Secure Database Exposition: Securely Exposing Services to the Public Internet with Cloudflare Tunnels

Benjamin Rader
IUPUI
Purdue School of Engineering & Technology

Abstract—This project explores the nuanced challenges of securing databases, focusing on threat modeling for self-hosted services, particularly the PostgreSQL database. The study investigates the risks associated with exposing services to the public internet and compares secure exposition methods, emphasizing the use of Cloudflare Tunnels. Implementation involves setting up both insecure and secure networks, with Nmap scans revealing the potential vulnerabilities of the former. The results highlight the effectiveness of Cloudflare Tunnels in mitigating risks and simplifying administration for smaller networks, making them a superior choice for secure self-hosting.

1 Introduction & Some Philosophy

The problem of securing databases is nuanced to say the least. The solution seems to be "well..it depends." Databases, by nature, are an abstraction and system for organizing and interacting with data. Oracle defines them as the following:

> "A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database [1]."

One must ask then if databases have problems that are uniquely applicable to most database systems as opposed to other abstractions of information technology such as applications, networking, hardware, authentication, authorization, cryptography, or common processes. Therefore, a more principled approach should be taken; one that can break out the relationship between various IT concepts while accounting for databases and the interests of those operating or even exploiting them. This can assist in threat modeling various use cases with databases involved.

1.1 Project & Motivation

This project involves hosting a database instance on my own on-premises hardware, specifically utilizing the PostgreSQL database to demonstrate threat modeling with exposed ports or services. In this context, the PostgreSQL database permits incoming traffic on a specific port, potentially allowing unintended outsiders to interact with the service if the "exposition" is not appropriately managed.

Databases vary across applications and implementations, with direct connections to virtualization layers like hardware, networking, operating systems, and platforms. Positioned at the top of these layers, databases link to components in lower layers. This layering aids in comparing cloud and on-prem solutions regarding management responsibilities. Database security enables us to identify specific areas and considerations unique to database security, distinct from network security, application security, and malware analysis. While compliance, standards, benchmarks, and legal cases related to databases may be considered unique, their comprehensive analysis is challenging within a semester. Moreover, assessing legal cases requires nuanced understanding and prior knowledge.

Extensive research exists on application-level attacks targeting databases, including injection attacks, cross-site scripting, and vulnerabilities specific to database implementations or versions [2]. Delving deeper into hardware-level attacks involves examining how database applications are segmented from the operating system and hardware. While common database issues like injection exploitation, access control, and post-access action statistics come to mind, none particularly captivated my interest.

In my professional role, I extensively engage with both cloud and on-premises technologies, navigating the intricacies of interfacing them through various layers of networking, protocols, and software. While these implementations can vary, I'm uncertain about the applicability of these skills. Unlike cloud technology, self-hosting (synonymous with on-prem) is less familiar to me. A significant challenge I encounter at work revolves around gaining access to resources. Although authorization is streamlined through a single sign-on system employing multi-factor authentication and specific protocols for on-prem or cloud systems, I've realized that the networking and transport-level protocols for authentication, authorization, and access across wide area networks can be complex even in their modern versions.

Motivated by these challenges, my objective is to explore

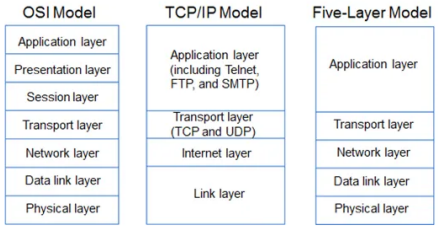

Fig. 1. Comparison of OSI Reference Model, the 4-Layer TCP/IP Model, and the 5-Layer TCP/IP Model

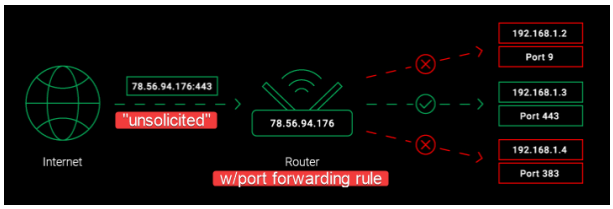

Fig. 2. Illustration of port forwarding: client from internet makes unsolicited request to server. Request is forwarded due to port forwarding rule on the logical firewall on the router.

how remote access is implemented in services or applications and identify common issues related to databases in this context.

1.2 Networking Concepts

To threat model thoroughly, it is productive to review some networking concepts that relate to hosting and exposing services.

1.2.1 The TCP/IP 5-Layer Reference Model

The TCP/IP 5-Layer Reference Model shown in Fig. 1 provides a structured framework for networking protocols. Of particular relevance is the Transport Layer, utilizing TCP (Transfer Control Protocol) and UDP (User Datagram Protocol). TCP establishes connections through system ports and ephemeral ports for servers and clients, respectively. Multiplexing and de-multiplexing processes occur through these ports, ensuring proper communication. The acknowledgment mechanism, crucial for critical data transfers like phone calls, introduces additional traffic [3], [4], [5].

1.2.2 Port Forwarding: Solicited & Unsolicited Connections

Port forwarding (Fig. 2) plays a crucial role in securely exposing services, especially when dealing with NAT (Network Address Translation). It addresses the challenge of enabling unsolicited incoming connections to a server within a local network. This feature, commonly found in routers, allows users to specify that all incoming connections on a designated port should be forwarded to a specific device in the local network [6]. Solicited connections involve TCP conversations initiated with outbound requests (SYN), while unsolicited connections start with inbound requests. Port forwarding rules and firewall settings on the router dictate the handling of inbound requests [7], [8]. Careful configuration is essential to prevent unauthorized access to local network resources, and secure port forwarding is often referred to as a ”firewall pinhole.” However, as will be shown, unsolicited connections can be abstracted to the application layer rather than the transport layer. In conclusion, if a device is getting unsolicited connections, then it starts to become a server. When machines do so, they immediately become a risk because adversarial motivations can manifest as exploits on the system.

2 Exposition Threat Model

To address the challenge of remote access, particularly for self-hosted services like a Postgres database, various security protocols are typically implemented at the network, transport, or application levels. This involves securing connections from public IP addresses to a domain name or FQDN, allowing users to remember a simpler identifier even if the underlying IP changes.

Authentication and authorization are key components of remote access, but their implementation varies widely based on use cases and organizational size. While larger corporations may share similarities, the security measures for a home network hosting self-hosted services differ significantly.

For smaller organizations or households, the effort required to implement and manage remote access solutions might not be justified. Outsourcing trust to external organizations or compromising on certain security aspects, such as strict TLS [9], can be reasonable, considering that most attackers prefer easier targets, especially if the potential gain is minimal.

2.1 Insecure Exposition Risks

The risks of exposing services to the public internet are easier to find evidence for than other parts of ”attack chains” such as privilege escalation or malware delivery. Such evidence takes extensive forensics whereas the risk of port scanning can easily be found by querying a store of logs and looking for patterns of IP addresses and ports. Reconnaissance, in this sense, is the riskiest part of exposing services. Malicious threat actors are profitable when they can find easy targets. Attacks have to ”chain” together multiple exploits, tools, methods, and workflows to obtain access on systems and use them for their own gain. The best way for them to be cost effective is to put together a good attack chain, then do ”recon” to find vulnerable targets. Port scanning is the best way to find vulnerable targets. Ports refer to places systems listen for inputs, so hackers simply find these open ports and attempt various inputs till they get the desired outcome.

2.2 Postgres Database Exploitation

Reconnaissance, by itself, is not detrimental to an institution’s IT operations. However, if an attacker does port scanning and finds an open port for a Postgres database, then it is only a matter of time before that database is compromised and has a monetary impact on the IT system. Port scanning can reveal valuable information about systems running the services or the version of the services themselves [10], [11], [12]. This includes but is not limited to operating system versions and service versions. Databases are gold for the

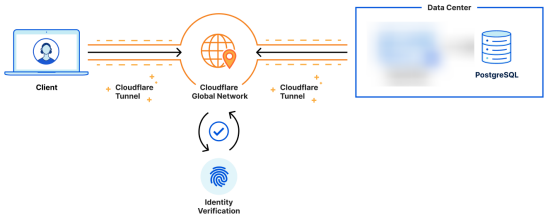

Fig. 3. Example of Cloudflare Tunnel with a Database

inexperienced script kiddie (amateur hacker) and APTs (advanced persistent threats) because they can be exploited in many different ways. When looking at secure exposition to the public internet and how that relates to exploitation based on service versions and OS information, there are two "actions on objectives" that stand out. Those would be ransomware and botnets. Botnets can be used for addition cybercrime or to mine cryptocurrency [13], and ransomware is the most profitable way to exploit sensitive data along with data extortion [14]. These are toxic ways that systems can be exploited because they are cost effective for almost any attacker. Databases have mission-critical data on them most of the time, so attackers have a higher chance of getting paid a ransom, and cryptominers are great on powerful systems such as those running databases and a DBMS.

3 Secure Exposition Methods

There are several methods or technologies that help implement secure exposition. The following are some examples:

1) **Cloudflare Tunnel** [15] - Operates at the application level by establish an agent on the webserver for each service without opening ports on firewall.
2) **Reverse Proxy with Firewall** - Proxy unsolicited requests to servers and host websites on the same IP address under different ports, then use firewall rule sets to accept or deny packets based on conditions.
3) **VPN (Virtual Private Network)** - implement secure exposition at the networking level by creating a tunnel into the private network.

These are a few examples of methods, but these are not comprehensive or exhaustive. I implemented Cloudflare Tunnels to show the simplicity of using application level protections along with an agent-based system.

4 Implementation

4.1 Tech Stack & Architecture

The two main components of this implementation are the secure and insecure networks which illustrate the threat model for exposing database services.

Below is a breakdown of the various components.

- (1) Insecure Exposition - runs on a virtual machine all on "localhost" to avoid the legitimate risks of exposing access to the public internet. Port scans are conducted with Nmap to show the risks of truly exposing the database to the internet.

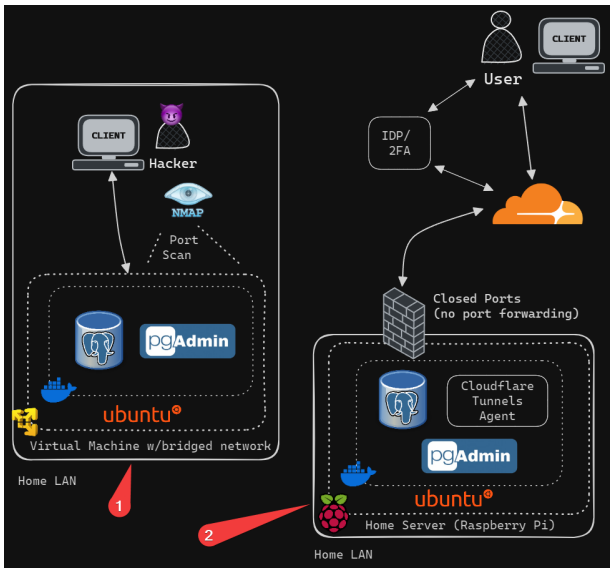

Fig. 4. Database Exposition Architecture & Setup: **1) Insecurely Exposed Services** - Postgres database is exposed on local network with no firewall and can be port scanning for its database service and OS information, **2) Securely Exposed Services** - uses a Cloudflare Tunnel docker container daemon to establish secure tunnels from the Cloudflare CDN while also implementing identity providers and authentication.

- (2) Secure Exposition - a Cloudflare Tunnel agent is established as a container running alongside Postgres and a Postgres management client (Pgadmin4). Nmap is also utilized here, but due to the security of this implementation (as will be shown) no information is found for the services adn therefore the risk is mitigated.
- Ubuntu - operating system using to run docker and service containers.
- Firewall - "brick wall icon." Logical process running on gateway router to WAN/public internet which would typically be used for port forwarding. This is used to show how Cloudflare Tunnel traffic works without port forwarding rules.
- Raspberry Pi - small computer hardware used to run Ubuntu Server and Docker containers. The purpose of using such a small computer was to illustrate a lightweight self-hosted use case of a database which is also common.
- All infrastructure for this project was set up on a home local network.

4.2 Secure Exposition Implementation

The general process for setting up the secure implementation was as follows:

1) Hardware & OS Setup
2) Container & Service Setup
3) Cloudflare Web & Agent Configuration
4) Nmap Setup

Setting up the infrastructure took 90+% of the project effort. I utilized a Raspberry Pi 3 for the secure exposition network, and in doing so I ran into numerous issues related to the computation and hardware of the Pi. For instance,

using the wrong power cord for the Pi resulted in issues with performance. At first, I also attempted to run an Ubuntu desktop on the Raspberry Pi, but it quickly crashed and I learned that the Pi was running on 1GB of RAM, so it could not do computationally expensive tasks such as running a desktop which seemed to necessitate at least 2GB of RAM and 4GB for normal performance. Instead, I imaged Ubuntu Server 22.04 onto the Pi, installed an SSH service, then accessed it remotely from my personal computer.

Container setup was straightforward and saved much space for the Raspberry Pi 3 as compared to other types of virtualization. I installed the "Portainer" container into Docker to make container management and monitoring easier. In terms of the database services and testing, I would typically setup the container on some port then access the Raspberry Pi's IP combined with the applicable port from within a browser. I could also do "port mapping" from within Portainer, so that certain ports resolved to specific containers even if separate services used the same ports internally. It's quite elegant to say the least.

An additional strength of Cloudflare Tunnels are that they integrate quickly with identity providers, authentication, and authorization logic which can be configured from within the Cloudflare Access interface on their web app. This allowed me to configure my Cloudflare Tunnel to only accept requests if a token was used (helpful for automated requests to databases) or a specific GitHub account or email were proven to be in the control of the user. This operated like a two-factor authentication enabled login screen, and it took 5 minutes to configure.

4.3 Insecure Exposition Implementation

I set up the insecure implementation from within a VM (using VMware Workstation Player 17) on my personal computer. I only had one Raspberry Pi, so I could not use another dedicated machine. I decided to emulate separate hosts by using a "bridged" network adapter from within VMware Workstaion Player. This allowed my local network to see them as separate devices which meant I could run port scans from my personal computer to the VM. There's an important dynamic here at play as well. With publicly exposed services, the point is that you are leaving exceptions in the firewalls rules which, by default, do not allow unsolicited connections from the public internet (outside the firewall.) If a device is on the same network and has open ports, then there is not a firewall at play and therefore no protections. However, there could be an internal firewall on the actual host which is often referred to as a distributed firewall. This is not the case here, but this setup mirrors that of a service being insecurely exposed to the public internet.

4.4 Port Scanning Results

4.4.1 *Exposed Database Services*

Figure 5 and 6 show the results of scanning the ports of the insecurely exposed services. Not only were the database-related services and their version properly recognized, but the operating system and it's version were also recognized. With this information, any attacker with time could form an attack chain to gain privileged access onto the machine

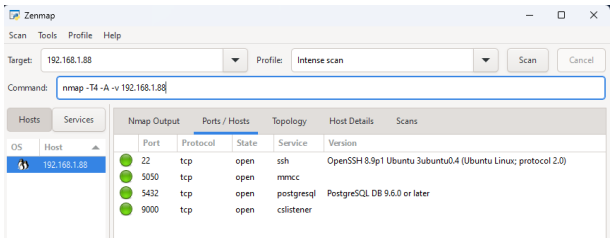

Fig. 5. Insecure Exposition: all services clearly being shown as open ports ready to receive malicious requests.

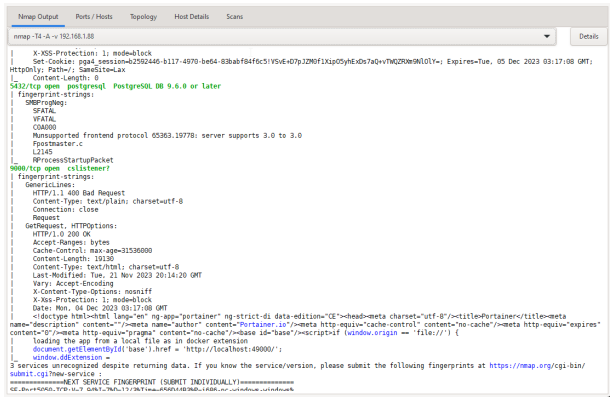

Fig. 6. Insecure Exposition: Postgres service scan results. Portainer is also shown on a banner being returned by a request to port 9000.

and deploy ransomware to encrypt the database data (from the OS level) or deploy something like "PGminer" to mine cryptocurrency.

4.4.2 *Secure Database Services*

The secured database services (Fig. 7) are being proxied by Cloudflare Tunnels and secured with an authentication and authorization layer implemented with Cloudflare Access. When scanned, the only ports that are open are those that return the login pages to authenticate to the database services. Therefore, no OS,services, or versions of each are exposed.

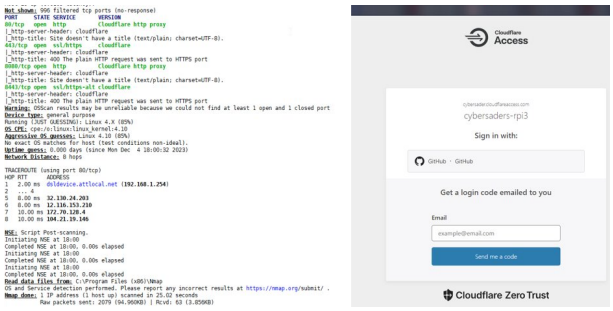

Fig. 7. Secure Exposition: returned a login screen asking to log in to GitHub as a form of authentication. Nmap results are the HTML fromt he authentication page provided by "Cloudflare Access" services.

5 Conclusion

5.1 Results

The results quickly show how reconnaissance activities can be countered with a simple agent-based system like Cloudflare Tunnels. The risks for exposing services to the internet are vast because it is the first step in any attack to find vulnerable targets which can be exploited. By exposing services and information about them or the OS they are running on, organizations risk great financial impact which could result from having data encrypted on the machine running the database or the database performing badly due to botnet activity.

5.2 Exposition Method Comparison

While VPNs and firewall rules offer robust security, Cloudflare Tunnels emerge as the optimal choice for smaller use cases, one-off projects, or uncomplicated networks. Unlike VPNs or firewall rules, which involve distributed administration and are typically accessible to entities with substantial resources, Cloudflare Tunnels simplify the process by consolidating administration within their web application.

Admittedly, Cloudflare Tunnels have the drawback of necessitating an agent for each service. Nevertheless, this agent-based system, utilizing Docker, proves highly efficient for smaller networks. In contrast, VPNs and firewall rules demand configurations across routers and interfaces spread throughout the network, making them less practical for streamlined administration.

In summary, Cloudflare Tunnels stand out as the superior option for self-hosting services on a home network, offering a secure means of exposing them to the public internet. This solution comes with the added benefits of simplified administration, authentication, and authorization at a minimal cost.

References

[1] Oracle, "What is a database?" www.oracle.com, 2022. [Online]. Available: https://www.oracle.com/database/what-is-database/

[2] HackTricks, "5432,5433 - pentesting postgresql - hacktricks," Hacktricks.xyz, 2023. [Online]. Available: https://book.hacktricks.xyz/network-services-pentesting/pentesting-postgresql

[3] GeeksforGeeks, "Tcp/ip model - geeksforgeeks," GeeksforGeeks, 08 2019. [Online]. Available: https://www.geeksforgeeks.org/tcp-ip-model/

[4] C. Limpalair, "Data encapsulation and de-encapsulation," Cybr, 09 2021. [Online]. Available: https://cybr.com/courses/networking-fundamentals/lessons/data-encapsulation-and-de-encapsulation/

[5] K. Mailsamy, "5-layer network model made simplified!" Medium, 07 2020. [Online]. Available: https://medium.com/@karthikayanmailsamy/5-layer-network-model-made-simplified-e813da0913ba

[6] J. Wherry, "What is port forwarding? what is it used for?" CyberNews, 09 2021. [Online]. Available: https://cybernews.com/what-is-vpn/port-forwarding/

[7] G. much!, "Nat — snat, dnat, pat port forwarding," Networks Security, 06 2021. [Online]. Available: https://medium.com/networks-security/nat-snat-dnat-pat-port-forwarding-b7982fab02cd

[8] R. Ostapiuk, "Introduction to tcp/ip (part 3) - client server model - developer help," Microchipdeveloper.com, 11 2023. [Online]. Available: https://microchipdeveloper.com/xwiki/bin/view/applications/tcp-ip/client-server/

[9] Jeremy, "Say goodbye to reverse proxy and hello to cloudflare tunnels," Noted, 06 2022. [Online]. Available: https://noted.lol/say-goodbye-to-reverse-proxy-and-hello-to-cloudflare-tunnels/

[10] j. nazario, "Awesome internet scanning," GitHub, 12 2023. [Online]. Available: https://github.com/paralax/awesome-internet-scanning

[11] Nate, "Recon tools," GitHub, 10 2023. [Online]. Available: https://github.com/nateahess/awesome-recon-tools

[12] 0xedward, "0xedward/awesome-infosec," GitHub, 12 2023. [Online]. Available: https://github.com/0xedward/awesome-infosec#recon

[13] S. Venkitesh, "How my server got infected with a crypto mining malware and how i fixed it," BigBinary, 09 2022. [Online]. Available: https://www.bigbinary.com/blog/how-my-server-got-infected-with-a-crypto-mining-malware-and-how-i-fixed-

[14] P. I. LLC, "Cost of a data breach study," www.ibm.com, 2022. [Online]. Available: https://www.ibm.com/sg-en/security/data-breach

[15] R. McNeil and V. Ravichandran, "Using cloudflare tunnel and access with postgres," The Cloudflare Blog, 06 2022. [Online]. Available: https://blog.cloudflare.com/cloudflare-tunnel-for-postgres/

[16] R. Pasnau, "Thomas aquinas," Stanford Encyclopedia of Philosophy, 2023. [Online]. Available: https://plato.stanford.edu/entries/aquinas/#:~:text=Aquinas%20believes%20that%20natural%20reason

[17] D. Fudenberg and J. Tirole, *Game theory*. MIT press, 1991.

[18] J. Roach, "Kant's categorical imperative," Veritas Vincit Tenebram, 05 2010. [Online]. Available: https://sophoslogos.wordpress.com/2010/05/19/kant%E2%80%99s-categorical-imperative/#:~:text=Something%20justifies%20our%20actions%20other

[19] "Certified devices," Ubuntu. [Online]. Available: https://ubuntu.com/certified/iot?q=&limit=15&category=Ubuntu+Core&vendor=Raspberry+Pi+Foundation

[20] C. Stream, "On the meaning of the latin "securitas" in seneca," Seneca and spiritual direction (philosophy as a way of life), 10 2022. [Online]. Available: https://medium.com/seneca-and-spiritual-direction-philosophy-as-a-way/on-the-meaning-of-the-latin-securitas-in-seneca-8dd3c354a94b

[21] "cyber- — etymology, origin and meaning of cyber- by etymonline," www.etymonline.com. [Online]. Available: https://www.etymonline.com/word/cyber-

[22] K. Shortridge, "What "security" means in the information society (track vi)," Kelly Shortridge, 10 2022. [Online]. Available: https://kellyshortridge.com/blog/posts/what-security-means-in-the-information-society-part-6/

[23] J. Pawlick, E. Colbert, and Q. Zhu, "A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy," *ACM Computing Surveys*, vol. 52, pp. 1–28, 08 2019.

[24] "Postgresql postgresql : List of security vulnerabilities," www.cvedetails.com. [Online]. Available: https://www.cvedetails.com/vulnerability-list/vendor_id-336/product_id-575/Postgresql-Postgresql.html

[25] Wiz, "Extrareplica - azure postgresql cross-account database vulnerability walkthrough," www.youtube.com, 2022. [Online]. Available: https://www.youtube.com/watch?v=FlYs05NXqOQ&list=PLAo444udA0qyan41bUMRNrH1idRk3GsrV&index=2&t=35s

[26] BlackHat, "Pwning cloud vendors with untraditional postgresql vulnerabilities," www.youtube.com, 2022. [Online]. Available: https://www.youtube.com/watch?v=X6CiUD3EcfQ&list=PLAo444udA0qyan41bUMRNrH1idRk3GsrV&index=4&t=549s)

[27] pentestmac, "Exploiting postgresql with metasploit and kali linux," www.youtube.com, 2021. [Online]. Available: https://www.youtube.com/watch?v=qtNkE1mHu-A&list=PLAo444udA0qyan41bUMRNrH1idRk3GsrV&index=7

[28] nixawk, "pentest-wiki," GitHub, 2020. [Online]. Available: https://github.com/nixawk/pentest-wiki/blob/master/2.Vulnerability-Assessment/Database-Assessment/postgresql/postgresql_hacking.md

[29] S. Yadav, "Ultimate guide: Postgresql pentesting," Medium, 04 2020. [Online]. Available: https://medium.com/@lordhorcrux_/ultimate-guide-postgresql-pentesting-989055d5551e

[30] Nick, "Exploiting postgresql with metasploit," pentesthacker, 12 2020. [Online]. Available: https://pentesthacker.wordpress.com/2020/12/30/exploiting-postgresql-with-metasploit/

[31] HackTricks, "Rce with postgresql extensions - hacktricks," Hacktricks.xyz, 2023. [Online]. Available: https://book.hacktricks.xyz/pentesting-web/sql-injection/postgresql-injection/rce-with-postgresql-extensions

[32] Cloudflare, "What is transport layer security? — tls protocol — cloudflare," *Cloudflare*, 2021. [Online]. Available: https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/
[33] S. Helme, "My tls conundrum and why i decided to leave cloudflare," Scott Helme, 02 2014. [Online]. Available: https://scotthelme.co.uk/tls-conundrum-and-leaving-cloudflare/
[34] R. Owl, "Raspberry pi home server - docker, portainer, plex, wordpress, and more," www.youtube.com, 2021. [Online]. Available: https://www.youtube.com/watch?v=yFuTAKq_j3Q&t=45s
[35] Emmet, "Dealing with the low voltage warning on a raspberry pi," Pi My Life Up, 09 2021. [Online]. Available: https://pimylifeup.com/raspberry-pi-low-voltage-warning/
[36] U. , "How to install ubuntu server on your raspberry pi," Ubuntu. [Online]. Available: https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#1-overview
[37] P. Frazao, "Happy pi day with docker and raspberry pi — docker," www.docker.com, 03 2019. [Online]. Available: https://www.docker.com/blog/happy-pi-day-docker-raspberry-pi/
[38] "Docker hub," hub.docker.com. [Online]. Available: https://hub.docker.com/_/postgres
[39] L. Systems, "Using cloudflare tunnels for hosting certificates without exposing ports on your firewall," www.youtube.com, 2023. [Online]. Available: https://www.youtube.com/watch?v=eojWaJQvqiw&t=346s
[40] C. Lempa, "How to use cloudflare tunnel in your homelab (even with traefik)," www.youtube.com, 2023. [Online]. Available: https://www.youtube.com/watch?v=yMmxw-DZ5Ec&list=PLAo444udA0qyan41bUMRNrH1idRk3GsrV&index=14&t=696s
[41] "Portspoof - a new approach to fight back port and service scanners." drk1wi.github.io. [Online]. Available: https://drk1wi.github.io/portspoof/
[42] "Nmap," GitHub, 10 2021. [Online]. Available: https://github.com/nmap/nmap
[43] "What is a reverse proxy? — proxy servers explained — cloudflare," *Cloudflare*. [Online]. Available: https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/
[44] T. M. Corporation, "Car-2021-01-001: Identifying port scanning activity," MITRE Cyber Analytics Repository. [Online]. Available: https://car.mitre.org/analytics/CAR-2021-01-001/
[45] "Cloudflare tunnel easy setup — crosstalk solutions," crosstalksolutions, 03 2023. [Online]. Available: https://www.crosstalksolutions.com/cloudflare-tunnel-easy-setup/