**Network Dump Script**

Justice Escalante

Linux Admin

For this script I used a handful of networking commands that I thought would be useful for network dumping such as ***netstat***, ***ip addr*** and cat. Overall, this script is quite simple, but in conjunction ***with fail2ban*** or other preventative tools you can have your server beefed up. First, I used ***ip addr*** to view network interfaces and ips. Monitoring this is very important as it provides essential network configurations that could help with diagnostics or attackers. As ***ip addr*** displays your whole set of configurations, this is handy for documentation and overall troubleshooting(you can even set up ***ip addr*** with scripting to do routine checks). The next part of the script involves the useof ***netstat*** and a few iterations to pull different stuff such as ports, routing tables. I decided that using open ports would be very beneficial as certain open ports could provide an entrance to attacker, so monitoring that is key. The kernel routing table is another great portion of information to use as it displays how packets are routed and to where. This is vital as you can look for unwanted network changes or disturbances that could ultimately be bad. There are other tools like ***iproute2*** that can also help immensely when you monitor routing as you can see socket stats, and configure network bridges. The routing table’s info alone is not sufficient to deem attackers, but combined with other tools or commands its really useful. To pull the routing ***table sudo netstat -tnlpr*** was used, using ***sudo*** with ***netstat*** is important as higher permissions will display more network info as it could be sensitive and more helpful(PID and its names). To display the ports I used ***netstat*** again, but used the operand ***-tuln for*** active connections and sockets + some and ports. Knowing the active ports and sockets is crucial as these could be entry points for potential unwanted attackers. In this script I have also included DNS info as an unproper management can cause massive breaches and even terrible network health. To pull the DNS information we can visit ***/etc/resolv.conf***, but to edit this we must use the actual configuration file that generates the data(***/etc/systemcd/resolved.conf***). For this script I tried to keep it on the simpler side and used more Linux specific tools for pulling data instead of distro specific. This can help a lot as we can run this script on multiple distros without needing to install a bunch of stuff. Even that certain tools/programs are distro specific so that can cause a headache.

**Bibliography**

<https://www.youtube.com/watch?v=uxq8YXQv-oI>

<https://blog.eduonix.com/shell-scripting/learn-to-write-linux-shell-scripts-for-network/>

<https://mirror.apps.cam.ac.uk/pub/doc/redhat/redhat7.3/rhl-rg-en-7.3/ch-networkscripts.html>

<https://www.youtube.com/watch?v=eXcrRr2o1HE>

<https://www.youtube.com/watch?v=Yr6qI6v1QCY>

<https://www.techrepublic.com/article/understand-the-basics-of-linux-routing/>

<https://www.cyberciti.biz/faq/what-is-a-routing-table/>

<https://www.cs.unh.edu/cnrg/people/gherrin/linux-net.html>

<https://greencloudvps.com/what-is-ip-address-the-importance-of-ip-addresses.php>

<https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en>

<https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address>

<https://smallbusiness.chron.com/importance-knowing-ip-address-69077.html>

<https://www.security.org/vpn/what-can-someone-do-with-your-ip/>

<https://www.upguard.com/blog/open-port#:~:text=As%20outlined%20above%2C%20open%20ports,has%20poor%20network%20security%20rules>.

<https://askubuntu.com/questions/152593/command-line-to-list-dns-servers-used-by-my-system>

<https://www.itsasap.com/blog/what-is-dns>

<https://www.digitalocean.com/community/tutorials/how-to-use-iproute2-tools-to-manage-network-configuration-on-a-linux-vps>