CS249i: The Modern Internet **Winter 2024**

Project 3: Internet Security

Part 1–2 Due: March 8 11:59pm PT **Part 3 Due:** March 15 11:59pm PT

Introduction

In this project, you will explore the services that live on the public Internet and how researchers and attackers search for those services.

In the first part, you will set up a non-interactive honeypot on the Linux virtual machine (VM) that you were provided with in Project 1 to collect all of the *unsolicited incoming Internet traffic* destined towards your VM. You will measure how much unsolicited scan traffic reaches your VM, and use it to identify the actors most aggressively scanning the Internet for vulnerable hosts and services.

In the second part, you will use the Censys, an Internet service search engine, to explore what services live on the public Internet. You will investigate where these services live, what they host and what fraction are exposed/vulnerable to attack. You will also analyze the TLS certificate ecosystem and identify the most popular certificate authorities responsible for securing the HTTPS ecosystem.

Internet Measurement Caveats. As in all previous assignments, you'll find that questions about the Internet do not have a single exact or correct answer, as the real-world Internet is nuanced and constantly changing. *Furthermore*, depending upon the days you choose to collect data, it is more likely than not that the scanning actors and what they are scanning for, will be different. You and your teammates must describe the methodology you use to answer each question in addition to providing your final answer.

The Importance of Starting Early. During this project, you will be tasked with collecting data for at least eight hours—this is a task that cannot be shortened (at the very least the course staff don't know how to time travel). In addition, you will also be asked to compare your data with another CS249i group's data, which will require advance coordination with another group. Thus, it is imperative that your group does not save this project for the last minute.

Ethics. In this project, you will be analyzing real systems on the Internet, some of which could be vulnerable to attack. Further probing these systems directly can result in severe penalties, up to and including expulsion, civil fines, and jail time. You may not attempt to probe or attack any system without prior explicit permission from the owner, and you must not use techniques learned in this class to violate anyone's privacy or security; violation of this policy will result in a failing grade for the term. Acting lawfully and ethically is your responsibility.

Conflict of Interest Disclosure. Censys, the search engine used in Part 3 of this project, was originally created at the University of Michigan in 2015 by Zakir Durumeric. In 2017, it became an independent company, in which Zakir Durumeric has financial interest and where he remains an adviser and board member. We use Censys because scanning the Internet requires extreme care in order to not advertently cause disruptions to destination networks, which we cannot otherwise guarantee. (Indeed, student groups trying to complete large scans using tools like ZMap have repeatedly taken down their universities' networks as well as studied hosts in the past.) Censys provides free data access to all academic institutions and non-commercial researchers. If you'd like more information on how Censys compares to other data sources, Greynoise presents an independent third-party analysis: A week in the life of a GreyNoise Sensor: The benign view.

Part 1: Collecting Internet Scanning Traffic

In the first part of this project, you will need to instrument your VM to collect unsolicited inbound Internet traffic (e.g., Internet scans and Backscatter).

Deploying `tcpdump`

To collect Internet scanning traffic on your CS249i VM, you will be using the `tcpdump` packet analyzer. `tcpdump` displays and filters TCP/UDP/ICMP/IP and other packets that are transmitted or received on an interface. While your VM does not have any publicly accessible ports open to the Internet (i.e., even the SSH port that your VM uses is accessible only through our SSH bastion), you can configure your VM to *listen* to all incoming traffic—including to services you don't have active. By listening to incoming traffic, you will be able able to record all the initial packets of a network handshake, with one caveat; due to a Stanford firewall, traffic destined to port 22 is blocked by the router and will not show up in your VM's `tcpdump` collection.

Since no real service is publicly accessible on any port on your VM, your VM will not engage in any TCP/UDP handshakes. Thus, `tcpdump` will not record any packets beyond the initial packet received (e.g., TCP SYN packet or first UDP packet). As a result, if the TCP protocol is used, you will not have insight into what the scanner was actually scanning for (e.g., attempting an SSH connection, requesting an HTTP page, etc), and will only see which port is being scanned. While we will not be using one for this particular assignment, interactive honeypots can be used to engage in a handshake and collect the data a client/scanner sends. If you want to understand more about how often probes are sent to the wrong ports, Stanford researchers recently published a paper on the topic: Cloud Watching: Understanding Attacks Against Cloud-Hosted Services.

To listen and collect all incoming scanning traffic, use the following `tcpdump` command:

```
sudo tcpdump -tttt -l -i ens9 -n \
not arp and not icmp and not icmp6 and not proto GRE and not src net 171.67.68.0/22 \
> honeypot.log
```

The `tcpdump` flags you're using here are:

- "-tttt": output date and time in human readable form
- "-l": use line-buffering to stream results into a file
- "-i": specify network interface
- "-n": turns off hostname and guessed protocol lookups (performing lookups generally substantially slows down real-time packet collection)
- "not arp and not icmp and not icmp6 and not proto GRE" : filters out arp requests, icmp requests, and generic routing encapsulation (i.e., tunneling) traffic
- "not src net 171.67.68.0/22" : filters out requests that come from the lab's network

`tcpdump` output

The `tcpdump` command above will save the output in a file named honeypot.log. Each line in honeypot.log will be of the following format:

```
Date Time Layer Source-IP.Source-Port > Destination-IP.Destination-Port: Protocol Details, Data-Length
```

For example:

```
2024-02-15 23:41:08.450536 IP 42.117.20.247.20920 > 171.67.69.34.23: Flags [S], ..., length 0
2024-02-16 01:04:40.975190 IP 14.1.112.177.38376 > 171.67.69.34.389: UDP, length 39
```

In the first line, the IP address 42.117.20.247 is scanning the IP address 171.67.69.34 on port 23 using the TCP protocol. The data length of the TCP packet is 0 bytes, as it is only a "SYN" packet. In contrast, in the second line, the UDP packet has a data length of 39 bytes. (Packets that don't say UDP and have Flags specified are TCP packets.) Note that we have configured `tcpdump` above to not record the data being sent, in order to minimize the resulting honeypot.log size. In general, the "-X" flag can be used to record the actual data being sent. Feel free to play around with this if you wish, but watch your disk usage!

Instructions for Collecting Traffic

Please run your honeypot for at least eight hours, to ensure that enough variety of traffic reaches your VM for a fruitful analysis.

Part 2: Analyzing Internet Scanning Behavior

In this part, we will use the `honeypot.log` generated from Part 1 to investigate Internet scanning behavior. When appropriate, please share your methodology (a high-level description will suffice) when answering each question.

1. The last question of Part 2 will ask you to compare your results with another CS249i group. Thus, please find and coordinate with another group in advance, to ensure that both of you have your assignments completed early enough to finish the last question. If there is an odd number of groups, it is ok for one group to coordinate with more than one other group. Please list the other group(s) here.
2. During what day and time did you run your `tcpdump` collection, and for how long?
3. How many total packets did you receive? How many packets on average does your VM receive per minute? Does the rate of scanning traffic differ over time? If you were running a vulnerable service on Port 445, how quickly would it have been found?
4. What fraction of incoming traffic uses TCP vs. UDP? Based on whether TCP packets are SYN packets or SYN-ACK packets (look into how to interpret the TCP Flags in the tcpdump output), what percentage of TCP packets are from scanning versus backscatter?
5. How many ASes did you receive traffic from? What are the top 10 ASes responsible for sending the most amount of traffic? What fraction of overall traffic received does each top-10 AS send? What are these ASes (e.g., ISP, cloud)? Do any of these ASes appear to have bad reputations (e.g., bulletproof hosting providers) or are they benign?
6. What are the top 5 countries responsible for sending the most amount of traffic? What fraction of overall traffic received does each top-5 country send? Are there any patterns or distinctions in the type of traffic you receive from each of these countries?
7. How many unique ports were scanned on your VM? What are the top 10 ports that are scanned? What fraction of overall traffic does each top-10 port receive? Are these ports IANA-assigned to any protocols? Can you hypothesize the reasons for scanning the top three ports that you see most commonly scanned?
8. Compare your group's answers to the questions above with another cs249i group. At a high level, do the scanning actors/ chosen-ports/ frequency of the scans appear to be similar with the other group's VM, or different? Why or why not?

Part 3: Finding Internet Services

In this part of the project, you will use the Censys Search Engine to analyze the types of services that are found on the public Internet. Censys scans 100% of the IPv4 address space on about 1,000 popular ports as well as attempts to predict the locations of services on the remaining 64K ports. More information about the scanning methodology can be found here: Censys Internet Scanning Intro.

Access to Censys Data

You will receive an invitation to join the Censys "CS249i - Stanford University" team that will give you access to the Censys search engine. Make sure to use this invite link to avoid quickly running into a query limit. *You must accept this invite ASAP, since it will expire within 48 hours.* Please check/post on Ed for details/questions regarding access to Censys.

To interact with the Censys search engine, you can directly query the Web UI and/or use the Censys Search API. You can find a short tutorial for the web UI here and the API here. You will find that some questions will be easier to answer by using web UI, while others will be easier to answer by using the API. Many of the questions in this homework will be easiest answered using Censys' *Report* functionality, which will provide the breakdown of a field (e.g., port or protocol) for all of the scan results in scope.

Censys has several tutorials (with example queries) and FAQs found here, which your group may at some point find helpful.

Analyzing Internet Services

Using the Censys search engine and/or API, please answer the following questions. **Please share the query/methodology you used to answer each question.**

1. What are the top 20 most common ports that host services in the IPv4 address space? What are these top-10 ports IANA-assigned to? We define "service" as an (IP, port) pair. What are the 20 most common network protocols? Should any of these protocols be cause for concern?
2. On each of the top 10 ports, what fraction of services are *not* hosting the IANA-assigned protocol (e.g., SSH, HTTP)? For each of the top-10 ports, what is the most popular non-IANA assigned protocol hosted? Why do you think that these services run on unassigned ports?
3. What are the five largest ASes hosting the most services? Who do these ASes belong to? What are the most popular ports (and their IANA assignments) for each AS?
4. How many hosts expose the MySQL or MongoDB database protocols? What ASes are the majority of exposed databases located in? What are the security implications of having publicly accessible databases?
5. Modbus, Siemens S7, PCOM, and DNP3 are network protocols commonly used by SCADA (supervisory control and data acquisition) and/or ICS (industrial control systems) devices. What ASes and countries are devices that run these control system protocols primarily located in? Explain the implications of having such devices exposed on the Internet.
6. Investigating the Stanford autonomous system, are there any services that you think pose a security risk to Stanford being publicly exposed? Why or why not?
7. The Censys Certificate Dataset is composed of certificates found in public Certificate Transparency servers. Based on certificates that are *currently* trusted, who are the ten largest certificate authorities (CAs) today? What are the set of CAs that are used for signing certificates used by sites under the `stanford.edu` domain or any of its subdomains?

8. Investigate the Issuer Strings on Trusted Certificates. Do you see any companies whose primary line of business is not issuing certificates? Who are these players? Which root CAs signed their intermediate certificates?

9. What are the most common reasons for a certificate being revoked? (Hint: Look into Certificate Revocation Lists, CRLs)