Practical Cyber Security for Cyber Security Practitioners

Prof. Sandeep Kumar Shukla

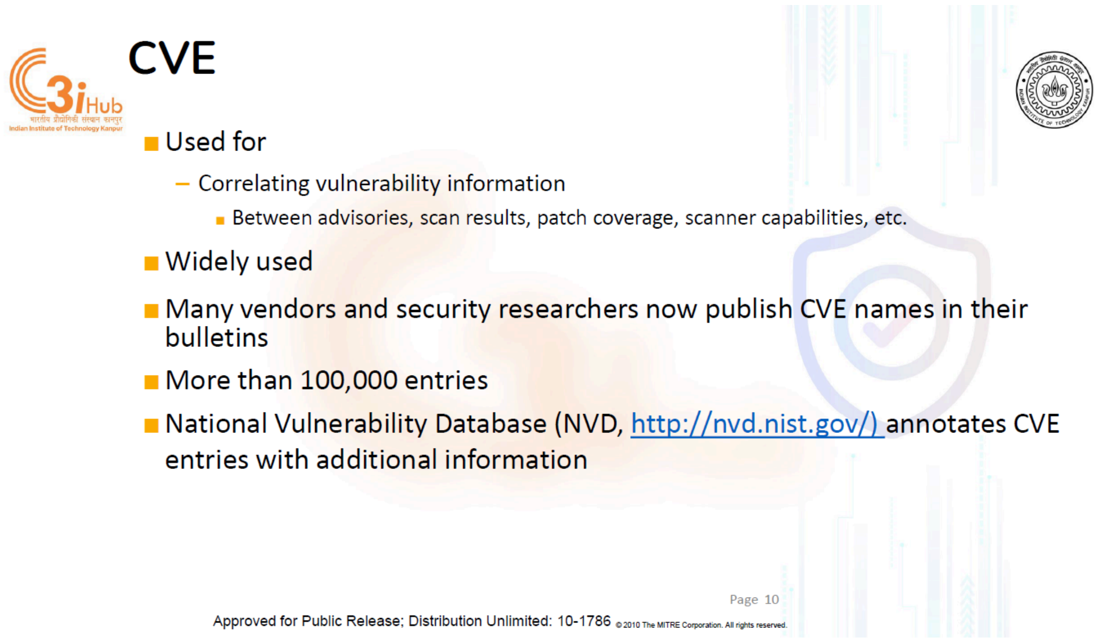
Department of Computer Science and Engineering

Indian Institute of Technology, Kanpur

Lecture 29

Deep Dive into CVE, CCE, CPE, CVSS Scoring, XCCDF, OVAL Languages - Part 1

Alright. So, we were talking about the SCAP which is we talked about various protocols that come under SCAP various standards. So, CVE is one of them CCE, CPE these are three enumeration type standards then we have languages XCCDF and oval and OCIL and then there is one metric we said is about the how to measure the severity of a vulnerability. So, that is the severity of vulnerability scoring system.



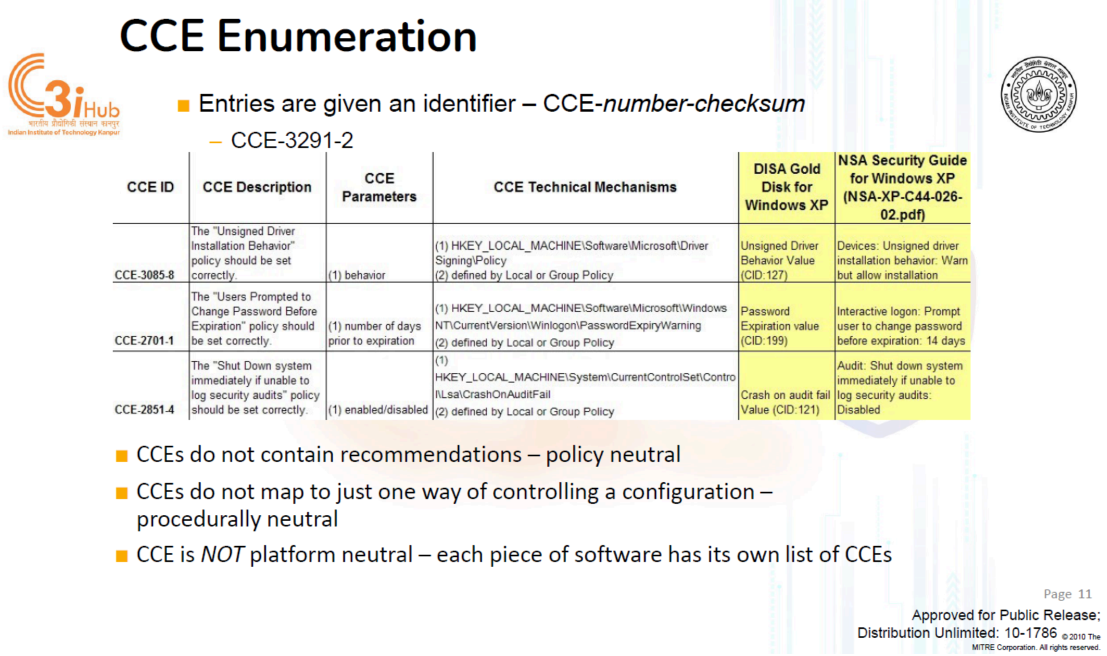
CVE

- Used for
  - Correlating vulnerability information
    - Between advisories, scan results, patch coverage, scanner capabilities, etc.
- Widely used
- Many vendors and security researchers now publish CVE names in their bulletins
- More than 100,000 entries
- National Vulnerability Database (NVD, http://nvd.nist.gov/) annotates CVE entries with additional information

Page 10

Approved for Public Release; Distribution Unlimited: 10-1786 ©2010 The MITRE Corporation. All rights reserved.

So, so it is we were discussing CVEs and we know that CVEs are actually way to uniquely identify vulnerabilities and we know that now all the CVEs that have been disclosed and and and analyzed and checked and given the right CVE score and also the vulnerabilities that are you know identified with the right platforms on which that vulnerability can be reproduced or CPEs and the configuration that is required for that vulnerability to be reproduced. So, CCEs all that stuff could be part of this database and as of yesterday, it seems that there are so many vulnerabilities being disclosed these days that NVD is 6900 backlogged on analyzing and publishing CVEs.



CCE Enumeration

- Entries are given an identifier – CCE-*number-checksum*
  - CCE-3291-2

CCE ID	CCE Description	CCE Parameters	CCE Technical Mechanisms	DISA Gold Disk for Windows XP	NSA Security Guide for Windows XP (NSA-XP-C44-026-02.pdf)
CCE-3085-8	The "Unsigned Driver Installation Behavior" policy should be set correctly.	(1) behavior	(1) HKEY_LOCAL_MACHINE\Software\Microsoft\Driver Signing\Policy (2) defined by Local or Group Policy	Unsigned Driver Behavior Value (CID:127)	Devices: Unsigned driver installation behavior: Warn but allow installation
CCE-2701-1	The "Users Prompted to Change Password Before Expiration" policy should be set correctly.	(1) number of days prior to expiration	(1) HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\PasswordExpiryWarning (2) defined by Local or Group Policy	Password Expiration value (CID:199)	Interactive logon: Prompt user to change password before expiration: 14 days
CCE-2851-4	The "Shut Down system immediately if unable to log security audits" policy should be set correctly.	(1) enabled/disabled	(1) HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail (2) defined by Local or Group Policy	Crash on audit fail Value (CID:121)	Audit: Shut down system immediately if unable to log security audits: Disabled

- CCEs do not contain recommendations – policy neutral
- CCEs do not map to just one way of controlling a configuration – procedurally neutral
- CCE is *NOT* platform neutral – each piece of software has its own list of CCEs

Page 11

Approved for Public Release;
Distribution Unlimited: 10-1786 ©2010 The MITRE Corporation. All rights reserved.

So, new CVEs are being published being reported and they have to be published, but before publishing they have to be analyzed, the vendor has to be usually they give time for vendor to actually patch the vulnerability before they are publicly made available. So, therefore, it is the situation is that the number of CVS are getting very very fast reported and therefore, it has become a problem and people are saying that there should be better way of handling this than just having a worldwide sync centralized database, but for now we can assume that see we will find the CVEs or the near future in the NVD database. Now, the next one is CCE or common configuration enumeration. So, now the question is there are many many configurations for every software right.

So, for a software like windows there would be tens of thousands of different way of configuring windows. And when we say tens of thousands of different configuration, we mean for its various components. For example, how you actually set up the password system, there would be many different things that you can do, how you set up the backup system, whether you use something like a some kind of a encryption mechanism of the disk or part of the disk whether you enable that this all kinds of configurations are possible. when you do all this different kinds of configurations So, normally in a document like XCCDF. So, we said that XCCDF is the standard for configuration benchmarking right.

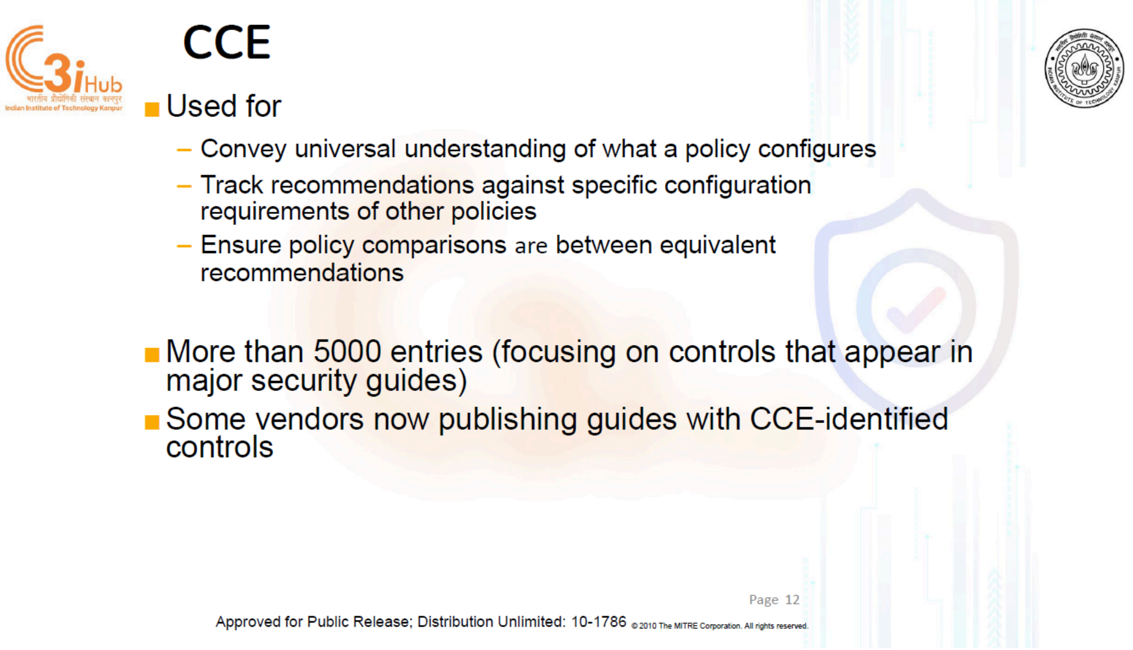
So, I want to say that this particular configuration for IIT Kanpur CSE department should always be this and this configuration should always be this right. So, I want to say such things in an XCCDF document so that it becomes a benchmark. So, then everybody can actually check that against that benchmark whether they are up to the required compliance of what IIT Kanpur CSE department requires right in terms of the configuration of all their Windows 11 machines or Red Hat so and so version machines right. So, there will be separate benchmark for each of this kind of systems and each of each kind of software. Now, how do you identify uniquely which configuration you are talking about and that is where an identification number is given through this CCE enumeration.

So, you can see here that for example, Here is a policy which has been given a unique name right. So, this policy is about what should be the behavior when an unsigned driver is being tried for installation right. So, suppose somebody is trying to install a driver software which is unsigned right. So, unsigned means that it has no digital signature on who created this driver. Now that is a pretty dangerous thing to do.

Now in a in a regular laptop when you are actually buying and using it at home you do not think about this configuration right. Most probably this configuration is permissive in your machine that is it is saying that you know any driver you want to install if you are thinking you should install it it will it will let you install or maybe it will ask you for permission right. So, one of the two will be the behavior. So, here this configuration setting is named CCE3085-8 here and it is you see that what should be the behavior. So, you can set this behavior in this registry key right.

So, it is a registry key and it is often defined by the local or group policy we are talking about windows here. And then so, but this CC does not does not say what should be the behavior right, it only says that this is something you must configure and when you configure this you can use this string to identify what you are configuring. Now, particular benchmarks for example, this is a defense benchmark, this is the national security agency benchmark. So this one saying is that you know it will tell you what should be the value for this key right and this guy will say what should be the value for this key. So in this case you see that the behavior is that you warn, but allow installation.

So a warning will be issued to the user that you are installing a driver which does not have signature, but if the user wants to go ahead it will be installed. But in a very critical system you may actually have this that deny, right? So, if it is a unsigned driver then you have to deny it, right? So, that would be part of the benchmark. So, similarly here you are saying ,you are talking about a particular configuration that users prompted to change password before expiration right. So, the number of days prior to expiration is what is the parameter to this policy setting right. So, what that means is that if your password is expiring in 2 weeks.



CCE

- Used for
  - Convey universal understanding of what a policy configures
  - Track recommendations against specific configuration requirements of other policies
  - Ensure policy comparisons are between equivalent recommendations
- More than 5000 entries (focusing on controls that appear in major security guides)
- Some vendors now publishing guides with CCE-identified controls

Page 12

Approved for Public Release; Distribution Unlimited: 10-1786 ©2010 The MITRE Corporation. All rights reserved.

If your policy is set to be like 2 weeks you will start getting the warning that your password is about to expire in 14 days, 13 days, 10 days and so on. and as soon as you fix it as you change the password the warning will go away because then the, this thing will not be triggered. So, this is the password expiration value may be set by your organization or by the standard benchmark like for example, DISA benchmark and similarly NSA if you are setting according to NSA standard then it will be 14 days. Similarly, shut down the system immediately if unable to log security audits, right. If you are not able to log, sometimes you see your log is not being stored or something for some reason, then what should be the behavior? One is to actually shut down the system, another is to actually continue, right.

So, this is enabled or disabled would be the parameters for this particular setting. And, then it would be also can be done through the registry keys and you see that in case of NSA it says shut down the system immediately if unable to lock security audits this has been disabled right. So, in this case this is disabled means that you will continue to operate even if the log is not being stored. So, this is a choice. So, the main point here is that CCE does not tell you what should be the policy, it only gives you a handle on the setting that you have to decide on.

So, how many days before password expiration you will get get the warnings or whether you are allowing unsigned drivers to be installed or whether you are going to continue when there is no log being stored etcetera. These are the these are what the part of the

CCE. CCE does not say that allow or disallow unsigned installation or make it 15 days or 30 days that is not part of CCE. CCE just gives a name to this policy settings. What should be the value of those policy settings of the configuration settings is based on the benchmark or the or the what we call STIG or security technology implementation guide that will tell you what should be the setting right.

And you could also write a manual in PDF or other form to tell what the settings should be, but then somebody has to go and sit down and actually do this all this setting and there would be thousands of settings it will be very difficult. So, that is where A tool like WasteCap will allow you to automatically apply those settings based on the automation that you do through the benchmark like stick files or security technology implementation guides. So, CCEs do not actually have recommendations. So, they are policy neutral. They do not decide what should be the value of this settings, but they only identify the settings.

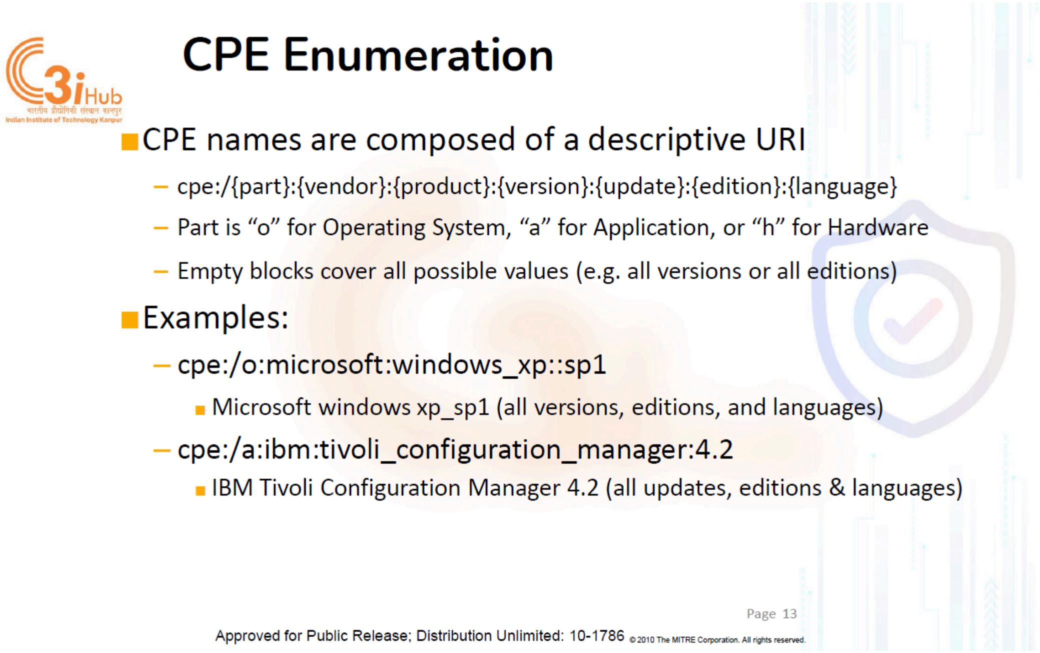
And CCEs do not map to just one way of controlling a configuration, it should be procedurally neutral. Whether you want to do this through a registry key or you some other mechanism by running some kind of a script or something that is up to you. And it is it is not platform neutral of course, because different platforms have different configurations. So, this the configuration names these names are specific to windows. for Linux and all there will be other CCE numbering.

So, it gives you a way to understand what the policy is supposed to configure. Yes. So, it keeps it keeps getting added. So, today what you have in CCE One month later, there will be some new ones added for new software and all that stuff, right? .

I see. Okay. So, there is a NIST page on CCE and it has a list of the CCE numbers and what they mean and all that stuff. Is that what you are asking? . No, no. So, that is an example, right? So this what you are seeing as technical mechanism is not part of CCE. This is just this table is not part of CCE.

This table is just to show you like see this is basically these 3 columns are basically specific to CCE, right? that what is the id, what is what it is, what parameter, what configuration we are talking about, what are the possible values of the what are the things that you need to set right. The rest of it is just to illustrate what this could be or how to set it or when it is set what would be then example setting things like that ok. So, you can you can actually search for it and go to the website and look at more examples of these right. So, CPE is another way of is a way of uniquely identifying the platforms Platforms regarding you know, what you are talking about, right? whenever you are talking about some CVEs for example, when you are saying that there is a vulnerability you have to uniquely identify which platforms. For example, you can have a vulnerability that only applies to certain version of Windows 11 not every version of Windows 11.

So, you have to specifically say what versions of Windows 11 are affected by this CVE and which versions are not affected by the CVE. So, therefore, this CPE is a common platform enumeration. the way it is done is you basically use the term CPE and then sometimes you actually also see version because CPE has gone through versioning. So, you can say CPE 2.1 or something then you say what is the part that you are talking about whether it is a operating system, whether it is a hardware, whether it is an application etcetera then you have to say what vendor it is from.



CPE Enumeration

- CPE names are composed of a descriptive URI
  - cpe:/{part}:{vendor}:{product}:{version}:{update}:{edition}:{language}
  - Part is "o" for Operating System, "a" for Application, or "h" for Hardware
  - Empty blocks cover all possible values (e.g. all versions or all editions)
- Examples:
  - cpe:/o:microsoft:windows_xp::sp1
    - Microsoft windows xp_sp1 (all versions, editions, and languages)
  - cpe:/a:ibm:tivoli_configuration_manager:4.2
    - IBM Tivoli Configuration Manager 4.2 (all updates, editions & languages)

Page 13

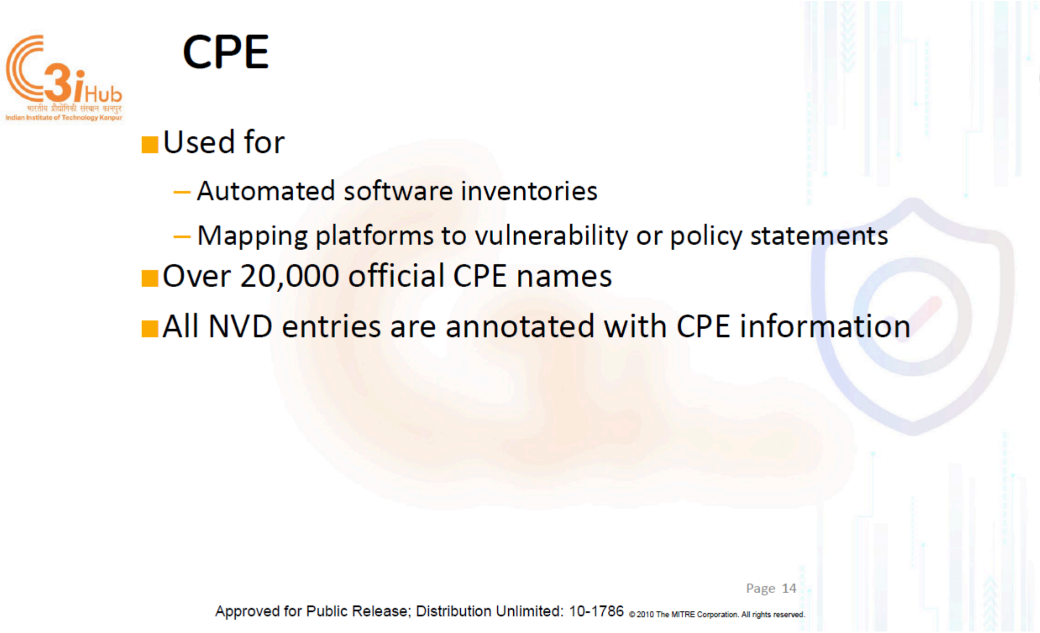
Approved for Public Release; Distribution Unlimited: 10-1786 ©2010 The MITRE Corporation. All rights reserved.

Then you have to say what product it is, then you have to say what version you are talking about and which update of that version because each version may have multiple different updates, it can may also have multiple different editions. and it may be actually a language version. So, for example, you can have a Chinese version of Windows versus a German version of Windows versus a US version of Windows and stuff like that. And then for the part the choices are O for operating system, A for application and H for hardware. and if you leave any of these things empty that means basically that is the wild card that means that any like if I keep the version field empty then the then I am talking about all versions of that particular vendor produced product right.

So, examples So, here is let us say Windows XP service pack 1 right. So, we are talking about part is operating system, the vendor is Microsoft, the product is Windows XP. then version you are skipping the version. So, there is an empty block they are skipping the version and then update.

So, this is a service pack 1. So, those who have I guess most of you have not seen Windows XP. So, they there was service pack 1 at some point and then they had service pack 2. So, that added features and fixed certain things and all that stuff. So, this is service pack 1 and rest is empty which means that it is true for all editions and languages and all versions right. So, basically we are talking about Windows XP service pack 1 irrespective of what version, what language, what edition etcetera.

Similarly, here is an application. So, here is A for application, the vendor is IBM and then this is Tivoli configuration manager. and this is the version is 4.2 and all updates additions and languages we are talking about. So, if you see this as a CPE for a particular CVE that means that CVE applies to all the versions and all the additions and languages if they exist if a different language versions exist. So it is used for automated software inventories.



CPE

- Used for
  - Automated software inventories
  - Mapping platforms to vulnerability or policy statements
- Over 20,000 official CPE names
- All NVD entries are annotated with CPE information

Page 14

Approved for Public Release; Distribution Unlimited: 10-1786 ©2010 The MITRE Corporation. All rights reserved.

So when you actually do software inventory, you do various kinds of asset inventory, you can actually use CPE for uniquely identify them. Also you can map the platforms to vulnerability or policy statement. So you can talk about both vulnerable CVEs associated with CPEs. Also when you do configurations CCE, you can say you know which version we are talking about. there are many CPE names there is also a database there and also all the NVD entries are annotated with the CPE information.

So, this is now CCE version 5 going on. So, all platform groups combine file. This is an excel file. so you can see CCE number and name you know all these things. Now, this also tells you where this which security implementation guide like talks about this configurations like for example, PCI DSS which is the which is the security standard. So, this is the like for example, this is showing in PCI DSS which is a standard for credit card card related information anybody who is taking credit card for payment on their e-commerce interface or whatever they have to comply with PCI DSS standard right.

So, this says what are the settings that are associated with the PCI DSS not all of them are related to PCI DSS. Similarly, it says you know what HIPAA, HIPAA is the health related information privacy protection standard in the US. So, HIPAA talks about certain configurations, COBIT talks about certain configuration, ISO 2700, let us see this is 27002 talks about this configurations, this is 62443. So, it is a very comprehensive mapping of all the different configurations and whose which configurations must be set to specific value according when you are when you are trying to comply with one of these. So, if you are trying to comply into with HIPAA then these CCEs are relevant to you and there is what setting they should be should be is defined here and similarly for PCI DSS or 2700 206 2443 etcetera.

So, this is a very relevant document for cyber security teams that are trying to be compliant with certain standards or required to be compliant with certain standards. So, now if we go to the CPE list. So, let us see, you know.. now this you can see also that there are different tabs for different types of software for which this CCEs are defined right.

So, you can see that you know therefore, you know Apache and Firefox and HPUX and various versions of browsers, macOS, you know various versions of macOS. So, they all these different software operating system and you know applications they have they have certain designated configuration parameters that have been identified here. And based on this configuration parameters if you are going to be compliant with certain standard then you have to set those configuration parameters correctly. So, now if we look at the CPEs. So, this is CPE dictionary as you can see that even in a zipped format compressed format it is about 20 meg.

So, it is has a lot of different CPEs lot of different platforms enumerated in this and if you go to the NVD database. and let us say look at some vulnerabilities, let us let us search for vulnerabilities. So, let us look at this CISA known exploited vulnerabilities and then let us choose. So, we just randomly pick one. So, this is let us look at a critical vulnerability right.

So, this is a critical vulnerability 10 CVE score, this is a brand new vulnerability and it is coming from the cyber information security agency ,sorry critical information security agency CISA and you see that we will talk about how the CVEs is computed soon, but let us see what are the CPEs. you see that this is Palo Alto network right a particular OS pan OS and particular version, but all editions and everything is included right. So, this particular and then version 11 also. So, this particular CVE of severity 10 is going to be, is affecting all these different platforms. So, if your asset inventory had your Palo Alto firewall, identified with this CPE enumeration numbers, then you can have an automatic script tell you as soon as this vulnerability is discovered and announced, it can tell you on your dashboard that your this and this software system has now a criticality 10 vulnerability that you need to patch right.

It also tells you it is in the CISA's known exploited vulnerability catalog. You can find more information like the from the vendor, you can find some information about exploits and you can also find third party advisory on this. Now, coming back to.. it is undergoing some reanalysis and seems like they are not very sure about this vulnerability something has probably come up, but right now as far as they are concerned it is criticality 10 and this is the what we call the vector of attack and we will come to this vector of attack very shortly.

So, that is the that is how the CPE is related to this vulnerability CVEs. Now, there is also another thing this is called a common weakness enumeration right. So, the common weakness enumeration is not associated with a product. So, CVE is always associated with a particular product and its particular version etcetera right. Common weakness enumeration is another product from MITRE which basically tells you what are the common weaknesses in software.

So, for example, improper neutralization of special elements used in a command. So, it is a command injection, right. So, it is CWE 77. Now, you are leaving the we are going into the MITRE, MITRE website where there is a common weakness enumeration. So, it tells you more about the types of mistakes in software that you keep right.

So, this means that so, command injection happens when you actually type in something through maybe a web interface or through some other interface or through an API for example, for an API call where you pass a string right and the string is eventually used as part of a command on the shell right or on the system. Now, if you do that now the software on the back end when the user uses a command, the user uses a string either in the API call or in the in a form in case of a web application etc that goes to the backend and in the backend that particular string is split or something and used as a part of a command, right. Now if you do not check if the string that is being written by the user as it goes to the backend, in the backend if you do not check that this is not a malicious string, if I use it as a part of a command it can actually backfire on me. If you do not check that then we say that we did not do the input sanitization, we did not sanitize the input before actually using it in the in the back end. So, CWE 77 talks about that kind of mistake in the back end program.

So, it is a improper neutralization of special elements such as you know scripting command for example used in a command. So, so this is a command injection vulnerability. So, similarly you can have a whole list of CWEs. So, top 25 for example, will tell you what are the different mostly, most common mistakes in software that people make like out of bounds, right? So, this is common like if you are doing writing in a loop and you have an array if you do not or if you are not careful you might write beyond the boundary of the array and that could corrupt your stack. improper neutralization of input during web page generation, improper neutralization of special elements used in an SQL command.

So, SQL injection use after free that is for pointers, improper neutralization of special elements in an waste command, this is a waste command injection, improper input validation out of bounds read. So, these are the 9 sorry 25 most common software weaknesses enumerated. So, CWD CWE is not part of the standard it is just a is just a list that has been researched and created by the MITRE corporations to actually and it is very commonly used and as you can see that even in the NVD database corresponding to a CVE they mention what CWE it is right. So, what weakness was actually there causing that particular vulnerability, but CWE to remember is that CWE is not specific to a particular software or platform or application etcetera.

So, now we go back to the SCAP and so we talked about CPE. So, now let us see. CVSS, we talked about three enumerative standards right. So, we talked about CVEs for vulnerability enumeration, we talked about platform enumeration CPEs and we talked about CCEs or configuration enumeration. Now, the next standard we are talking about is how to measure the vulnerabilities danger, danger of a particular vulnerability. So, I have given a vulnerability I have to score and say this vulnerability is of low severity or medium severity or high severity that is what I want to do.

Now to do this there are three different parts of this one is called the base, another is called temporal, another is called environmental. So, it is better explained when I actually do it right. So, I go to the CVSS website. So, I go to a CVSS calculator. So, CVSS calculation also has gone through number of different versions.

So, right now we are in version 3. So, you see that when I get a vulnerability let us say I find a vulnerability in a software. So, how do I go about assigning severity to it right. So, I am going to see whether this first thing that I will check is whether this vulnerability can be exploited from the internet if my software is somehow accessible from the internet or is this software is this vulnerability only exploitable if you are in the adjacent network that is you are in a network segment that is connected to the next network segment where the software is being run or is do you have to be local which means you have to be in the same network as the as the software or do you have to be physically on the machine on which the software resides. now which one is more dangerous,right? If the vulnerability can be exploited from the network itself that is much more dangerous because lot more people have now ability to exploit it. If it is not accessible from the internet or cannot be exploited from the internet, but can be exploited from the adjacent network, then also the number of people who can potentially exploit it is higher.

If you have to be on the local network, then the number of people is still reduced. And if you have to be physically on the machine to exploit the vulnerability, then that will give you a much lesser opportunity to exploit. So, let us for just example I say that my vulnerability is exploitable from adjacent network not from the internet and if it can be done from the adjacent network of course, it can be done from the local and physical access right. Now I want to also see whether the complexity of the exploit. To exploit it do I have to know a lot of technology, lot of techniques and so on or technology complexity or is it like very simple right.

So, I can just type in something on the network and on the internet some interface and it will be done. So, let us assume that it is of high complexity. High complexity means this vulnerability is difficult to exploit. So, its severity will be again reduced right.

Then I am saying privilege required. Do you need to have be a high privileged user to exploit the vulnerability or it you can be a low privileged user or you can be no privilege that is you may be from the network from the internet right. You do not have any account on the system or anything and you can still exploit it. If that is the case then that is a big problem right. So, let us put this in none that I do not need any special privilege. And then I have to say whether it requires user interaction or it does not require user interaction and by user interaction we mean victims interaction right.

So, 0 click is it a 0 click vulnerability or not. So, let us say it does not require user interaction. Now there is something called scope. Scope means scope has two choices unchanged or changed. So, unchanged means that if the vulnerability is in machine A will the effect be on other machines or will it be only on that machine if you exploit it.

So, let us say the scope is changed that is I can affect others. And then I am asking whether if it is exploited will the effect on confidentiality be high low or mid or none. Let us say the confidentiality effect is low and integrity impact is high and availability impact is low right. So, let us see what is the score.

So, with this I am getting a score of CVSS base score of 7.5. and the what we call the vector the see the vector basically says attack vector is adjacent network AC that is the attack complexity is high, the peer privilege required is n that is none, the user interaction is none and then scope is changed. and the confidentiality impact is low, integrity impact is high and the availability impact is low right. Now, let us do some changes just to for illustration let us say I say it requires high privilege. see the severity went down to 6.8 because if the attacker needs high privilege that reduces the possibility that an attacker any run you know Tom Dick and Harry can do it.

Now let us say I say that you have to have physical access to the machine to exploit this. So, as soon as I do that it becomes 6.

1. Now if I say that confidential impact is none it becomes 5.5. If I now say availability impact is none it becomes 4.7 and then if I say user interaction is required then it is 4.6. So, you get the idea that more difficult it is to exploit by requiring physical access to the machine for the user to have high privilege for the attacker to have high privilege, the attacker the attack complexity being high, user interaction is required, scope gets you know if the scope is if I make scope unchanged then it is even less 3.

8 right. So, if the scope is unchanged that is only affects the machine on which you are doing the attack all that then you can actually the severity becomes low,now let us maximize the let us maximize the score at a complexity is low privilege required is none user interaction is none and then scope is changed the impact is high and high. So, now, I have a severity 10 vulnerability right. So, this is the base score. See the base code basically says that I do not know anything about what is happening in the world or I do not know much about your own systems configuration or your network configuration and so on.

I am just thinking of the vulnerability in the worst or worst case or best case. Now temporal score basically says that over time a particular vulnerability may be exploited more and more because the exploit becomes easily available right. Anybody can write an exploit or anybody can buy the exploit or anybody can download an exploit. So, let us say exploit code maturity, let us say the exploit code maturity becomes high and remediation level that is the fix from the vendor has not come in yet let us say fix is unavailable right. And the confidence of the report that whoever has reported this vulnerability they are not very confident that this is indeed a vulnerability.

So, let us say unknown right. So, now you see that I have because of that unknown my thing has reduced to 9.2 right. Now if I may if I make it confirmed then it will go back to 10. So, now if I say official fix is available, but I say that temporary fix then it has become 9.

6. Now if I have if I say that the exploit code is just unproven exploit exists then it becomes 8.8. So, you see that this now if taking into account the possibility that yes there is a vulnerability that can be easily exploited from the network and does not require the user interaction, it does not require the attacker to have high privilege, scope is you know broader than just that machine. So, I see a high severity, but then I see that actually exploits are not that commonly available and I am seeing that the, but on the other hand the fix that is available is also not a proper fix it is just a temporary fix and the, but the report is saying that it is confirmed then the score becomes small lower.

Now, if I say that the fix is not available, fix is not available then it becomes 9.1, but if I say official fix is available then it becomes 8.7 right. So, official fix is available means that anybody who wants to protect themselves can protect themselves right. So, temporal factors will reduce or increase the severity usually it will reduce the severity except that when maybe the official fix is not available etcetera.

Now the third part is the environmental score. The environmental score means that within your organizational environment what is the importance of this vulnerability and what is the how they attack vector will be affected. I may have a very good perimeter security. So, even though my original vulnerability could be accessed from the network, but in my organization it you have to be in the local network to exploit this vulnerability, this is possible. Similarly, at a complexity it may be that a complexity may be low, but in my organization if you want to access that particular software you have to go through a number of hoops. So, at a complexity also becomes high and then in my organization you do not access that particular part of the network unless you are a high privilege user.

And then you in my organization you may require no interaction, but the scope will be unchanged because I have very micro segmentation of my network. So, if you do exploit me you do not spread. And then here I basically say the data that I have in that machine on which this vulnerability exists maybe these things are low right see the importance of that machine is not that high and the confidentiality requirements are also low low and low. So, now let us look at what is the see now you see that just because of environmental a 10 severity base score has now come down to 2.

1. So, environmental factors will also be very important because of this, because your environment may be very well designed and well secured, it has been properly

segmented, it has proper use of firewall rules and maybe intrusion detection system and high level of user authenticate two factor authentication and only high privilege users have access to then segment where this particular software is located and all that mitigating factors could actually bring down your severity tremendously right. So, that is what I wanted to say about the CVSS. So, ok. So, let us next class we will talk about the rest of this Scap ok.

I think usually that is the case. However, in some cases I think in case of temporal if the maturity of the exploit goes high yeah. So, I think still it will be less than the base core right. So, base the base core gives you an upper bound and then you have to play within that. So, base code assumes that you have no base code assumes that you have exploits available, access available all that stuff right. So, then you have this environmental and temporal gives you more information about the context.