“Design of the Kali Linux Operating System Performance Evaluation and Analysis.”

By Eric Webb

**Abstract:**

Every so often there will be a new flavor of Linux that sets a standard of what an Operating System can be. This literature will discuss one of those flavors being Kali Linux. It will also begin to cover specific technical analyses of default programs embedded within the Kali operating system and evaluate their performances. This will be conducted through peer reviewed research provided from the IEEE or ACM journal digital libraries. Academic study will be applied to previously written content and projects to develop an educated analysis and evaluation about the programs and their subject matters at hand along with an opinionated discussion about the future of this open source project.

**Introduction:**

Kali is a distribution of Linux designed for penetration testing and digital forensics. This security-focused operating system is managed through Offensive Security LTD and is essentially an upgrade from its predecessor called Backtrack (Gaddam & Nandhini, 2017. Pg. 3).

Kali Linux has become a powerful tool among many technical spectrums. Having mastery in many of these technical facets can lead to what can be compared to as a very experienced driver behind a very fun but dangerous well-tuned car. Meaning Kali Linux can be great for learning and offensive security practices within ethical hacking, but can ultimately be used for malicious intent with the wrong user.

This flavor of Linux has become popular within the ethical hacking community because it has the ability to boot off of an external device such as a USB device, CD-ROM, or a Virtual Machine. This has become beneficial because security professionals can now carry these tools within Kali’s operating system anywhere with them. This ability to boot up Kali and conduct research from a multitude of devices instead of a single dedicated device makes your testing more portable and available.

The tools already embedded within Kali’s Debian based operating system contain many techniques and practices for digital forensics and network security. Giving Kali so many options across these spectrums gives it the flexibility to meet the demands of penetration testing. This literature will now discuss these features of Kali Linux, where they currently fit in society, why they are important, and the potential future endeavors of the mentioned technology.

**Use Case 1: Packet Crafting with Hping3.**

The Kali operating system comes with the ability to packet craft right out of the box. One means of achieving this is using the Hping3 tool that is already pre-installed. Hping3 allows you to create a multitude of packets to be sent out in different configurations. This is important to the network security and ethical hacking realm because it allows administrators to test their network entry points and see if these entry points can be exploited.

A real world case example can be demonstrated in a network lab where a virtual machine running Kali Linux can create multiple TCP packets via a Spoofed IP address using Hping3.They then try to create multiple connections to a TCP port. Doing this TCP flood fast enough can exhaust resources if the network is not properly configured and can cause a denial-of-service attack to occur. This DOS attack that can be accomplished by Hping3 packet crafting is powerful tool that can be taken advantage of in the Kali Linux Operating system. It is important because it allows administrators to offensively probe their security scope but can also be used nefariously to bring down systems (Cui, Liu, Zheng, & Huang, 2018. Pg. 3).

As packet crafting software becomes more popular administrators will have to design their systems to protect against high amounts of traffic. The installation and use of load balancers will become ever more present in network design so that TCP floods don’t effect the network

**Use Case 2: Intrusion Detection with Snort.**

Not only does Kali offer strictly offensive security tools embedded within its out of the box operating system, but it also offers techniques for network security as well. This is apparent in a tools called Snort. Snort is an intrusion detection system that allows the administrator to set up rule tables to allow alerts for nefarious activities. These alerts can also trigger events down the line to execute procedures to mitigate these attacks. Alerts can also be set as logs for auditing purposes.

This can be demonstrated by using Snort to set up rules to alert administrators of possible harmful traffic within your network. For example you can set up Snort on your server to allow you to get notifications from events on your network such as OS fingerprinting attemps, Buffer Overflows, Stealthy Port scans, and more (Gaddam & Nandhini, 2017. Pg. 3). One can use the Intrusion detection in many ways. Some examples would be creating snort based rules of of a honeypot results. Another was to create rules based off of efficient port scanning detection rules by getting the results of port scans. Snort is not just limited to network and host based intrusion detection but can also be applied in cloud environments (Gaddam & Nandhini, 2017. Pg. 4).

This is important because you would typically have your statefull Snort IDS behind your stateless firewall. The stateless firewall handles and filters a lot of traffic independent of whats inside the packets. Once the initial filtering is complete the importance of the Snort IDS is applied. Snort will then take those accepted packets perform a statefull analysis fully analyzing the content of the packet. A lot of times using signature or anomaly based detection they can detect a packet that made it pass the firewall that was nefariously crafted. That is why it is important for the IT infrastructure to actively have traffic passing through an IDS to protect against malicious packets that have intruded your network.

As packet crafting becomes more prevalent and new ways are created to breach networks IDS’s will have to grow. This will most likely mean larger and more cumbersome libraries of known signatures. Since Snort can use a signature based libraries to detect known intrusion it will no doubt increase in size as time goes on making scans more resource and time exhausting.

**Use Case 3: Network Attack with Ettercap.**

Another powerful tool in the Kali Linux OS distributions’ arsenal is a multipurpose tool called Ettercap. This tools has many capabilities relating but not limited to packet sniffing, packet filtering, DNS spoofing, forging SSL certificates, and even offers a suite for attacking the Point-to-Point Tunneling protocol. Because this tools has many capabilities it has been known to discover exploits in networking technologies which has led to the advancement of best practices in security. This literature will looks at a use case examples using Ettercap.

One documented example of this tools with another tool called SSL-strip were used to exploit an implementation of PPTP. Essentially SSL-strip forces requests to HTTP pages that would normally be redirected to HTTPS pages, to stay pointed at their non-secure HTTP counterparts. This means that the login information would no longer be sent over the secure PPTP tunnel but now be sent via the insecure methods of HTTP. After the SSL-strip removed the secure redirect the Ettercap was able to listen in and monitor network traffic in clear, revealing the credentials (Pingle, Mairaj, & Javaid, 2018. Pg. 2).

This is important because it revealed to the world a vulnerability that you did not have to overcome the PPTP tunnel but simply had to stop the user from implementing it via stopping its redirects. It is also a great example of how one of the Kali Linux operating system pre-downloaded features can be used to help benefit the advancement of network security. Because of this flaw of not checking for redirects to HTTPS standards were introduced to further check logins are sent to their HTTPS secure locations. (Pingle, Mairaj, & Javaid, 2018. Pg. 2).

As the future of MITM attacks unfolds more and more protocols and standards will be developed to mitigate them. Every time an exploit is discovered a new procedure will be instilled to protect users against it. This will lead to more cumbersome libraries, procedure, and protocol traffic as time but will be needed for the sake of security.

**Use Case 4: WarDriving with Airmon, BlueNMEA and Kismet.**

As the development of wireless local area networks becomes all more increasing the number of access points left open for vulnerabilities and exploits becomes more prevalent. This leaves attackers with more options to choose from. With some of the tools available in the Kali operating system it is possible to perform war driving reconnaissance and enumeration in a geographical area.

An example of war driving performed in the Kali Linux OS is seen with the following example. Where exploiters were able to use Airmon, BlueNMEA, and Kismet to drive around and gather WLAN information from local entry points. This was done by initially bootking up Kali Linux and enabling monitor mode to sniff wireless packets using airmon-ng. Blootooth and GPS sensors were paired to the instance of Kali. The GPS device was sending GPS information to the Bluetooth device and was being relayed back to the Kali instance viea BlueNMEA an application for sending GPS information. This information was being recorded and read via Kismet to extract this data. From doing this WarDriving exercise the ones administering this were able to find a multitude of WLAN networks and clients. While most of the clients were protected by the secure WPA2 security, some environments did no encryption implementations that could be taken advantage of. (Akram, Saeed, & Daud, 2018 Pg. 3)

This is important because it shows how administrators can set up a scenario using Kali Linux where they could drive around their network perimeter probing the network for weak spots. It also shows how a criminal could theoretically drive around and probe areas for easy entry points with little to no security. Essentially WarDriving for low bearing fruit. Another example is how this can be applied in a militaristic or police stand point. Where authorities could probe the area and get an idea of the number of devices belonging to a target and get an idea of how many communication channels are being used within that target network.

In the future WarDriving practices will become more prevalent and will have its complexity deteriorate as pre-written graphical user interfaces become easier for the novice user to understand. Systems will have to develop self-identifying technology that will have to identify when they are being probed. As Kali Linux evolves there will be great strides in not just improving war driving technologies but also preventing it.

**Use Case 5: Structured Query Language Attack and Defense.**

After all the advancements of good coding practices and knowledge available about network security mistakes are still made and negligence is still applied. Because of this the concepts of attacks like SQL injection are still a discussion. SQL injection allows for information to be obtained or SQL commands to be executed that should not be. This is of course huge security threats that must be dealt with accordingly. In this literature we will see examples of how Kali Linux provides offensive security and automated protection against SQL based attacks.

One example of a SQL injection attack happens when there is not sanitazition between the user input and the database. The information needs to be validated and user input verified. One infamous example using SQL injection can be seen where no sanitaztion occurs and a user can enters 1=1 in an input field. In SQL 1=1 equals true. So regardless of the question if 1=1 ,then the answer is true. This is used in the login where 1=1 is used in the ID field and returns true to being an admin. (Gudipati, Venna, Subburaj, & Abuzaghleh, 2016 Pg. 3)

(Gudipati, Venna, Subburaj, & Abuzaghleh, 2016 Pg. 3)

**- methods used to measure/analyze**

**- why its important**

**- future ?**

**Summary of findings**

**Prediction of future.**

**Work Cited.**

Akram, Z. A., Saeed, M. A., & Daud, M. A. (2018, April 6). *Wardriving and its Application in Combating Terrorism.* doi: 10.1109/CAIS.2018.8442035 (Akram, Saeed, & Daud, 2018)

Cui, Y., Liu, Q., Zheng, K., & Huang, X. (2018, December 27). *Evaluation of Several Denial of Service Attack Methods for IoT Systems.* doi: 10.1109/ITME.2018.00179(Cui, Liu, Zheng, & Huang, 2018)

Gaddam, R. T., & Nandhini, M. (2017, July 17). *An analysis of various snort based techniques to detect and prevent intrusions in networks proposal with code refactoring snort tool in Kali Linux environment.* doi: 10.1109/ICICCT.2017.7975177 (Gaddam & Nandhini, 2017. Pg. 3)

Gudipati, V. K., Venna, T., Subburaj, S., & Abuzaghleh, O. (2016, October 15). *Advanced automated SQL injection attacks and defensive mechanisms.* doi: 10.1109/CT-IETA.2016.7868248 (Gudipati, Venna, Subburaj, & Abuzaghleh, 2016)

Pingle, B., Mairaj, A., & Javaid, A. Y. (2018, October 22). *Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use.* doi: 10.1109/EIT.2018.8500082 (Pingle, Mairaj, & Javaid, 2018)

Scott, B., Xu, J., Zhang, J., Brown, A., Clark, E., Yuan, X,Williams, K. (2017, December 14). *An interactive visualization tool for teaching ARP spoofing attack.* doi: 10.1109/FIE.2017.8190531

(Scott, B. Zhang, J. Brown, A. Clark, E. Yuan, X. Williams, K., 2017)

Rehman, U. U., & Abbasi, A. G. (2014, December 9). *Security analysis of VoIP architecture for identifying SIP vulnerabilities.* doi: 10.1109/ICET.2014.7021022

(Rehman & Abbasi, 2014)