CS249i: The Modern Internet **Winter 2023**

Project 3: Internet Security

Part 1–2 Due: March 8 11:59pm PT **Part 3 Due:** March 17 11:59pm PT

Introduction

In this lab you will explore the Internet services researchers and attackers are scanning for and what services actually live on the public IPv4 Internet.

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-scanning traffic destined towards your VM. You will get to see how much unsolicited traffic reaches your VM, and will use it to identify the biggest Internet scanning actors.

In the second part, you will use the Censys Internet 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. You will also analyze the TLS-certificate ecosystem and identify who are the most popular certificate authorities.

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. Thus, it is important that you and your teammates 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 collect data for at least six hours—this is a task that cannot be shortened. You will also be asked to compare your data with another cs249i group's data, which will require advance co-ordination with another group. Thus, it is imperative that your group does not save this project for the last minute.

Conflict of interest disclosure. Censys, the Internet search engine used in Part 3 of this project, was originally started as a research project at the University of Michigan in 2015 as part of Zakir Durumeric's Ph.D. dissertation. In 2017, it became an independent company, in which Zakir Durumeric has financial interest.

Part 1: Collecting Internet Scanning Traffic

In the first part of the project, you will need to instrument your VM to collect incoming Internet scan traffic.

Instrumenting `tcpdump`

To collect Internet scanning traffic on your cs249i VM, you will be using the `tcpdump` data-network 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 (i.e., even the SSH port that your VM uses is accessible only through the a bastion and not to the outside world), you can configure your VM to *listen* to all incoming traffic. By listening to incoming traffic, you will be able able to record all the initial packets of a 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 publically 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. 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 assignment, interactive honeypots can be used to engage in a handshake and collect the data a client/scanner sends.

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

```
sudo tcpdump -tttt -q -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
- "-q" : minimize output
- "-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 Data-Length
```

For example:

```
2023-02-15 23:41:08.450536 IP 42.117.20.247.20920 > 171.67.69.34.23: tcp 0
2023-02-16 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 likely only a "SYN" packet. In contrast, in the second line, the UDP packet has a data length of 39 bytes. 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 six 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) for answering the 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? Is this more or less than you expected? Does the rate of scanning traffic differ over time?
4. What fraction of incoming scanning traffic uses TCP vs. UDP?
5. What are the domains of the top 10 IPs that send the most amount of traffic? What fraction of overall traffic received does each top-10 IP send? Do any of these domains indicate that the scanner is a researcher or benign actor?
6. 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)?
7. 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?
8. 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? Do any of these IANA-assigned protocols have known vulnerabilities?
9. 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 Internet Scanning Search Engine to analyze the types of services that are found on the public Internet. Censys scans 100% of the IPv4 address space on 3,500 popular ports using 100 different protocols, thereby providing one of the most detailed and up-to-date snapshots of the Internet.

Access to Censys Data

At some point within the next few weeks, 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 Censys Internet Scanning Search Engine 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.

The Censys API and web UI connects you with Censys' Universal Internet Dataset and Certificates Dataset; please take a moment to familiarize yourself with both data sets. You can find more information about how Censys stores information about its hosts here.

It is worth noting that Censys has a vast collection of 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 10 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.
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?
3. What are the five largest ASes hosting the most services? Who are these ASes? What are the most popular ports (and their IANA assignments) for each AS? Are any of these results surprising?
4. What fraction of all IPv4 services use the Remote Desktop Protocol or OpenVPN? Find a CVE that affects the RDP or OpenVPN protocol. What version(s) of either RDP or OpenVPN are vulnerable to that CVE? What fraction of RDP/OpenVPN services today are still vulnerable to that CVE? What country are the majority of those vulnerable services located in?
5. What fraction of all IPv4 services are exposed MYSQL or MongoDB databases? What ASes are the majority of exposed databases located in? Identify a CVE that affects each protocol; what does each CVE exploit? What are the security implications of having publically accessible databases?
6. What fraction of HTTP(S) pages on ports 80, 8080, and 8081 appear to be hosting publically exposed login pages (Hint: use regular expressions to filter the service.banner). How does the fraction compare across those three ports? Are publically exposed login pages more or less likely to be using TLS?
7. Modbus, Siemens S7, BACnet and DNP3 are all protocols commoonly used by SCADA (supervisory control and data acquisition) and/or ICS (industrial control systems) devices. Use google to understand the use of each of these protocols; what are the most common types of devices that use each protocol (e.g., smartgrids)? What ASes and countries are devices that run these protocols primarily located in? Explain the implications of having such devices exposed on the Internet.
8. Choose a CVE of your choice and describe the fraction of services that are vulnerable to this CVE. What countries and ASes host the largest fraction of services vulnerable to your chosen CVE?
9. Explore all the TLS certificates used by IPv4 services; who are the top 5 most popular Certificate Authorities?