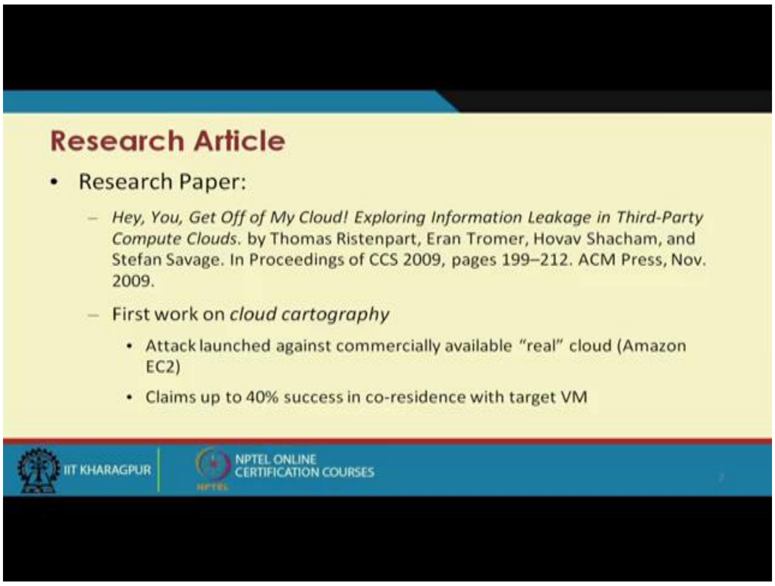
Cloud Computing
Prof. Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 28
Cloud Security – III

Hi. We will be discussing on Cloud Computing, primarily looking into the Cloud Security. So, this may be this series of lectures third talk on cloud security. So, in this discussion, we will be basically trying to look at a case study taken from a well-known research article. It might so happen that many of you have gone through the article or if not it will be good to go through these articles that is pretty one of the interesting article which shows that where these security in cloud computing varies, or how it differ from our generic network or computing security or what are the extra things we need to look at when we look at the cloud security per say.

So, it may not be possible to go for all deep into the technical details of this article this such article, but I will try to give you that a overview of the problem which will help us in understanding that how security matters even when you are using is a standard secured, trusted, well used public cloud computing platform.

(Refer Slide Time: 01:51)



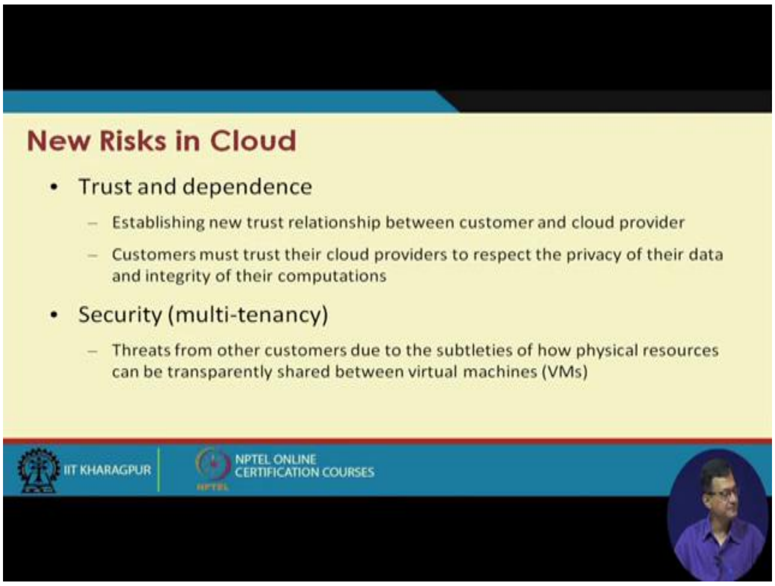
Research Article

- Research Paper:
  - *Hey, You, Get Off of My Cloud! Exploring Information Leakage in Third-Party Compute Clouds.* by Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. In Proceedings of CCS 2009, pages 199–212. ACM Press, Nov. 2009.
  - First work on *cloud cartography*
    - Attack launched against commercially available "real" cloud (Amazon EC2)
    - Claims up to 40% success in co-residence with target VM

So, one that article we are talking about is came in ACM CCS 2009 and title says that hey you get off my cloud exploring information leakage in third party compute clouds right. So, what they demand is the first work in cloud cartography, but apart from that it is interesting to see that what are the different way things will be there. So, our major objective of this particular discussion is to more look into the security aspects of the cloud; it is not on looking into any loophole of a particular cloud provider or security of any particular provider. We try to look at this paper which is there are some practical experiments which may help us in understanding the security aspect of the cloud in a better way that is our objective. It is not to analyze the work, but say but to take that work as an example case and see that how security it plays a important role in this cloud computing aspect.

So, the experiment that done in this particular work is this some sort of a so called quote unquote attack launched against a commercially available real cloud like typically Amazon EC2. And what they claimed that 40 percent are success in co-residence with the target VM right. So, if we remember our earlier lectures, what we are telling that one of the major issue is that if you whether we can co-residence a particular what we say attacking VM or a malicious users VM two way target VM, so that is a very challenging task. Because I really do not know how a cloud provider allocates the VM to other things.

(Refer Slide Time: 04:00)



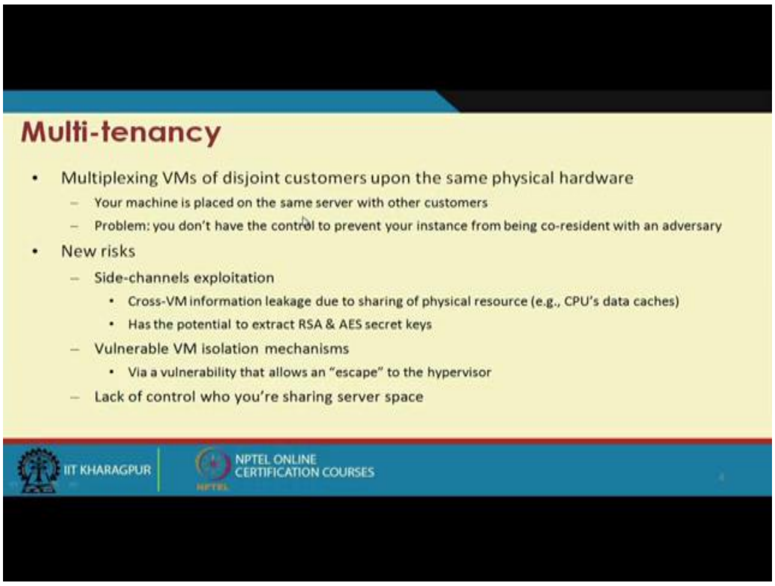
New Risks in Cloud

- Trust and dependence
  - Establishing new trust relationship between customer and cloud provider
  - Customers must trust their cloud providers to respect the privacy of their data and integrity of their computations
- Security (multi-tenancy)
  - Threats from other customers due to the subtleties of how physical resources can be transparently shared between virtual machines (VMs)

So, what we have seen in the new risk in cloud, one is that trust and dependency right. Establishing new trust relationship between the class customer and cloud provider that is important because I am basically as a customer leveraging all my means most of my data most of my processes on the cloud and I am somewhat going dependent on that cloud infrastructure. So, customer must trust their cloud provider to respect the privacy of the data and integrity of the computations. So, when we look at the security point of view, the customer must trust the cloud provider for the preservation of the privacy of the data and the integrity of the computation. If there is a process, the process is supposed to is performing the way it is supposed to perform that is one of the objectives of any customer right, so that is expected.

Now, the other problem, so that is how much you trust and dependent on the thing; other thing is the multi tenancy right. Threats from other customer due to the say they are basically deciding on the same VMs and physical resource can be transparently shared. So, what is happening that the virtual machine what I have been allocated the in the same physical machine some other customers are also allocated. So, what is the what is the chance that there is a there is a path establishment between these two VM; and if there is a malicious VM or this processes running in a malicious VM, what is the chance that my VM or my process is likely to be compromised. So, that is the usual problem or multi tenancy when you have multi tenant data, these are the things which becomes a big issue.

(Refer Slide Time: 05:59)



Multi-tenancy

- Multiplexing VMs of disjoint customers upon the same physical hardware
  - Your machine is placed on the same server with other customers
  - Problem: you don't have the control to prevent your instance from being co-resident with an adversary
- New risks
  - Side-channels exploitation
    - Cross-VM information leakage due to sharing of physical resource (e.g., CPU's data caches)
    - Has the potential to extract RSA & AES secret keys
  - Vulnerable VM isolation mechanisms
    - Via a vulnerability that allows an "escape" to the hypervisor
  - Lack of control who you're sharing server space

IIT KHARAGPUR

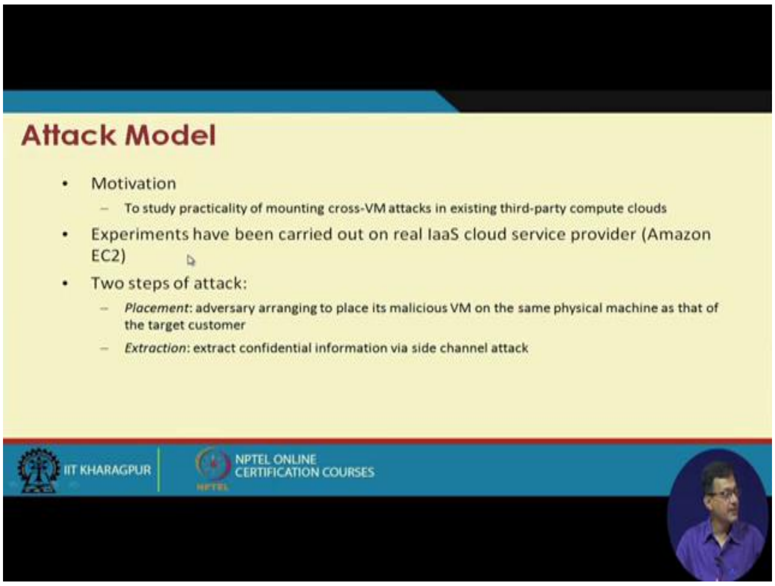
NPTEL ONLINE CERTIFICATION COURSES

So, multi-tenancy as we have discussed earlier multiplexing VMs of disjoint customers upon the same on the same physical machine. So, your machine is placed in the same server with other customer problem you do not have the control to prevent your instant from being co-resident with the some adversary instant. So, if it is a multi-tenant, so this multi-tenancy, how it will be residing that is at that is the logic the cloud provider doing. So, for the cloud provider point of view it is one of the major thing is resource management right it has a limited resource or it has a particular resource and it has to manage the thing, so that it optimize the performance of the customer level. So, based on that it basically try to try to deploy the VMs based on this analysis of that how resource can be properly managed and maximum performance level can be provided to the respective customer vis-a-vis their service level agreements right.

So, they here the with this with the multi tenancy some of the new risk factor came into picture, one is that slight channel exploitation that means, cross VM information leakage due to sharing of the physical resource, so that is another a big challenge. Across VM information leakage sharing of the physical resources is there. Has the potential to extract RSA and AES secret keys. We do not know, whether there is a potential to extract RSA and AES secret keys of this cross channel all this side channel exploitation. There are vulnerable VM isolation mechanisms like via a vulnerability that allows an escape to the hypervisor.

So, if there is a hyper, if there is a vulnerability, so if that can be exploited and this hypervisor some extent can be compromised. Lack of control you are sharing the server space. So, lack of control who you are sharing with. So, you do not have any control that who you are co-residing with right or this you are saying the simple way. So, these are the new risk which come into play.

(Refer Slide Time: 08:11)



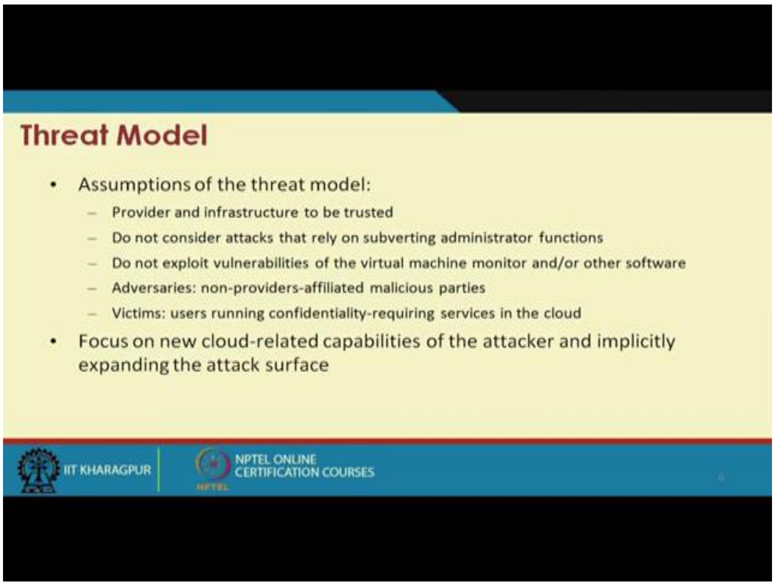
Attack Model

- Motivation
  - To study practicality of mounting cross-VM attacks in existing third-party compute clouds
- Experiments have been carried out on real IaaS cloud service provider (Amazon EC2)
- Two steps of attack:
  - *Placement*: adversary arranging to place its malicious VM on the same physical machine as that of the target customer
  - *Extraction*: extract confidential information via side channel attack

And the attack model specifically the attack model which in being also followed in this particular work we are what we are discussing is that the one of the motivation of this attack model is to study the practicality that whether it is practical or mounting cross VM attacks in existing third party compute clouds. So, if I am having existing third party compute cloud, is it possible to do to launch some sort of a cross VM attacks right if on the things right. So, what they did the experiments have been carried out on realize cloud provider like I am as an EC2 and this can be carried out to any type of IS provider.

So, there are two steps and this is these two steps are irrespective whether is a cloud or network or anything that is one is placement, adversary arranging to place the malicious VM on the same virtual machine as that of the target customer this is important. If I want to do across some sort of attack or something, first thing I have to do is that whether I can physically place my VM into the same space or the same physical machine or the same server space where the adversaries machine is there. So, that is one important thing And secondly, it is a extraction thing. So, once I am placed, so extract confidential information by side channel attack. So, these are the two type of.

(Refer Slide Time: 09:45)



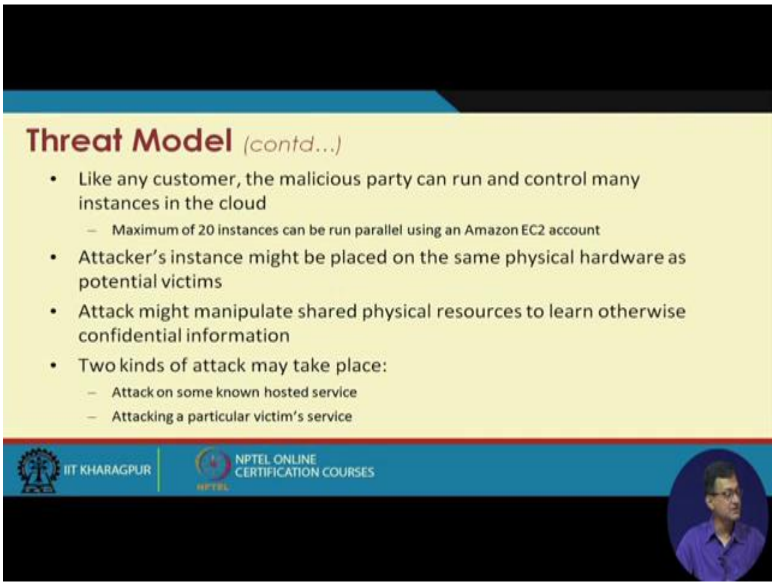
Threat Model

- Assumptions of the threat model:
  - Provider and infrastructure to be trusted
  - Do not consider attacks that rely on subverting administrator functions
  - Do not exploit vulnerabilities of the virtual machine monitor and/or other software
  - Adversaries: non-providers-affiliated malicious parties
  - Victims: users running confidentiality-requiring services in the cloud
- Focus on new cloud-related capabilities of the attacker and implicitly expanding the attack surface

So, next is the threat model like assumption of threat model is that the provider and the infrastructure to be trusted right. So, what we do when we this is one of the basic assumption is the provider and the infrastructure need to be trusted, do not consider attack that that rely on subverting the administrator functions all right do not exploit vulnerabilities of the virtual machine monitor or others software. So, it will not exploit that hypervisor and other things.

So, adversaries non-providers affiliated malicious parties. So, advisories are not provider affiliated they came as a user or customer. Victims, user running confidentially requiring services of the cloud. So, victims are running some of the operations which needs some basic privacy and confidentiality is not public operations of public data and services through the cloud. So, focus on new cloud related capabilities of the attacker and implicitly expanding the attack surface right. So, we try to see that what are the things and try to see that what type of other attacks.

(Refer Slide Time: 10:54)



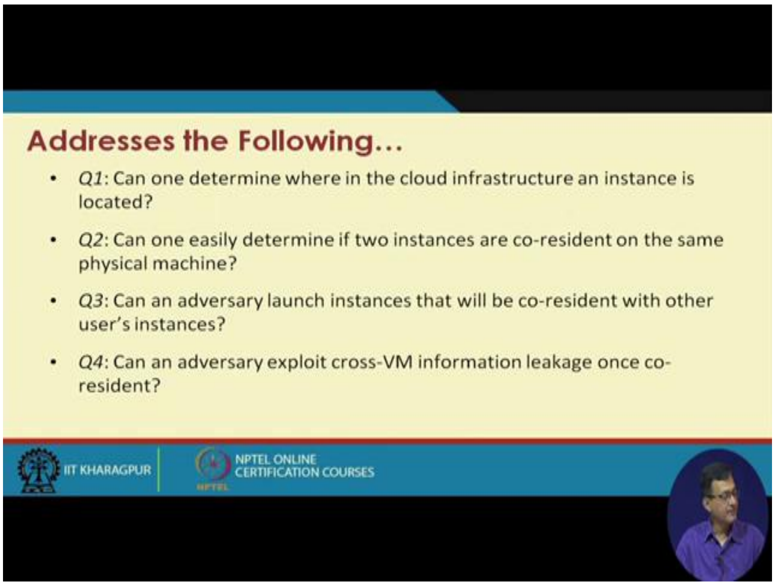
Threat Model *(contd…)*

- Like any customer, the malicious party can run and control many instances in the cloud
  - Maximum of 20 instances can be run parallel using an Amazon EC2 account
- Attacker's instance might be placed on the same physical hardware as potential victims
- Attack might manipulate shared physical resources to learn otherwise confidential information
- Two kinds of attack may take place:
  - Attack on some known hosted service
  - Attacking a particular victim's service

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, there are other threat models consideration like any customer a the malicious party can run and control many instances of the cloud, so that is another thing. Attackers instance might be placed on the same physical hardware as the potential victims are. Attacks might manipulate the shared physical resource to learn otherwise confidential information, so that attacker can basically do some surface cause VM attacks. So, two type of attacks can take place, attack on some known hosted services or attack on a particular victim services. So, one attack is that I know that these are the hosted services I want to attack on the things or I want to have particular victim service to be attacked. So, it is more what we say targeted attack.

(Refer Slide Time: 11:45)



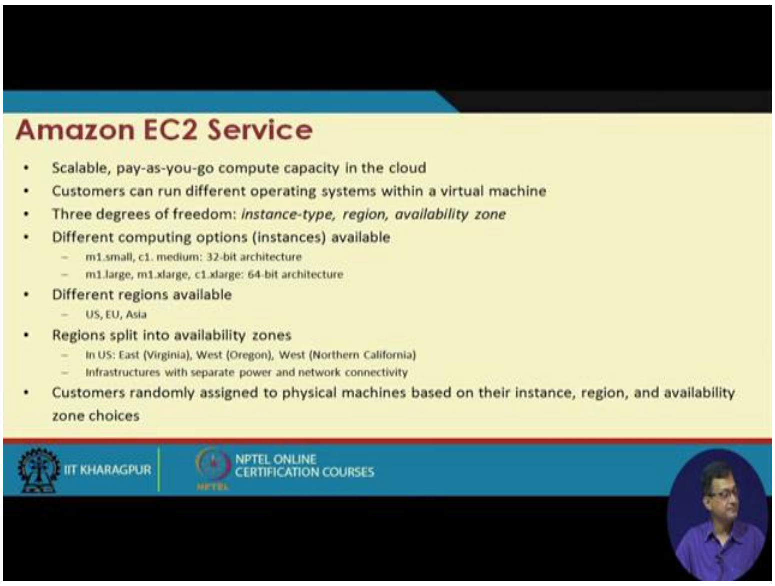
Addresses the Following…

- *Q1*: Can one determine where in the cloud infrastructure an instance is located?
- *Q2*: Can one easily determine if two instances are co-resident on the same physical machine?
- *Q3*: Can an adversary launch instances that will be co-resident with other user's instances?
- *Q4*: Can an adversary exploit cross-VM information leakage once co-resident?

So, in order to do that, so what they proposed or what they did is basically need to answer a few questions, one is that can one determine where the cloud infrastructure and instance is located right. So, is it possible to determine that where a particular instances located, very, very difficult not only difficult, something apparently impossible. You take a login from Amazon or Azure or Google platform or any other sales force or anything and that is they are way of handling the things like at the back of his management or the backbone management things. Question two, can anyone can one easily determine if two instances are co resident on the same physical machine right. One is that finding that where the one instance is there, another is that whether it is possible that I can determine that whether these two instances are co-resident right.

Number three, is that can a adversary launch instances that will be co resident with the user instances right. So, other question is that whether the adversaries launch instances, so that you want to co-residents with some targeted instances. And number four can an adversary exploit cross VM information leakage one co resident. So, if it is a co-resident somewhere rather whether there is a possibility that then I can have a cross VM information leakage. So, these are what they did the experiment or what we are trying to it get overview of the whole thing is primarily working on one of the very popular cloud provider, and it follows all possible best practices. Even with that, whether it is possibilities they are or not that is that thing what we are trying to look at.

(Refer Slide Time: 13:42)



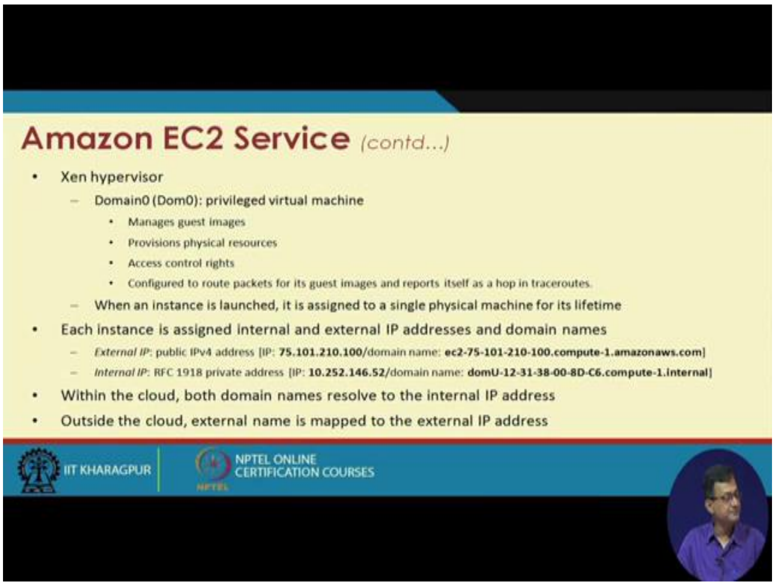
Amazon EC2 Service

- Scalable, pay-as-you-go compute capacity in the cloud
- Customers can run different operating systems within a virtual machine
- Three degrees of freedom: *instance-type, region, availability zone*
- Different computing options (instances) available
  - m1.small, c1. medium: 32-bit architecture
  - m1.large, m1.xlarge, c1.xlarge: 64-bit architecture
- Different regions available
  - US, EU, Asia
- Regions split into availability zones
  - In US: East (Virginia), West (Oregon), West (Northern California)
  - Infrastructures with separate power and network connectivity
- Customers randomly assigned to physical machines based on their instance, region, and availability zone choices

So, if you look at the Amazon EC2 service per say. So, it is a scalable pay as you go compute capacity of the cloud customer can run different operating system within the virtual machine, three degree of freedom instance type, region and availability zone. So, when you do select. So, there are three degree of freedoms. So, you can have instance type what type of instant or which region you want to launch, and the availability zone in the things. So, three computing options instances are available one is m1 small, m1 medium 32 bit architecture, m1 large, m1 extra large and so on and so forth. So, these are the different instances are available.

So, there are different region available right US, EU and Asia, this is the time one what the paper tells that is what when they are the came up in 2009. So, region split into availability zone. So, if you look at that UCI in us it has a east in the East Virginia, West Oregon and west another thing is Northern Carolina right. So, similarly infrastructure will separate power and network capacity connectivity. So, these are the different physically different located different places. So, they have not only different power line, that means, they are not on the same power backbone that in other sense that is they are not subject to failure if there is a power failure of one instances, and it is likely they are network also is different. So, that means, the IP block used in one will be different another end type of things. Customers randomly assigned to a physical machine based on their instance region and availability zone choices. So, customer has this option of choice in taking a make a choice of this, and based on that they are given to the different machine.

(Refer Slide Time: 15:43)



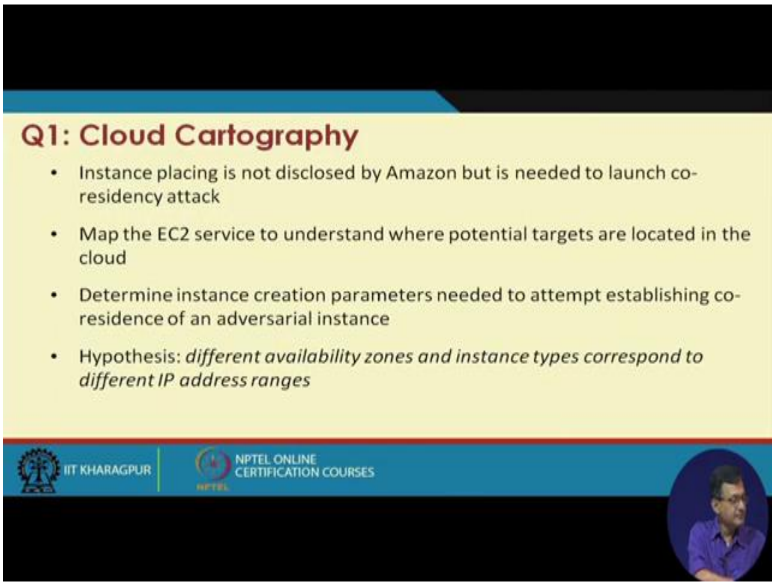
Amazon EC2 Service *(contd...)*

- Xen hypervisor
  - Domain0 (Dom0): privileged virtual machine
    - Manages guest images
    - Provisions physical resources
    - Access control rights
    - Configured to route packets for its guest images and reports itself as a hop in traceroutes.
  - When an instance is launched, it is assigned to a single physical machine for its lifetime
- Each instance is assigned internal and external IP addresses and domain names
  - *External IP*: public IPv4 address [IP: **75.101.210.100**/domain name: **ec2-75-101-210-100.compute-1.amazonaws.com**]
  - *Internal IP*: RFC 1918 private address [IP: **10.252.146.52**/domain name: **domU-12-31-38-00-8D-C6.compute-1.internal**]
- Within the cloud, both domain names resolve to the internal IP address
- Outside the cloud, external name is mapped to the external IP address

So, typically Amazon EC2 service using hp hypervisor and if you a XEN hypervisor sorry; so if you look at the XEN hypervisor, so there is a Dom0 what we say portion what is the privileged virtual machine which have manages guest images, it provisions physical resources access control rights configure to route packets in its guest images and reports itself as a hop to the trace route right. So, it routes the things and that can get it is a hop on the thing. So, when an instant is launched, it is assigned to a single physical machine for its lifetime. So, the instant particular anything.

So, secondly, each instance is assigned internal and external IP address and a domain names, so that is the philosophy of Amazon. So, external IP address something internal IP address based on some standard and respective domain name. Within the cloud both the domain names resolve the internal IP address. So, within the cloud both whatever you have seen internal external add thing outside the cloud external name is made to the external ip address. So, when we go to the outside the cloud then the external name is mapped to the external IP address.

(Refer Slide Time: 17:01)



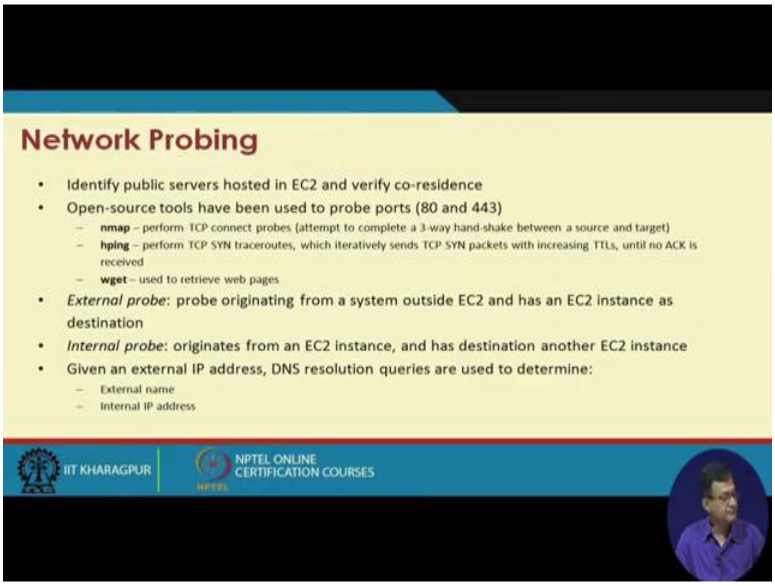
Q1: Cloud Cartography

- Instance placing is not disclosed by Amazon but is needed to launch co-residency attack
- Map the EC2 service to understand where potential targets are located in the cloud
- Determine instance creation parameters needed to attempt establishing co-residence of an adversarial instance
- Hypothesis: *different availability zones and instance types correspond to different IP address ranges*

Now, if we look at the is different aspects or the different queries which we which we raised or where which the article raised that try to address those and those queries. The query one is the cloud cartography, instant placing is not disclosed by Amazon, but is needed to launch whole residency attack. So, if I want to do some sort of a co-residency attack then I require the instant to be placed into the victims things, but the Amazon will definitely not disclose this. And map the EC2 service to understand where the potential targets are located in the cloud. So, we need to map or what they have shown they have tried to map the EC2 service to understand that where the targets are located in the cloud.

So, determine the instance creation parameters needed to attempt establishing co-residence at an adversarial instance. So, it is needs a create a parameter to attempt establishing a co-residence on the thing. And the basic hypothesis different availability zone and instance types correspond to different IP address right. So, if I have different availability zone and different instance types, it is likely that they are in the different IP ranges. So, whether we are able to whether we can there is a possibility of exploiting this.

(Refer Slide Time: 18:29)



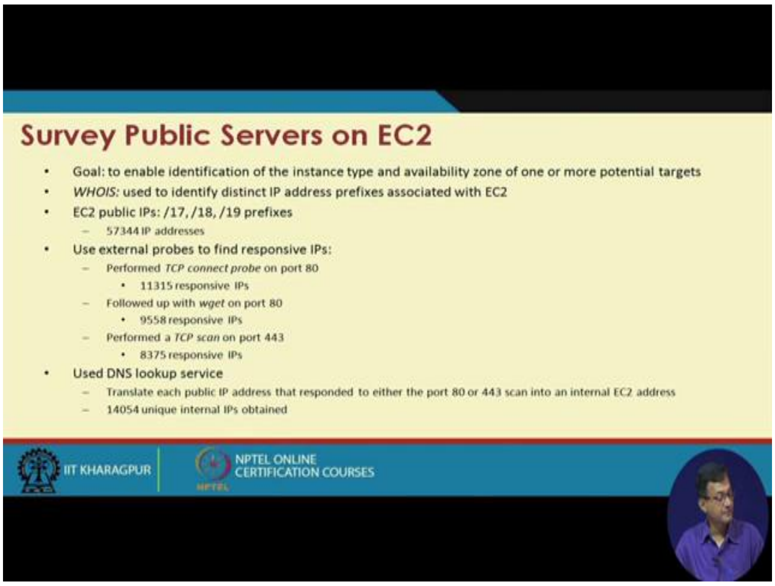
Network Probing

- Identify public servers hosted in EC2 and verify co-residence
- Open-source tools have been used to probe ports (80 and 443)
  - **nmap** – perform TCP connect probes (attempt to complete a 3-way hand-shake between a source and target)
  - **hping** – perform TCP SYN traceroutes, which iteratively sends TCP SYN packets with increasing TTLs, until no ACK is received
  - **wget** – used to retrieve web pages
- *External probe*: probe originating from a system outside EC2 and has an EC2 instance as destination
- *Internal probe*: originates from an EC2 instance, and has destination another EC2 instance
- Given an external IP address, DNS resolution queries are used to determine:
  - External name
  - Internal IP address

So, in order to do that those who have worked on network security or networking per se you know that there are different type of network probing tools are available. So, network probes are available and which are many of them are open source and fairly able to map the thing. So, similarly here also we require a network probing. Identify public servers hosted in EC2 and verify the co-residency. So, open source tools have been used to probe the ports port 80 and 443 that is the http and https secure port right RSN port. So, because these are these are mostly used for external access and likely that they will be opened and allowed the things.

So, one such tool is nmap, other is hping and wget right. So, these are the three popular there are several others tools and it is sometimes someone can write their own tool and type of things, but they are using the tool of probe. So, external probe, probe originating from a system outside EC2 and has an EC2 instance as the destination, so that can be the external. And internal originates from EC2 instant and it has destination another e c two instance right. So, this is internal and external thing. So, given external IP, DNS resolution queries are used to determine external name and internal IP address right. So, this is by the DNS query

(Refer Slide Time: 20:03)



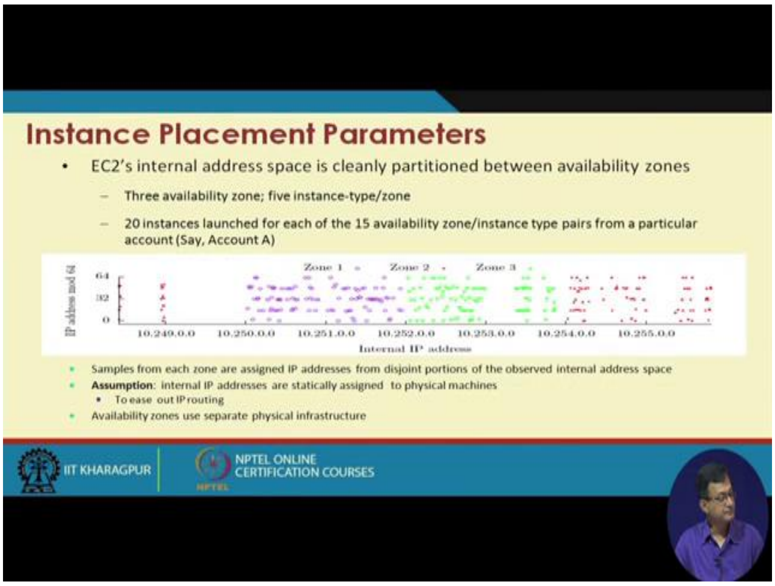
Survey Public Servers on EC2

- Goal: to enable identification of the instance type and availability zone of one or more potential targets
- *WHOIS:* used to identify distinct IP address prefixes associated with EC2
- EC2 public IPs: /17, /18, /19 prefixes
  - 57344 IP addresses
- Use external probes to find responsive IPs:
  - Performed *TCP connect probe* on port 80
    - 11315 responsive IPs
  - Followed up with *wget* on port 80
    - 9558 responsive IPs
  - Performed a *TCP scan* on port 443
    - 8375 responsive IPs
- Used DNS lookup service
  - Translate each public IP address that responded to either the port 80 or 443 scan into an internal EC2 address
  - 14054 unique internal IPs obtained

So, survey a public server on EC2 because to have a if we survey a goal to enable identification of instant type and availability zone of one or more potential targets. So, our primary or the primary goal of this particular work is to whether I can basically identify the instant type and availability zone of one or more potential targets that is one of the major thing. EC2 public IPs are in this prefixes like and there are public IPs of those tuned as reported by the particular article use external probes to find the responsive IPs right which are responsive for from TCP connect probe on port 80 and followed by wget port at port 80 and performed TCP scan at port 443 and then they see that what are the IPs which responses.

So, use DNS lookup translate each public IP that correspond to either port 80 or port 443 to an internal EC 2 address and then do again the probing on the things. So, some 14,000 odd unique internal IPs are obtained

(Refer Slide Time: 21:20)



Instance Placement Parameters

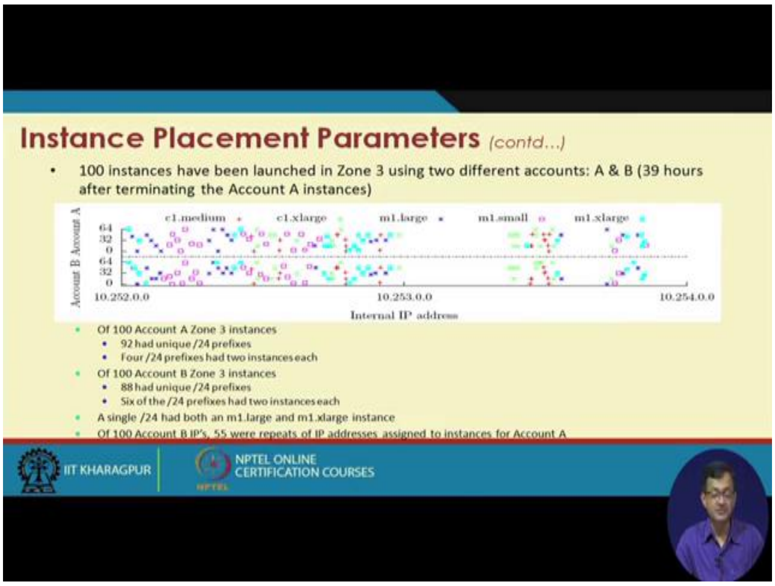
- EC2's internal address space is cleanly partitioned between availability zones
  - Three availability zone; five instance-type/zone
  - 20 instances launched for each of the 15 availability zone/instance type pairs from a particular account (Say, Account A)

- Samples from each zone are assigned IP addresses from disjoint portions of the observed internal address space
- **Assumption**: internal IP addresses are statically assigned to physical machines
  - To ease out IP routing
- Availability zones use separate physical infrastructure

Now, next is the instance placement parameter, what parameter you need to say that the things. Now, EC2s internal address space is cleanly partitioned between the availability zone. They are partitioned into thing three availability zone five instance type of instance type and zone as we have seen. 20 instance launched for each of the 15 what the experimental things they have done, and they have shown that samples from each of the zone are assigned IP address from disjoint portion of the observed internal IP address spaces. Assumption, internal IP address are statically assigned to physical machines to ease out IP routing otherwise it will again routing parameters will be there. Availability zone are used physical infrastructure. So, these are the things which are there.

So, in other sense what we try holistically see there are these are the different zones and different type of instances they are on different IP block. In other sense if I somehow select the same zone and etcetera whom I am targeting, it is likely I may be in the same IP block. If I know that in the same IP block then whether it is possible to launch again some of the probes and some of the attacks with the same IP blocks.

(Refer Slide Time: 22:49)



Instance Placement Parameters *(contd…)*

- 100 instances have been launched in Zone 3 using two different accounts: A & B (39 hours after terminating the Account A instances)

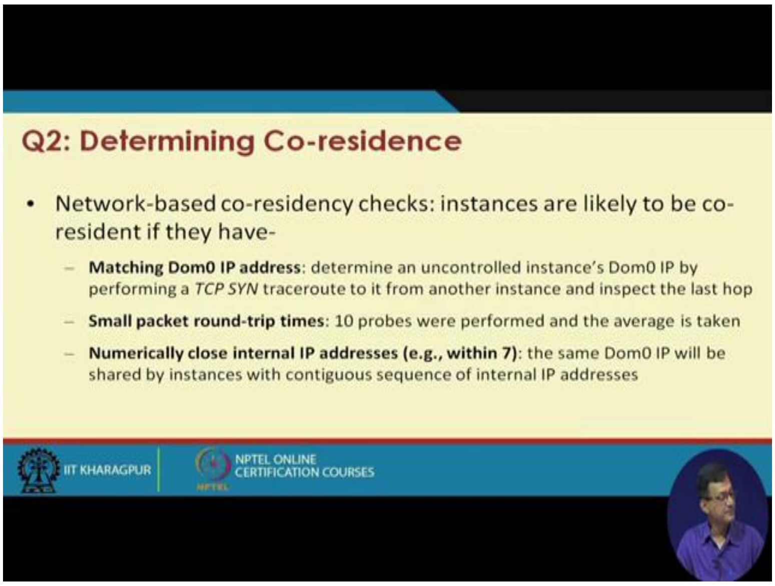
- Of 100 Account A Zone 3 instances
  - 92 had unique /24 prefixes
  - Four /24 prefixes had two instances each
- Of 100 Account B Zone 3 instances
  - 88 had unique /24 prefixes
  - Six of the /24 prefixes had two instances each
- A single /24 had both an m1.large and m1.xlarge instance
- Of 100 Account B IP's, 55 were repeats of IP addresses assigned to instances for Account A

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, they what they experimentally they shown that hundred instances has been launched in zone three using two different account A and B, 39 hours after terminating the account instance A and of hundred zone 3, 92 has a unique slash 24 prefixes, 4 prefixes has two instances each. So, these are their resultant. Out of 100 b zone three instances 88 unique and 6 this. A single slash 24 had both m1 large and m1 extra large instance. Of 100 accounts of these IPs 55 were repeats of IP address assigned to the instance of that account A that is interesting. So, out of that assign thing which are A and B.

So, what they are launched in different time scale like A and after terminating 39 hours of B, so I can I got that address etcetera. So, if you look at it gives some sort of it tries to give in some may be very grossly some cartography of or in other sense that the IP address blocks and etcetera how they are spared and so on and so forth.

(Refer Slide Time: 23:58)

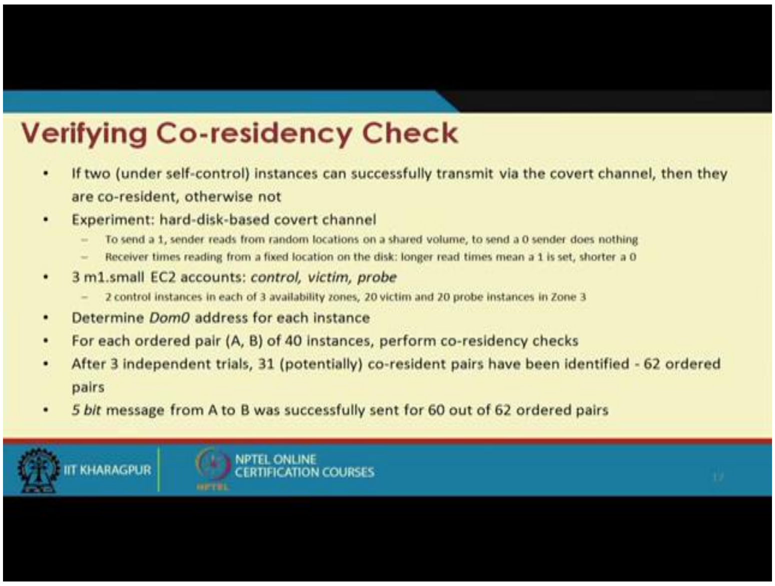


Q2: Determining Co-residence

- Network-based co-residency checks: instances are likely to be co-resident if they have-
  - **Matching Dom0 IP address**: determine an uncontrolled instance's Dom0 IP by performing a *TCP SYN* traceroute to it from another instance and inspect the last hop
  - **Small packet round-trip times**: 10 probes were performed and the average is taken
  - **Numerically close internal IP addresses (e.g., within 7)**: the same Dom0 IP will be shared by instances with contiguous sequence of internal IP addresses

Now, if I have this type of things like roughly know that these are the IP address blocks and type of things are there then whether determining co-residency can help or what can be done. So, network based co-residency at a checks instances likely to be co-residence if they are matching Dom0 IP address that as we have seen that domain zero is that primarily do we their management part. Small packet rounder trips if I have a round-trip times, so that will be the small packet in is in the same block, numerically close IP range typically within seven, so that is another things what we are having.

(Refer Slide Time: 24:43)

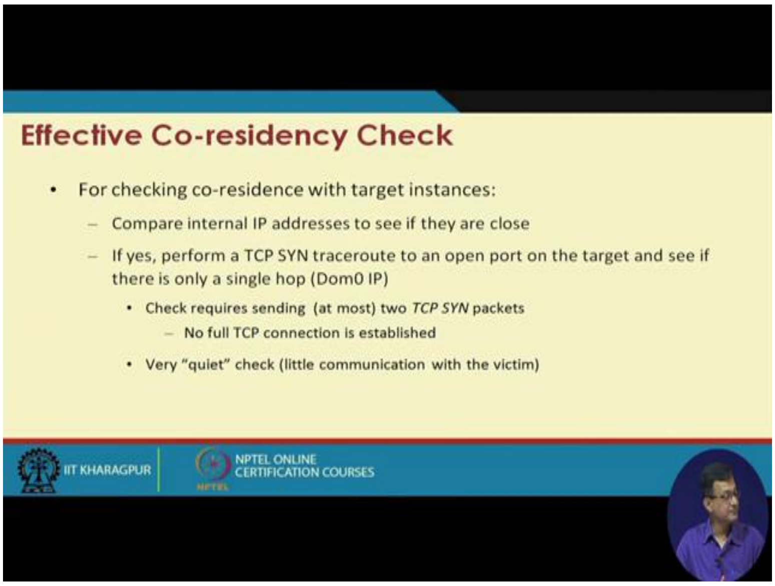


Verifying Co-residency Check

- If two (under self-control) instances can successfully transmit via the covert channel, then they are co-resident, otherwise not
- Experiment: hard-disk-based covert channel
  - To send a 1, sender reads from random locations on a shared volume, to send a 0 sender does nothing
  - Receiver times reading from a fixed location on the disk: longer read times mean a 1 is set, shorter a 0
- 3 m1.small EC2 accounts: *control, victim, probe*
  - 2 control instances in each of 3 availability zones, 20 victim and 20 probe instances in Zone 3
- Determine *Dom0* address for each instance
- For each ordered pair (A, B) of 40 instances, perform co-residency checks
- After 3 independent trials, 31 (potentially) co-resident pairs have been identified - 62 ordered pairs
- *5 bit* message from A to B was successfully sent for 60 out of 62 ordered pairs

So, verifying co-residency is check is another challenge that if two under self control instances are successfully transmit via the covered channel they are co-residents and so on and so forth. You can, if you can connect them experiment the hard disk base covert channels they have shown. So, three m one small account control victim probe what they need determine dom0 address for each pair of A B. So, what they try to do that what is a checking the co-residency of that 2 to VMs and type of things

(Refer Slide Time: 25:21)

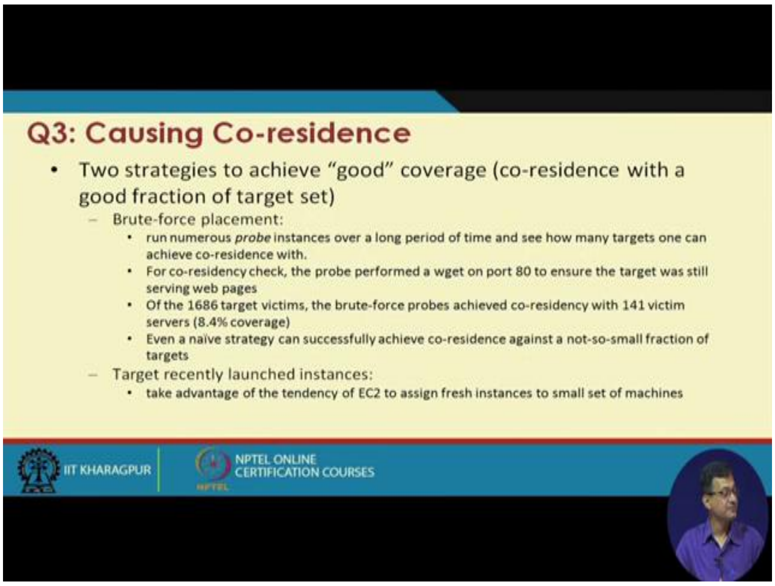


Effective Co-residency Check

- For checking co-residence with target instances:
  - Compare internal IP addresses to see if they are close
  - If yes, perform a TCP SYN traceroute to an open port on the target and see if there is only a single hop (Dom0 IP)
    - Check requires sending (at most) two *TCP SYN* packets
      - No full TCP connection is established
    - Very "quiet" check (little communication with the victim)

So, effective co-residency check for checking constancy with the target instances compare internal IP address to see if they are closer. So, if it is within the within the typical seven thing then it is a closer if he has performed TCP sync trace route to open port to the target and see and see if there is only one single hop dom0 IP. Check requires sending at most two TCP SYN packets, no full TCP connection is established very quiet check little communication with the victim. So, these are the checks which are done known some sort of a quote unquote non invasive manner that means, the victims is not aware of the things. So, you are basically cross checking that whether it is some co-residency thing.

(Refer Slide Time: 26:13)



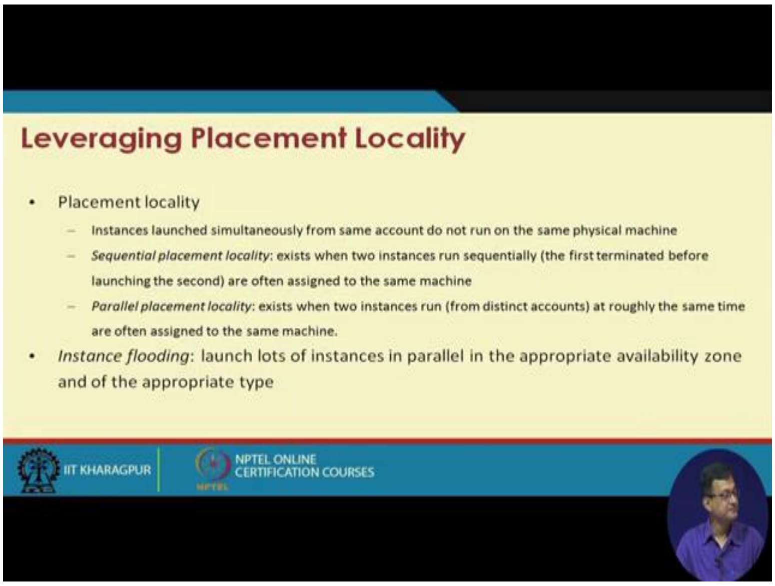
Q3: Causing Co-residence

- Two strategies to achieve "good" coverage (co-residence with a good fraction of target set)
  - Brute-force placement:
    - run numerous *probe* instances over a long period of time and see how many targets one can achieve co-residence with.
    - For co-residency check, the probe performed a wget on port 80 to ensure the target was still serving web pages
    - Of the 1686 target victims, the brute-force probes achieved co-residency with 141 victim servers (8.4% coverage)
    - Even a naïve strategy can successfully achieve co-residence against a not-so-small fraction of targets
  - Target recently launched instances:
    - take advantage of the tendency of EC2 to assign fresh instances to small set of machines

And the third thing is causing co-residency two strategies to achieve good coverage co-residency with a good functions target one is brute force placement you want to do some brute force placement of the things by run numerous probe instances and find out that where things are there, and you do a brute force. Other is target recently launched instances take advantage of the tendency of EC2 to assign fresh instance a small set of machines.

So, if it is after studying etcetera that is it can be think that the service provider are doing that is that close which are very instances launched within a particular small time span are placed into the thing into the same type of hardware or server. Then there is a chance of doing a target recently launched, instances and try to co-residency.

(Refer Slide Time: 27:05)

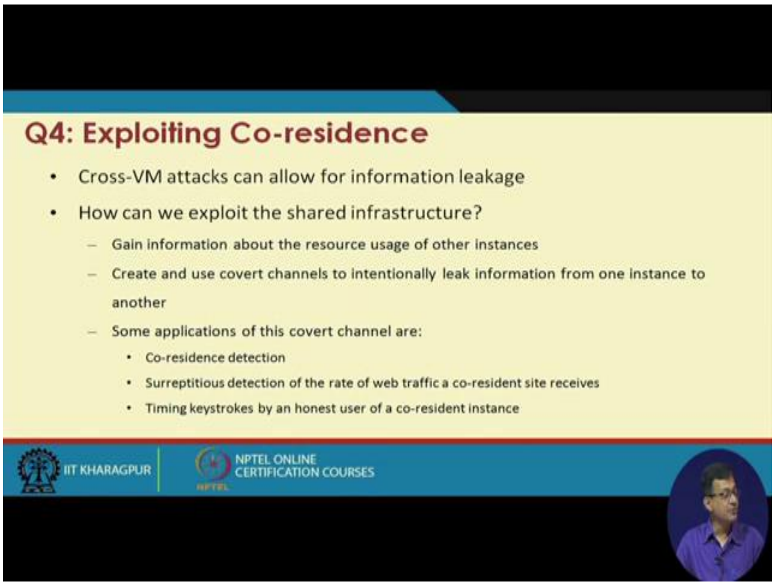


Leveraging Placement Locality

- Placement locality
  - Instances launched simultaneously from same account do not run on the same physical machine
  - *Sequential placement locality*: exists when two instances run sequentially (the first terminated before launching the second) are often assigned to the same machine
  - *Parallel placement locality*: exists when two instances run (from distinct accounts) at roughly the same time are often assigned to the same machine.
- *Instance flooding*: launch lots of instances in parallel in the appropriate availability zone and of the appropriate type

So, leveraging placement of locality, one is the placement of locality instances launched simultaneously from the same account, do not run on the same physical machine. Sequential placement locality existing when two instances run sequentially as we have seen that previously A and B. Parallel placement, locality exist when two instances run at roughly the same time are often assigned to the same machine. So, these are the things which can be exploited. And there is a other ways that instance flooding launch lots. So, parallel instances in the appropriate availability zone and appropriate type and try to see that what they happen.

(Refer Slide Time: 27:53)



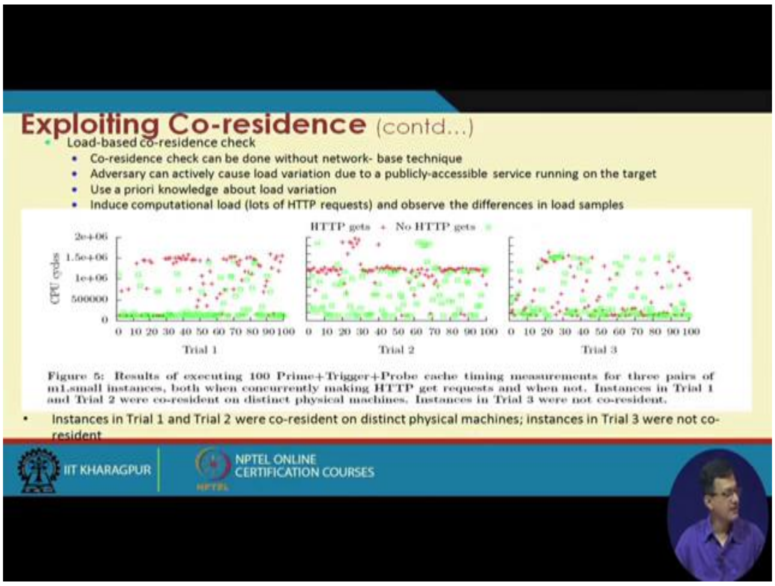
Q4: Exploiting Co-residence

- Cross-VM attacks can allow for information leakage
- How can we exploit the shared infrastructure?
  - Gain information about the resource usage of other instances
  - Create and use covert channels to intentionally leak information from one instance to another
  - Some applications of this covert channel are:
    - Co-residence detection
    - Surreptitious detection of the rate of web traffic a co-resident site receives
    - Timing keystrokes by an honest user of a co-resident instance

Similarly, they did a lot of experimentation to see that how this locality can be exploited. And finally, exploiting co-residents that is cross VM attacks can allow information leakage how can we exploit shared infrastructure is one that it means if I am co-resident then how to exploit this like gain information about the source uses of the other instances. Create and use covert channels to intentionally leak information from one instance to other. So, these are the other things which can be exploited by the attacker. Some application of these covert channels are co residence detection whether it is the thing that is whether I can have some secret detection scheme to look at it whether timing of the keystroke allow me to look at the password and so on and so forth.

And other type of techniques which are there in other cases other this type of covert channel attacks is one is that measuring the cache uses that and try to see that the what is the normal pattern and whether there is a any attacks etcetera there. And try to map that what sort of processes are they are based on the cache uses pattern.

(Refer Slide Time: 29:08)



Exploiting Co-residence (contd...)

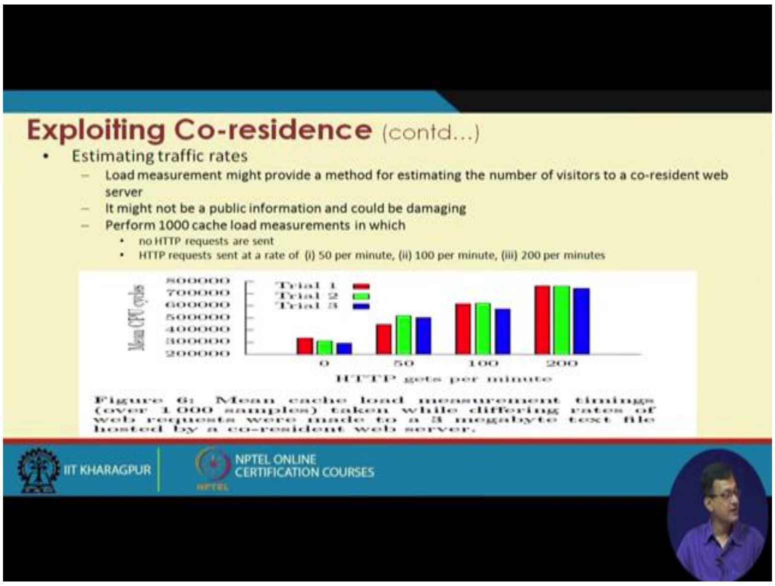
- Load-based co-residence check
  - Co-residence check can be done without network- base technique
  - Adversary can actively cause load variation due to a publicly-accessible service running on the target
  - Use a priori knowledge about load variation
  - Induce computational load (lots of HTTP requests) and observe the differences in load samples

Figure 5: Results of executing 100 Prime+Trigger+Probe cache timing measurements for three pairs of m1.small instances, both when concurrently making HTTP get requests and when not. Instances in Trial 1 and Trial 2 were co-resident on distinct physical machines. Instances in Trial 3 were not co-resident.

- Instances in Trial 1 and Trial 2 were co-resident on distinct physical machines; instances in Trial 3 were not co-resident

So, one is that exploit a load-based co-residence checking that exploit in co-residency. So, co-residency check can be done without network base network or adversary because I am co-residence. So, I do not require again now the resident a bigger other network infrastructure I am on the same machine. So, here they have shown this with the experiment that the trial one and two were co-resident on distinct physical machine instant, three were not co resident. So, that what if you look go through the paper there this has been shown.

(Refer Slide Time: 29:42)



Exploiting Co-residence (contd…)

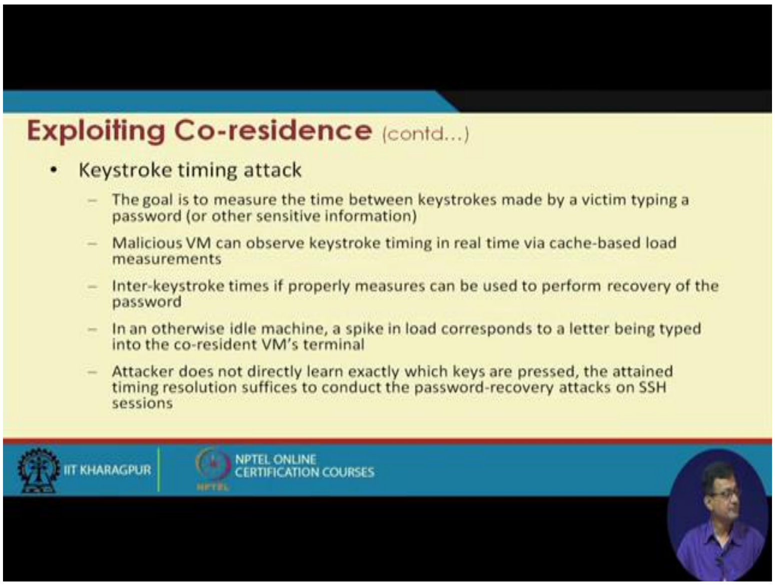
- Estimating traffic rates
  - Load measurement might provide a method for estimating the number of visitors to a co-resident web server
  - It might not be a public information and could be damaging
  - Perform 1000 cache load measurements in which
    - no HTTP requests are sent
    - HTTP requests sent at a rate of (i) 50 per minute, (ii) 100 per minute, (iii) 200 per minutes

Figure 6: Mean cache load measurement timings (over 1000 samples) taken while differing rates of web requests were made to a 3 megabyte text file hosted by a co-resident web server.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And it has been shown that estimation traffic rates that what sort of traffics are there with no http, with http connection and so on and so forth. These are the figures and data taken from that particular research article.

(Refer Slide Time: 29:57)



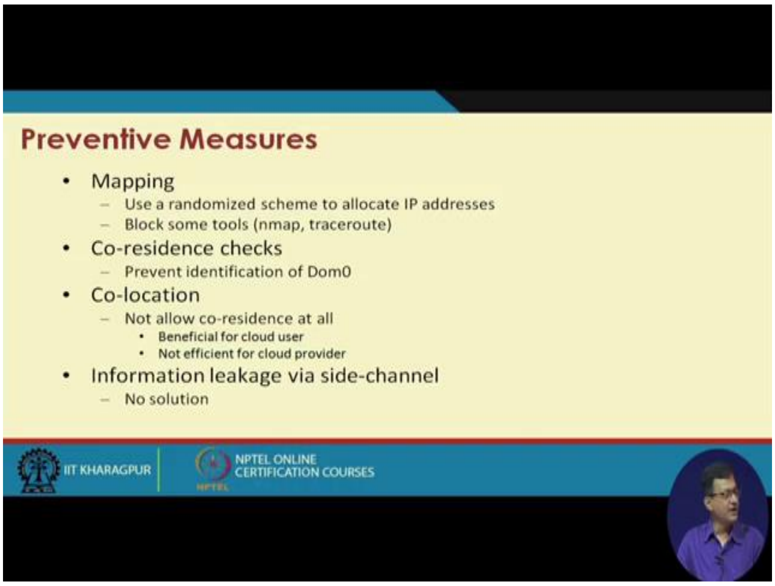
Exploiting Co-residence (contd…)

- Keystroke timing attack
  - The goal is to measure the time between keystrokes made by a victim typing a password (or other sensitive information)
  - Malicious VM can observe keystroke timing in real time via cache-based load measurements
  - Inter-keystroke times if properly measures can be used to perform recovery of the password
  - In an otherwise idle machine, a spike in load corresponds to a letter being typed into the co-resident VM's terminal
  - Attacker does not directly learn exactly which keys are pressed, the attained timing resolution suffices to conduct the password-recovery attacks on SSH sessions

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And other things are keystroke timing attacks right. So, based on that keystroke and this is a very popular or very well known type of things based on that whether you can basically look at that passwords and against the password and try type of things.

(Refer Slide Time: 30:18)



Preventive Measures

- Mapping
  - Use a randomized scheme to allocate IP addresses
  - Block some tools (nmap, traceroute)
- Co-residence checks
  - Prevent identification of Dom0
- Co-location
  - Not allow co-residence at all
    - Beneficial for cloud user
    - Not efficient for cloud provider
- Information leakage via side-channel
  - No solution

So, finally that whether the what type of preventive measures we can think of one is the mapping use randomized scheme to allocate IP address block some tools like in map and trace route. So, blocking tools may be, but randomized allocation may basically may be going against and some cases that the optimization or resource management of the things right. So, what they are from the prospective of the service provider, so that is that may be a challenge. So, co-residency check prevent identification of the dom0 that may be one of the way. So, what of that, so that that they cannot check that what is the domain zero.

Co-location not allow co residence at all right so, that means, beneficial for cloud user, but not efficient definitely not efficient for the cloud provider. And information leakage via side channel still is a big challenge like different type of sites challenge things are there and it is not only that you create a covert path like that you basically judge different other parameters look at like maybe the as well looking at that the looking at the cache behavior or the how the cache uses pattern, I am trying to look at that what sort of activities are going on.

So, with this we end our talk today that that new risk from the summary of that security thing that new risks from cloud computing they are which is little different from our conventional computer or information or network security. Shared physical infrastructure may and most likely will cause problems exploiting software vulnerabilities are not addressed properly here. Practical we have not there may be some software vulnerability if that the SaaS level cloud etcetera which are not being addressed. Practical attacks are in that particular paper they have shown that particular attacks are performed and some counter measures also proposed in this work

So, I encourage you to go through this paper, so that it is a good again I am repeating it is not to particularly look into particular service provider or look at the they are loopholes etcetera, more things we want to see a look at an overview that these are the things possible or this at the things which open up new risk etcetera, which are not there in our traditional network or computer or information security.

So, with this we will end our talk today.

Thank you.