

Project 3: Internet Services and Security

Part 1–2 Due: March 9 11:59pm PT **Part 3 Due:** March 16 11:59pm PT

Introduction

In this project, you will explore the services (e.g., DNS protocol running on 8.8.8.8:53) that live on the public Internet as well as how researchers, defenders, 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* directed towards your VM. You will track how much unsolicited scan traffic reaches your VM, and use that data to identify the actors most aggressively scanning the Internet for vulnerable hosts and services.

In the second part, you will use **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 collect data, it is possible that the actors and what they are scanning for will be different. Like the past two projects, 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. 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 inadvertently cause disruptions to destination networks, which we cannot otherwise guarantee with many student groups operating simultaneously. (Indeed, student groups trying to complete large scans using tools like **ZMap** have repeatedly taken down their universities’ networks as well as scanned 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 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 public services are 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 visibility into what application layer protocol 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. Instead, we will assume that traffic destined to a port is intended for the default protocol on the port (e.g., TCP traffic destined to TCP/22 is scanning for SSH and traffic destined to TCP/80 is HTTP traffic).

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. If you collect data for a short period, you might end up collecting skewed data.

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. During what day(s) and time did you run your `tcpdump` collection, and for how long?
2. Let’s characterize the overall amount of traffic you saw:
 - (a) How quickly after you started to collect traffic did you see the first packet?
 - (b) How many total packets did you receive?
 - (c) How many packets on average does your VM receive per minute?
 - (d) Does the rate of scanning traffic differ over time?
3. We can filter out some of the traffic you received to better isolate scan traffic targeting your VM.
 - (a) What fraction of incoming traffic uses TCP vs. UDP?
 - (b) 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**? What characteristic or heuristic are you using to determine that a TCP packet is a scan probe?
 - (c) If you were running a vulnerable service on TCP Port 445, how quickly would it have been found after you first connected your host to the Internet? (Consider only scan probes. We will define a scan packet to be either a TCP SYN packet or a UDP packet for the remainder of the assignment.)
 - (d) How many unique ports were scanned on your VM?
 - (e) What are the top 10 ports that were most commonly scanned on your VM? What fraction of overall traffic did each top-10 port receive? What protocols are associated with these ports as defined by the **IANA-assigned ports list**? Provide a table that summarizes your results.
 - (f) For the top three ports that you see most commonly scanned, can you hypothesize the reasons for scanning these ports, based on your understanding of the current security landscape?
4. Next, let’s investigate where scan traffic originates from. Oftentimes, researchers first aggregate traffic by geographic region and origin ASN (that the sender belongs to) to understand high-level traffic characteristics.
 - (a) How many unique ASes did you receive traffic from?
 - (b) What are the top 10 ASes responsible for sending the greatest amount of TCP-based scan traffic, and what fraction of overall traffic received does each top-10 AS send?

- (c) For each of the top five ASes, do you think that traffic from the ASN is malicious? Base your assessment on the ports you see targeted and what you can find out about the ASN and its owner online. For example, you might consider whether any of these ASes appear to have bad reputations (e.g., bulletproof hosting providers).
5. Next, let's investigate what countries most frequently scanned your VM. Using the same data and methodology from Project 2, geolocate the IP addresses that sent you scan traffic.
- (a) What are the top 5 countries responsible for sending the greatest amount of traffic, and what fraction of overall traffic received does each top-5 country send?
 - (b) Separate the scan traffic originating from (1) the United States, (2) Germany, (3) China, (4), Russia, and (5) all other countries. What are the top 10 ports broken down by country?
 - (c) In about a paragraph, describe and interpret the similarities or differences in what protocols are targeted by these countries.

Part 3: Finding Internet Services

Public-facing Internet-facing services are not only critical to how we connect to devices in practice, but they are the number one way that attackers first gain a foothold in networks to attack them (e.g., to deploy ransomware or steal intellectual property.) 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 is one of a number of Internet search engines that provide this service.

Censys scans 100% of the IPv4 address space on about 150 popular ports daily as well as attempts to predict the locations of services across all 65K 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.

Important! Many of the questions 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. Let's start by looking broadly at the services on the public Internet.
 - (a) What are the top 20 most common *ports* that host services in the IPv4 address space? What protocols are these top-10 ports [IANA-assigned](#) to? We define “service” as an (IP, port) pair. Note: You can exclude IPv6 hosts by specifying `not labels: ipv6` in your query.
 - (b) What are the 20 most common network *protocols* in IPv4, counted across all ports where they appear?

- (c) Consider the implications of having services running the protocols from part (b) exposed on the public Internet. Should any of these protocols be cause for concern?
 - (d) What are the five largest ASes hosting the most IPv4-based services? What types of networks are these—e.g., individual companies? residential last-mile? cloud providers?
 - (e) For each of the networks you identified in part (d), describe what types of devices or services drive their high service deployment (e.g., cable modems? cloud services? web servers?). Explain the queries you use, and ground your interpretation in your data.
2. Let's zoom in on one particular type of protocol. Modbus, Siemens S7, and DNP3 are network protocols commonly used by SCADA (supervisory control and data acquisition) and/or ICS (industrial control systems) devices. These devices have increasingly come under attack in an effort to cause damage to physical infrastructure (e.g., water and wastewater systems). We'll focus on Modbus as it is the most widely used of these protocols.
- (a) What are the top 10 countries where exposed Modbus devices are located?
 - (b) What are the top 10 ASes responsible for running exposed Modbus control systems?
 - (c) Why are Modbus services concentrated in these providers? Hint: Look beyond the AS name to see what companies own these autonomous systems, and consider how devices like a water tower might be connected to the Internet.
3. Next let's investigate Stanford's network more specifically, and examine what Stanford exposes to the Internet.
- (a) What is the breakdown of top 20 protocols exposed by devices in Stanford's main autonomous system?
 - (b) Which of these protocols do you think could pose a potential security risk to Stanford simply by being exposed on the public Internet?
 - (c) What is one protocol that you don't recognize that runs on the Stanford network? Describe what this protocol is for, based on what you can find online. Given the protocol's purpose, do you think there is a good reason for Stanford to expose this host (or hosts) on the public Internet given that it can be connected to by an adversary? Or, do you think that this service was more likely exposed as an oversight, and should have been accessible only via Stanford's internal infrastructure?
4. Censys also lets you explore all of the certificates in public Certificate Transparency logs.
- (a) How many certificates are *currently* trusted on the public Internet?
 - (b) Based on certificates that are *currently* trusted, who are the ten largest certificate authorities (CAs) today by number of issued certificates? What percent of current certificates do these ten providers issue in total? Provide a table with the breakdown across the ten providers.
 - (c) How many of the top CAs in (b) offer any type of free certificate on their webpage? For any of them that don't, why do you hypothesize they might be so popular?
 - (d) What is average lifespan of certificates issued by Let's Encrypt versus other browser trusted certificates? What do you think accounts for this difference?
 - (e) Based on currently trusted certificates issued to `stanford.edu` or any of its subdomains, what top five CAs does Stanford most commonly rely on?