Practical Cyber Security for Cyber Security Practitioners

Prof. Sandeep Kumar Shukla

Department of Computer Science and Engineering

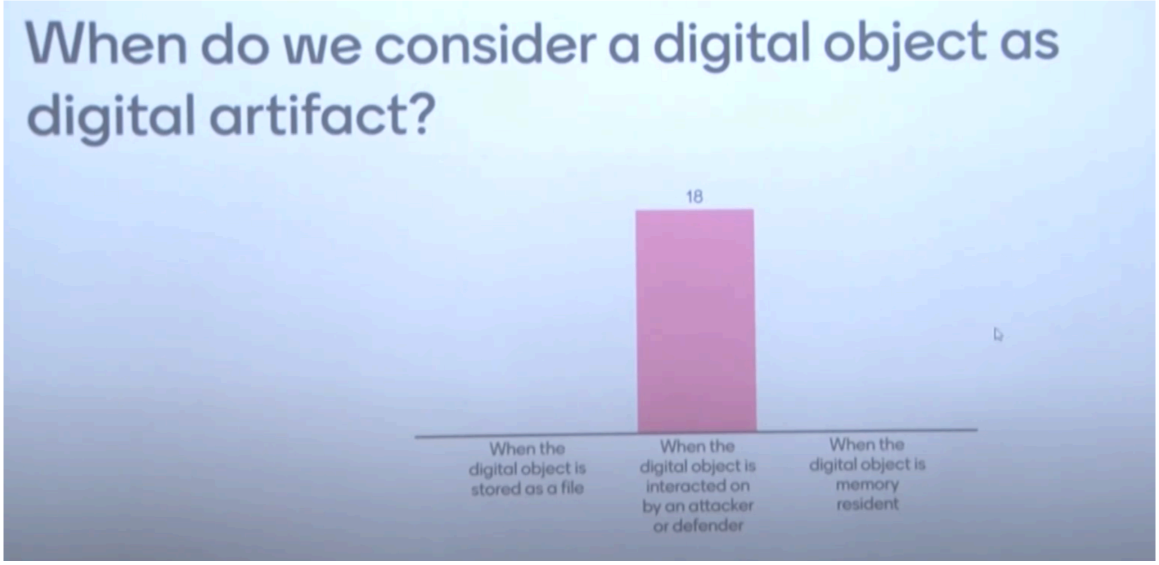
Indian Institute of Technology, Kanpur

Lecture 16

Deep dive into MITRE DEF3ND framework-I

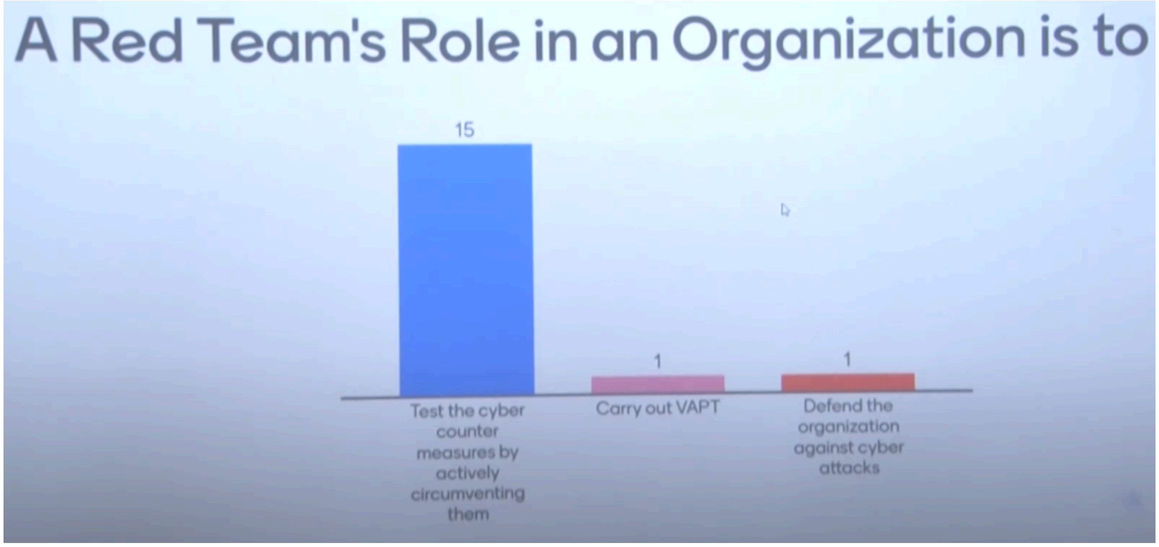
We were talking about MITRE DEFEND, and we discussed about digital objects, digital artifacts, and how digital objects and artifacts kind of connect the offensive model of MITRE, which is the MITRE ATT&CK model, and the defensive model of MITRE, which is the MITRE DEFEND model. So as usual, before we do that, let's look at how we are doing in our understanding of things. So get your phone out, menti.com. The code is 15286361.

So when we consider a digital object as digital artifact, okay yeah so we discussed this last time the digital object anything at URL and IP address a file all of those things can be digital objects but we consider it as digital artifact if that particular object is either being created during an attack, or used by an attacker, or used as artifact for detection by the defenders, or as forensic, for forensic reasons, so then it's a digital artifact. So almost everybody, actually everybody got that correct. So let's look at, So what is the role of a Red team in an organization? Is it to carry out VAPT? Is it to test the countermeasures or defend the organization against cyber attacks? Okay, so Red Team is actually meant to test the countermeasures, the controls that you have put in. So VAPT is vulnerability assessment and penetration testing.



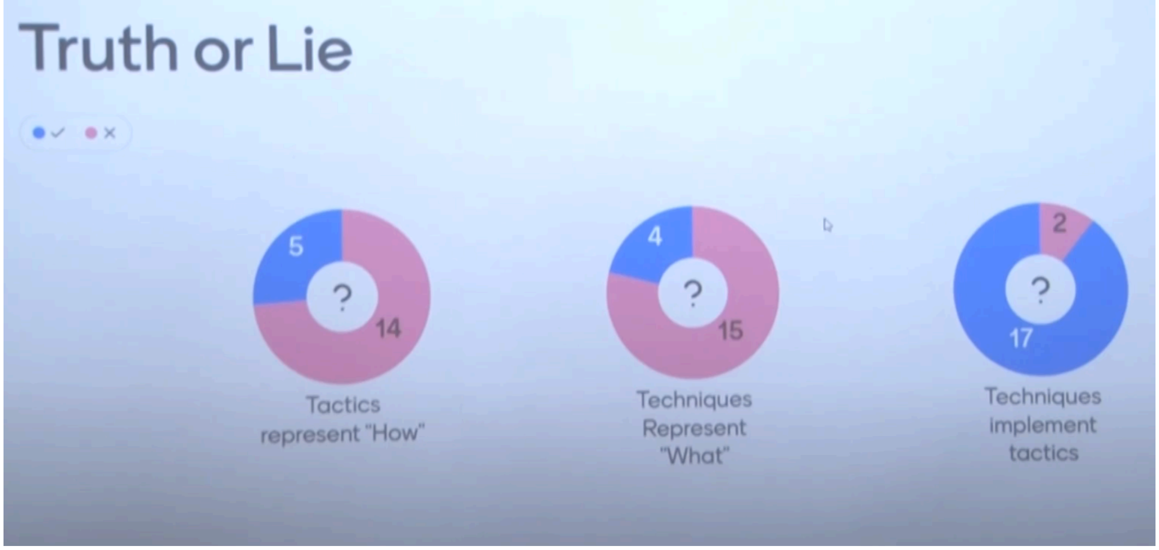
When do we consider a digital object as digital artifact?

In vulnerability assessment, we actually use some kind of a tool to scan the network, to scan the applications, to scan the existing configurations and we figure out if some known vulnerability CVE is actually there in the system. So known vulnerability is found through vulnerability assessment usually done using tools like Nessus But penetration testing is a more active process of trying to bridge the various applications, various network and operating system, firmware, etc. but it does not think in terms of the countermeasures.



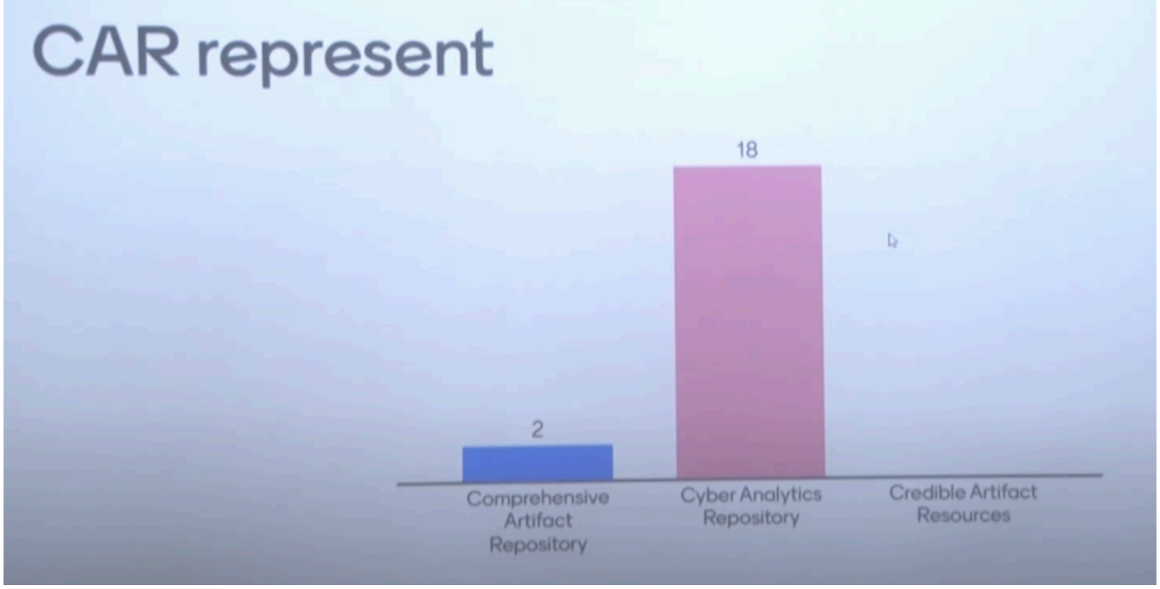
A Red Team's Role in an Organization is to

So the red team's job is to actually work against the blue team and they're aware of the countermeasures and they try to evade the countermeasures to show to the blue team that their countermeasures are not effective enough. So the red team is different from the VAPT team.



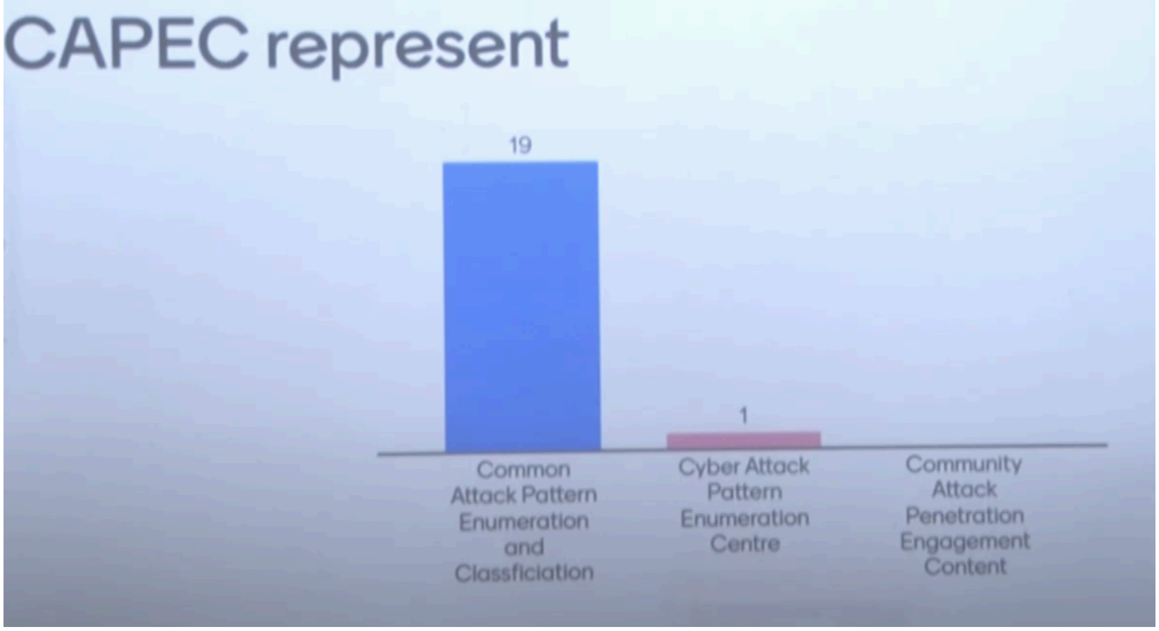
Truth or Lie

Now we have three true or false questions. So tactics, techniques and one statement about technique, implementing tactics. okay so this was also discussed in the last class that tactics are basically about WHAT and techniques are HOW and that is why techniques implement tactics right because so if you are saying that techniques implement tactics but at the same time you are saying that techniques is what and tactics is how how is that possible right what say how is how implements how is the method of implementation for something which is what right so techniques and tactics are you know techniques are what's a house and and tactics are what's now we discussed in the above the CAR and we showed you car.mitre.org website as well so what is CAR Yeah so it's a cyber analytics repository and we showed you how to get there and you can see what you know different ways in which you can detect various techniques.



CAR represent

Cyber analytics repository is a set of tools, scripts, various rules that you can implement inside your network intrusion detection system or host intrusion detection system to actually detect various techniques, whether somebody is trying to use those techniques in your system, like somebody is trying to do something in the, to persist something and to create execution in the system and so on and so forth, right? or trying to do literal movement, or trying to do command and control. What are the tools or scripts, et cetera, available? So that's Cyber Analytics Repository, or CAR. And you can find those in car.mitre.


CAPEC represent

org. So we also showed you the CAPEC website. So CAPEC is what? Yeah, so common attack pattern enumeration and classification, right. So that's the capec.mitre.

org. You have, that's where the various common attack patterns you can find. You know, it has been cataloged. So if we go there, like, this is the CAPEC website. And here you can find various attack patterns. Like for example, they have been numbered, so mechanisms of attack.

So if you now look at this, like engage in deceptive interaction, abuse existing functionality, manipulate data structures, manipulate system resources. Now if you go into one of these, software integrity attacks, then if you go to this malicious software update, So this is malicious automated software update via redirection or routing of SIM cards, malicious automated software update via spoofing. So these are the various common patterns, attack patterns and each of them has a description. You have mapping friendly versions and then you have the corresponding weakness, common weakness that is being used and so on and so forth. So you can actually get this whole catalog, a knowledge base of various common attack pattern enumeration.

So you can actually refer to this as and when needed when you are doing some defensive mechanism or trying to analyze a particular attack. Okay, so coming back to the different framework. So we actually discussed that digital artifacts basically are the connection between attack and defense models. So for example, suppose there is an exploitation for client execution or exploitation for privilege escalation. So you have either exploitation of a remote service or process injection.

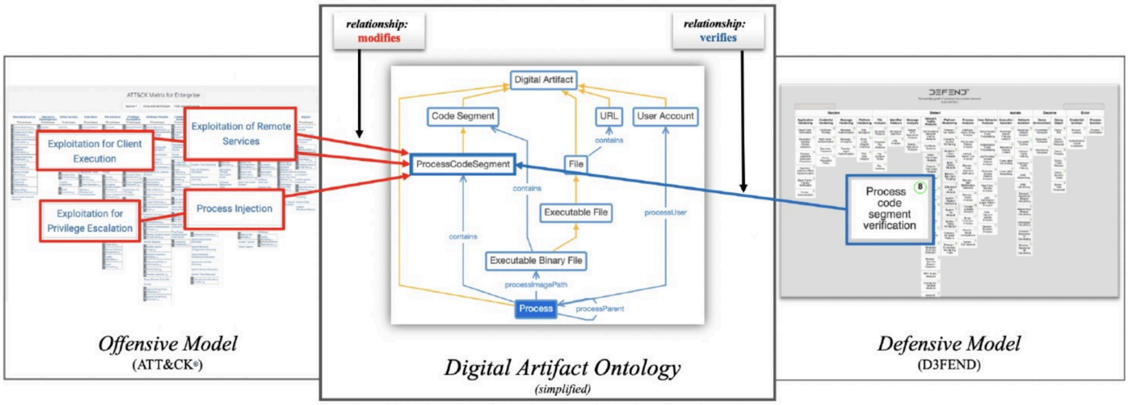


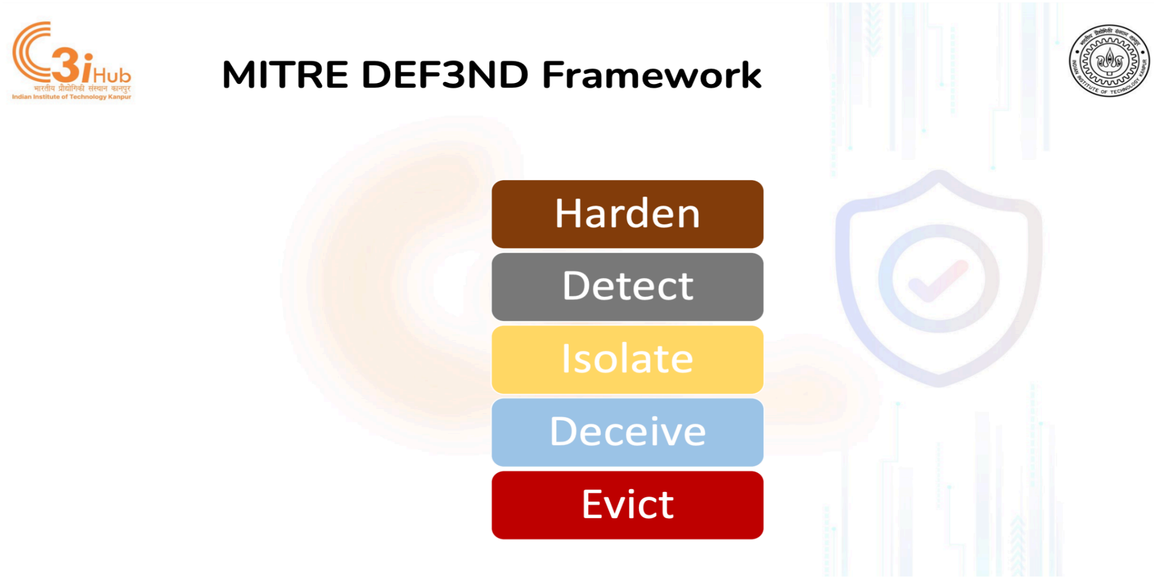
Fig. 8. Mapping via Inference Through the Digital Artifact Ontology

Now the artifact in which you will find some indicator that you have a process injection is somebody has injected some code inside the process. running process so that it does not have to run as a separate the mischief or the malicious code is not running as a separate process but somebody injected code into a running process or through a DLL injection. or there is a remote service that is being exploited by let us say exploiting a buffer overflow and then or some kind of you know code execution. the indicator will be there inside a code segment of a process right. So you know that every process has virtual memory associated with the process and the virtual memory is divided into these various segments and the code segment is where the code is loaded in the memory right.

So process code segment, so now if I know that that is the artifact, that is the digital artifact which I should inspect in order to know that something like that is happening, then inside the DEFEND, as we go into DEFEND, we haven't gone into the DEFEND yet, but in DEFEND there would be a particular technique called process code segment verification. So that basically looks at the processes code segment and the integrity of the code segment whether the code segment is what it is supposed to be versus what the code segment is currently being loaded. if there is a variation you know that something one of these might have happened right so this is the way digital artifacts connect the offensive model with the defensive model. The DEFEND model will obviously have certain parts of the defensive model that will not depend on an artifact being generated from the attack model because for example hardening right. Hardening is when you actually proactively do things to change the processes or change the configuration of software operating systems or you change the various methods like for example you change into two factor authentication instead of single factor authentication.

So these are hardening processes right. So the hardening process happens before you know, you look for attacks. After you have hardened a system and you have deployed the system, you are hoping that since your system is hardened, a lot of the attacks can be stopped. Like for example, if you have two factor authentication, the possibility of somebody using an user credentials to access your system, you know, unauthorized user using an authorized user's credential, is lowered the probability is lowered that that would happen so hardening reduces the probability of a particular attack but a hardening does not eliminate the possibility of an attack therefore we have to also look for detecting whether an attack is happening and if detection is a very important part And as you will see that in all cybersecurity frameworks, like if you look at NIST framework, in NIST framework, we have various functions, identify, protect, detect, right? And then you have to respond and recover, right? So there are five different functions in the first version of the NIST framework.

NIST 2.0 came out like the day before yesterday officially came out so it has 6 functions instead of 5 but the point I am trying to make is that no matter how much hardening you do no matter how many different controls you put in. you have to always assume that there will be clever attackers who will actually circumvent those possibilities and therefore there will be attacks in your system and therefore we have to be able to detect when attack is happening and then accordingly we have to either deceive the attacker or we have to isolate the part of the system which has been attacked or compromised already or we have to actually evict the attacker from the system. So those are the different tactics, top level tactics of the MITRE DEFEND framework. So now we go into the MITRE DEFEND framework. So the MITRE DEFEND framework has these five top level tactics.



MITRE DEF3ND Framework

Harden

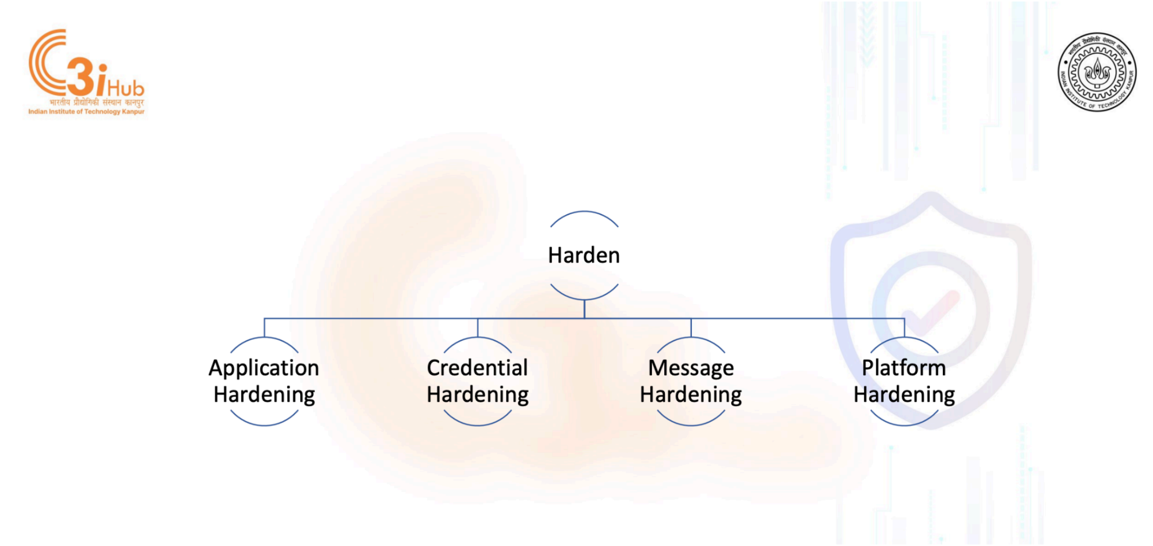
Detect

Isolate

Deceive

Evict

So harden, detect, isolate, deceive and evict. So in case of attack we had 14, 12 plus 2, 14 different tactics, here we have 5 different top level tactics easier to remember but this if you are ever going to be a cyber security personnel inside an organization These are the things that you would certainly do. So these are things from which you will start. So you have to think about what I am supposed to do and these are things that will come to your mind. So now we'll go into this top level tactics to this, you know, they are now in a DEFEND framework.



Harden

Application Hardening

Credential Hardening

Message Hardening

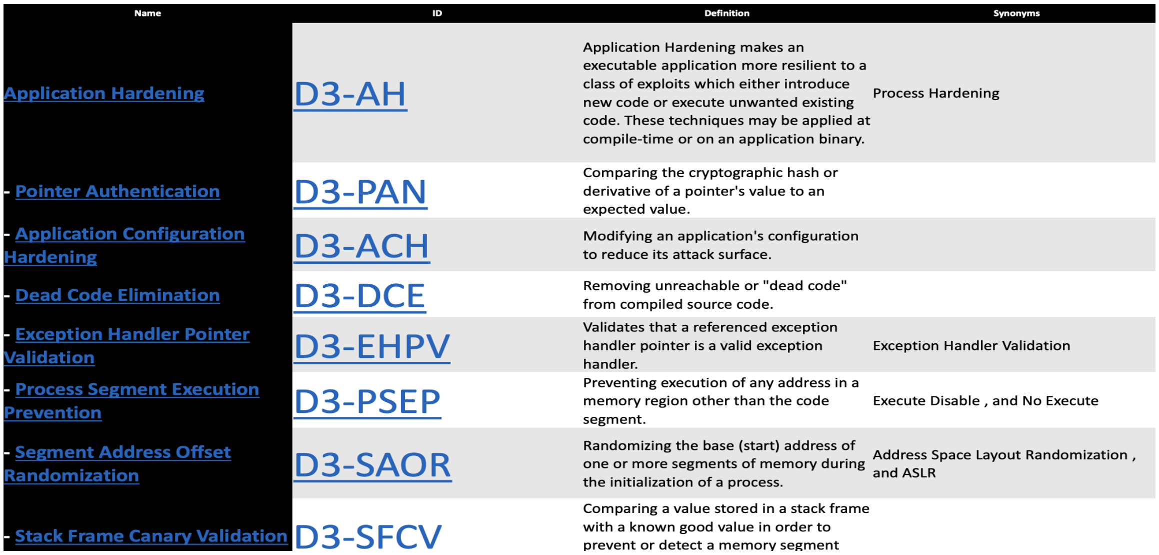
Platform Hardening

They are, you know, have these different sub tactics which are application hardening. credential hardening, message hardening, and platform hardening. So when we say we are hardening our system, we are basically saying that we are trying to make the configurations of our systems in such a way so that the possibility of somebody breaching the system is reduced. And then they say, OK, so if I want to harden, what are the things that I can harden? So application hardening. how to make your application such that even if the application has certain vulnerabilities such as buffer overflow vulnerability we can still you know obviously we can still stop an attacker from exploiting it right credential hardening is when we are trying to make sure that the credential of users legitimate users cannot be misused by an unauthorized user or unauthorized personnel because we probably will require two-factor authentication we may need a digital certificate pinned to a particular user's device.

So this kind of thing will allow us to stop unauthorized users from just stealing somebody's password and using it to enter our system. Message hardening. So messages in a system may have many meanings. For example, when we have an API call, you can call that as a message. We are basically sending a message from the client to a service.

So an API call could be a message, it could be an HTTP request, could be considered a message or any kind of two different processes. They might be using a message passing protocol, application level message passing protocol. So, those kinds of things are or you can have a socket connection. So, you can have messaging between two processes which are establishing a TCP IP TCP socket or UDP socket. So, these messages can be intercepted, they can be replayed, they could be actually modified and then sent to the service you know as intercepted in the middle of its journey.

And that could create various cyber attack possibilities. But if we do message authentication, if we do message encryption, then it becomes harder for a man in the middle attack. Somebody intercepting the message and then changing some of the parameters or changing the message. It becomes difficult if you add authentication and encryption on the messages, right? So message hardening is one such thing. Platform hardening is very important because many times we actually make mistakes in configuring things.



Name	ID	Definition	Synonyms
Application Hardening	D3-AH	Application Hardening makes an executable application more resilient to a class of exploits which either introduce new code or execute unwanted existing code. These techniques may be applied at compile-time or on an application binary.	Process Hardening
- Pointer Authentication	D3-PAN	Comparing the cryptographic hash or derivative of a pointer's value to an expected value.	
- Application Configuration Hardening	D3-ACH	Modifying an application's configuration to reduce its attack surface.	
- Dead Code Elimination	D3-DCE	Removing unreachable or "dead code" from compiled source code.	
- Exception Handler Pointer Validation	D3-EHPV	Validates that a referenced exception handler pointer is a valid exception handler.	Exception Handler Validation
- Process Segment Execution Prevention	D3-PSEP	Preventing execution of any address in a memory region other than the code segment.	Execute Disable , and No Execute
- Segment Address Offset Randomization	D3-SAOR	Randomizing the base (start) address of one or more segments of memory during the initialization of a process.	Address Space Layout Randomization , and ASLR
- Stack Frame Canary Validation	D3-SFCV	Comparing a value stored in a stack frame with a known good value in order to prevent or detect a memory segment	

We actually keep many ports open. We keep the operating system unpatched. We keep the applications unpatched. We keep the firmware unpatched. We often do not configure things like, for example, some encryption for your important files.

We may not configure them. Various things we may make mistakes about which would eventually cost us an attack. So, therefore platform hardening is also an important part of hardening. Now if you go into the ATT&CK website DEFEND website defend.mitre.org you can then go into the further details of all of these things for example here I am just giving an example of application hardening right.

So application hardening may have multiple different techniques by which you harden an application. So what does that mean? For example, pointer authentication. So those of you who have taken CS628 would remember how a buffer overflow attack happens, right? So when a buffer overflow attack happens, So you have a program which has a stack program stack where there is a you know a buffer a array variable right, so it is a character array, so that character array if you write on that character array are very large sequence of characters it will overflow the space allocated for it and it will start writing over other elements on the stack and then go beyond the stack to write on the memory locations beyond that stack segment of that particular function. and in this process if you do it right then you will eventually write the return pointer right the return. So when you make a function call on the stack the return pointer says in the main C main function you are calling a function F.

when it goes to the function f it creates a stack segment right and then it creates a you know it pushes a element on the program stack where all its variables are stored but when it finishes this is the function f finishes it returns it has to know where to come back so it has to save the return pointer return place where it has to return so if during buffer overflow if you rewrite that address then when you come when you finish this function you will go to a different place not the place where you are supposed to go. So this is what buffer overflow does now there are various ways to stop this. For example there is something called DEP that is the data execution prevention so if you turn on data execution prevention then data you cannot have execution from that point. So, but if you actually authenticate the pointer that if you store the pointer in another place to compare so when you finish the function call you get the return pointer but you do not trust it so you actually check it against a stored value of the function pointer then you will get the you are basically authenticating that pointer or that pointer may be signed by a particular you know can be hashed and signed right. In that case you can check whether the stored value of the pointer matches the pointer you are seeing at the end of the function call. So that is what pointer authentication is.

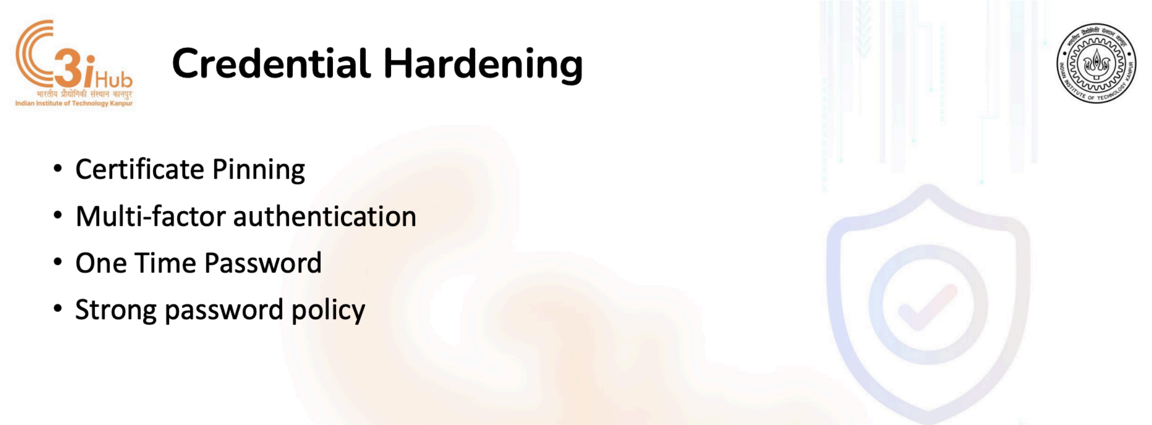
So configuration hardened. So many complex applications have many configurations that you can do. So they are usually a configuration file. So if you have ever you know compiled a very large program. usually it also generates a configuration file which you have to change right in which you have to write specific configuration information. It could be a JSON format. It could be a specific format so the program as it starts up reads the configuration file to know what configuration it should set up for the particular run of the program. Now, usually for testing and for novice users, there are certain configurations which are left default, right? Now, if you run your program, run your application with default configuration, maybe there are certain vulnerabilities.

So you have to change the configuration to change that possibility, right? So that is what application configuration hardening is. Dead code elimination, so many programs are very large especially very large programs they might have a lot of files and a lot of code segments which are in the system in the source code directory which are no longer in use but in your makefile you still compile them and link them right. an attacker may find that and say that okay so this part of the code even though it is not useful for them I can use it I can redirect I using a buffer overflow or something I can redirect to that point of the program and then it can execute from there and it will do something that is useful for me. So having dead code inside your code inside your application compiled into your application is a bad idea because it can be misused against you. So, eliminating any kind of dead code now for a very large you know 1 million line 2 million line application the source code application finding the dead code is not non-trivial right because those of you who actually have taken theory of computation should appreciate that finding dead code is actually a undecidable problem right.

Because it is a reachability problem whether the code ever reaches that point right whether the application ever reaches that point. So dead code elimination is non-trivial but the application developers might be able to find a better way to find the dead code and remove it from the compiled code. exception handler pointer validation is an exception so one of the ways attackers may use buffer overflow against you especially if you have turned on DEP the data execution prevention is to use the exception handlers right to make you jump to the exception handler and then write a malicious code in the exception code. So that exception handler pointer validation. So you actually have to know what your exception handler pointers are from the beginning so that somebody who has inserted extra code as an exception pointer should be detected.

So these are all runtime methods process segment execution prevention that is what the DEP is all about and then segment address offset randomization. So this is ASLR address randomization so that the attacker cannot use the buffer overflow easily by calculating where the return pointer address is and things like that. Stack frame canary validation, so many times we put a canary so that if there is a buffer overflow it will also rewrite the canary and then by checking the canary I can detect whether there is a buffer overflow. So buffer overflow, so all these things that we just talked about is about assumption that application code that is used in today's world cannot be guaranteed to be free of mistakes like buffer overflow. So buffer overflow mistakes are very very common and therefore as a cyber security defender, I want to make sure my runtime validation against you know possible exploitation of buffer overflow are all configured.

So I want to do address space layout randomization or ASLR. I want to have DEP or data execution prevention. I want to be able to have canary. These are all runtime system checks against buffer overflow exploitation.



Credential Hardening

- Certificate Pinning
- Multi-factor authentication
- One Time Password
- Strong password policy

So I want to have all these. So this is what application hardening is all about, right. Credential hardening is another hardening. Now when we are giving you these examples this is not the exhaustive list of possible hardening rights. So the hardening I just showed is possible. These are all possible hardening but tomorrow somebody might add three more application hardening techniques into the knowledge base. So that is why it is a running program. It is a living knowledge base at MITRE Similarly, credential hardening we are talking about 4 different techniques.

So certificate pinning right. So you may want to actually so one of the suppose you want to go to your bank right and it is an online SBI bank. and your local DNS resolver right so so when you type in SBI online SBI.com what happens your browser if it already has recently been to SBI then it will actually have the IP address corresponding IP address so it will send the HTTP request to the right IP address but if it does not then it will go to the DNS resolver right now local DNS resolver might be, so we actually, when you configure or if you go and look in your computer and do an ipconfig or ifconfig, you see a DNS address, right? So I think it's 131, 172 something, something, 131 is the DNS address. So that's our local DNS resolver, right? So the local DNS resolver will actually show you the IP address for SBI. Right now suppose soumitri here decides to take your money right you do not manage the DNS do you yeah so somebody in the DNS in the CC decides that they will change the DNS translation of SBI to their malicious server that they have put up.

So every time somebody new types in SBI, it goes to that address and he makes a nice SBI webpage which looks like SBI. You will not see the difference. Now the only thing that will save you is a SBI certificate. Right whenever we go to SBI it sends us a digital certificate to our browser saying that proof as a proof that it is SBI because this certificate is signed by some certificate authority which tells us and which our browser can check against the known certification authority signatures that this is SBI right. Now suppose Soumitri also manages to get a certificate from another organization, a rogue organization, and it actually happened.

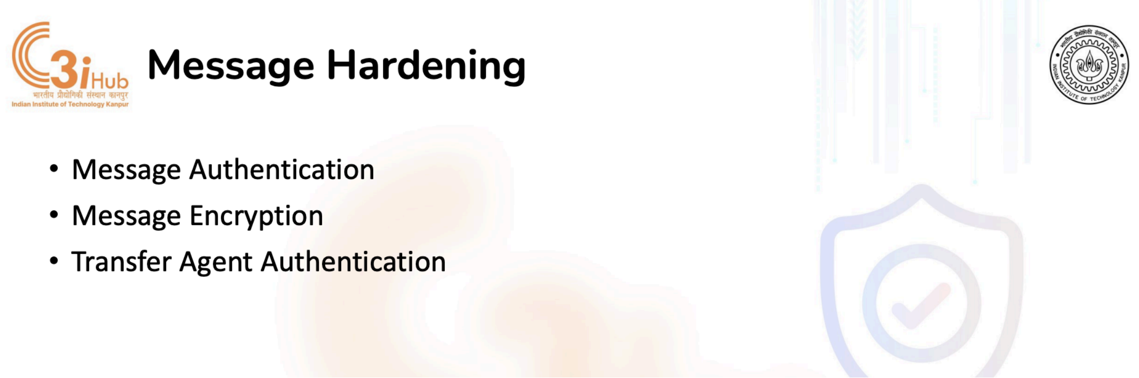
CDAC, which is a government organization, gave digital certificates for both Yahoo and Google domains in 2012 to some random people, right? So then for a while, The google.com and yahoo.com were translated to some random IP addresses within India, right? So let's say, and this has happened. So certificate authority people may have gone rogue and they give a certificate of SBI.

So now Soumitri can prove to your browser that he is SBI. This will go on for a while. It may not go on for a long time because eventually SBI will get complaints and they will do something. But for a while, it can go on. And then at that time, you type in your username, password, and all that stuff.

Now, thankfully, SBI also requires OTP. So he won't be able to get your money because the bank, he will get your username and password. He can actually do social engineering to get your OTP also, but then Somitri has to do other work also at CC. So he doesn't have that much time, right? But this can happen, right? I mean, I'm not saying Somitri will do it, but somebody can do it not necessarily for the bank, but for some other place where there is no OTP. right so there is no second factor authentication then this can be easily done. So what you do is you pin that your browser can pin the certificate of SBI so that no other certificate of SBI no other thing that claims to be certificate of SBI will be recognized by your browser right.

So that is certificate pinning right so you can actually save yourself by doing certificate pinning. So another thing that we do in our hub is that for important applications where we want only a certain number of people to even see that such an application exists, such a web application exists. We actually ask them to pin a certificate in their browser of their own. So unless they present that certificate to our application, our application does not even show up, right.

So that kind of certification pinning. Multi-factor authentication, all of you understand why it is important. OTP one time password as I said that this is very important strong password policy of course is something that you can enforce for example when you when somebody is changing their password If they do not do certain combination like one capital, at least one small, one capital, at least one letter, at least one number, at least one special character, total length at least 14, 12, whatever, that is strong password policy, you can enforce it, right? So that way you can do credential hardening. So these are some of the examples of techniques for credential hardening. So message hardening as I said that we always use messaging in various forms like API calls, HTTP requests.



Message Hardening

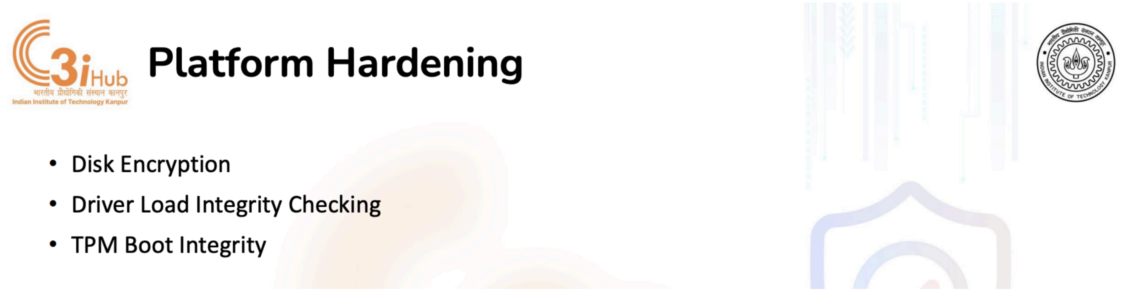
- Message Authentication
- Message Encryption
- Transfer Agent Authentication

So we can do socket level messaging. All this has to be, can be intercepted in the middle like HTTP. is the one of the biggest problems because HTTP is not secured, it is not encrypted, it is not authenticated right. So when I am doing an HTTP communication with a web application all the thing I am doing is interceptable and interceptable and changeable right because there is no authentication right. So, one of the other problems is that many times you will find people using self-signed certificates. They think that if I have to do HTTPS, I can self-sign a certificate and use HTTPS.

But the problem with self-signed certificates is that suppose I run an application which basically sends a self-signed certificate to your browser. So what does your browser do? 10 years ago, browsers wouldn't care. Certificate is certificate, right? Now, Chrome, Firefox, all of them will complain. They will say, this is not safe. Do you still want to go to that website? Because we cannot guarantee safety.

Because they cannot recognize that certificate is signatory. Because signature is done by you, like by IIT, somebody inside IIT or something. So why is the browser complaining? Because the browser is saying to you that I do not recognize it. So now somebody in the middle, so my application and your browser, somebody intercept the messaging between us and present another certificate to you as me. Your browser doesn't know that from the certificate that I was giving, right? Because both certificates are unrecognizable to the browser. So browsers will say that if you are going to accept a certificate from Sandeep, why not a certificate that is now presented as Sandeep's domain as somebody else? So browsers will accept it, right? and then that person is now having a will get your username password whatever you type in you will get it right so therefore certificates has to be recognizable by the browser right so self-signed certificates usually that is why is not acceptable by current browsers so like chrome etc so messaging has to be authenticated and authentication means digital signature, right? How do you know that this is Sandeep's digital signature? I have to also give you my certificate which is authenticated by a known certificate authority so that the browser can check that, okay, whatever Sandeep is presenting as a digital certificate is actually a valid digital certificate.

His public key is the valid public key of Sandeep, right? Otherwise, anybody can claim that this is Sandeep's public key. And message encryption, so that nobody can intercept in the middle and change the message. Transfer agent authentication. So when we have, for example, in the mail system, we have these mail transfer agents, right? So these mail transfer agents have to be authenticated. Like, is it this IP address, from which this domain can send email, right? So this is the SFP check right.



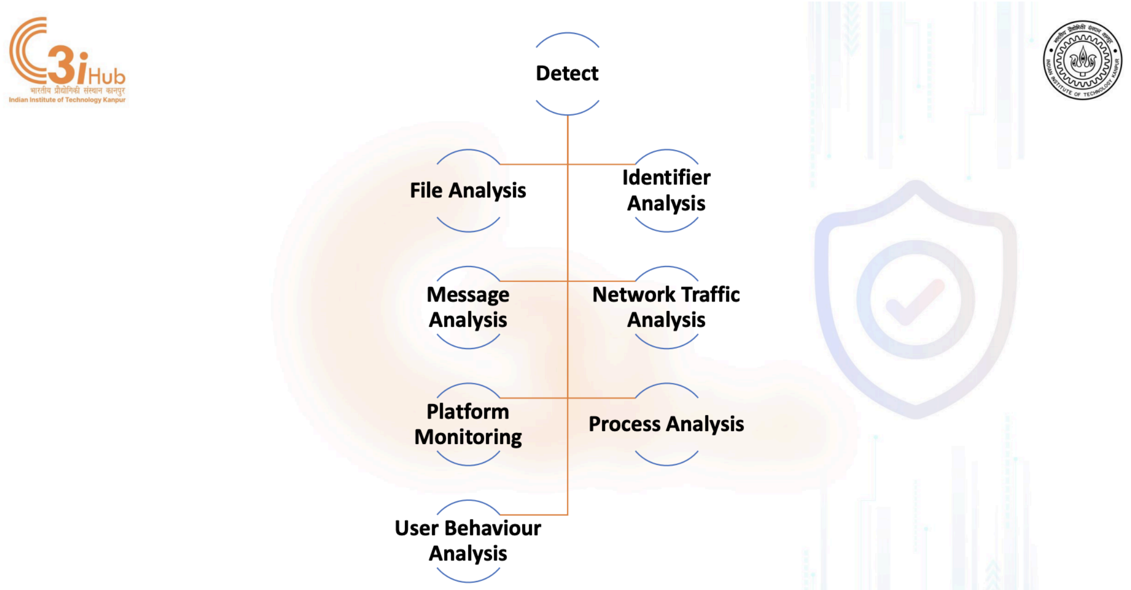
Platform Hardening

- Disk Encryption
- Driver Load Integrity Checking
- TPM Boot Integrity

So message hardening is one of the other things to reduce the possibility of attack. And platform hardening has all these things like encrypting your disk for example or part of your disk where you have critical files etc. When you load a driver right so one of the major attacks now it has reduced quite a bit is the driver you know you get a driver for your printer or your you know various devices and that driver itself has you know is actually malicious. So now you check the signature on the driver. So nowadays Microsoft, if the driver is from Microsoft, Microsoft will hash the driver binary, then it will sign it with its private key, right? And then you know Microsoft's public key.

How do you know Microsoft's public key? Because Microsoft has pinned the Microsoft certificate inside the operating system. So you can retrieve Microsoft's public key from there. Use that to check whether the driver is really signed by Microsoft. Then you know that the driver has been written by Microsoft, compiled by Microsoft, and while transferred from Microsoft to you, it has not been tampered with.

Its integrity has not been compromised. So the driver loads integrity checking, then boot integrity. When your machine boots, if somebody has changed the boot code, to do various malicious things then it will start to completely compromise your operating system. So usually we also check the integrity of the boot code. How do we check the integrity of the boot code again? We have to check the check against a signature. Now who signed this is signed by a private key that has been put by Microsoft or whoever is responsible for the boot code firmware, right.



Detect

File Analysis

Identifier Analysis

Message Analysis

Network Traffic Analysis

Platform Monitoring

Process Analysis

User Behaviour Analysis

So this is usually based on the trusted platform module, right. So some of the major keys in a system nowadays in today's system like in high-end phones for example they have a TPM or trusted platform module so certain keys are actually put into the trusted platform module so and any kind of signature that has to be done using that key you do not bring the key and put it on your application level or in the memory you actually send the content the hash or whatever you want to sign get signed you send it inside the TPM, TPM will give you the signed document so you will never see nobody inside your platform will ever actually see the key in memory right so that is how you save the integrity of the key. So these are platform hardening so when you are as a defender of your organization you will tell that all the assets, all the devices you have inside your organization, including servers, including desktops, engineering workstations, or any other device, you would like to get them hardened at the credential hardening level or at the operating system hardening level or all the applications that are put on them, hardening them and all that stuff. So you have done all that. Now you say okay so I have done all this so I can sleep better. Actually you cannot because that does not guarantee that you will not get attacked.

So now you have to think in terms of if I get attacked how do I detect right. So that is where the detect techniques will detect tactics and its various sub tactics will come in right. So what are the various detect tactics right? techniques so file analysis so this is somewhere you know that antivirus tools do file analysis they actually check for signatures inside files binary binary or any pdf or office documents or jpeg documents or anything that you download zip files and so on they try to find signatures of malware In more advanced file analysis would require machine learning, right? So machine learning based analysis for the existence of malicious code by extracting features from the file and so on. So you do file analysis, you can also do, you know, hash, file hash and use that against a list of hashes you know as malicious, right? So you can do that as well. identifier analysis so identifier analysis so it it might have multiple different meaning for example when you have an URL or IP address you may actually check it against a list of malicious URLs or list of malicious IP addresses there are machine learning techniques by which you can detect whether a particular URL is suspicious So, this is identifier analysis the other part of identifier analysis is like let us say you have a browser and you are you click on a link right and the link goes to some site that actually looks like a known website, but there is a slight difference an A there is replaced by an alpha.

These are called homoglyphs. So you have a URL that looks almost like the legitimate URL, but it actually has some slight difference. So it's going to a different IP address. So this is one way of phishing and one way of sending it to a malicious website by using what is called homoglyphs. So detecting homoglyphs. that you are yours now as a human you may actually be very very conscious and you can every time you go to a click on a URL you check one by one letter by letter am I be falling victim to a homoglyph attack or not nobody does that right so system has to do it system has to look for homoglyphs before redirecting you to a particular website it should check for possibility of homoglyphs so this is part of identifier analysis Message analysis is when you have messages of various kinds including email messages.

You may have to do analysis before you deliver it to the final destination like a user to see whether there is malicious content in the message or in case of API calls, you want to see whether it has a sign of malicious API call, things like that. network traffic analysis is very very important what kind of network and detecting various signs of intrusion various signs of anomaly those kind of things monitoring the platform that is the host monitoring right so not only monitor for file malicious files but you also look for signs like sudden change in performance, strain on CPU, certain strain on memory usage. You may want to also look at various other things in the platform, like whether some system files or privileged files are being changed, or you may want to look at whether a particular binary is being replaced by a different binary. This kind of stuff is in the platform monitoring or host intrusion detection. Process analysis is when a running process is being checked for possibility of process injection, DLL injection, or going through an exploit like a buffer overflow exploit.

now nowadays also the insider attack is very important like the insiders may do things which are harmful for the system so user behavioral analytics there are lots of tools nowadays they analyze what the users of your organization are doing in terms of their you access to various resources their their login times their usage of external resources like websites ftp ssh all these things and they try to find suspicious activities of users so so these are currently the list of different sub tactics of detect tactic right which has much more elaboration in the system in the defend.mitre.org website.