Practical Cyber Security for Cyber Security Practitioners

Prof. Sandeep Kumar Shukla

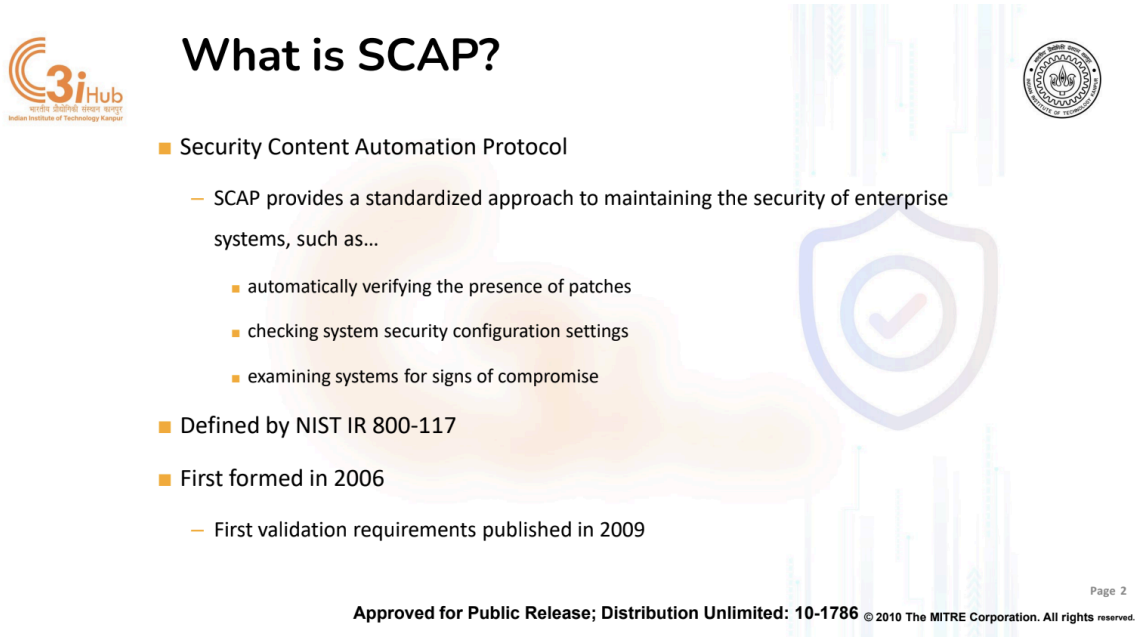
Department of Computer Science and Engineering

Indian Institute of Technology, Kanpur

Lecture 28

Introduction to SCAP, CVE and CCE

So, now that we have finished talking about the data model for sharing threat intelligence with STIX, I have some time. So, I have like 4 classes including today. So, I am going to use that to give you some introduction to SCAP which is the Security Content Automation Protocol. So, this is something that came out of NIST. So, STIX is actually some security content right. So, you are basically sharing threat intelligence in some stylized format which can be parsed tools like firewalls and intrusion detection tools, endpoint detection tools to make various kinds of configuration changes and various kinds of actions they can take.



So, STIX is part of security content automation, but we are not talking about threat intelligence. We were talking about checking the, you know, devices' endpoints with an automated tool for presence of vulnerability for presence of configuration mistakes and for presence of any kind of lack of patching this kind of information I want to automatically check on every device on my network. So, if I could do that, then I could basically have a console from which I connect to all the my devices in the network and then I will run a scanner which will check whether every device has the right

configuration, whether every device has the right patch or whether every device has any or whether the device version or application version etc have known vulnerabilities and report to the console. that in your 1000 devices in your system so and so device at so and so location has these problems.

So, I will be able to see that and then I will be able to actually try to do patching and other activities. So, that is what we mean by automation. So, I want to automate checking for vulnerabilities and configuration mistakes and I want to also automate checking patches and all that stuff or also other things like signs of compromise and so on. So, to do that first of all what do I need right to do that first of all I have to if I want to automatically check that this Windows 10 machine has any vulnerability or not whether its configuration is correct or not. whether it has the latest patch or not, what do I need right.

I need first of all a definition of what it means to be the correct configuration right. So, I have to have a configuration benchmark or a standard configuration for Windows 10. Similarly, within that Windows 10 machine I might be running Microsoft Edge or you know Firefox or Chrome, I might be running various other applications and for each of these applications there may be thousands of different possible configurations. So, how do I know what by automatic scanning that the configuration of all the applications the most secure possible secure configuration, whether it is compliant with let us say standards like HIPAA or standards like PCI DSS or any other kind of standards that requires the configuration to be of a particular type, then I have to have a way of writing what is the correct configuration. So, that is one thing I need to be able to write in a way the configuration correct or secure configuration or acceptable compliant configuration using some kind of a common language such that the tools can read that and then compare that against the existing configuration of the machine that it is checking right.

So, I need to be able to define the benchmark configuration. So, that is one requirement. Second requirement is that if I want to check for vulnerabilities in common vulnerabilities, then I need to be able to express what I am looking for and what kind of vulnerabilities I am looking for. So, I have to have a way of defining vulnerabilities that I look for in devices. So, I need a language for defining vulnerabilities and how to test for those vulnerabilities.

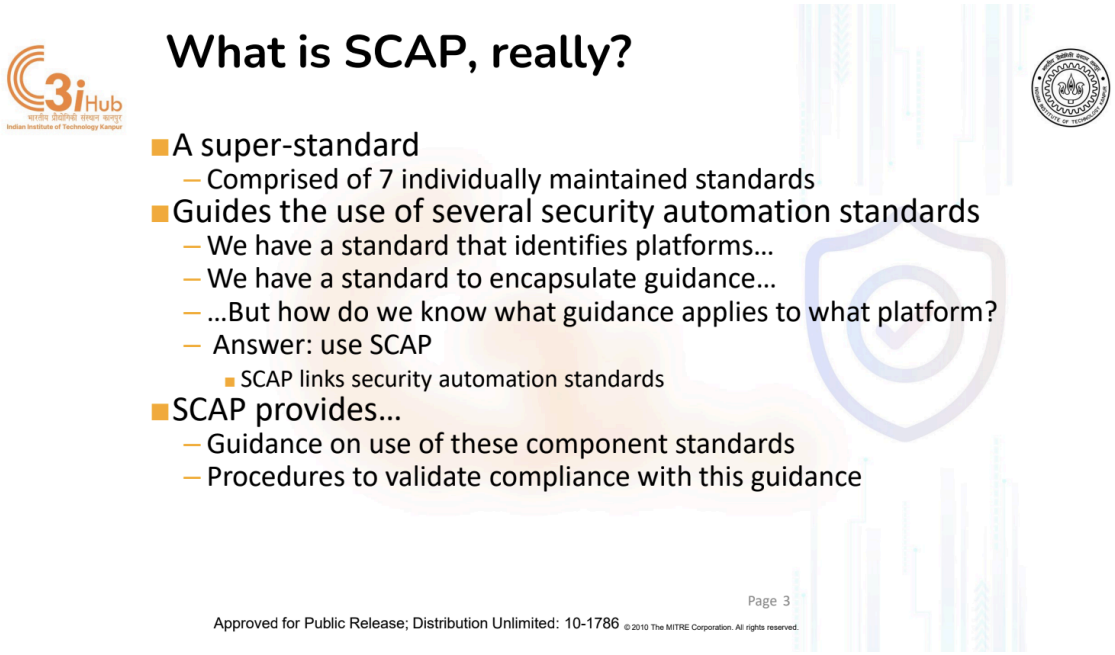
So, I have to have a way to, I have to tell the tool that this is what you should check or this file you should check or this configuration you should check to know whether that particular vulnerability is there. So, I also need to also know that if I need to also know whether it has the right patch level, then I have to go and look up the exact version that is running on that device and then I have to go to some kind of a database which tells me

that what is the latest patch for this thing and I should not make any mistake about that right. So, Windows 10 might have many many versions right. So, Windows 11 may have many versions including different versions for different languages. There is a Chinese version of Windows 11 and maybe a German version of Windows 11 and an English version of Windows 11. Similarly, it may have multiple different patch levels and so on.

So, I need to know exactly which machine I am talking about and which application I am talking about and I have to be very very precise and exact. To do that I need some kind of a way to express the most uniquely and most precisely the software or hardware or application etc in some standard format. So, I need a standard way of expressing what it is, which application or which operating system or which firmware we are talking about. So, I need a standard way of expressing what we call a platform. So, which platform I am talking about. So, I have to have a way to do platform enumeration.

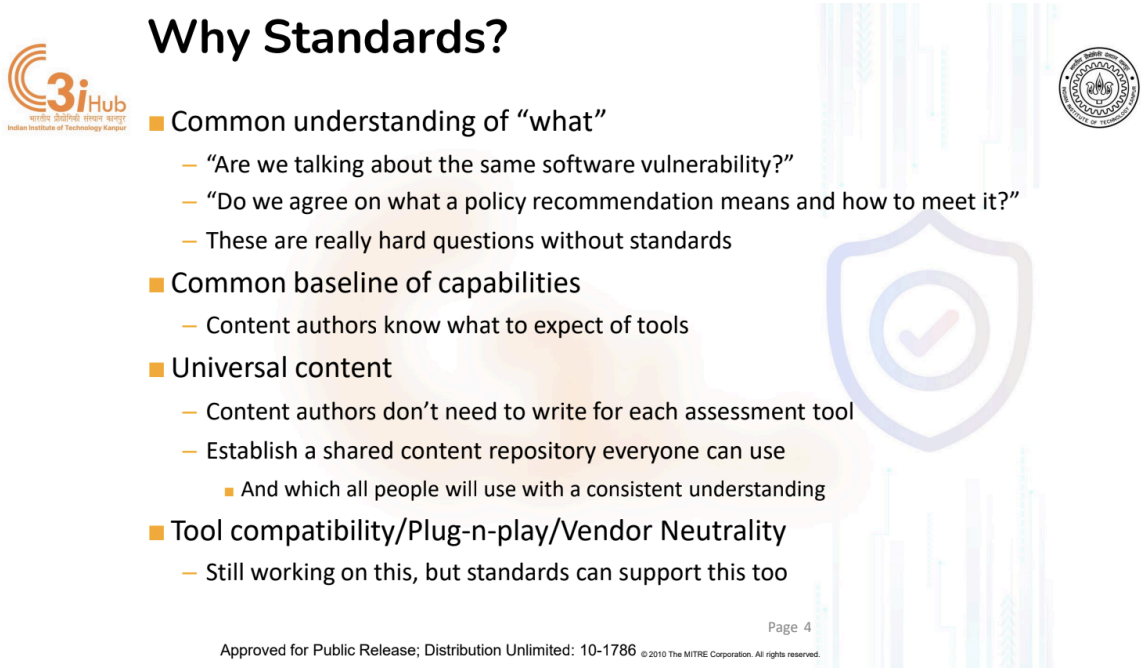
Similarly, when I am talking about configurations like I did in the beginning, I said that I need a language to express which configuration settings are correct. So, but to talk about configurations like which configuration I am talking about, am I talking about the configuration about you know what is the longevity of passwords in the system or am I talking about whether you know a particular registry key has to be set in a particular way. So, these configurations which configuration I am talking about that also has to be standard there should be a standardized way of saying I am talking about this configuration and that configuration. So, I have to have a standard way of talking about configuration. So, I need a language for configuration enumeration.

So, all these things are part of this SCAP standard or SCAP vocabulary and this is why it is actually a collection of standards. So, it is not a single standard, but it is a collection of standards as we will see which has multiple different components for naming the platforms, for naming the configurations, for naming vulnerabilities, for measuring importance of vulnerabilities, for expressing what are the correct configurations, for expressing what vulnerabilities we are looking for. So, I need different standards for all these things and this collection of standards is altogether called SCAP or Security Content Automation Protocol. And it was defined by NIST way back in 2009. in the NIST IR 800117 document and we will see that there have been a number of other things that have been created.



So, as I said that this is a super standard 7 individually maintained standards put together is called SCAP. So it guides how to identify a platform uniquely and unambiguously right. So if I write a platform in the way they ask us to write then I will uniquely identify that ok I am talking about you know the Windows 11 so and so patch number and so on. It has a you know it is a standard to encapsulate guidance and by guidance I mean the benchmarks like how to best configure your application or how to best configure your operating system or how to best configure your hardware and so on. So, that guidance has to be written in such a way that it can be automated, understood by a program, and used to take action to set the configuration correctly.

And so, how do you know which guidance applies to which platform? That is where the use of the SCAP protocol comes in, as SCAP essentially links the various security automation standards. It provides guidance on how to use the component standards and how to validate compliance with this guidance, and there is a tool available. In the next class, we will look at the tool, and you can also download it; it is a free tool called OS CAP. You can download and test the benchmark to determine whether your machine is running the Windows or Linux version according to the standard benchmarks. We will obtain the standards from the US Department of Defense, or we can use CIS standards.



There are many standard benchmarks for secure configurations. So, regarding standards, we haven't talked about them much in class, right? We have discussed ATT&CK, which is a framework; we also talked about MITRE DEFEND, another framework; and we discussed NIST CSF, which is yet another framework. Additionally, we talked about the Lockheed Martin kill chain, or unified kill chain, which is also a framework. And then we talked about various languages, such as STIX, for example, and we discussed risk assessment. During the discussion on risk assessment, we talked about some standard ways of conducting risk assessments.

We briefly mentioned COBIT and various other methods, and we also demonstrated how NIST approaches risk assessment. You know that guidance is used for risk assessment. We also talked about resilience there; it is a maturity model, right? So, we talked about an RMM, or Resilience Maturity Model. Now, standards are actually very important not only for security but also in general, right? So, for example, when we buy a phone, and we have Bluetooth, nowadays all smartphones have Bluetooth, right? Non-smartphones also have Bluetooth. Now this Bluetooth will work with any device, let's say an earpiece, headphones, or something else that is Bluetooth enabled, and your phone is, let's say, a Samsung, while that other device is from another company, like Boat.

They connect immediately, right? How do they work? Because Bluetooth has been standardized and the standard has evolved. Currently, there is low-power BLE (Bluetooth Low Energy), which is what everyone is implementing. Before Bluetooth, there were a few other options like Zigbee, among others. But the fact is that earlier, companies used to have their own protocols for communicating with peripheral devices. Now, when

companies start having their own protocols, you cannot do plug-and-play; you cannot buy new equipment and connect it to that device, which makes it very difficult and also makes you bound to a particular vendor.

Like Apple tries to do for its users, the standards allow industry players to come together and define a common vocabulary and protocol that everyone will implement. This makes it easier for others to plug in and also check whether they comply with a particular set of rules or engagement protocols. So, standards like the 27001 standard we talk about, right? So, the ISO 27001 standard is actually what we call the ISMS standard, or Information Security Management System. How do you manage an organization's information security? So, if you want to be satisfied that the company you are working for, for example, is supplying software to your organization. And you want to know if that company is following proper security measures to protect the data you share with them, as well as the intellectual property you share with them, and so on.

How do you do that correctly? One way to do that is to go check for yourself. So, you send someone from your organization and say, "Okay, go and look into their cybersecurity management system, and if everything is fine, then we will do business with them." However, that will be very onerous for companies because a single company might work with many suppliers. So, if you have to go and check with every supplier individually, that will be very difficult. So, what you need is to see whether those companies you work with have actually implemented their cybersecurity according to a standard such as ISO 27001.

Now, if they are compliant with ISO 27001, then you need to know whether they have truly implemented it. So, you need to look at their audit report. If you claim to be implementing ISO 27001, then you have to maintain an audit where auditors will come and check your system and identify the areas where you fail to meet the standard, and you will need to address those issues. And once they fix it, they call back the auditors, who double-check and say, "Okay, now we are satisfied." So, we now certify that you are 27001 compliant, right? So, that is how the standards are used, right? Similarly, in the case of systems like OT systems, such as industrial control systems, the standard is IEC 62443 or ISO 62443.

So, if you claim to be compatible with or implementing 62443, then you have to get a qualified auditor who can certify that you are indeed compatible with it. So, standards are a good thing, right? So, standards are a way of, you know, addressing a problem. You know, I can give you a handle on the problem, right? So, if you want to check the security of other organizations—if you are a regulator, for example, the Reserve Bank of India, SEBI, or the Central Electricity Authority (CEA), and you have industry regulation as

part of your charter—then you may want to have auditors. That they are compliant with some standard or guideline, right? So, not all guidelines become standard; they become standard when there is a consensus across the industry that this is what is good for everybody. So, now standards also provide you with a vocabulary, right? So, by using standard terminology, we can communicate with each other effectively.

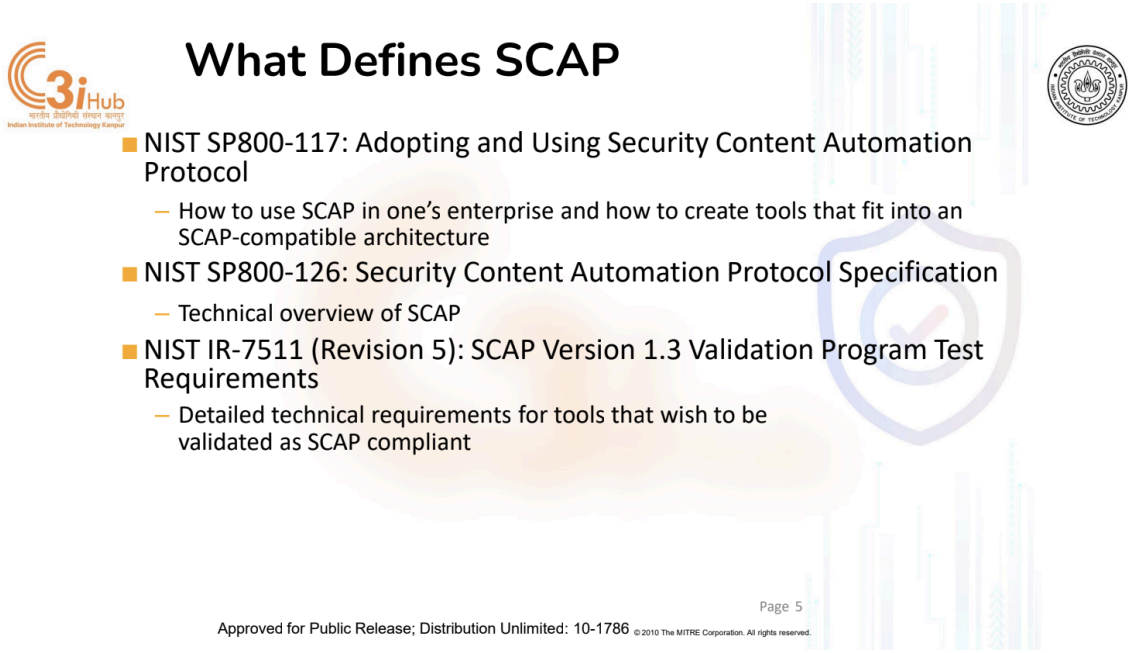
For example, let's say I am talking about a Log4j vulnerability, and you are also discussing a Log4j vulnerability, but there were multiple Log4j vulnerabilities, right? So, which one are we talking about, right? I need to know a standard way of naming vulnerabilities so that when two people discuss the same vulnerability, they use the same name or string to express it. A slightly different vulnerability will have a different name, allowing us to clearly identify which one we are referring to. Similarly, when we are providing a policy recommendation, we should do so correctly. So, whether that policy recommendation means what it truly means. So, what does it mean to conduct risk assessments and risk-driven security, or what does it mean to manage assets and similar tasks? So, without standards, we will be talking at cross purposes.

So, we will probably be saying the same thing in different languages, or we will be saying different things in the same language, which will lead to huge confusion. So, it gives you a common understanding of what we are talking about. And then it also provides a common baseline of capabilities, right? So, when we are writing a benchmark or specifying what the configuration should be or what vulnerabilities should be checked, I should have a common baseline of capabilities that I will discuss. The biggest advantage of this is that if I write a benchmark configuration for Windows 11. And then I can actually make it available to anyone and everyone in the world who uses Windows 11.

So, if I had my own proprietary language to write the baseline configuration, I would not have been able to share it, right? Because if I shared it, you would have to interpret it using your own knowledge and understanding, and you would probably interpret some of the things incorrectly. Certainly, tools cannot be interoperable; therefore, the same guidance or benchmark cannot be used by multiple different tools. But if you have a standard language, then I can write a benchmark and create a shared content repository so that everyone can use it. We will see that the US Department of Defense has a very extensive content repository that is open to everyone. You also have the CIS benchmark, which is a very good repository of benchmarks for secure configuration.

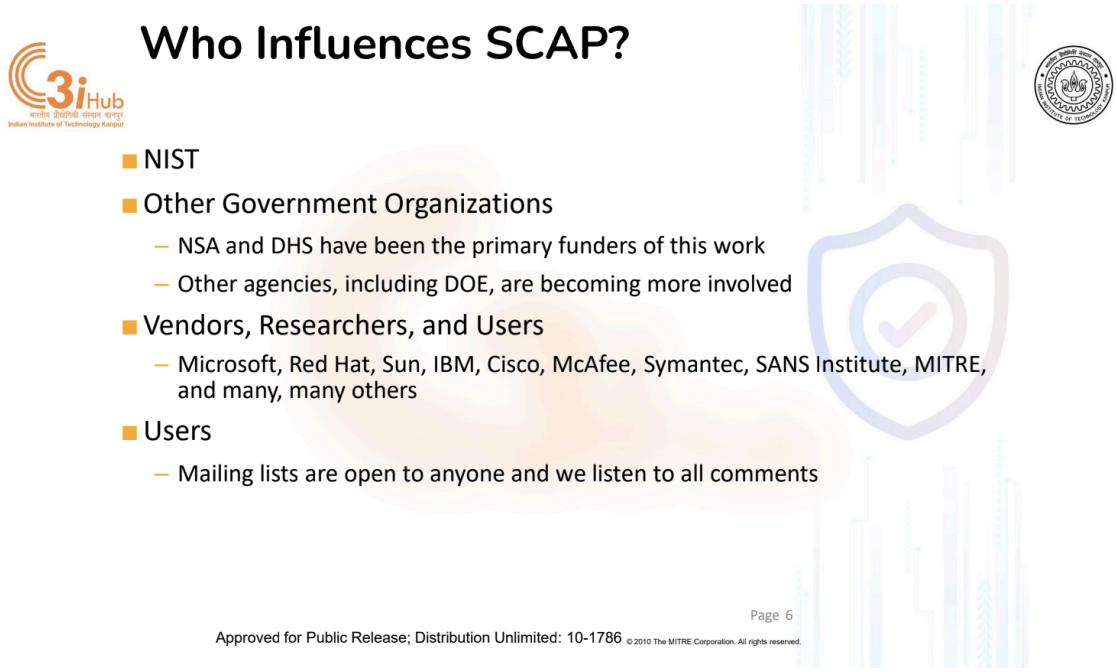
Now, one problem with all of this is that, suppose you use the US Department of Defense standard benchmark for configuration, and you find that all your organization's Windows machines need to be upgraded to comply with the US Department of Defense

benchmark. Now, how do you know that they are not actually hiding the proper benchmark and then showing you a weaker benchmark? So, you leave some of your configurations weaker, right? You do not know that, right? So, you need to have the ability to check that benchmark and customize it according to your organizational needs, right? So, you can modify the benchmark, and there is a way to do so. You can add additional properties to the benchmark to make it more customized for your organization, provided you know the standards and the language correctly. So, as I mentioned, if you have a standard, then all the content you create, including benchmarks and everything else, will be compatible with the tools. Therefore, you would have vendor neutrality and would not have to rely on just one particular vendor all the time.

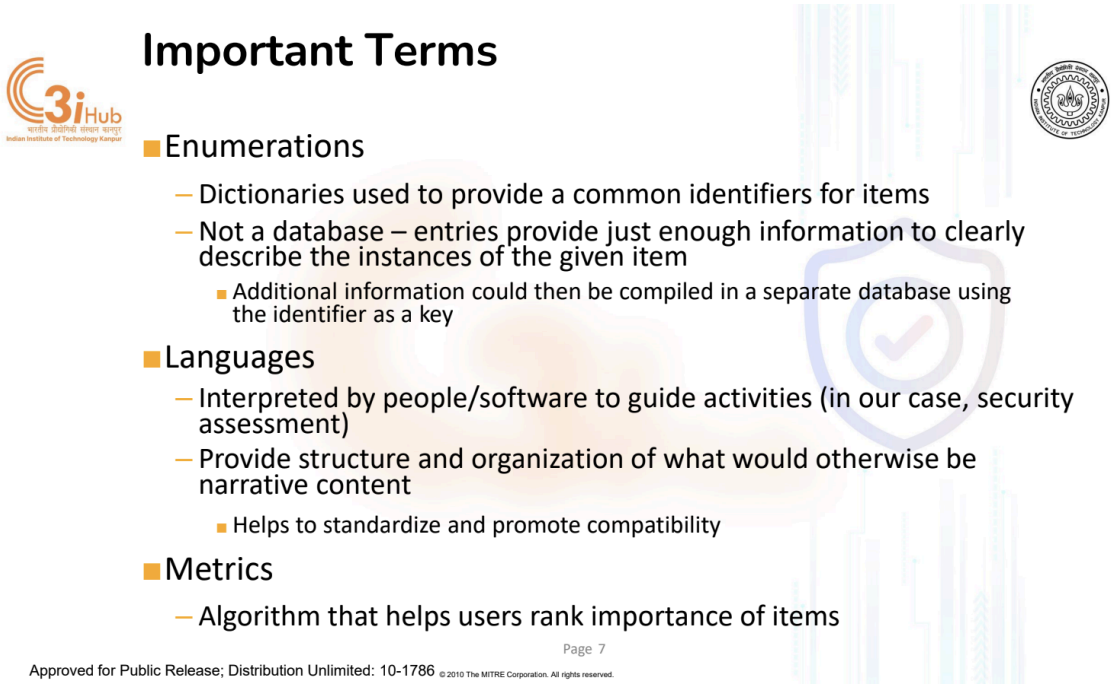


Here is the main document where they first discussed adopting and using the SCAP protocol. It explains how to use SCAP in an enterprise and how to create tools that fit into this SCAP-compatible architecture. They then published Special Publication 126, which provides a full specification of the protocol. So, this is more like a user guide, and this is actually the technical specification of the standard. And then, this 7511 is actually an upgrade, like a newer version of SCAP, which is SCAP 1.

3, that includes instructions on how to validate whether you are compatible with the requirements to be SCAP compliant. So, that is the set of documents that is relevant here. For this class, we do not need to go through those documents and so on; I just want to help you become familiar with and understand the basic ideas behind this. So, I will not ask you to write SCAP content for this class, but you will be familiar with it because it is going to be very useful moving forward if you are in the cybersecurity field. So, the NIST, or the National Institute of Standards and Technology, is obviously a major organization that spearheaded this effort.



The National Security Agency and the Department of Homeland Security were the primary funders, and the Department of Energy also serves as the regulator for energy systems. And then there was also the involvement of Microsoft, while Red Hat and Sun are no longer there. So, Sun is now part of Oracle, along with IBM, Cisco, McAfee, Symantec, the SANS Institute, MITRE, and others. And then they also had a mailing list through which users could comment as the original project was being developed. Now, there are three types of things, as I mentioned, and there are seven standards, right? So, the seven standards—not every standard is alike, right? So, for example, some of the standards are what we call "enumerations." Or their dictionaries, for example, how to uniquely name an application, how to uniquely name an operating system, or how to uniquely name a hardware system, right? So, that is basically what we call enumerations.



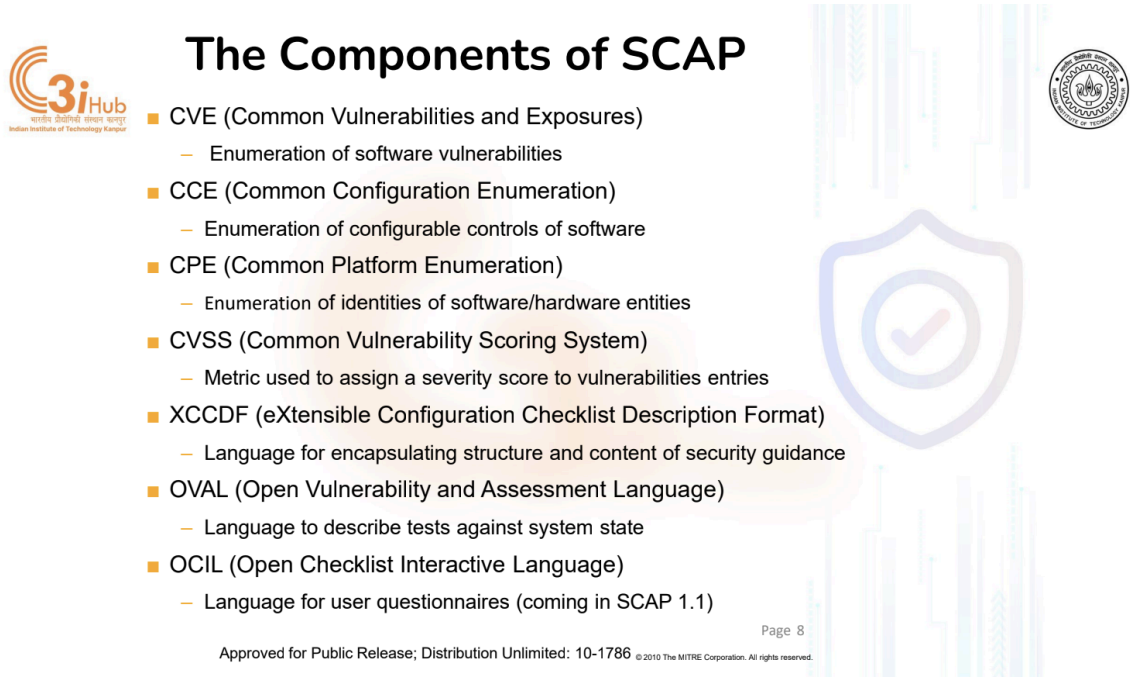
So, that is, you know, dictionaries used to provide common identifiers for items. It is not a database; it has enough information to clearly describe the instances of a given item, and you can have additional information in a separate database that is not part of the standard. Now, there are a couple of enumerative standards, right? So, how do we name things? Then there are some standard languages, such as STIX, which is a standard language. Similarly, here we have three different languages: one is called XCCDF, another is called OVAL, and the third is called OCIL.

So, there are three different languages. So, XCCDF is used to write common benchmarks for standard configurations. The oval is for writing tests for vulnerabilities, and OCIL is for interacting with the users. It is not that important; OCIL is the least important of all. So, it can be interpreted by either people or software.

To guide activities, such as how to change an unsafe configuration to a safer one, for example. It also provides a structure and organization that would otherwise be narrative content. As we saw in the case of STIX, we could have written natural language stating that this is the threat actor, this is how the threat actor is using the malware, and this malware has these indicators. We could have written that as an English narrative, but to share it and quickly ingest that information into tools, we created a language called STIX. Similarly, there are languages that make it easier to achieve compatibility.

One of the standards is a metric; it is an algorithm that helps users rank the importance of items. In our case, it will be the importance of vulnerabilities, specifically which

vulnerability is more severe than the others. So, here are the names of the standards that are part of SCAP. So, this is one you might have seen already: this is CVE, or Common Vulnerabilities and Exposures. This is basically a unique way of naming each vulnerability that is discovered around the world, right? Typically, they are named like CVE-year of discovery followed by a number.



This number, which we will discuss later, becomes unique after the numbering is done, allowing all tools to recognize the specific vulnerability associated with that number. And this is very easy because the vulnerability database is the NVD (National Vulnerability Database) maintained by MITRE. This national vulnerability database has an API. You can actually call the API and look up every vulnerability. You will know exact information, such as the severity of the vulnerability, the platforms on which this vulnerability has been found, the configuration mistake, and the configuration that might actually expose this vulnerability, and so on.

So, CVEs are a way of enumerating software vulnerabilities. Actually, it should not be just software; hardware vulnerabilities are also assigned CVEs. So, it should actually be an enumeration of vulnerabilities. The second one is the CCE, or Configuration Enumeration. So, when we talk about configuration, such as how many times a password can be incorrectly entered before your system locks out, that is a configuration, right? Some systems configure it for three incorrect attempts and then lock it.

Some configure it for 5 incorrect attempts, while others set it to allow as many attempts as the user pleases, which is very dangerous and essentially invites brute force attacks. The question is, after you lock the system due to incorrect password attempts, how long

does the lock last? Is it 10 minutes, 1 hour, or 24 hours? That is also configuration information. So, similarly, how often is the password required to be changed? Is it every 6 months, every 3 months, every 15 days, or can you keep the same password for as long as you want? So, that is a configuration of the system. Similarly, what is the length of the password? Is it 6 characters, 8 characters, or 12 characters? That is a configuration of the system. What are the different characters that are allowed, disallowed, or required in the password, which is part of the configuration information? Whether you need two-factor authentication or if single-factor authentication is sufficient, and whether you allow single-factor authentication after initially using two-factor authentication for the same browser or machine, that is configuration information.

So, there are numerous configuration settings related to passwords. So, you can think of other configuration information, such as what should be included. For example, for a browser, what would that be? Do you know how the browser chooses the encryption algorithms? So, if it allows you multiple encryption algorithms, which one should you choose? So, there are many such configurations, right? Now, if I want to tell you that you should look into securing your system. You must expire your password every six months, have a password that is 12 characters long, include a mixture of different types of characters in your password, and enable two-factor authentication. So, if I want to tell you these things correctly, I can express them in natural language.

Now you will have to go and find in your system—if you go back and try to find in your Windows—how to set these things up. You can do it; I mean, you can Google it or use ChatGPT, and you will find out how to set these things up. If you want a machine to do this right, and if you want software to go and check all these things and say okay, then.

.. So, your password never expires. You have a mistake in your configuration, and the mistake is that it contradicts my advice. My advice is that you have to expire it every 3 months, but you are either not expiring it at all or you are expiring it in 1 year. So, you are not compliant with my advice. To say all this automatically, I need two things: a way to express my advice in a machine-readable format, right? So, I should be able to write what I am advising you in a machine-readable way, and that is what I call benchmark configuration. So, in the configuration benchmark, I will write in a machine-readable way that these are the things that I think are a secure configuration.

Then you will have software like OSCAN and OSCAP, which will take the benchmark file, run it on your system, and check whether each of these cases and pieces of advice is actually being implemented in your system. If not, it will give you a list of the issues. The list of non-compliances does not mean that my advice was correct. Maybe Microsoft has shown that if I change the password too often, it is actually unsafe. This is because people

tend to take various kinds of shortcuts if they are pushed to change their passwords very frequently.

But that is not the point. When I create a benchmark, it does not mean that I know what is best. However, I have created a benchmark, and I expect you to be compliant. Whether I am right or wrong is another issue. To be right or wrong usually means it will be the CISO's team that creates the benchmark correctly. He will probably derive the benchmark from the US DoD or CIS benchmarks, but he or she, along with his or her team, will create the benchmark.

Once the benchmark is established, it is mandatory for every computer in that organization to comply with it. So, I need to be able to specify which configuration I am referring to. Am I talking about password length, password expiry time, the number of failed attempts that will lead to a lock, or how long the lock will remain in place? So, I have to uniquely name each of these configurations. So, that is what the Common Configuration Enumeration is about.

Then the next one is platform enumeration. This is the one I was talking about, and I need to know which software and exactly which version it is. Not every version of the software has the same problems or configuration settings. So, I should be uniquely able to talk about which software or hardware I am referring to. So, these three are enumeration-type standards: how to name the vulnerabilities uniquely, how to name the configurations uniquely, and how to name the platforms uniquely.

So, these are three enumerations. The next one is a metric. You might have seen that this vulnerability has a severity of 10, or that this vulnerability has a severity of 8.5 or 7.1, etc., right? So this is how to measure the severity of a vulnerability using CVSS. This is also a very important one because it allows you to decide which vulnerability to fix first.

If you have a severity 10 vulnerability and a severity 7.1 vulnerability, you will certainly fix the severity 10 vulnerability first. Now, the question is why this metric indicates that a severity 10 vulnerability is more important than a severity 7.1. We will see how this CVSS is calculated, and then it will be clear to you.

However, this CVSS has also come under criticism now. CVSS is often used to assess a vulnerability. For example, suppose you have a Windows 10 machine that has few vulnerabilities, but you never connect it to the internet. In that case, it does not matter whether those vulnerabilities are present or not. However, there may be a vulnerability related to plugging a USB into your system that has a lower vulnerability. Since you do not connect to the internet but often connect USB sticks, this vulnerability may be more

important for your device to address.

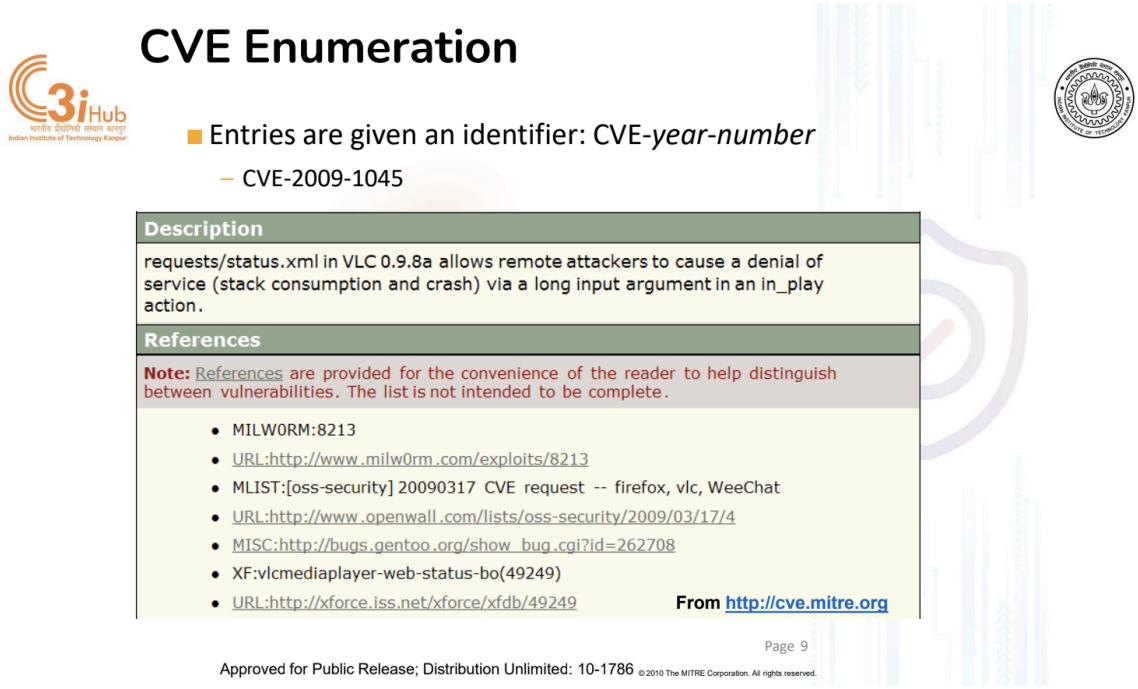
So, there is a new notion called the EPSS exploitability score, right? "See, exploitability is different from severity, right? Whether a particular..." The exploitability depends on the context in which this device or software will be used. But we will not get into that; I just wanted to say that this is the only one in the SCAP suite of standards that is a metric, rather than an enumeration or a language.

So, three enumerations, one metric, and now the languages. The most common first language is XCCDF, which stands for Extensible Configuration Checklist Description Format. So, the checklist is as I was saying: my advice is that you know about this password configuration. Have you completed this configuration? So, that is the checklist, right? How do you do the checklist? How you write the checklist is related to this language, right? So, this is the language in which you can write the configuration checklist. It is extensible in the sense that I can take your checklist and modify or extend it to suit my purposes.

So, that is why when I take the one from the Department of Defense or from CIS, I can extend it. So, that is why it is extensible; however, it is a configuration checklist description format. So, it is a language for describing benchmarks. The benchmark that is created is often called STIG, or Security Technology Implementation Guide.

So, that is one language. The second language is OVAL. It is an open vulnerability and assessment language. It basically explains how to test for specific vulnerabilities. I can write a benchmark about the vulnerabilities that I want you to test. This includes whether your system has a specific vulnerability, whether it has a Log4j vulnerability, and whether it has SSL vulnerabilities such as X, Y, or Z.

I want to check all of that. So, I will write a file in this language, and I will use a tool that accepts this file as input, then runs those tests on your system and tells me whether you have those vulnerabilities. And then there is an open checklist interactive language. This is more for user interaction, and it is not very important.

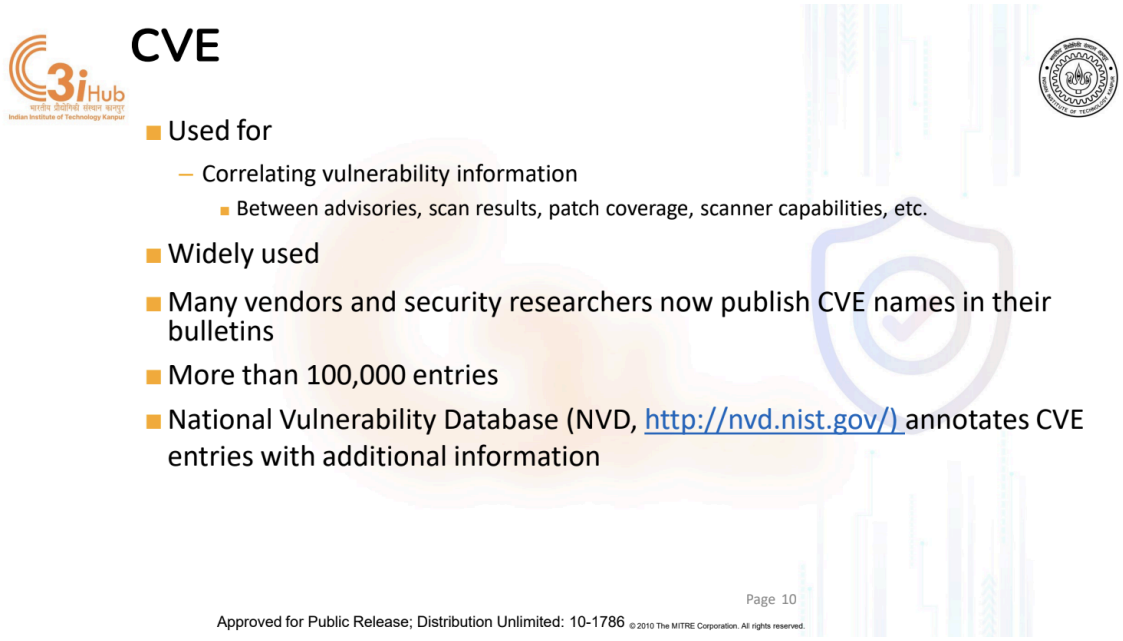


So, we will not discuss this in detail. So, all of you probably know about this CVE. CVE is the Common Vulnerability Enumeration language. So, CVE is written as CVE, followed by the year and then the number. The way the numbering is done is actually quite varied. So, earlier, everybody had to go to MITRE to report a vulnerability, and then MITRE would contact the vendor of that software to confirm the vulnerability.

Then MITRE will actually assign a number to the vulnerability and publish it in their database, so that the whole world can see it. However, now the number of vulnerabilities has increased so much that they have given a chunk of numbers a sequence, creating a kind of window of numbers for various organizations. For example, in India, CERT-IN, the Computer Emergency Response Team, can provide you with CVE numbers. Once they have filled in all the numbers given to them, they can proceed. If you discover a vulnerability, you can go to CERT-IN and say, "Okay, here is a vulnerability." They will check it, and if they are satisfied that it is indeed a security vulnerability, they will assign a number from the set of numbers provided by MITRE. Once you are satisfied, you inform MITRE that this has been assigned and provide the details of the vulnerability. Then, MITRE will include it in the database. So, that is how this CVE numbering is currently happening, right? Yes.

It's actually, I think NIST gives funding to MITRE, and MITRE maintains it. So, that is the CVE enumeration. We can see that it is used for correlating vulnerability information, and it is widely used; many vendors and security researchers publish CVE names in their bulletins. You know this is a low estimate; I am pretty sure it is much larger than this. This is probably a number of 100,000 vulnerabilities per year. The National Vulnerability

Database annotates the CVE entries with additional information, and this is what we are going to look at in the next class.



So, you can, okay? So, here it is. You can actually go to the NVD database, where you can look up vulnerabilities and search for them. So, there are APIs through which you can connect your tool to this database, and you can also look for specific vulnerabilities.