Practical Cyber Security for Cyber Security Practitioners

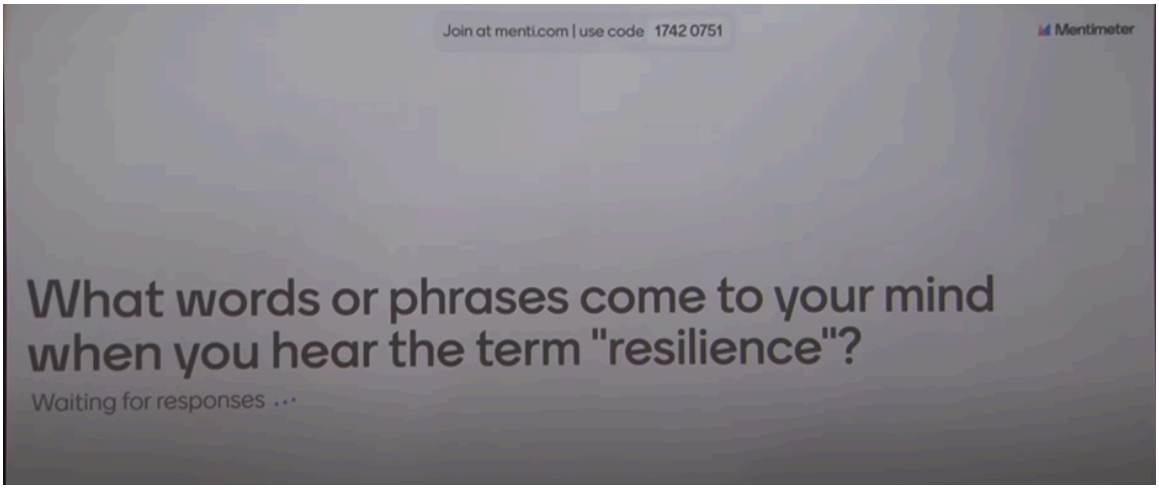
Prof. Sandeep Kumar Shukla

Department of Computer Science and Engineering

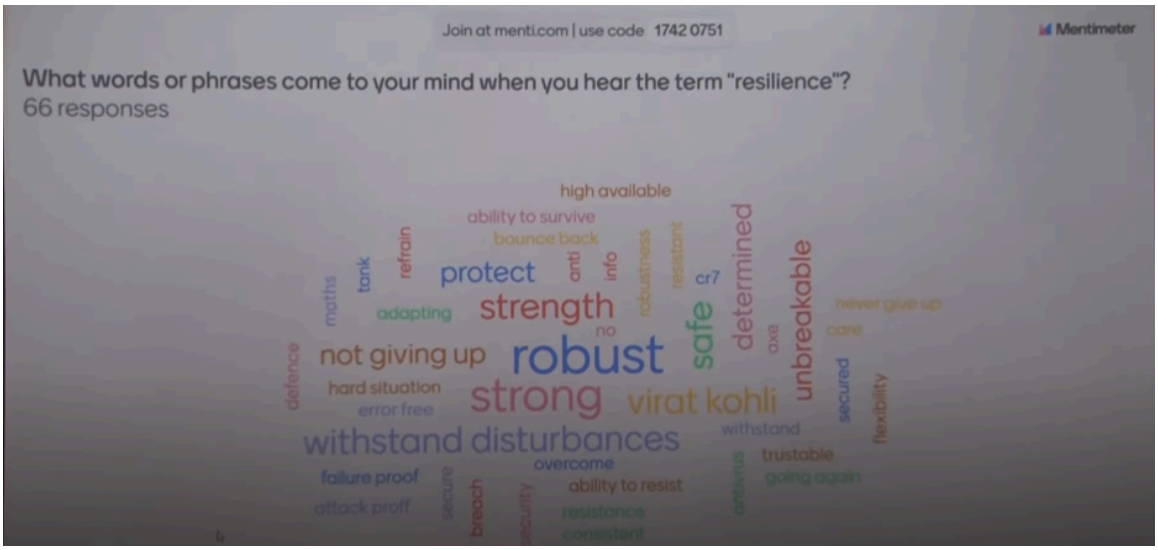
Indian Institute of Technology, Kanpur

Lecture 02

Introduction to Cyber Kill Chains - Lockheed Martin Kill Chain



So, tell us what you think. What comes to your mind when you think of 'Resilience'?



Okay, it seems like 'robust' has been conflated with 'resilience.' But in system design terms, 'robust' and 'resilient' are very different. For those of you who have studied control

theory, you must have studied robust control, but you do not study resilient control, right? For example, an oak tree is robust, meaning you cannot easily uproot it. However, if you look at the mangroves in the Sundarbans, when a hurricane comes, the mangrove actually lowers its branches to survive the storm. It is not robust, but it is resilient. You can easily uproot a mangrove plant, but in a catastrophic event or unprecedented challenge, the mangrove performs better than an oak tree. Oak trees can be uprooted by a storm of high potential. So, resilience is the ability to withstand unlikely or unprecedented disturbances and recover quickly.

Robust means that when you design something, it is considered robust when you account for a number of different scenarios that may occur to that system. You have to design the system so that, irrespective of all those scenarios—whether they are normal, nominal, or expected deviations from the original system's inputs—the system can still function. We say it's 'robust' if it can handle noise, spurious inputs, or certain types of disturbances that are accounted for in the design itself.

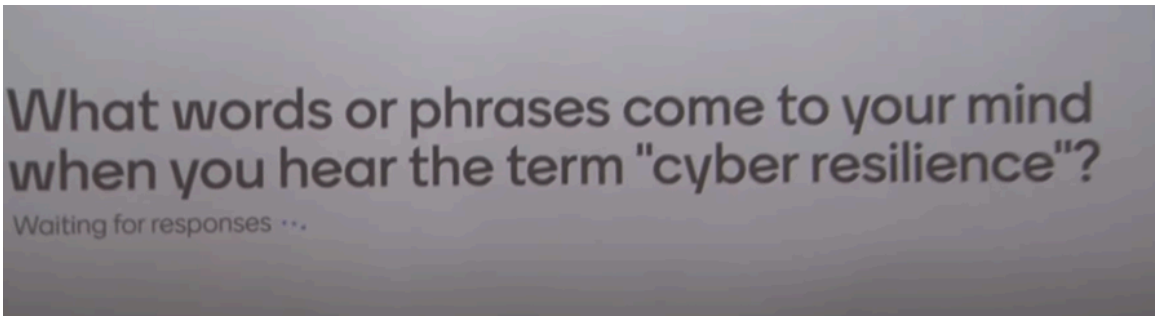
We create a product or system to be 'robust' against certain already envisioned disturbances or excursions from nominal behavior. That is what we call 'robust.' 'Resilience' is different; it applies when we cannot imagine all possible scenarios, especially very catastrophic ones. We haven't specifically designed our system to withstand those, but when they happen, the system should have enough ability to withstand them—perhaps with reduced functionality—but without completely collapsing. And then when it comes back up, it comes back up like an elastic spring that comes back up by itself by gradually recovering its abilities or its functionalities. So, when you design a city, for example, we try to make it robust against, say, rain. So, we have rainwater drainage system. We make it robust towards traffic, certain kind of traffic conditions. We create diversions and so on.

But most cities today are not resilient. For example, there was a flood in Chennai a couple of weeks ago that tested the city's resilience, and it turned out that the resilience was lacking. Everything came to a standstill; things didn't work until the disturbance was withdrawn by nature. This highlights the difference between resilience and robustness. When designing an enterprise IT architecture or enterprise operational technology architecture, we aim to make the design robust against scenarios we can anticipate.

However, if a crisis occurs and the system is not designed to be resilient, it collapses. For instance, if there is a ransomware attack on the IIT system, and systems like Pingala and OARS are completely encrypted with no backups or manual processes to manage student

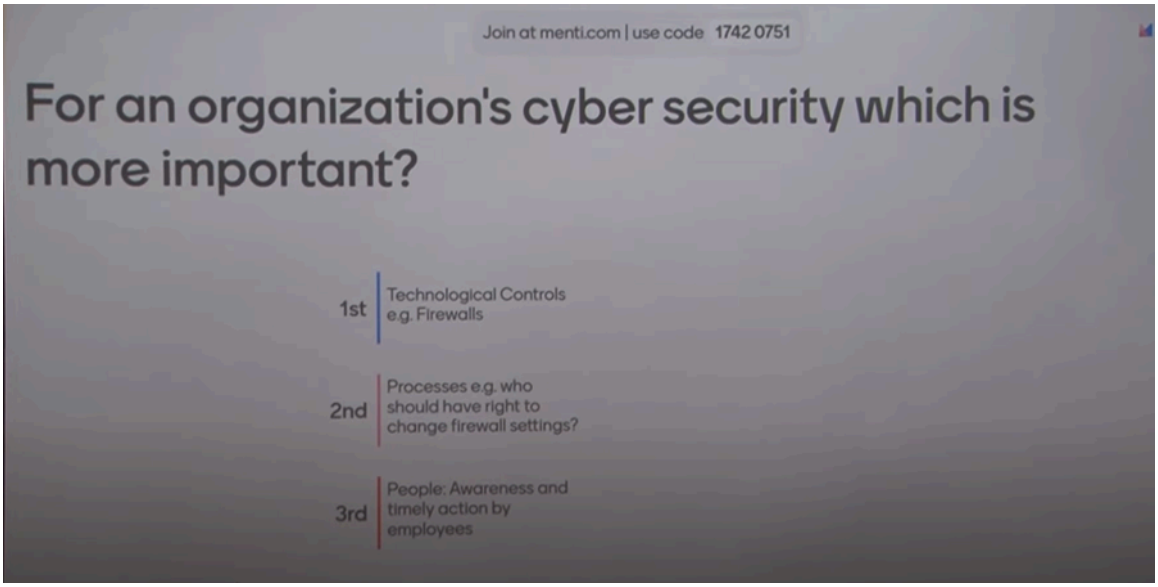
registration or salary processing, we would say that such a system is not resilient. The inability to operate with reduced functionality results in a complete standstill, necessitating crisis management to eventually restore all functionalities and recover data. This process is manual, strenuous, and challenging for all stakeholders, including students and employees. This is what differentiates resilient design from robust design.

Robust design has been practiced for a long time, focusing on responding to expected disturbances. In contrast, resilience involves the ability to operate with minimal functionality during a disturbance and to recover quickly afterward. The speed of recovery serves as a measure of resilience.



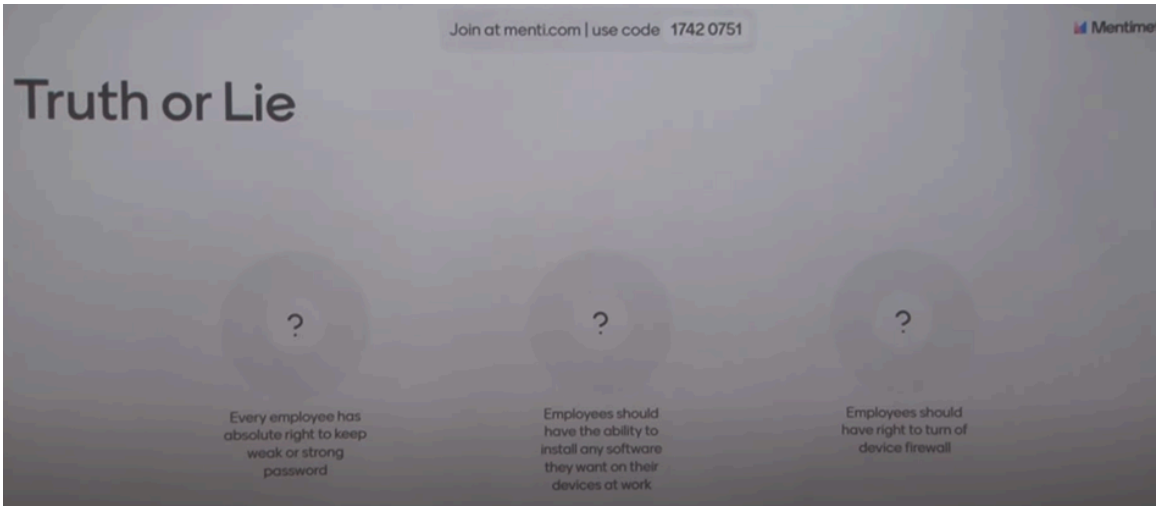
Now, considering resilience in regular system design, such as in cities, products, or IT systems, what do you think about cyber resilience? Although I've already touched on cyber resilience, it's still worth discussing. For instance, a firewall is not considered a part of resilience.

A firewall is part of regular protection, right? Okay, so I think you understand the concept of resilience. We'll discuss resilience, resilient system design, and other related topics, particularly in cybersecurity, under the term 'cyber resilience.'



Let's move on to the next question. This is something that you should better understand by the end of the course, but I want to gauge the baseline now. Consider the following: you need to decide which factor you think is more important and which is less important. It seems that most of you believe that people awareness and their timely actions are the most important aspects of an organization's security. Technology controls and processes are also crucial, but they are considered equally important alongside processes that need to be followed. So, cybersecurity involves people, processes, and technology. It's not just about technology like firewalls, antiviruses, endpoint security, network monitoring, or security operation centers. While those are significant, people play the most critical role.

We'll delve deeper into the role of people in cybersecurity later.



For now, let's move on to the next topic. I have three statements, and you need to decide whether each one is true or false. The first statement is, 'Every employee has the absolute right to keep a weak or strong password.' The second is, 'Employees should have the ability to install any software they want on their work devices.' The third is, 'Employees should have the right to turn off the device firewall.'

It seems about 40% think that employees have the right to choose their password strength, which reflects how IIT has operated so far. Academics often feel they have absolute rights, but in today's world, for a resilient system, that cannot be allowed. It should be part of your policy to specify what kind of password is allowed. You cannot allow a weak password policy.

You must have a password that meets certain criteria, such as a minimum length, including special characters, numerical characters, at least one capital letter, etc. However, this doesn't necessarily ensure complete protection. For example, someone could use a simple pattern like a name followed by '123' and a special character, which would meet these conditions but still be weak. Many organizations use a backend password strength checker that evaluates the strength of the chosen password. If the password is deemed weak, it is not accepted. They also do not accept passwords that have been used in the last three password cycles. Employees or students do not have the absolute right to choose their passwords freely because they are not only putting themselves at risk but also the entire community.

This is a matter of civic responsibility. Just as the constitution grants certain fundamental rights along with responsibilities, the cybersecurity policy of an organization must ensure a robust password policy that applies to everyone. For example, many years ago, when I was a graduate student, around 1994, my system administrators cracked my password on

a VAX mainframe system. I was called into the IT office and required to take a course before my account was restored. My password was 'Bhaiya,' which was easily cracked using a dictionary attack. Dictionary attacks can match hashed passwords with hashed dictionary words, even from different languages. At that time, password salting was not common, making it easier for such attacks to succeed.

The IT team detected that my password hash matched a known word's hash and identified 'Bhaiya' as my password. They called me in and held me accountable. Such enforcement mechanisms are necessary to ensure strong passwords. Moreover, two-factor authentication can enhance security, but we'll discuss that later.

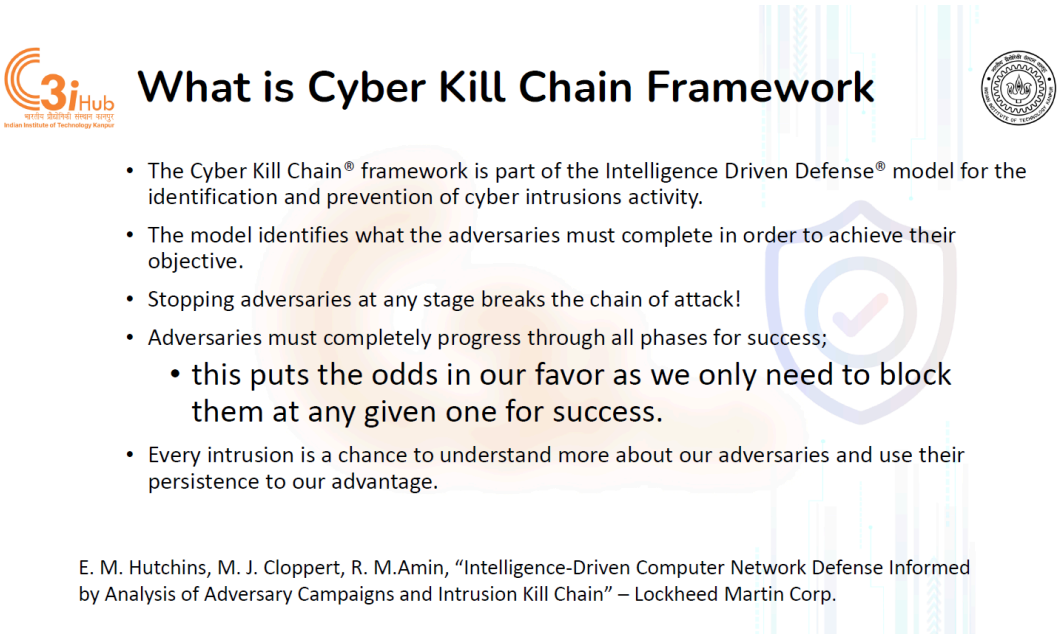
Another issue is software installation. When I worked for companies like Intel and Verizon, I couldn't install any software on my desktop Windows machine without submitting a formal request to the IT group. The IT group would review and approve the software for security reasons before installation. This doesn't guarantee that all threats will be detected, but it helps mitigate known risks. The IT group had administrative control over my machine, a common practice in organizations concerned about cybersecurity and data loss.

Lastly, the question of whether employees should have the right to turn off the device firewall arises. This, like the other aspects, is a matter of policy and security control within the organization.

Many people install random software, and when the firewall creates problems, they turn it off because they have administrative access to the machine. This makes their machine a potential launching pad for malware, which can spread through the network, make lateral movements, and attack other machines. It's not just an individual matter; it's a community concern.

Every machine should have visibility to the Cyber Operations Center team. This includes monitoring what is happening on the machine—except for personal data—such as any program that is running, any network access made to or from the machine, and any open ports. All this information should be visible in real-time on the cybersecurity operations screens so that immediate action can be taken if something unusual happens. The idea of absolute individualistic rights to do whatever one pleases is no longer valid in today's cyber world.

Even back in 1994, the level of control in US universities wasn't as stringent as it is now. Nowadays, companies like Microsoft and Intel have strict cybersecurity measures that employees must comply with. With this baseline understanding, let's dive into the main topic.



In the first module, we introduced various kinds of threats and what cybersecurity entails, focusing on educational institutions. In module two, our job is to understand the attacker. To protect your infrastructure from attackers, you must understand their methods, how they infiltrate systems, and what they do once inside. We need to identify common indicators of compromise, which signal that a system has already been breached. Attackers, including state-sponsored groups from countries like China or Russia, often remain undetected in systems for years. They establish persistence, communicate with command and control servers, and may bring in more malicious payloads, awaiting instructions to execute an attack.

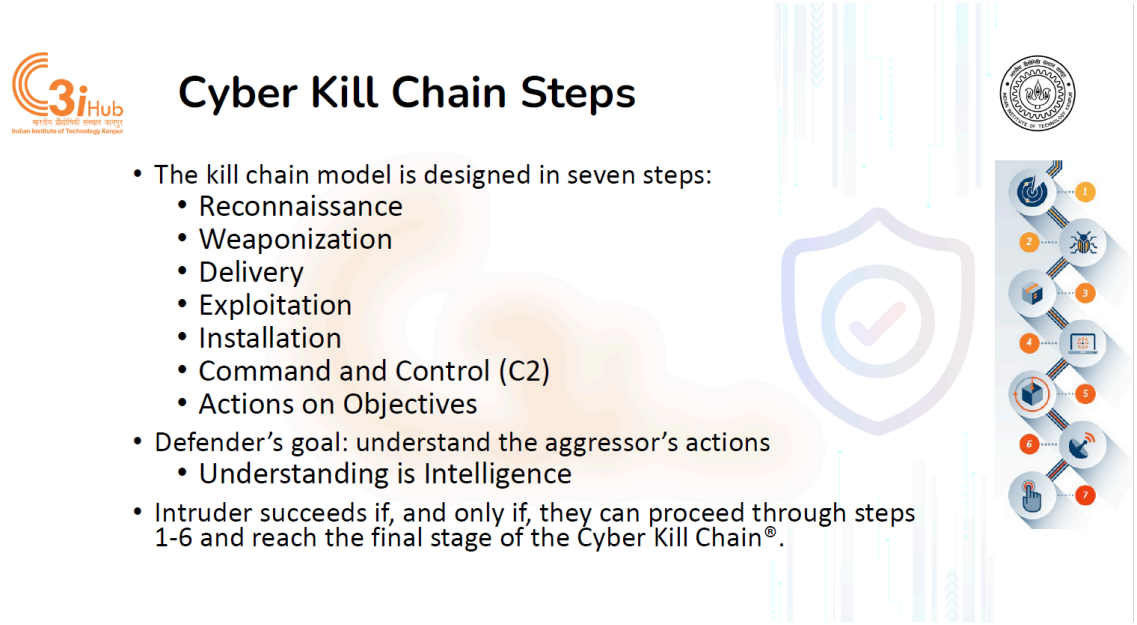
These threats can be triggered by various factors, such as geopolitical tensions. For example, during the 2020 border skirmish with China in Galwan, there were concerns that hidden Trojans in critical infrastructure like power systems, water treatment plants, and oil and gas facilities could be activated. Such attacks, however, are considered acts of war and can escalate conflicts. For instance, during the Ukraine-Russia war and the Hamas-Israel conflict, there have been significant cyber activities targeting not only the direct combatants but also allied countries.

This phenomenon includes pre-placed threats within infrastructure that can be activated at opportune moments. The framework for understanding attacker behavior was first developed by Lockheed Martin, which provided a comprehensive analysis in a well-known paper.

Paper link :
https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

This paper introduced the notion of the cyber kill chain. Since then, the cyber kill chain has been improved, and other more sophisticated kill chains have been developed. To get an idea, we'll start with the Lockheed Martin cyber kill chain. So, what is this kill chain? It's basically a way of thinking, a way of conceptualizing what the attacker might have done or might be doing. This does not mean that the attacker is necessarily doing it this way, but it is a theory. For example, Newtonian mechanics was a theory to understand motion, but it failed when considering motion at the speed of light, which led to Einstein's theory of motion, and that again failed in the quantum space. Theories are not necessarily exact explanations of what happens in the real world; they are approximations that may need refinement as we understand things more.

So, think about the cyber kill chain (CKC) in that way. Don't assume that this is the only way attackers operate. This theory, first proposed about 15 years ago, has been refined quite a bit since then. It essentially identifies how adversaries work on our infrastructure.



It suggests that there are seven stages through which an adversary gets into the infrastructure, communicates back to its command and control server, brings back more payloads, and waits for commands from the server before executing the final impact. This is how they explained what adversaries actually do.

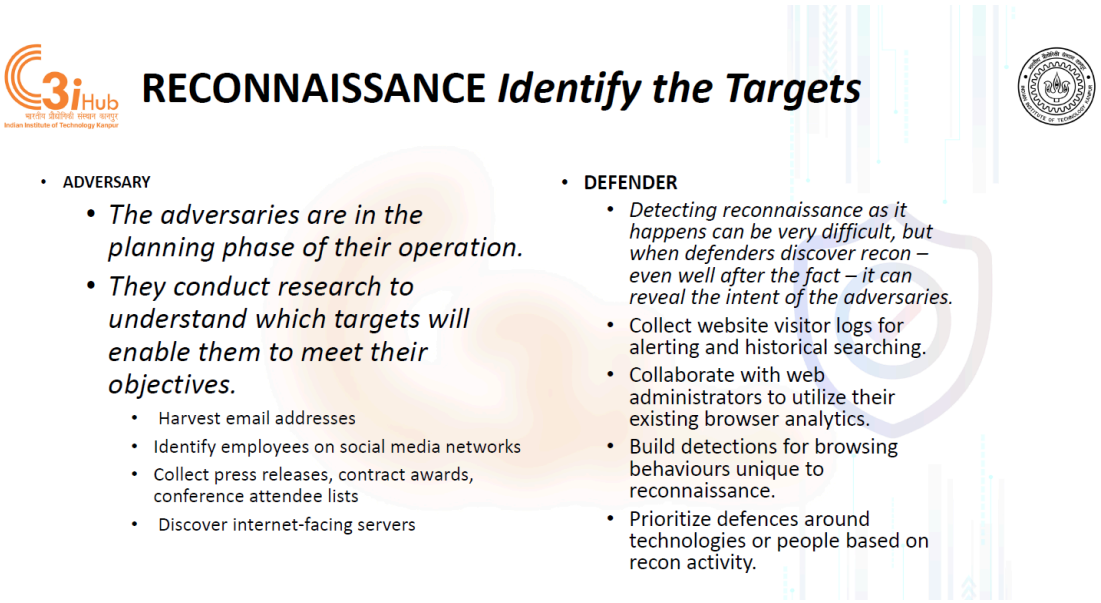
The idea is that if it is a chain of events that the adversary needs to carry out, then stopping or cutting the chain at any point can stop the adversary. This is one of the weaknesses of the cyber kill chain theory. The assumption is that if you know the steps—first this, then this, then that—then stopping the attacker at the first step can prevent the attack entirely. Alternatively, stopping them at a later stage can at least mitigate the damage. The theory posits that by interrupting at least one phase or stage, the adversary can be completely defeated.

But that is not how it really works. Adversaries can do a lot of harm by circumventing many of these defenses. So, you have to take it with a grain of salt. However, this gives you an idea of how to think about what the adversary is doing. This is intelligence-driven computer network defense, informed by the analysis of adversary campaigns and the intrusion kill chain.

The paper outlines seven steps in the kill chain: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. We discussed this in lecture two in another context, specifically the MITRE context, but the CKC is

more simplified with its seven stages. The defender's goal is to understand how these stages are carried out, as different adversaries may execute these stages in different ways.

Not every adversary operates in the same way. Therefore, it's crucial to understand all possible adversaries and how they carry out these seven stages. By doing so, you can identify when a stage is being executed by observing events in your network and devices. Based on these observations, you can attempt to stop them.



That's the idea. Let's delve into each of these stages, starting with reconnaissance. For example, if someone wants to attack the IIT Kanpur network, how would they even know how to get in? First, they have to find a device that is vulnerable and exposed to the internet where they can install something. Initially, this malicious program may not have all the functionalities but will establish a foothold in the infrastructure. Later, they can bring in more payloads. This initial foothold can be achieved by scanning the network with tools like Nmap. Many of you may have used Nmap to identify open ports and potential vulnerabilities

And if you specify a range of IP addresses and port numbers for scanning, using TCP packets or UDP packets, you can see whether any ports are open and gather some information. This process can be complicated by the need to use machine learning or signatures based on the responses you receive. You might scan by pinging or sending SYN packets (TCP-SYN packets) to all the ports. Most ports are typically closed, as we do not open all of them. However, most machines, especially those facing the internet, will have port 443 (the HTTPS port) open.

There may also be other open ports, like port 25, which is typically used for SMTP. By analyzing the responses from these ports, you can match signatures and identify whether the system is running Unix, Linux, or Windows. If it's running Windows, you can determine the specific version. Similarly, if it's running Linux, you can identify the version.

Many times, they will find systems running versions like 16.04, 18.04, or 20.04 of Linux, or maybe Windows 7, Windows 10, etc. They may also identify weaknesses, such as an unpatched Apache web server. Once they know about these vulnerabilities, they can exploit them. For example, they might send a well-formed payload or packet to exploit a remote code execution vulnerability. Alternatively, they might target web applications exposed to the internet.

For instance, if you have a web application with a form where users can enter information or a username-password field, and if it's poorly designed, the application might not properly validate the input. This could make it vulnerable to SQL injection, command injection, or other types of attacks. If your web application is not secure, attackers may figure out how to infiltrate your server. However, if your security measures are robust, including up-to-date patches for all applications, operating systems, and middleware facing the internet, and your web applications have been thoroughly tested for vulnerabilities through penetration testing, the likelihood of successful exploitation is significantly reduced.

Then the attacker might think, 'Ah, I cannot get in this way. So, what I should do is target an employee who is in a good position, potentially with high privileges, especially if they are an IT person or administrator.' They may figure out the employee's email address and WhatsApp number through various means, such as purchasing credentials dumped and sold on the dark web or exploiting information from previous data leaks. They will identify someone vulnerable and send a phishing email or message, which includes a URL leading to a malware-laden website. The goal is for the target to click the link and download something malicious. Alternatively, they might send a Word document, PDF file, or another file with embedded malware. This tactic is called 'phishing.' If the attack is delivered via SMS, it's called 'smishing.'

If the attack comes through WhatsApp, it doesn't have a specific name, but it's essentially a form of phishing. The messages are often crafted to be hard to resist, possibly claiming to be a bill or a notice from tax authorities, compelling the recipient to click on a link or attachment. In the past, phishing emails were often easy to spot due to grammatical and spelling errors. However, with tools like ChatGPT, these messages have become more sophisticated and harder to distinguish from legitimate communications.

But with tools like ChatGPT, they can craft impeccable emails. If they gather more information about you from your social media accounts, LinkedIn profile, and so on, they can tailor these emails even more effectively. They might see that you are interested in certain topics or hold specific roles, and then they can instruct ChatGPT or another AI tool to write a convincing email for a particular purpose. This becomes problematic because, even if you've designed robust perimeter security and patched all vulnerabilities, a person inside your organization who is unaware of phishing or not very careful can still compromise the system.

From the adversary's point of view, they aim to find someone through whom they can gain access to the system. Now, the issue is, as a defender, what actions should you take?

If you want to break the chain at the reconnaissance stage, you need to make reconnaissance difficult or impossible. Suppose you aim to do that. You would start by robustifying all applications and operating systems facing the internet and monitoring the logs of all accesses through those ports. By doing this, you're halfway there; you've made it more challenging for reconnaissance to occur through those channels. You should also check the access logs of your websites, particularly where employee directories and similar information are available.

Look for patterns indicating that someone is attempting to download the directory or gather similar information. For example, the IIT Kanpur directory is accessible from outside, so if someone wants to find my phone number, office number, or email, they can. Additionally, I have a personal website that can be accessed. It's not a big issue for me. However, there may be someone in the finance department who doesn't have a personal website or LinkedIn, but their information can still be found because IIT Kanpur directories are accessible from outside.

Therefore, we need to monitor the logs of access to the directory. If we notice suspicious access from countries like Russia, China, Vietnam, North Korea, Iran, etc., we should identify those IP addresses and be cautious. While we can't do much until an actual phishing email arrives, we can take proactive steps. We should provide training to every employee, student, and everyone involved to raise awareness about potential phishing threats and proper cybersecurity practices.

Why? Because if even one person connected to the network opens a malicious attachment, they might compromise the entire network, or at least the portion they have access to based on their privilege level. They may not have full access to the IITK network, but they could compromise the part of the network they can access. A bigger problem could arise if the malware is designed to exploit vulnerable applications on the victim's system. Even if the victim doesn't have administrator permissions, the malware could still attempt privilege escalation. There are programs in systems, including Windows system programs, that might have weaknesses exploitable for root access.

If the malware achieves administrative privileges, it could access sensitive data, penetrate other network segments, or cause further damage. Therefore, employee awareness training is extremely important. It helps prevent these kinds of attacks by educating users about the dangers of phishing and other threats.

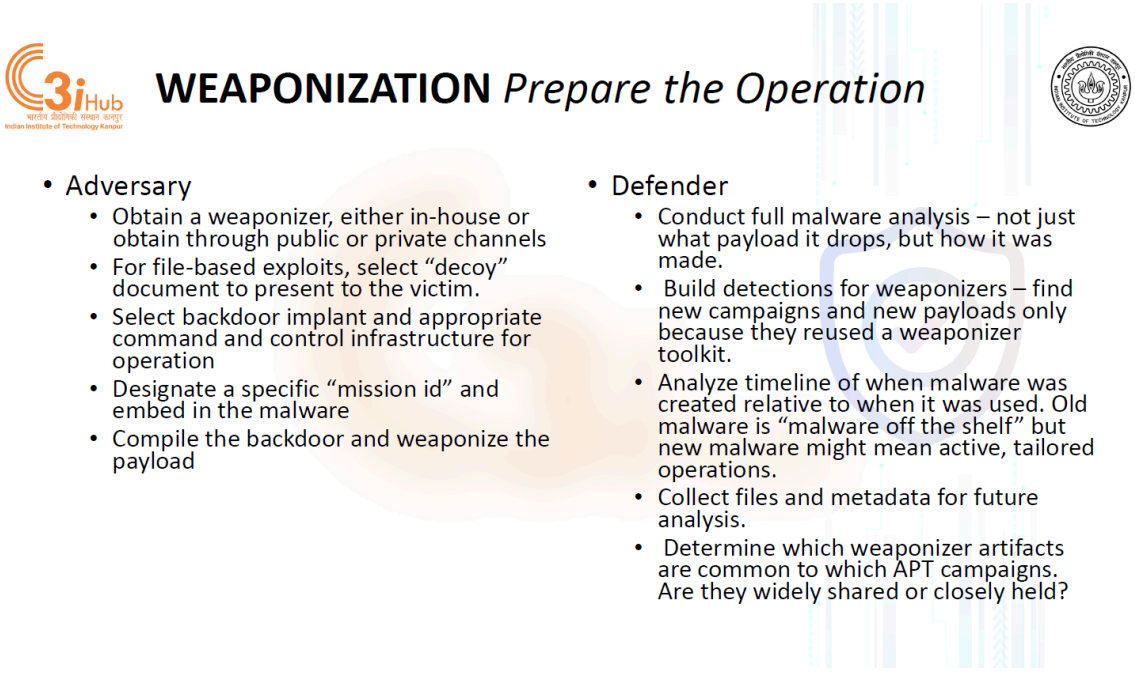
Companies also conduct what is called a phishing drill, where they send fake phishing emails to employees. If an employee clicks on a link or downloads an attachment from these fake emails, they are immediately redirected to a website that reprimands them, highlighting their lack of awareness. The website typically provides a document explaining why their actions were wrong and instructs them on proper behavior. Moreover, the company compiles a list of employees who clicked on the attachment or link.

These individuals are then required to take a course. In subsequent drills, if someone repeats the mistake, they receive more severe reprimands, such as doubling the course load they must complete, and so on. This practice is called a phishing drill. It's a proactive measure that defenders can use against reconnaissance. For the first type, where the network is scanned to identify weaknesses in internet-facing services and software, protection can be ensured through patching and other measures. The next critical step is vigorous training for employees and all users, including repeated phishing drills and continuous awareness efforts.

We also need to implement other detection measures using tools. Most companies now use detection agents on every device, including those used at home to connect to the company network. If the agent is not running, your VPN will not work. The agent continuously collects information from your device, sending all events, logs, and IP address connections to the central security operations center. If you turn off the agent, your VPN connection will be disabled. This system is designed to be robust, though not

foolproof, as a hacker might still manage to bypass it by hacking the VPN to function without the agent.

At least the company is doing the due diligence required. Detecting browsing behaviors of users is one such measure. For example, in the US, individuals with a security clearance have access to national security-related activities. Declaring your security clearance level on LinkedIn is prohibited because it would make you a target for adversarial countries. Companies may impose restrictions on disclosing specific job positions or roles within the company. These company policies, technologies, and processes are crucial for defending against reconnaissance activities.



Okay, so I'll just introduce this, and we'll continue tomorrow. Once you've figured out how to gain a foothold in the system and access at least one device in the organization, the next step is to determine what payload to use. The type and size of the payload depend on the method of entry. For example, if you find a remote code execution vulnerability in a web server, exploiting it may require more effort. However, these exploits can also be purchased on the dark web.

You could go to the dark web and find the right forum—though I'm not suggesting you do this, and you shouldn't. There are very deep dark web forums where you can request an exploit for a specific version, such as Windows 11 with a particular patch. If an exploit

exists, someone might offer it to you for a price, typically in Bitcoin or other cryptocurrencies.

It's challenging to track these transactions, making it possible for individuals without advanced technical skills to carry out severe attacks simply by purchasing exploits. Once you've acquired the exploit, you can execute the attack. However, if you're using phishing, the payload is often simpler, as the user will typically click on the malicious link or attachment, initiating the attack within the organization.

The weaponization step involves deciding what payload to deploy once you've gained access to the organization's network. You might consider installing a backdoor to maintain access and control over the device, or you could deploy a Trojan that performs malicious activities while reporting back to a command and control server. Alternatively, if the goal is financial gain, you might opt to use ransomware to encrypt files and demand payment for their release.

So, these decisions and the process of building the payload constitute the weaponization process. This weaponization can be done by the attackers themselves or by purchasing the exploit from the dark web. There are also resources available on platforms like GitHub. Additionally, tools like Kali Linux or Rapid7, which are intended for penetration testing, can be repurposed by malicious actors to inject harmful payloads into an organization's network.

We'll continue from here in the next class, as we've run out of time for today. It will take some time to cover these topics thoroughly, so we'll pick up from this point next time.