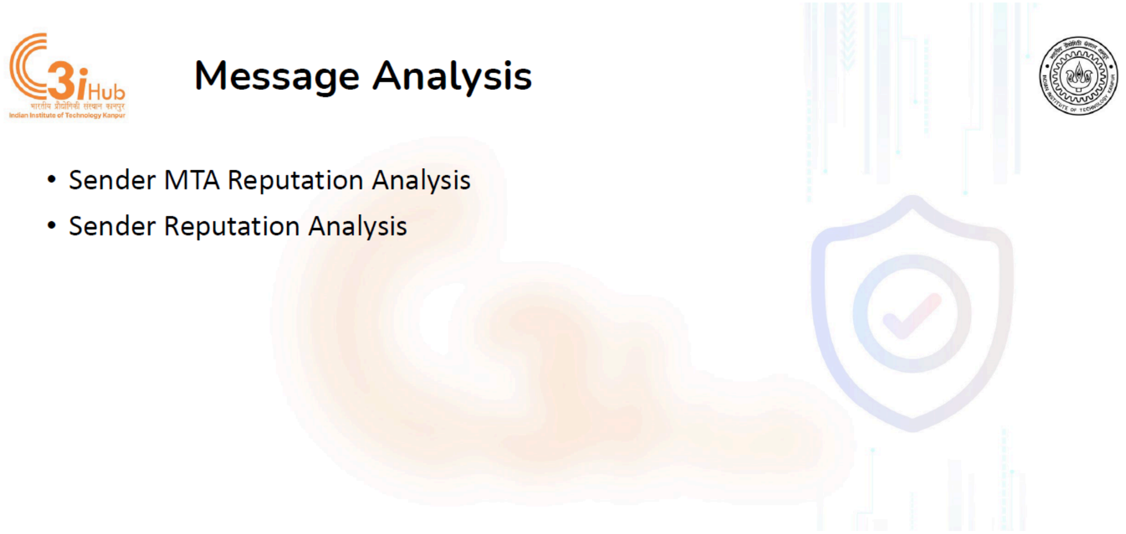
And then homoglyph detection is one technique by which we actually do this. We can check whether the users are being sent to malicious websites. analyzing URL for various other things. For example, suppose by monitoring the network traffic, you see there are certain URL being communicated to, then you may want to analyze the URL to see whether this URL is a malicious one or It's an algorithmically generated one, which is a sign that some command and control communication from inside your system is being attempted to an URL that is temporary.



Identifier Analysis

- Homoglyph Detection
- URL analysis

And so you can then... block the command and control channels to reduce the harm done by the persistent malware that is sitting inside your system and then work on removing that malware afterwards. Message analysis, when you get a message, whether it is an email message or whether it's a HTTP request or whether it's a REST API call or some other API call, you may want to know where this is coming from. So this is a message transfer agent reputation analysis, like where it is coming from, whether this IP address is malicious, whether this IP address is supposed to be the one that is associated with the domain from which this message is supposed to come.



Message Analysis

- Sender MTA Reputation Analysis
- Sender Reputation Analysis

So this kind of analysis can be done. There is also in the email security, there is a security mechanism called SPF that is actually associated with this kind of checks. Sender reputation analysis, so you can also maintain a reputation score, especially in an environment where you are aware of possibility of a sender system being compromised. So by looking at its behavior over time, you can actually increase the reputation or decrease the reputation. And then based on the reputation, you may decide to not accept some messages or some API calls, et cetera.

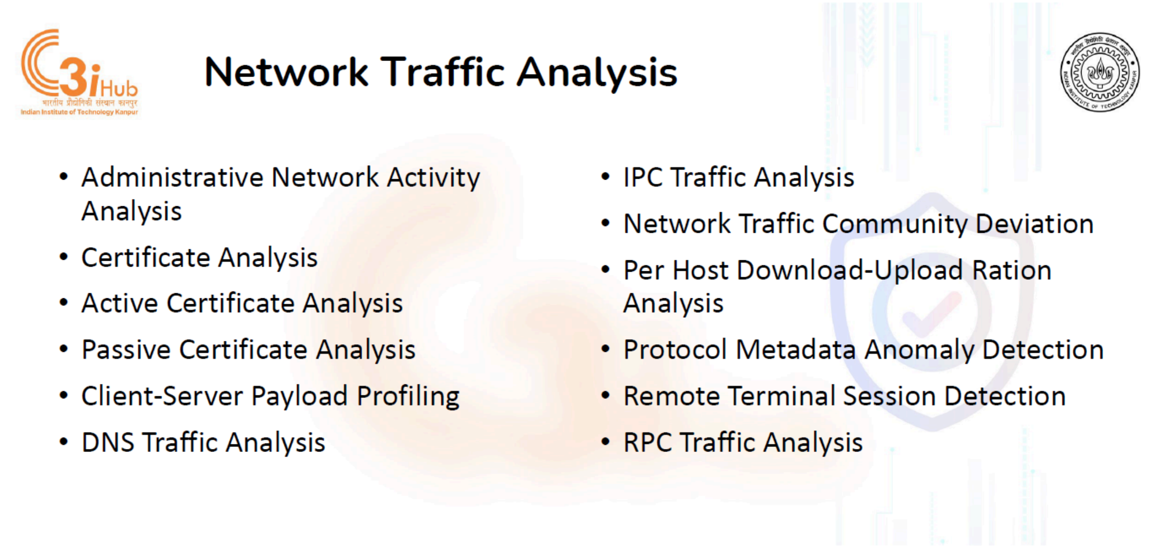
For example, often iitk.ac.in gets blacklisted by some reputation, some agency that keeps reputation because somebody inside the IIT Kanpur system might get infected and a lot of spam email might be generated from iitk.ac.in email addresses. When that happens, then this reputation score keeps going down and eventually it gets blacklisted.

Then the system administrators of the iotk.ac.in has to go and talk to this reputation, this blacklisting organizations to actually get out of the blacklist. Otherwise, a lot of different companies, they actually check this reputation scores or these blacklists before they accept emails. So oftentimes, you send an email and they keep getting bounced from certain domains because of this reputation problem and then we have to fix the reputation problem.

So this is a defensive posture and all these defensive postures come with certain disadvantages with respect to convenience and with respect to usability. So security is always compromised between usability, user convenience, and actually security. confidentiality, integrity, et cetera. So whenever you actually, when as an organization you decide to consult the blacklists of domains and reject the emails, there may be some important emails for some of your users that may be rejected and that could be very annoying and very inconvenient. Sometimes it might be actually pretty problematic for business, etc.

Therefore, it is important that if you are into business, if you want to keep your business working properly, you want to have your domain's reputation intact and you do not want any email addresses, any user inside your system getting compromised. Because if they get compromised, their platform may be used for spamming their platform may be used as a bot for a botnet. And then once that is noticed by others, they will start blacklisting your domain. And then other regular users of your domain who are not at fault will actually get their email bounced. So it is important to actually keep the system's reputation up and always keep the reputation.

So there are companies which will also tell you whether your domain has a reputation problem, whether it is being blacklisted, whether it is being misused by other entities, or whether there are data leaks from your domain into the dark web. CloudSec, for example, is an Indian company that does this kind of business. So it's a brand reputation kind of protection type of service that they provide. So network traffic analysis, all of you know about tools like network monitoring tools, such as, you know, Zeek, for example, or Snort, or Sudikata. These are network traffic analysis tools.



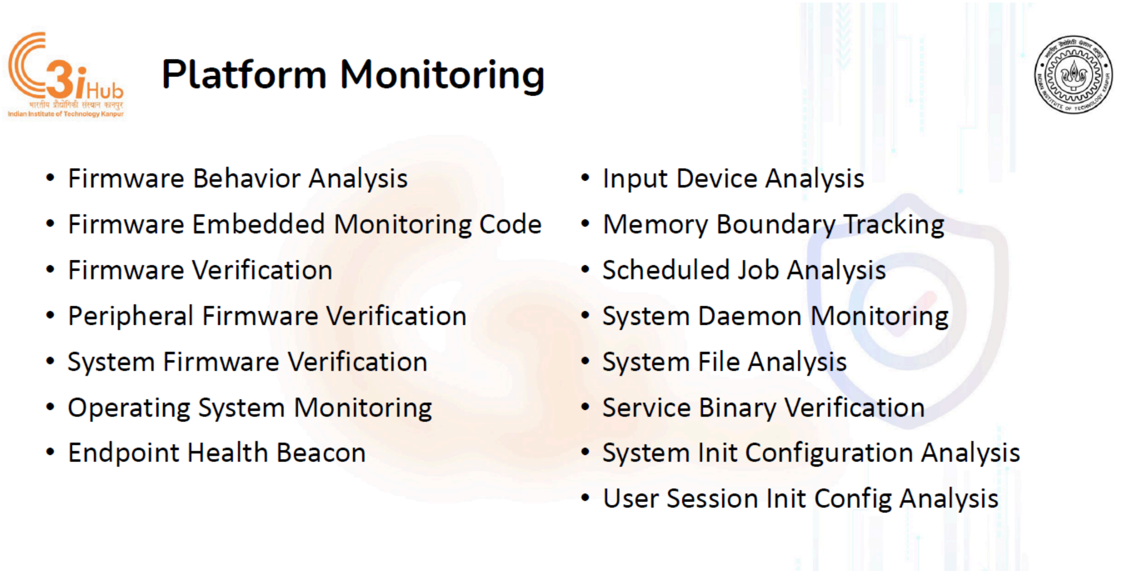
Network Traffic Analysis

- Administrative Network Activity Analysis
- Certificate Analysis
- Active Certificate Analysis
- Passive Certificate Analysis
- Client-Server Payload Profiling
- DNS Traffic Analysis
- IPC Traffic Analysis
- Network Traffic Community Deviation
- Per Host Download-Upload Ration Analysis
- Protocol Metadata Anomaly Detection
- Remote Terminal Session Detection
- RPC Traffic Analysis

So they actually mirror the main ports, main ingress port of your organization, and then they start analyzing each packet, right? And depending on how well you want to do network and traffic analysis, you might terminate the encryption at the main router so that the network traffic analysis tool can actually see decrypted packets in case you are using, for example, the IPsec or any kind of encrypted in any any kind of encrypted or tunneling system or you could actually do a you know shallow inspection of packets where you do not see the payload because it is encrypted but you see the headers and so on so depending on how you want to do this you may see different level of depth, different level of analysis. But you want to do the analysis of digital certificates that are coming into your system because they actually come in through the network when somebody connects to a, let's say, HTTPS server. And then you want to analyze the certificate to see whether any fake certificate is coming in to fool the users at the browser level, although nowadays most browsers can check for that. You can also do client server payload profiling, which, for example, if one of your users or one of the clients inside your organization is communicating a huge amount of data to a server, outside, you may want to know why is it they're communicating insider information like intellectual property, etc. outside their exfiltrating IP to outside this kind of stuff, you may want to see what kind of DNS requests are going because if there is a command and control traffic, there will be DNS traffic You may want to see the regular IP traffic analysis.

You can see any kind of traffic deviation for a particular set of endpoints or power host download upload analysis. Protocol metadata anomaly detection, the metadata means in the header, the data that is in the header. Remote terminal session, if somebody is connecting to a remote terminal, any kind of remote procedure call or API calls, this kind of analysis. So all kinds of analysis will give you some idea about what's going on at the macro level. It will give you what is going on from inside your organization to outside and from outside the organization into your system.

what kind of data is coming in, what kind of activities are going on, whether there are some unusual activity being carried out by a host or by a user like exfiltrating large amount of data or connecting to malicious websites or malicious FTP sites and so on. So you could do various kinds of analysis for knowing and generate alerts. These alerts will then be shown on the in the SOC, the Security Operations Center screens, so that the operation center analysts then will take notice of that and do something about it or escalate it to higher levels for taking some kind of an action. So platform monitoring is more about monitoring the endpoints. and may have a software agent sitting at every endpoint and detect certain activities and report certain activities to the Security Operations Center and then, if necessary, generate alerts.



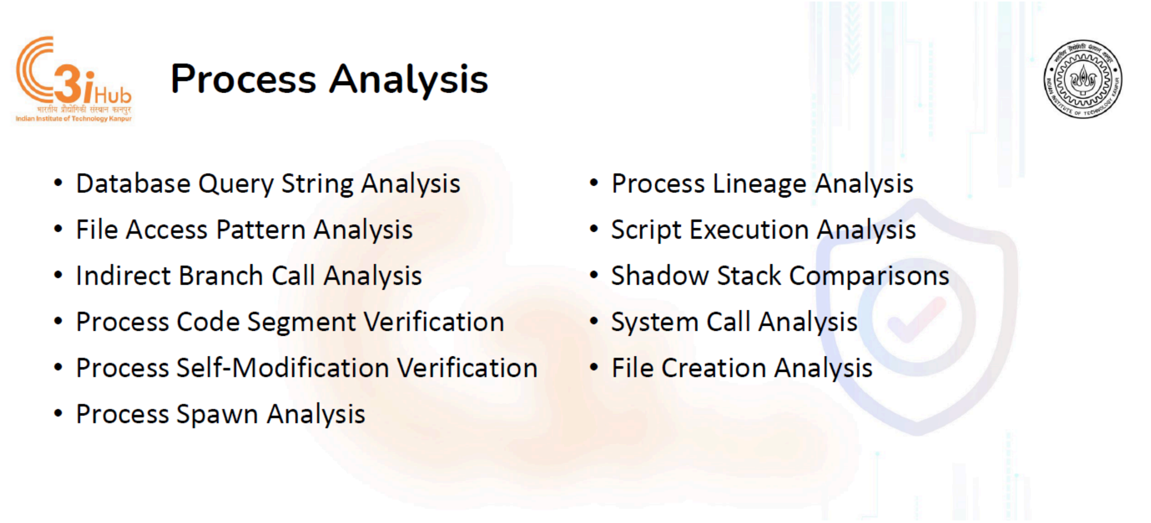
Platform Monitoring

- Firmware Behavior Analysis
- Firmware Embedded Monitoring Code
- Firmware Verification
- Peripheral Firmware Verification
- System Firmware Verification
- Operating System Monitoring
- Endpoint Health Beacon
- Input Device Analysis
- Memory Boundary Tracking
- Scheduled Job Analysis
- System Daemon Monitoring
- System File Analysis
- Service Binary Verification
- System Init Configuration Analysis
- User Session Init Config Analysis

For example, in the platform, you could have the firmware or application their behavior, like if their behavior has some anomaly in it. Whether there is, you know, whether the firmware that is running, whether its signature has been matched with the originator of that firmware, right? So you may want to make sure the digital signature matches. The same thing with the peripheral firmware, that is the plug and play, form where you may want to do operating system monitoring, like whether an operating system is behaving in anomalous ways, whether there is a huge amount of CPU usage, a huge amount of memory usage, It is thrashing this kind of stuff. So endpoint health beacon. So there may be some check of certain things, for example, the configurations.

And then accordingly, based on, let's say, PCI DSS or based on some standard requirements that for every endpoint, whether endpoint is actually losing data, et cetera. So memory boundary checking, whether an application is going over memory boundaries, so using things like canaries and so on. checking what schedule jobs are there, whether something unknown schedule job is there, system demands monitoring, system file analysis, binary verification by signature, configuration analysis of various kinds. So all these things are part of the platform monitor. So you can find more detail at the different website.

But you get the basic idea that you have to monitor not only the network traffic and try to find anomalies, what kind of files are passing through the network, whether there are fake digital signatures being passed, whether certain malicious files are being passed. whether there are certain unusual amount of data being uploaded, downloaded, all that stuff is good to monitor, but also you have to monitor what is happening on each of the endpoints. That's what the platform monitoring is all about, right? So you have to put agents in each of the endpoints and these agents will then start collecting data and then send this data in real time. to a server where this data will be analyzed and correlated and then accordingly alerts will be generated if certain activities seem suspicious. The process analysis, as I said that when you monitor the endpoints, there are processes running.



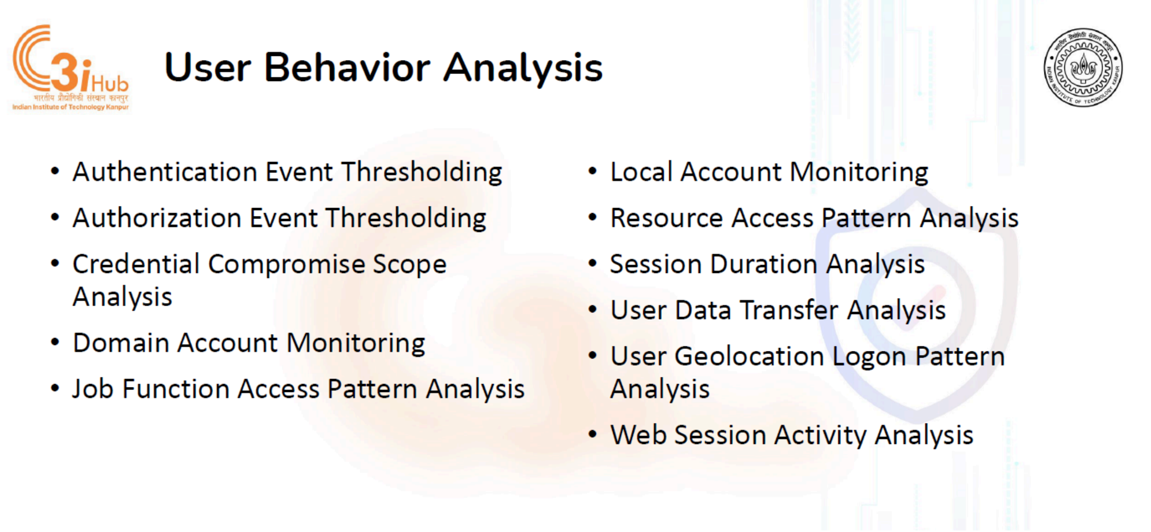
Process Analysis

- Database Query String Analysis
- File Access Pattern Analysis
- Indirect Branch Call Analysis
- Process Code Segment Verification
- Process Self-Modification Verification
- Process Spawn Analysis
- Process Lineage Analysis
- Script Execution Analysis
- Shadow Stack Comparisons
- System Call Analysis
- File Creation Analysis

And you want to know whether the running processes are actually doing certain things that are not good, that are signs of some exploits being run and so on. Like, for example, we discussed earlier process code segment verification. So you want to see whether the code segment of a process is being violated by, its integrity is being violated and certain additional code is being injected, malicious code is being injected and so on. Whether it's self-modification, if a process is modifying its own code, which is usually not a normal thing. Whether a process is pawning other sub-processes or child processes, looking at the process tree and figure out where this process is coming from.

script execution analysis, shadow stack comparison. This is some kind of a technique for knowing whether the program stack has some anomalous activities. checking what system calls are being made, what files are being created, and so on, what kind of database queries are being made. So if you really want to analyze everything that applications are doing in your system so that you can actually detect an application from being a rogue application, then you can do all this. Normally, this is not necessarily done in every system unless the system is extremely sensitive for security purposes or privacy purposes.

Because if you do all this, you are going to slow down your application to such an extent that it might not be very useful. So as I always say that security is actually is a compromise between between user convenience and security. It is also a compromise between, if you want to talk about user convenience, you may also want to talk about performance of the application that the user wants executed. And if you want very high security, you want to do all these runtime checks, then you are going to also see a pretty good amount of slowdown, which would be, again, the compromise against user convenience and requirements of the user. And the user behavior analysis, as I said, that 30% of the attacks nowadays are actually happening because of the insider attacks.



User Behavior Analysis

- Authentication Event Thresholding
- Authorization Event Thresholding
- Credential Compromise Scope Analysis
- Domain Account Monitoring
- Job Function Access Pattern Analysis
- Local Account Monitoring
- Resource Access Pattern Analysis
- Session Duration Analysis
- User Data Transfer Analysis
- User Geolocation Logon Pattern Analysis
- Web Session Activity Analysis

An insider attack is basically most dangerous because insider is already in the system. So all the firewalls and all those strong authentication of network traffic, et cetera, for intrusion, those things are usually already evaded by having an insider do things inside the system, inside the firewall. And if that user has a higher privilege, then it is even worse, right? So user behavior monitoring is very important nowadays, right? So you can get an early sign that somebody is trying to do something that is not supposed to be done by that user. For example, a user who does not necessarily have a reason to have an administrator access on a machine on a server, if he or she gains administrator access, you have to be suspicious. So similarly, if a user has no reason to go to a particular database or make queries to a particular database, if you see that that particular user account is doing that, you have to be suspicious.

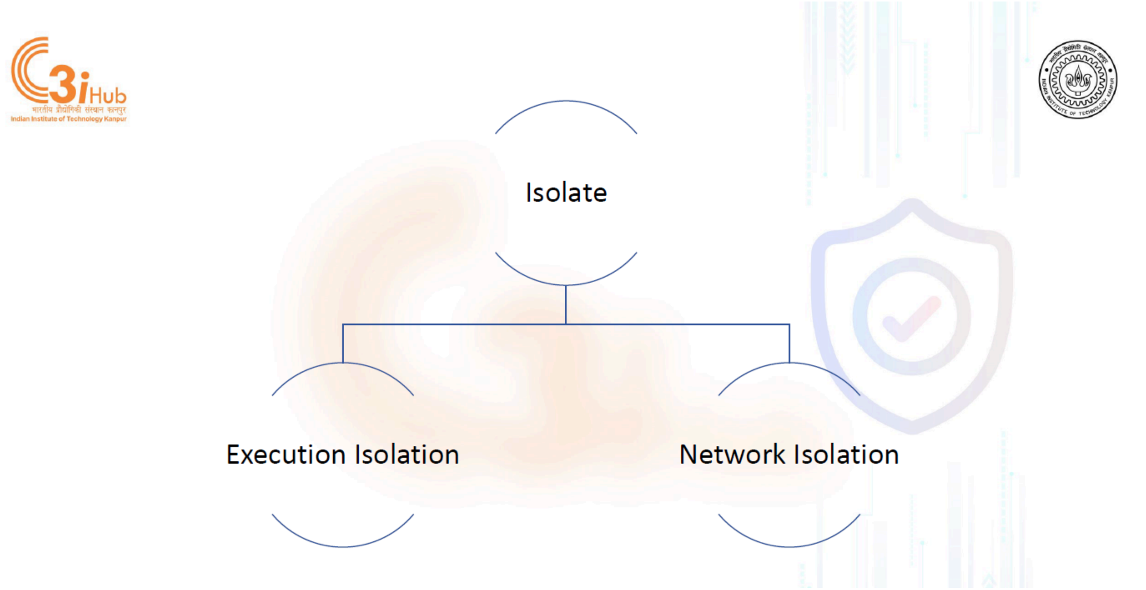
Similarly, if the user is transferring a huge amount of data to outside the organization's network, then you have a reason to be suspicious. You have a reason to be suspicious if there is a user session to a particular server or particular resource. uh is longer than usual or shorter than usual so then you have to be careful and see what is going on so all these things are not necessarily indicator of you know the guilt of a particular user he might have a perfectly good explanation of why he is accessing a resource that he normally or he or she doesn't normally require to access, there may be a good reason that why he or she is making a query to a database, which they are not usually doing. But then you are, again, checking and being cautious. So you may actually want to automatically block that acc ess and then ask the user, or you may ask the user first, whatever.

But there has to be a way for the security operation center analysts to see whether some user is behaving in a way that is outside the normal behavior of that particular user. and is not in synchronization with the role of that user or access rights of that user. So you can see whether somebody is trying to authenticate himself and then if they're trying to authenticate three, four times without success, whether they want an authorization to get into a particular resource Then see whether if the user has credential compromise, what is the scope? What is the role and privileges that the user has? Based on that, you have to figure out what is the role, what is the scope of the compromise? What can that particular credential be used for? and then accordingly take action by actually removing that credential from your Active Directory or LDAP or wherever. certain access to a particular domain, then monitoring how often they access that, access pattern analysis based on job function, resource access pattern analysis, session duration analysis, data transfer analysis, geolocation logon pattern analysis.

Some people are working from home. So if they are connecting from a particular geolocation and then suddenly, without prior information, they suddenly change their geolocation, you have two species that somebody compromised their credential and logging in from different geolocation web session activity analysis, you have to actually check these things. Now, there is a compromise between security and privacy here because as an employee of an organization, you expect certain amount of privacy about what you are doing, which websites you are browsing or what kind of sessions you are making or connecting to and what is the duration of your session, what database access you are trying to do and all that. But in today's world, this is considered security in this case trumps the need of privacy of the user because when you are in the organization's network, you are using organization's platforms and resources, you are bound to be monitored nowadays. for every keystroke you are making and everything you are doing. So therefore, it is also advice nowadays not to do your private activities, like for example, your private banking activities, private browsing, or if you're doing any part-time job or if you're doing a part-time studies and so on.

keep that totally outside the realm of the organizational network and do not use the organizational network to connect to personal activities because all the stuff that you're doing inside the organizational network using an organizational platform will be subject to monitoring for user behavior analysis. So this is an important part of today's security. The next different tactic is actually about isolating. Isolating is important because once you detect you have a system being compromised or has been compromised, you do not want that lateral movement or privilege escalation and then lateral movement or discovery, reconnaissance, etc.

to the rest of the system and so on. So you want to create isolation. There are two types of isolation. One is that applications may be isolated from each other to begin with. So that you have, if one application gets compromised, it is isolated and it cannot really go and compromise the rest of the system or rest of the network or other processes and so on.



Isolate

Execution Isolation

Network Isolation

So that's the solution isolation. And then there is network isolation where we actually do network segmentation. So we segment the network. into micro segments such that between two network segments, there are firewall and there are access controls. So you cannot, if one part of the network gets compromised, it doesn't automatically mean that the other part of the network gets compromised. And important resources such as servers, et cetera, will be in a very highly protected segment where only a very few have access.

And that access control has to be properly enforced so that even if other segments get compromised, that critical segment does not get compromised. So execution isolation has many different aspects. So for example, You might have, as an organization, for all the machines that your employees are using, you may actually enforce something called executable allow listing. That means in that platform, only certain applications will be allowed to run.

No other application that is unknown can be run on that platform. Now this is okay for users who are not doing coding and developing. For developers, software developers, this will not work because they have to always compile new executables and they also have to execute the executables on the platform. So this will be different. So the application developers they will be in a different segment network segment than regular network segment where most other users will be there. So that if developers somehow due to non-usage of this whitelisting of applications, if they get compromised, their network segment is isolated.

So they cannot easily affect the rest of the network. Deny listing is blacklisting basically. So blacklisting means you actually say, okay, these executables are not allowed to execute on this platform. That's harder to do because people can always change the way the particular executable is recognized by the system. So if the system says, you are not allowed to execute on my platform, I just changed the cover in such a way so that the system allows me to execute. So blacklisting is not a, or deny listing is not an effective way of doing protection.

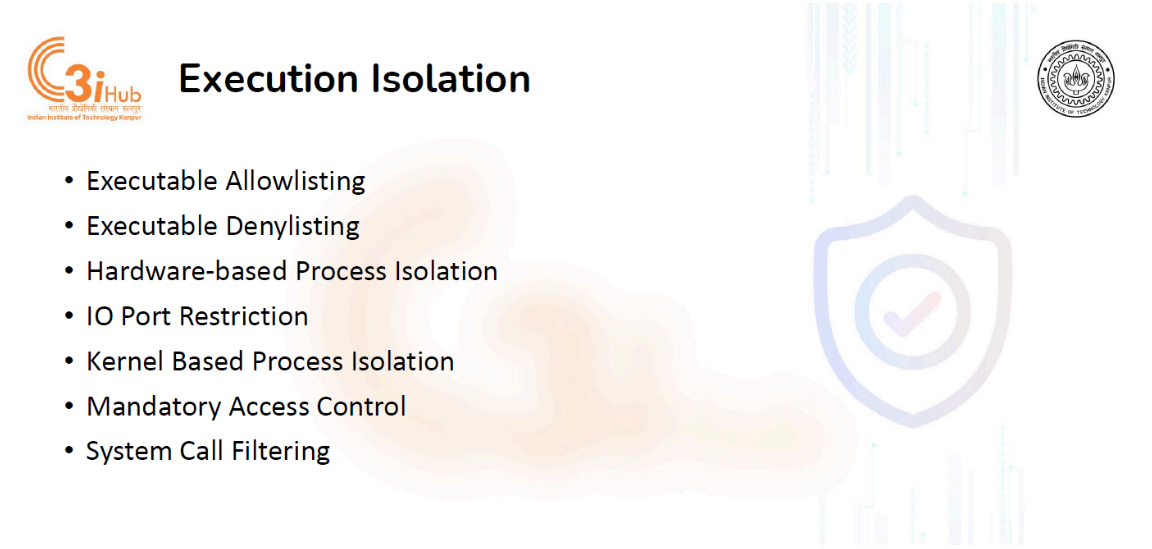
Allow listing is better because there you are actually only allowing the ones that are in the list. Hardware-based process isolation You can do hardware emulator, hardware platforms. Or you can have different applications running different hardware platforms. different servers, but that's very expensive to do nowadays. And nowadays the servers can run both an email server and a web server and an FTP server and database server on the same server because of the amount of processors in there, the number of amount of memory in there, et cetera, can support such a multiple different large, very large scale applications on the same server.

So hardware based process isolation is probably not a very successful one here. You can also do restrictions on what IOPorts are allowed to be used. Or you can also do process isolation. So we already have Linux and Microsoft.

They all have basic process isolation. So one process is isolated from the other.

They have different virtual memory space. They have all the different... uh you know memory uh they don't uh you know access each other's memory and so on so there is already a process isolation uh you know in from the definition of the process but you can do more by uh you know various kinds of uh process isolation within processes for example by by isolating threads and so on mandatory access control is the ability to control access of resources where you are actually, even as a user, you are not allowed to violate that. So for example, the opposite of mandatory access control is the discretionary access control. In a discretionary access control, when the user gets access, let's say, to a file, even though the file may be private or confidential, the user can easily copy the file and give it to somebody who doesn't necessarily have the rights to access that file. Mandatory access control is such that that kind of passing of access is not possible by the system itself. So you can think of Android security, each Android app is, is actually running as a separate user ID, right.

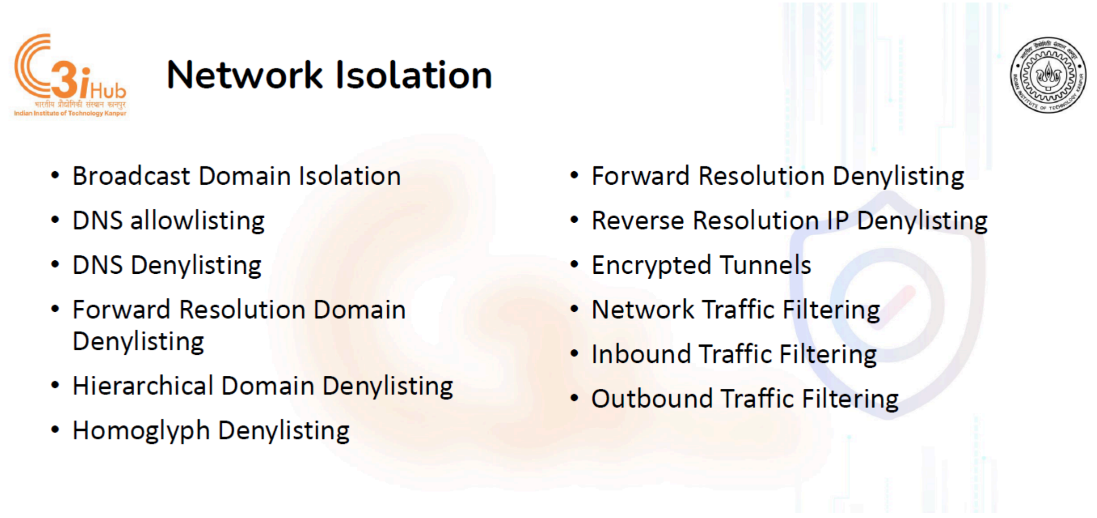
And this, each of this app, you know, can access another data from another app or any kind of other resources such as the file system or the camera or the audio, et cetera, under the control of the Android operating system, systems mandatory access control. So it has to be, it is a system which allows or disallows the access. A user cannot pass the, a particular program cannot pass the access to another application. System call filtering is another way of doing execution isolation. If there are some system calls which look suspicious, you can do a system call interposition and then stop that particular system call being executed by an application.



Execution Isolation

- Executable Allowlisting
- Executable Denylisting
- Hardware-based Process Isolation
- IO Port Restriction
- Kernel Based Process Isolation
- Mandatory Access Control
- System Call Filtering

So these are different ways of checking that the executable is denied unrestricted access to resources, unrestricted access to data without proper provisioning for those. So either it is given certain access rights or whether it is mediated through a system called interposition or it is mediated through mandatory access control, only then such accesses are possible. So this is what the execution isolation is all about. And network isolation is again, Similarly, you can have domain isolation.

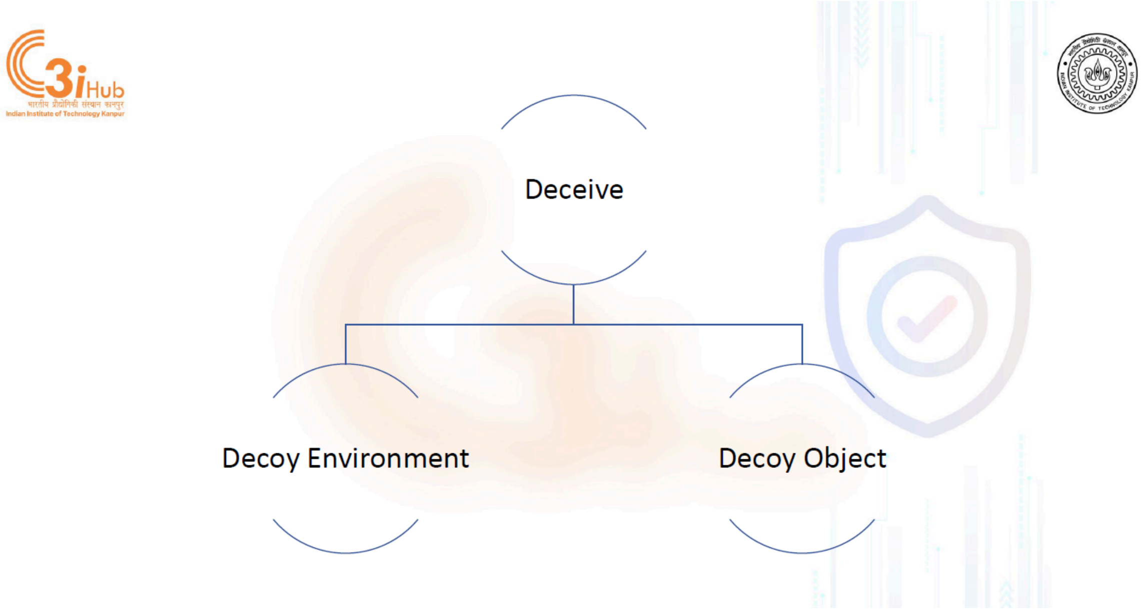


Network Isolation

- Broadcast Domain Isolation
- DNS allowlisting
- DNS Denylisting
- Forward Resolution Domain Denylisting
- Hierarchical Domain Denylisting
- Homoglyph Denylisting
- Forward Resolution Denylisting
- Reverse Resolution IP Denylisting
- Encrypted Tunnels
- Network Traffic Filtering
- Inbound Traffic Filtering
- Outbound Traffic Filtering

For example, you have the VLANs. They have domain isolation, broadcast domain isolation. You have certain DNSs are allowed and you can filter the DNSs. You can deny certain DNSs. Again, allow listing is better than deny listing for obvious reasons.

So similarly, forward resolution for domains may be denied. Forward reverse resolution IP may be denied for certain things. Similarly, domain deny listing, inbound traffic filtering, outbound traffic filtering. These are all parts of the network isolation. So of course, you can also do network isolation through domain isolation and through by creating segmentation of the network. Okay, so in the next lecture, we will talk about deceive and decoy environment and decoy objects and also we will talk about eviction.



Deceive

Decoy Environment

Decoy Object

But I would suggest strongly that you go to the def3nd website and actually look at the various MITRE website. and then you actually study these things, right? So like for example, here, execution isolation, here you have like IO port restriction, right? So it tells you how it works and what digital artifacts are associated with it and so on. So you will get a better idea about each of these, although a basic idea of all this tactics like hardening, then what is involved in platform hardening and what is involved in application hardening or what is involved in credential hardening or in case of detect what is involved in platform monitoring versus network monitoring, et cetera. In case of isolation, what does it mean to isolate execution versus isolate network and so on.

All these basic ideas are obviously most important. But knowing these things, at least going through each of these will actually give you a better idea in case you are going to take security as your profession, you need to know all these things. Okay, so we'll talk more about the rest of the different metrics in the next class and then we'll move on to the next topic. Thank you.