Practical Cyber Security for Cyber Security Practitioners

Prof. Sandeep Kumar Shukla

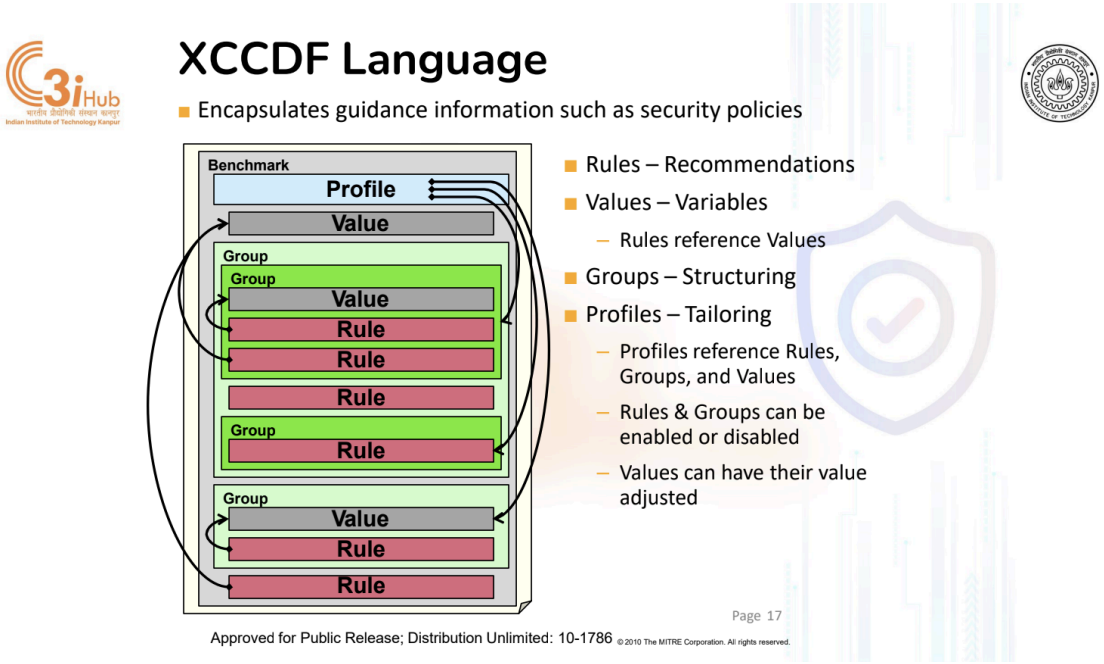
Department of Computer Science and Engineering

Indian Institute of Technology, Kanpur

Lecture 30

Deep Dive into CVE, CCE, CPE, CVSS Scoring, XCCDF, OVAL Languages - Part 2

All right. So, we were talking about SCAP. So, we talked about three enumerative standards that are CVE for vulnerability enumeration, CCE for configuration enumeration and CPE for platform enumeration and then we looked at CVSS which is for vulnerability severity scoring for that matrix. Now we are looking at the language part of these standards. So, XCCDF is the standard for benchmark configurations and will not go too much into the detail of how the language is structured. It basically is a set of rules and each rule tells you something about a specific configuration.



And so, in order to write the rules they have to also talk about some variables like some registry key variable or some configuration settings such as the number of times a password can be tried before locking out and things like that. And then the rules are put into groups that are a structural issue right. So, how to structure a set of rules into a group and then they have this notion of profiles which contains a number of groups and variables and rules. The idea of profile is interesting because let us say you have US military for example, would come up with a benchmark for let us say windows 11.

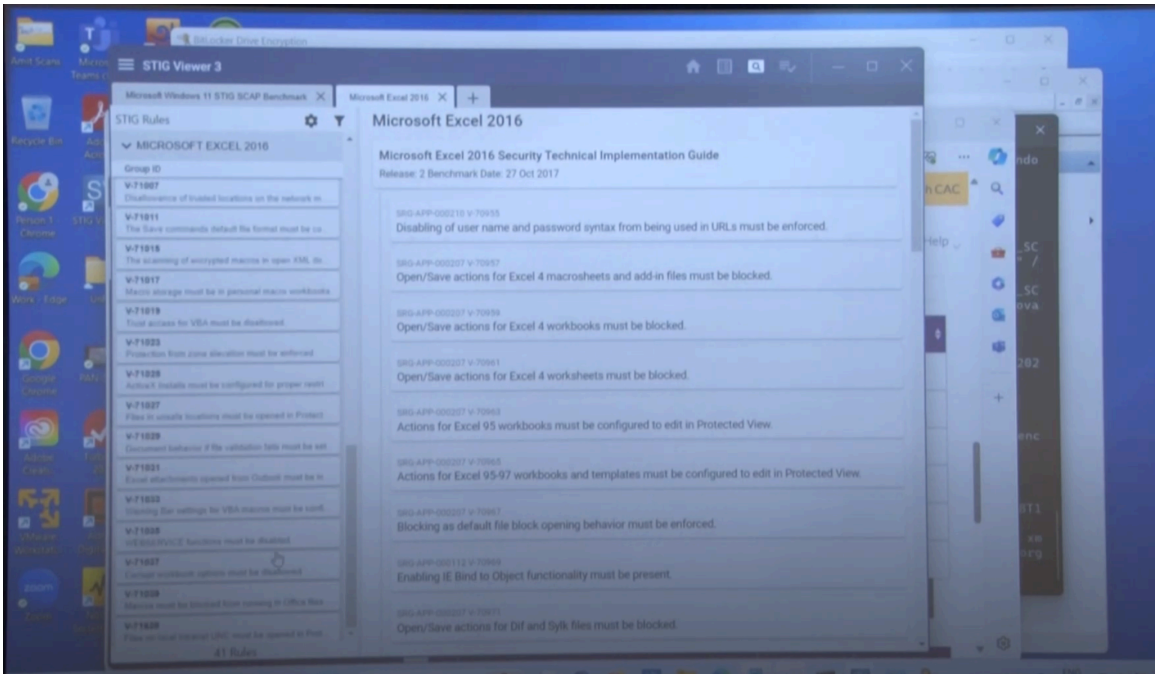
Now, within the US army,you may have computers which are used in critical missions, then there are also computers which are public facing, there are computers which are being used by regular users. So, not all of them have to be equally secure or their benchmark for their security configuration should not need not be the same. Some have very strict security configuration, some have relaxed security configuration. So, what they do is that they put the rules that correspond to let us say the most classified computers where Windows 11 is being used, they will put them in under one profile and then for public facing computers and in the same organization they will put them under a different profile. So, when you are actually measuring your configuration with respect to this benchmark you mention the profile of the device or or the endpoint that you want to be benchmarked against.

STIG VIEWER DOWNLOAD LINK: https://public.cyber.mil/stigs/srg-stig-tools/

So, I will show you how that all works. So, here is first of all there is DOD the Department of Defense has this tool called STIG viewer right. So, let us go and look at the website of STIG So DoD cyber exchange, so this is where you can find the XCCDF files for all kinds of software as recommended by the Department of Defense. Now another organization may have their own benchmarking. So now if you go here you will see that they have this thing called STIG viewing tool called STIG viewer.

So, STIG viewing tool you can have that for Linux or Windows version and STIG viewer tool you can download and this is the tool that I was just showing you. In this tool you can actually, you know, import or export a particular stick file and then you can actually, so you can open multiple different stick files or single stick files and whatever. So, open stick, so I can go and go to wherever I have this, and then I will, let us say, so u underscore ms, so I am looking for, so I do not, yeah. So let us say Microsoft Edge, the stick file is not there. So let us say I want to get to US, UMS, Excel.

So, I am opening an excel configuration file as recommended by for excel 200, 2016 version of excel and here I see that these things are under various groups and within groups there are various rules right. So, for example, let us look at some of the rules. For example, the first one says, so you have a group ID, the group it belongs to, a rule ID and then this is this ID of the STIG. STIG stands for Security Technology Implementation Guide. This is for unclassified, you know this that is why we are able to actually access it.



So, disabling username and password syntax from being used in URLs must be enforced. So, basically this is for Microsoft Excel. So, it is saying that username and password syntax used in URLs must be enforced. So, you can actually say that you know many URLs actually in the URL string you would give an username password combination like this right. Now that is very insecure practice.

So, if you have an excel file where you have URLs then you cannot have this you know user name password in the URL. So, here the next rule says open save action for excel form macro sheets and add in files must be blocked. So, this is about the macros, Then you have, let's look at another one. blocking as default file block opening behavior must be enforced. So, if a user can open a view or edit excel files that has to be that is the policy setting.

So, if you enable this policy setting you can have blocked files not opened, blocked files open in protected view or cannot be edited and blocked files open in protected view and can be edited. you have to choose what you are going to allow right. So, these benchmarks basically set for you the for anybody anybody's machine which has excel 2016, certain settings that then cannot be changed by those users and or at least if they change that will be a policy violation right. and in order to actually enable these settings in each of the machines you have to use some automation. So, you use this benchmark file and use something like OScap and then you can actually set these settings for that machine.

And the idea here is that this and this also says how to check whether the setting is there

and then if not then how to fix it right. So that also is there. So sometimes there will be scripts. So this is human readable. This STIG viewer allows you to actually look at this file in human readable form. So if I actually go into this file for example, so if I now go and look at this file actually, XML file, you would not be able to read it properly, right, because this is XML.

So, what this STIG viewer tool gives you is a human readable formatting of the same right. So, you can look at. So, you can understand what the rules are. Second thing that you can do is you can actually customize. So, you can actually have STIG editors. You can actually use a STIG editor to change some of the settings right.

So, you say I use STIG viewer to read each of these rules and I note down which ones I agree with and which ones I need to change in my organizational setup and then I can go back and actually change this file. in order to add or change some of the settings. Also I can actually, so this is kind of an object oriented language. So, you can actually also derive from that and then from an existing rule and then you can actually add additional constraints on the rules and so on. So, there are a lot of things you can do.

So, usually organizations actually take these benchmarks from CIS for example or DoD and then they make their own changes as necessary for their organization and then within the same organization in the same file you can have multiple profiles right. So for example if I let me so I use OSCAP and then I say info and then I say the file, So, you see that it will tell you. What are the different profiles? So, here you have an ID. So, there is a MAC 1 classified profile, there is a sensitive profile, there is a public profile, there is an administrative classified profile and that administrative sensitive profile right.

So, these profiles basically indicate the different set of settings for different profiles. different endpoints based on where those endpoints are, what their purposes are and so on. So, whenever I am going to automate the setting, checking that the settings are correct according to the benchmark or when I am trying to fix the settings when I find that there are some you know missing settings, then I have to mention the profile according to which I need to check or according to which I need to set configuration. So, now how do I do the checking right? So, I go back to the one where I have.

So, here I have the benchmark for Windows 11, this is for a specific version. So, I am going to do this, I will first check what profiles are there. So, I am going to check what profiles are there. So, I see that, so I am basically saying OSCAP info, then the benchmark file and then it will tell me what profiles exist right. So, mission critical classified, then you have mission critical sensitive, then you have administrative classified, administrative sensitive, mission support classified, mission support public and

so on.

So now if I try to check whether my computer which I use normally for teaching classes or sending emails and so on and I am not doing any critical classified activity on this computer I may not want to benchmark the settings of this machine on critical classified or sensitive etcetera. So I am going to do a mission to support public rights. So, I am going to check the benchmark against mission support. So, I have the command here. So, I am saying OSCAP XCCDF evaluates according to this profile.

This is the public profile and then the results are posted. This result can be in the XML format and also you can get a basically human readable form of the result in HTML format and here is the benchmark right. So, I can run the benchmark. Now it is running against each rule that is referred to in the profile that I mentioned, right. So, this whole benchmark file has multiple different profiles.

Profiles may have common rules, right? So, many rules will be written first and then they will be linked to the profiles, right. So, not necessarily each profile will have a disjoint set of rules, they might have common rules. And when I am So, I am seeing that I am passing some, but I am also failing some. So, you know this So, for example, my machine to get into my machine I do not do a LDAP check right.

So, I just have a local check right. So, I am failing this. So, there would be a lot of these things and this will take some time, maybe half an hour or so, to get this going, but I have run it once before. So, I can show you the result right.

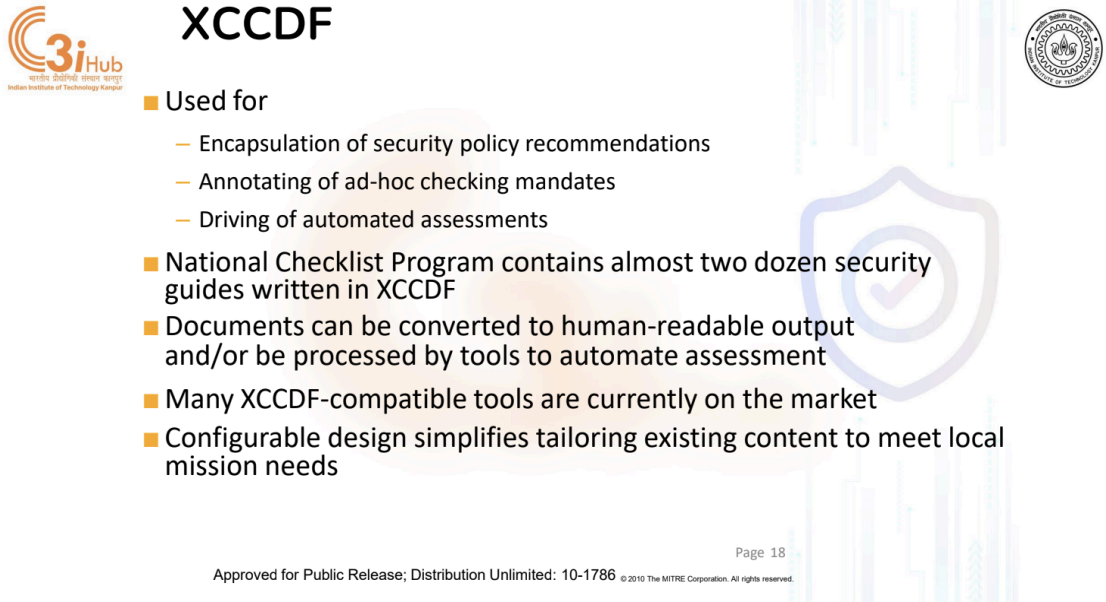
So, I will show you. So, here is the HTML report. So, once you have run it you will get an HTML report which will tell you which benchmark you ran and what is the profile against which you have run and then it will tell you how many rules I have failed in that profile. So, I have failed in 24 rules, I have passed 122 rules and then I have 54 others for which I am not. They were inconclusive. And then among the ones which have failed which ones are what of what criticality. So, certain settings are highly critical.

So, there are 2 such critical settings that we have failed and then 19 are medium and 3 are of low severity. So you have a score of 61 out of 100. Now your organization might have a threshold that in all computers must make 95 percent or 98 percent of the benchmark passing right. Otherwise you have to decommission that machine and fix all these things and then do this. So here you can also get a very detailed report on what it is that I have failed and you can actually go and check in further details and sometimes you may have a remediation script.

So it will not have the remediation scripts. But, you can also write like this: it tells you the remediation description right. So, enable full disk encryption on all information systems using BitLocker. So, I have disabled BitLocker on my laptop. So, it is giving me a failure, but it will also tell me what I need to do and then it could also have a PowerShell script that would have actually appeared here that would have done this.

So, that is something that you can get from this. So, that is basically how you are going to use this thing. Now, if you want to, this is for configuration, right. So, if you want to check for vulnerabilities, let us kill this. So, if you want to check vulnerabilities, you can actually do this.

I do not know if it will work. So, this benchmark also contains many OVAL, OVAL is the vulnerability tests benchmark ok. So, I need an OVAL file. I do not have the oval file here, but the process is very similar. then you have to use OVAL eval and then you have to give an OVAL definition file. An OVAL definition file will basically describe how to test for certain vulnerabilities right. So, instead of certain configurations it will tell you about certain vulnerabilities ok.

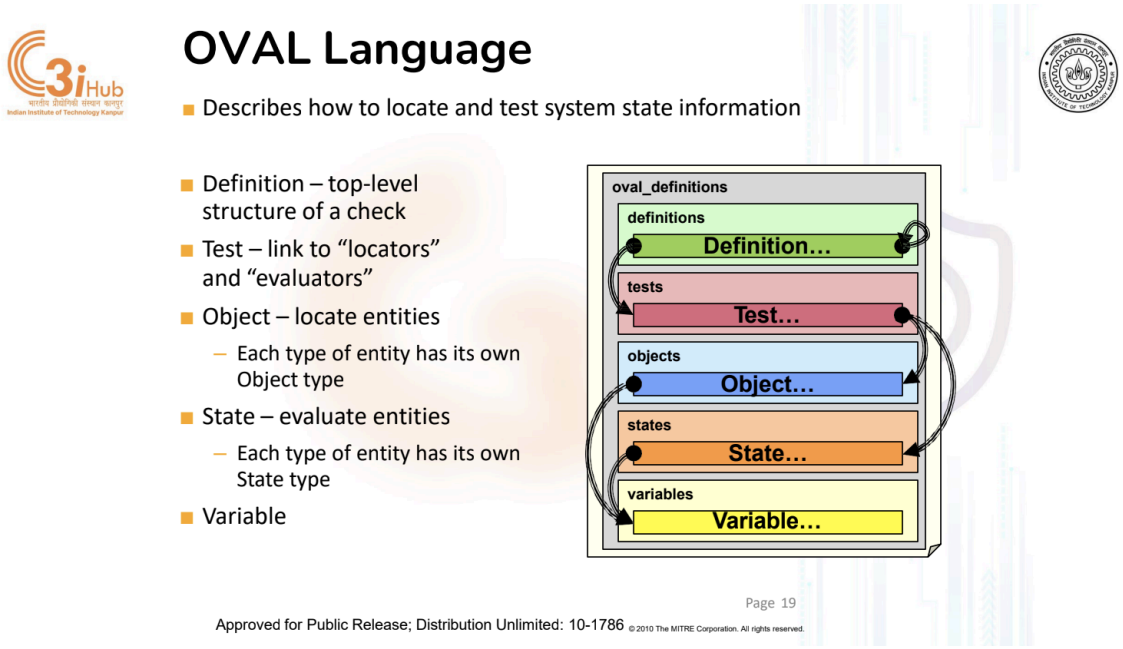


So, that is basically what it is to use this XCCDF. So, basically XCCDF is used for encapsulation of security policy recommendations. So, you have you know and what is a policy for an endpoint, for an endpoint the policy would be settings of various things that you can configure right. So, anything that you can configure either the policy would say that you have to set that particular configuration in a certain way or it will say we do not care right. So, if you do not care then there will be no rule for that particular policy for configuration setting right.

So, you can actually annotate ad hoc checking mandates. So, this is basically for audit and driving of automated assessments that we just saw using the OSCAP tool, but there are other tools which use STIG files to compare the settings policy settings compared to what you already have in your end point. So the national checklist program contains security guides written in XCCDF. So this is something that we believe that organizations like NCIIPC and CERT-IN and others need to do is to create a checklist. They can start from CIS benchmark or from the DoD benchmark, but they can then customize it in case there are, at least for government organizations it would be good if for example NIC for example runs these benchmarks against their servers and machines and various applications plus applications that are written by organizations here they have to write their own checklist right they cannot depend on like you know for example if you are thinking in terms of an application like you know NIC written application that you use for various government activities they have to write their own checklist right.

And then something like stick viewer can convert this checklist into human readable output and can be processed by tools for automated assessment like OSCAP and there are other tools like OSCAP is a free tool, but there are other tools commercial tools as well for checking compatibility against XCCDF and configurable design simplifies tailoring existing content to meet local mission needs. So, this is important and this is why this has been done in a language. So, XCCDF, so you can actually learn XCCDF pretty easily, it is not that difficult. then you can actually configure it so that it is relevant to your organization. For example, in IIT Kanpur we might come up with all the windows machines anybody who is using the internet has to run this benchmark.

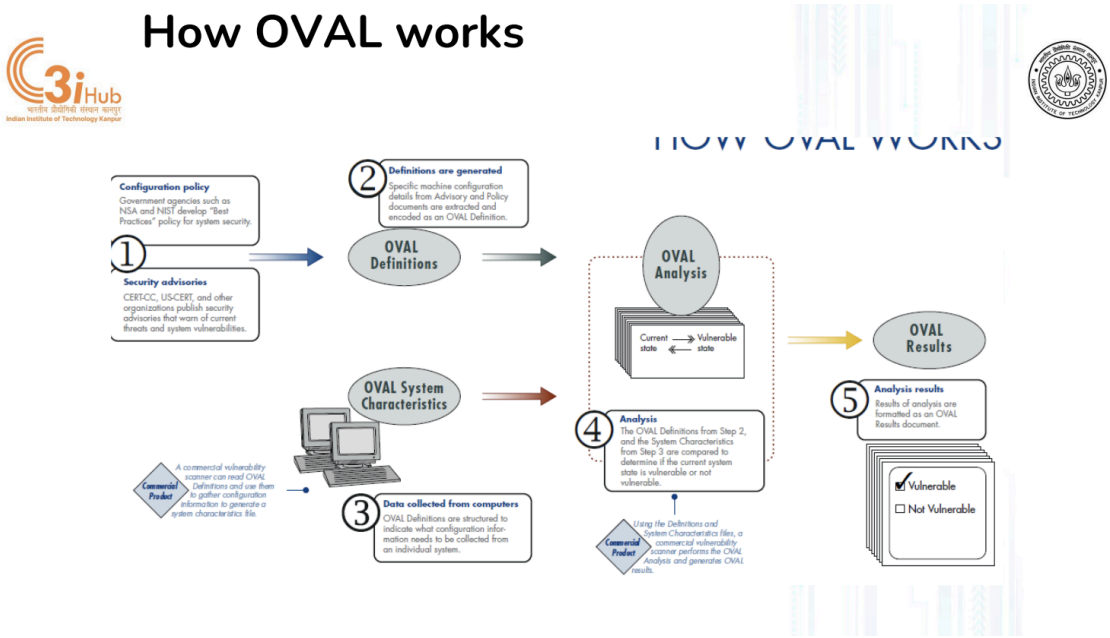
report to CC what is the how many configurations they are missing or failing. So, CC can decide whether to disconnect that machine from the internet or not. So, things like that can be done. So the next standard, so XCCDF is there for the configuration benchmarking exchange format. OVAL is an open vulnerability assessment language.



So if you want to know that your system has certain vulnerabilities, how do you know that your system has certain vulnerabilities? The first thing is that you have to check you can have vulnerabilities from two conditions right. So, one is that you can have vulnerabilities when a vulnerability is reported and the CPE and CCE associated with that CVE matches with your platform and your configuration is correct. So, not every vulnerability applies to your system right. So, your system has to be matching the CPE and CCE corresponding to that CVE. So, I have to check whether whenever I have a vulnerability discovered I need to check whether that vulnerability affects my system.

and in order to know whether the vulnerability affects my system sometimes I do not know all the components in my system. I am a regular user. I downloaded a lot of software. I purchased a lot of software. They have various components: Java libraries, this and that and so on. So, I may not be able to quickly figure out that the vulnerability has been also affecting my system. So, therefore, nowadays whenever CVE is discovered or published oftentimes a test in OVAL language is associated with that CVE. So, if you search in the NVD database you will find in many CVS there is also an oval definition of the test which will tell you if you run on your system using OSCAP, then that will tell you whether you are affected or not right.

So, that is what the OVAL language is about right. So, you basically create certain tests and these tests will be dependent on certain definitions and certain objects, variables and state. So, let us look at this diagram. I do not know if you can read this. So, this is what the idea of OVAL is. That let us say you have now missed a configuration setting is also a vulnerability right. So, you may want to know what is the configuration policy that you must abide by right.



So, if you are in the department of defense or in this case they are saying NSA or NIST right. So, this configuration policy and then you have the security advisories that are coming from CERT or CERT you know CC or US CERT etc. So, this CERT is basically saying that oh there is a new vulnerability that is coming that has come up and here is a definition in oval language on how to test for that. So, here either you take directly the definition given by them or you write the definition based on the policy etc., that is what I am going to test to check whether that particular problem exists.

In parallel, the software the OSCAP will also collect associated variable values. So, there are certain values of variables. For example, what is the current version of Windows you are running? What is the current version of let us say Chrome you are running or what is the current version of some software you are running or things like what is the what is your you know particular version like for example, what is your JavaScript version, what is your PHP version and so on. So, all this information is collected from the computer. So, you have the definitions which tells you what to test and here is you know the information you need to do that test.

and then you do the analysis. So, you do the analysis and this analysis will tell you whether let us say the definition says that a certain variable should have a certain value and you see that this one has a lower value right. So, from that you will say that this machine is affected by that vulnerability because the definition says that to be free of that vulnerability I have to have a version number greater than this, but my version number is so and so which is not greater. So, therefore, I will get a failure and I will get a vulnerable output. So, this way a document will be generated and an HTML document will be generated which you can view and see what are the vulnerabilities you have. The OVAL

file may also contain the information about how to fix it right.

So, like whether you have to download a new version, but that will come from here right it will not automatically be there. So, if that information is here then this final result document will give you guidance about how to go about fixing the problems right. So, that is what the OVAL is about. So, this is what the OVAL kind of looks like.

So, here is a test. So, you have an id for the test and then it basically has some meta information like check all. then what object we are talking about now this is an example. So, in this case we are talking about a tutorial object and what state we are talking about. So, this state is also a synthetic state variable right.

So, this variable and object are you know not important for us. Now, a test referencing an object and what value the object must have. So, for example, here you have a test where you have a tutorial object 1 and this object basically says that the key value This is a registry key value should have a certain value right. So, this is just to give you an example about how the registry object looks like. So, a registry object will have this hive and key and the actual name of the key.

OVAL example continued

```
<registry_state id="oval:tutorial:ste:1">
 <value>Hello World</value>
 </registry_state>
```

State/Value that the Object Must have

```
<definition id="oval:tutorial:def:1">
        <metadata>...</metadata>
        <criteria>...</criteria>
</definition>
```

Definition holds it together

So, that is how the registry key is referenced in a test. So, now you have a state named STE1 and let us say its value is hello world. So, this is just to give you an example and then you have a definition which has various other fields, for example, a test criteria that you are testing right. So, here criteria is saying that the value of the registry key should be equal to hello world this is the criteria in this test right. So, this is how this language looks, not very important for us right now to know exactly how this language is. I have a I am I probably have videos where I explain how to about the whole language of XCCDF as well as oval. I can probably make the links available if you are interested, but for this class it is not important I am just trying to give you.

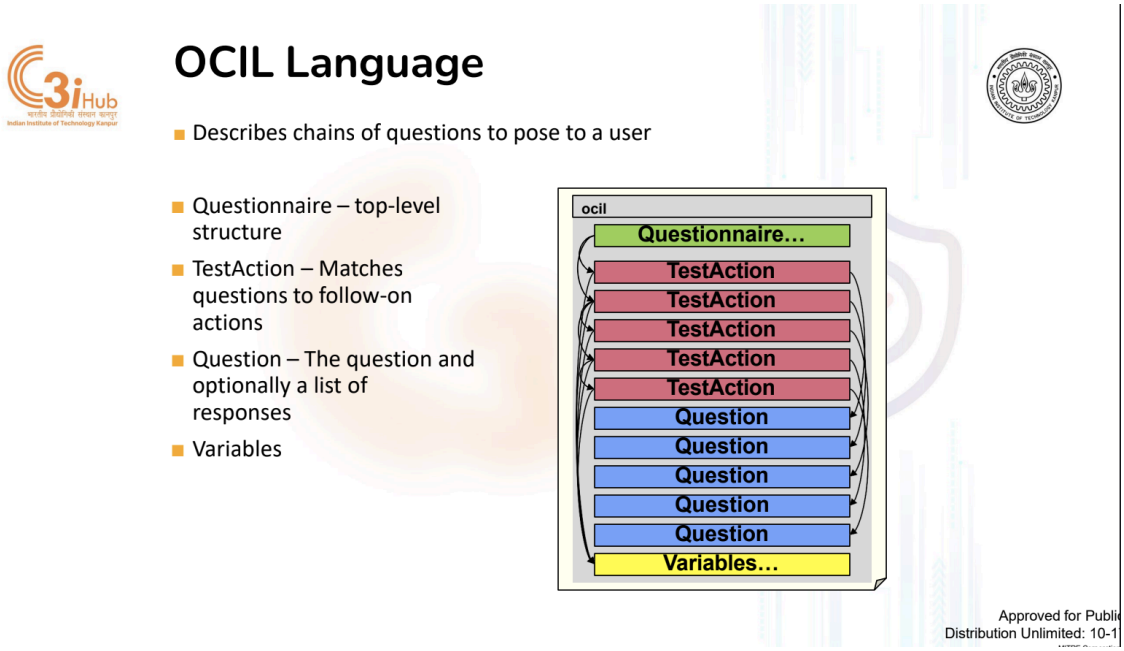
OVAL Example Continued

```
<metadata>
        <title>Hello World Example</title>
 <description> This definition is used to introduce the OVAL
Language to individuals interested in writing OVAL Content.
</description>
\</metadata>


<criteria>
<criterion test_ref="oval:tutorial:tst:1"
comment="the value of the registry key equals Hello World"/>
</criteria>
```

the idea about security content automation right. So, security content automation has this

benchmark for configuration XCCDF and the fact that XCCDF benchmarks are many of them are publicly available for well known software and operating system etcetera and the fact that you can customize it and the fact that there are tools like waste cap which can be used to test your endpoints against those benchmarks for certain profiles will be good enough you know information that you need to need to remember from this class because this is something that should be much more widely practiced in India than it is right. So, you will not see many organizations using XCCDF benchmarks or you know tests and so on which is a shame because it is there is lot of content that are already available that one can reuse and get at least benchmark not if even if that benchmark is not necessarily made a requirement, but it can be tested just to even know where you are with respect to that.



Then the last one is actually a later addition to this CAP benchmarking which is the OCIL language right. So it is an interactive language for questionnaires for asking questions to security personnel right. So security personnel are often needed. Remember that security is about processing people and technology correctly.

So, a lot of things that you know happen, you have to ask these questions to the or to the personnel about their security practices when you are actually figuring out the security posture of that organization. So, OCIL is a language that was designed for that. and it is for non-technical policy issues right. Remember when I said cyber security policy always has some technical policy parts such as the ones we were talking about in terms of configuration like how you know the system should be configured, whether endpoints should have this property or that property and so on. There should also be policies which are unrelated to real endpoints or networks etc.

OCIL

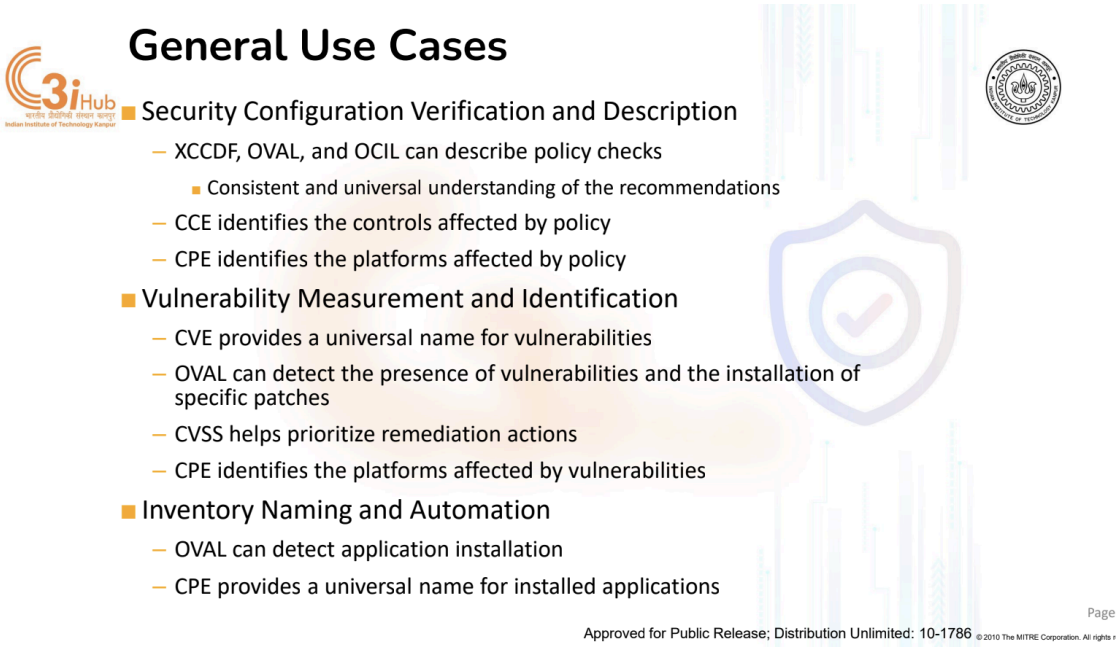
- Used for
  - Queries regarding non-technical policy recommendations
  - Manual collection of artifacts that provide evidence of security posture
- OCIL will be part of SCAP 1.1, released in January 2011

, they are mostly related to the behavior for example, like you know whether you use social media while on your work computer or whether you download software on your work computer games you download and then you may violate policy right. So say the policy is clear about what you can or cannot do and then you can ask these questions to the people involved. So that's this thing. So where we can get more about SCAP and more resources from SCAP, one is of course the National Vulnerability Database and its search engine, it has APIs. So APIs will allow you to have your own tools in which you can actually get information directly from the NVD database.

What Resources Does SCAP Have?

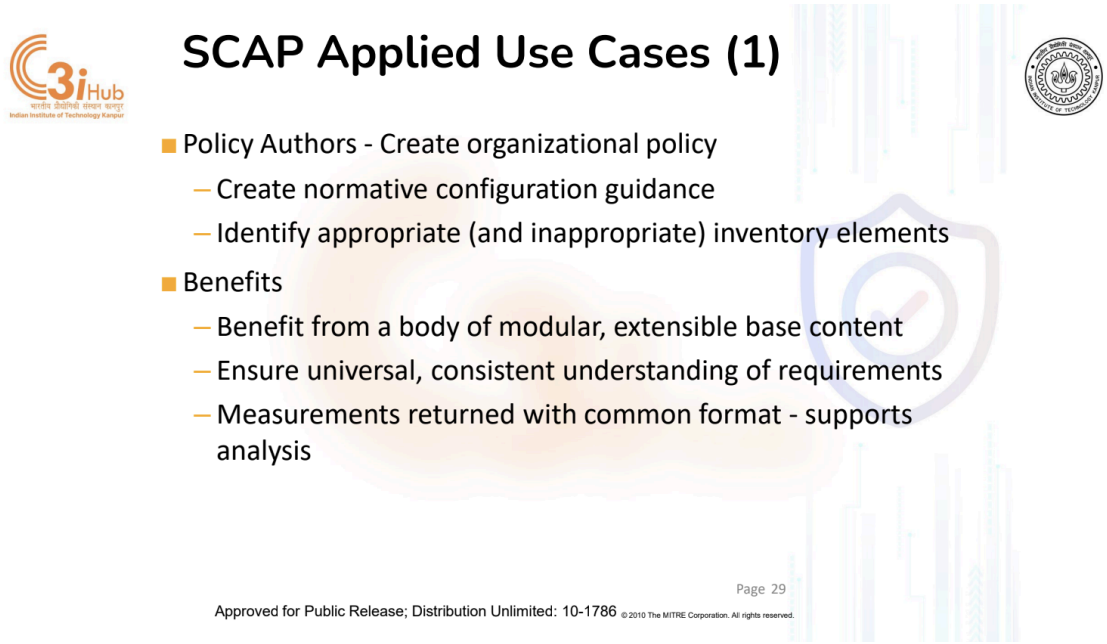
- National Vulnerability Database
  - Vulnerability Search Engine
    - Annotated CVE entries include CVSS scores and vectors, CPEs, and other information
  - National Checklist Program Repository
    - Guidance for many applications and operating systems
    - Many guides use SCAP – usable by SCAP compatible tools
    - Includes STIG, FDCC, USGCB, and vendor benchmarks
  - CPE dictionary
    - All official CPE names for platforms
- Component standard sites
  - OVAL – OVAL repository with over 7000 definitions
  - CCE – The official list of CCE entries
  - Documentation, use cases, and other information on all sites
- Mailing lists and archives

You have the national checklist program repository. So, the one that we saw is the DOD checklist repository. So, it includes various benchmarks for the configurations for various applications and operating systems. You have the CPE dictionary that we looked at in the previous class on the MITRE website. You also have an oval repository over 7000 definitions of XML files, you have the CCE the official list of the CCE entries and you have various documentation use cases and other information and there are other mailing lists and so on. You also have XCCDF oval and OCIL. You know this for universal understanding of recommended practices.



CCE for identifying control affected by policy, CPE to identify platforms affected by policy, vulnerability measurement and identification. So, you have CVEs for naming the vulnerabilities, oval for detecting the presence of vulnerabilities and installation of specific patches, CVSS gives you the prioritization because you actually prioritize based on CVSS. However in recent times people are moving away from CVSS and going for something called EPSS which is the exploitability severity score right. So how exploitable a particular so you may have a vulnerability of severity 10 but it might be very difficult to exploit so especially in your organization.

So therefore people are moving to EPSS but that is not part of SCAP. CPE tells you which platform is affected by which vulnerability and then you can also have the inventory naming and automation. So, this is for asset management. You can use Oval and CPE for naming the applications and then detect the application you know, problems that are associated with the application. So who are the users of SCAP? So policy authors, people inside the organization who create the organizational policy. Now one thing is to write a policy in a PDF, in a word file and circulate it in as a text file to everybody, right.

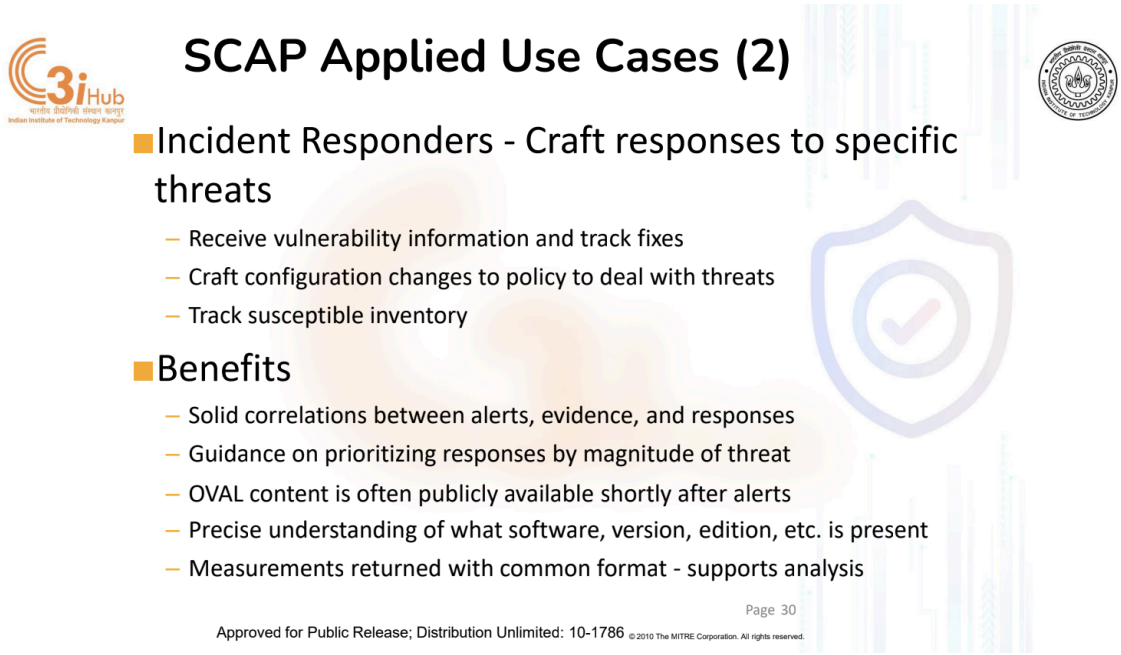


And usually these policy documents are pretty long. And, you need many many awareness and training sessions of your employees to get them familiar with the policy, because a policy document is a very boring sub very boring document to read. So, you will not find many people reading the policy document. And therefore, there will be a lot of violation of policy rights. So, therefore, it is up to you the CISOs office to continue to give training sessions and make it mandatory to attend those training sessions and so on. So, people get familiar with the policy, but the bigger important issue is that once you have the familiarity with the policy, but still technical technological policy like the configuration setting etcetera.

cannot be manually done like you have thousands of machines, some are mission critical, some are less mission critical. So, you have to automate the process of configuring those machines according to the policy. And policy might also include things like you have to be compliant with 27001 or we have to be compliant with PCI DSS and so on. So, these benchmark files usually contain all those details like what is it that you know what configuration setting will make you PCI DSS compliant and so on. So, create normative configuration guidance for automation this is the XCCDF and identify appropriate inventory elements or inappropriate inventory elements on which this policy has to be implemented.

So, benefits, benefit from a body of modular extensible base content. So, you can actually create your own extension of XCCDF etcetera, ensuring universal consistent understanding of requirements. So, requirements remain unambiguous, everybody in the organization will run OSCAP with XCCDF files. So, nobody has to interpret the English language command on what configuration should be. measurements returned with

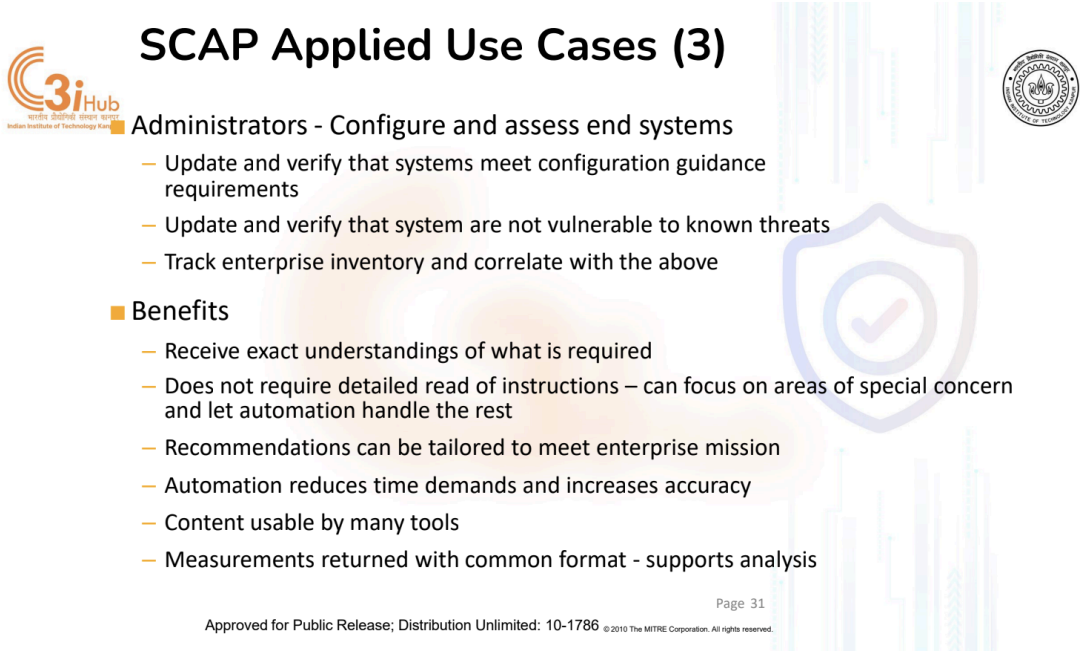
common format supports analysis so once everybody runs or you can actually there are ways to run this thing directly from a console to all the machines in the in the organization right so you do not necessarily have to go to every machine and run this manually right so when you do that you can get back all the results together and you can combine the results and get a holistic picture of your organizational security conditions So, incident sorry incident response incident responders. So, receive vulnerability information and track whether fixes are done configuration change tracking to make sure that the policy according to policy the configuration change has been applied.



and track susceptible inventory like any kind of asset that has been susceptible to vulnerability you can track and see what to do like whether to decommission them or whether to temporarily shut them down and things like that or temporarily disconnect them from the network and so on. So, using benchmark and using oval you can basically identify automatically across your organization all assets which are affected by certain vulnerabilities and you can also see how many of them are being fixed and so on. So solid correlation between alerts, evidence and responses can be obtained because when you get alerts you actually know exactly which machine it is talking about and then you can actually know that there is some problem with respect to that particular configuration or particular vulnerability. Guidance and prioritizing responses by magnitude of threat.

So, this is what they are talking about CVSS. So, using CVSS core you can tell which vulnerability you must fix. Overall content is often publicly available shortly after alerts. So, this is what I was talking about. Nowadays in the NVD database many CVEs come with the corresponding overall content. So you can just cut and paste, put it into your overall file and rerun your overall test, right.

In fact, tools like Nessus, they basically do this kind of stuff. precise understanding of what software version edition etcetera are present and measurements returned with common format. So, it supports analysis. So, again the output of these measurements using oval or XCCDF can give you a good picture of what is going on inside your organization. So administrators, so they can configure and assess the end system.

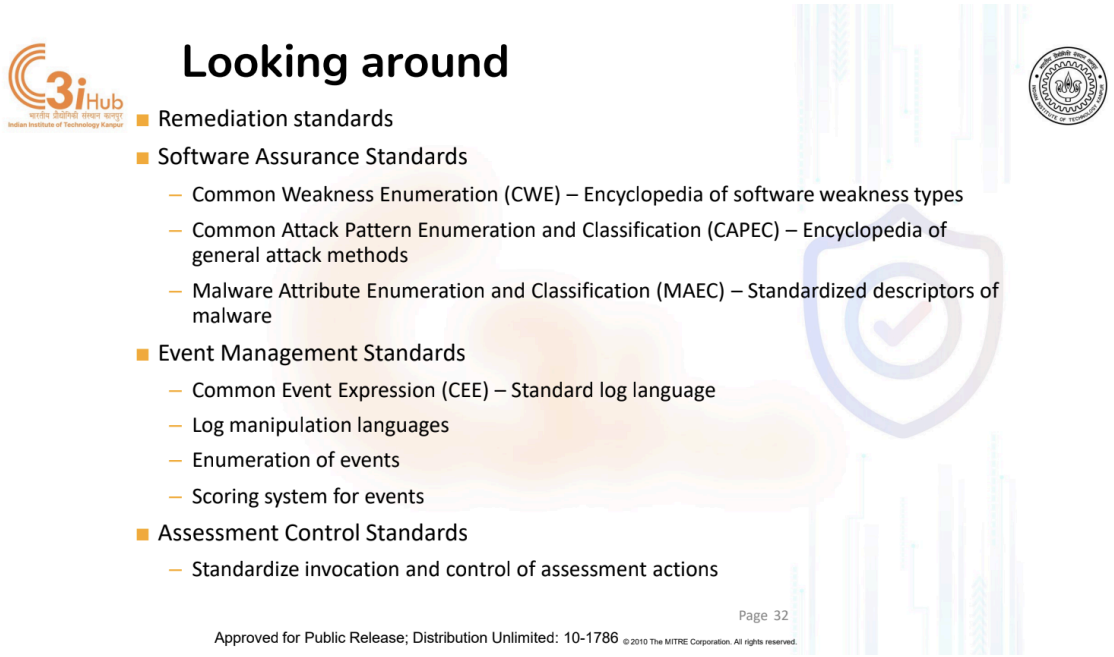


So update and verify the system, meet configuration guide. So administrators, of course, nowadays, you know, without, in absence of these things, they have to manually configure, right. So they may forget to configure certain settings and so on, so forth. So this will basically allow them to automate this, plus they don't have to use their own brain. to actually decide what configuration setting should be what is secure they can actually depend on the benchmark. And so they can also check whether vulnerabilities they are using are running the overall thing and then they can track the enterprise inventory and correlate with the above.

So which assets are having all these vulnerabilities? So, of course the administrators as I said do not have to use their own brain to understand what needs to be done, what configuration needs to be done, which is part of the benchmark. It does not require detailed reading of instructions because it is basically automated. Recommendations can be tailored to meet enterprise missions. So, that is what you do: you take the benchmarks and then you tailor it. Automation reduces time demands and increases accuracy of course you cannot forget to check something or fix something because it is automated.

Content usable by many tools, so there are lots of tools other than OSCAP that you can use and measurements written in common format of course supports analysis that gives you a good idea about where we are. So, there are other related standards or I will not say

standards, but these are now becoming kind of de facto standards such as we talked about common weakness enumeration right. So, many CVE now come with common weakness enumeration. So, it is an encyclopedia of software weakness types like we were seeing yesterday in the previous class that one of the CVE was associated with CWE that talks about the incorrect handling of the or inadequate handling of the inputs, command inputs leading to command injection.

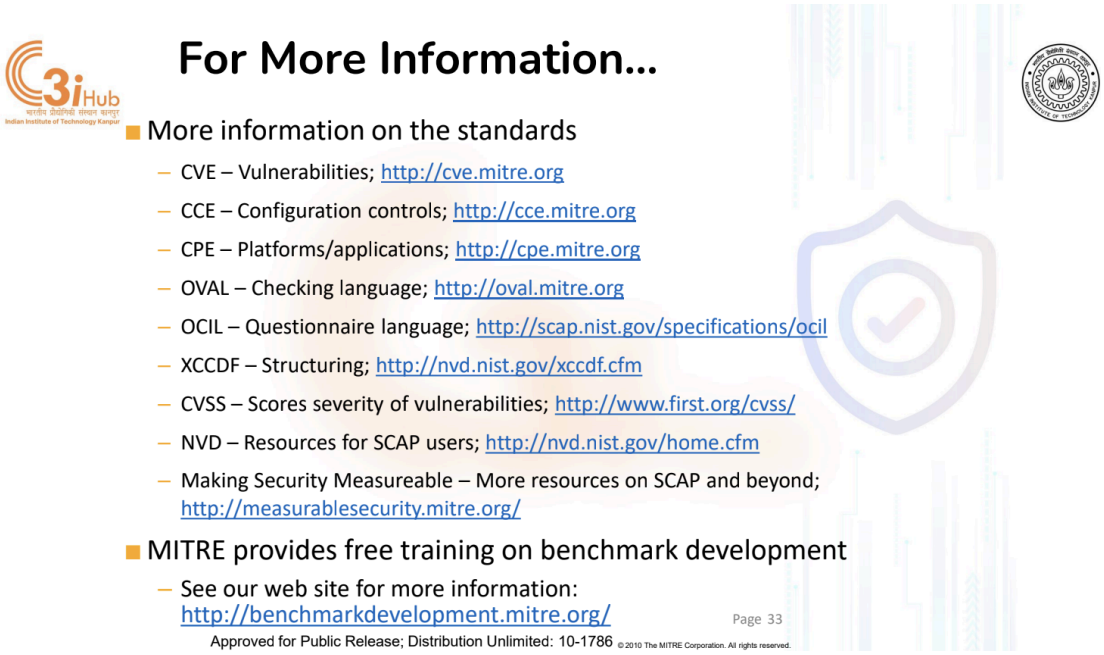


There is also software attack pattern enumeration and classification CAPEC. So, this is a general attack method enumeration. So, we already saw this in the context of MITRE ATT & CK. And then there is also malware attribute enumeration and classification or MAEC standardized descriptors of malware. So, these are all from MITRE. So these are additional knowledge bases I would say which every security professional should be familiar with.

And then there are event management standards like common event expression, it is a standard log language, log manipulation language, enumeration of events, scoring system for events. So, these are event management standards and then there are assessment control standards. So, these are all additional things. and above SCAP right. So, SCAP so what the point of this slide is that SCAP by itself has a lot of powerful you know standards and then because of its because of the effort by NIST on SCAP the number of tools have come up, number of benchmarking languages have come up and this benchmarking languages are very useful for automation of configuration setting, vulnerability checking, etc.

But there by themselves are not enough, there are more things out there which are also complementary to what SCAP does and there are a lot of other things which have a lot of

value as you know in the arsenal of a security professional. And then one final thing is that you have all these places where you find more about these things. We just scratch the surface and you can also get how to develop the benchmarks. So, benchmark development you know tutorial is there.



So, if you are interested in developing benchmarks you can use this. So, that is the thing that is it. So, we are done with the SCAP. So, tomorrow I will just have the next class. I will just summarize what is it that we learnt and what is it that you are expected to remember and what is it that you can forget.