ReconChimp: Fast Reconnaissance Framework

1st Kushal
B.Tech Student, SCOPE
Vellore Institute of Technology
Vellore, India
kushal2018@vitstudent.ac.in

2nd Siddhartha Varma
B.Tech Student, SCOPE
Vellore Institute of Technology
Vellore, India
siddhartha.varma2018@vistudent.ac.in

3rd Akshit Kumar
B.Tech Student, SCOPE
Vellore Institute of Technology
Vellore, India
akshit.kumar2018@vitstudent.ac.in

***Abstract*—An automated reconnaissance framework meant for fast enumeration of an organization. It will have customizable scan engines, which can be used to enumerate the sub-domains, endpoints, or gather information. The beauty of ReconChimp is that it will gather everything in one place with a great speed and will be less resource intensive. This framework will be more focused on passive reconnaissance techniques such as google dorking, javascript file enumeration, censys and wayback url results to speed up the complete process of reconnaissance rather than active reconnaissance techniques like direct brute forcing etc. It's a smart recon framework which is made with speed and on the point results in mind. It will have a pipeline of reconnaissance, which will be highly customizable. Additionally, it'll also be offered as a service which can be consumed as an API.**

***Index Terms*—passive reconnaissance, ethical-hacking, JS file enumeration**

I. Introduction

We found out that a lot of security researchers, cyber security evangelists use different open source platforms for osint/recon so that they can acquire information about a target machine. This information includes IPs, subdomains, hosts, ports, crawler information, redirects, HTTP status codes and host certificate information. But this is a long and timetaking task because we have to use numerous tools for getting this data. Hence, we came up with the idea of reconchimp, which compiles all of these services into an API, a CLI and an interface so that a researcher is able to focus on the task of fixing vulnerabilities while leaving these high effort, highly timetaking tasks to ReconChimp.

II. Literature Review

This paper aims at finding best tools that are performing in stealthy scan ie. a scan that is able to find all the assets and domains for any target and the second comparison done in this paper is to find which tool is making least noise so to remain undetected in the scanning scenario. It has compared many tools on these 2 parameters namely Nmap, BeeF, Nikto scan, WhatWeb etc, It has only used 2 parameters and has missed the another very important parameter of speed. [1] This paper aims at passive reconnaissance via social media of the user and then that passively mined data will be used in social engineering to trick the user in fraudulent activities. They try to find Personal Identity Information of the user from the sites by simplifying analyzing them and then after collecting the same from multiple sources they get to the point where it's possible to fool the victim and perform social engineering, One major limitation is that the audience targeted were from the college itself and hence a week sample space was used. It would be better if that was performed on very random targets and too on a bigger set of people. [3] This paper's main objective is to find best practices that we can use to perform nmap scan, it showcases many methods and commands to how to use nmap when dealing with a larger organization. It also analyzes performance of different commands and hence provides a full analysis of passive reconnaissance using nmap, The very limitation of this paper is that it only analyzes nmap and has excluded other variants or modified versions of the nmap that is rustscan which is a highly performant variant of nmap and is open source. [5] This paper explores different kinds of reconnaissance techniques that are used by an attacker or hacker to collect information regarding the target. This study further determines which recon technique gathers the most information about the target while keeping its identity hidden, This paper has let us understand different reconnaissance techniques available. It helped us understand working and limitations of each method explored. [8] Treemaps have received wide attention due to its simplicity, reduced visual complexity, and compact use of the available space. A few different types of Treemap algorithms have been proposed, however the core idea is the same that is to divide the visual space into rectangles with areas proportional to some data attribute or weight. In this paper, authors propose a novel approach, called Neighborhood Treemap (Nmap), that seeks to solve this limitation by employing a slice and scale strategy where the visual space is successively bisected on the horizontal or vertical directions and the bisections are scaled until one rectangle is defined per data element. Compared to the current techniques with the same similarity preservation goal, this approach presents the best results while being two to three orders of magnitude faster but the Nmap only supports ethernet interfaces (including most 802.11 wireless cards and many VPN clients) for raw packet scans. Unless you use the -sT -Pn options and RAS connections. Nmap must send lower-

level ethernet frames instead. When using Nmap with WinPcap instead of Npcap, you cannot generally scan your own machine from itself (using a loopback IP such as 127.0.0.1 or any of its registered IP addresses). [15] This article is majorly focused on the reconnaissance phase, which is the basis for the totality of cybersecurity attacks. As a general trend, the evolution of smart devices, social media, and IoT capable applications, boosted the amount of information that can be gathered by an attacker and also multiplied the communications paths that can be used to reach the victim, Rethink the concept of privacy in a more broad manner to also include protection mechanisms against advanced and malicious data gathering campaigns. [20]

III. Architecture

A user starts by providing a hostname or URL to a CLI interface, which then connects to the ReconChimp API, sends the domain to the API, and asks user to check the details later. Now, the API communicates with the core logic module, which includes subprocesses like subdomain enumeration, redirect rules, status codes, certificate information, web crawler data, all coming from different services in our suite of tools. As soon as this is over, reconchimp then fires the report generation microservice which helps to create a document with all of the endpoint information compiled. As soon as that is over, the subprocesses fire up a notification microservice which sends a notification to telegram, informing the researcher that the scan is complete, and sends the report along with that. The user can then interact with the CLI or the UI to view to report. The generated report could also be sent to a Telegram group.

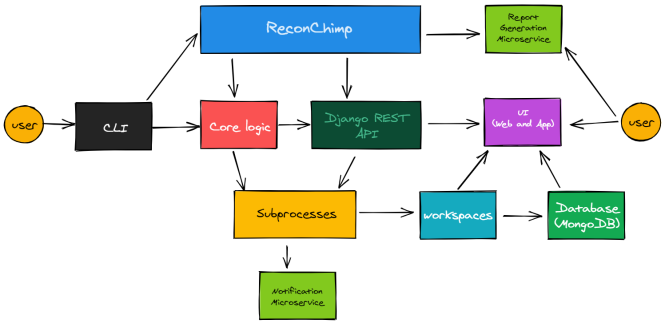

Fig. 1. Architecture of ReconChimp.

A. Core Logic

The core logic of the ReconChimp consists of several scripts that run different recon services simultaneously. The idea behind this was that one service o its own does not offer much insight into the various subdomains, IPs, host, certificate information. These scrapes are called by the CLI. The scraping is done and the results are stored in text files

B. Subprocesses

As part of the subprocesses we are doing subdomain enumeration from best passive resources publically available, we are using web-scraping techniques to extract the desired information. We are also doing Subdomain Filtration via httpx tool. httpx is highly fast and is a multi-purpose HTTP toolkit, that provides us features like running multiple probes using retry-able http library which is designed to maintain the result reliability with increased number of threads. Its simple and modular code base makes it easy to use. It has fast and fully configurable flags to probe multiple elements. It supports multiple HTTP based probings and also supports hosts, URLs and CIDR as input. It has smart auto fallback from https to http as default. It also handles edge cases doing retries, backoffs etc for handling WAFs. We are also doing directory Fuzzing using ffuf, a fast web fuzzer written in Go programming language. We are doing parameter mining using Arjun, It is an open-source tool that can find query parameters for URL endpoints.

C. Subdomain Enumeration

We are using multiple publically available tools for sub-domain enumeration. We are using Subdomainfinder.c99.nl, which is a tool which performs an advanced scan over the specified domain and tries to find as many subdomains as possible, while also checking whether the domain is tunneling through CloudFlare. It might also include some historical data since that might contain useful information, for more accurate results. We are also using crt.sh which finds hidden endpoints via digital certificate analysis. We are using DNSdumpster to find visible hosts from the attackers perspective is an important part of the security assessment process. We are searching through ThreatCrowd for any previously mentioned vulnerabilities. And finally we are also scrapping data from Shodan, which is the world's first search engine for Internet-connected devices.

D. Notifications

We also provide notifications microservice so as to generate alerts when the reconnaissance work is done. Along with the notification, the file(s) generated are sent to the user (via Telegram).

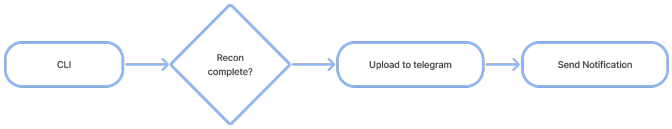

Fig. 2. Notification Flowchart.

E. Report Generation

ReconChimp will generate a detailed report contained all the scrapped and acquired data. It will contain all the end-points, sub-domains, and directories. The generated report will be sent to the user via the notification.

IV. Results and Analysis

This project aimed at creating a standalone tool for content discovery specifically focused for bug bounty hunters. It's comparison is best done with tools that are designed for similar purpose and uses passive recon where fuzzing the sub-domain is not an option. For our analysis we shall have some

parameters to measure the Accuracy and Digital Infrastructure discoverability which we are calling Coverage here.

$$Accuracy = \frac{Allcorrectreceivedresults}{Allthereceivedresults} \quad (1)$$

$$Coverage = \frac{Results}{MaxResults} \quad (2)$$

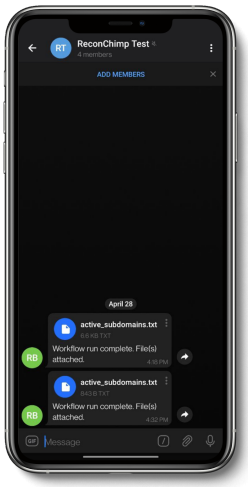

Fig. 3. Report being sent to telegram.

Tool Name	Results	200/403 Results	Accuracy	Coverage
Recon Chimp	33	11	33.3 %	100 %
subdomainfinder.c99	28	10	35.7 %	84.8 %
crt.sh	31	10	32.2 %	93.9 %
dns-dumpster	8	5	62.2 %	24.2 %
threatcrowd	0	0	-	-

TABLE I: Comparison Table among different tools after testing on "dcsvit.com"

Tool Name	Results	200/403 Results	Accuracy	Coverage
Recon Chimp	573	421	73.4 %	100 %
subdomainfinder.c99	437	352	80 %	76.2 %
crt.sh	428	421	98 %	74.6 %
dns-dumpster	154	87	56.4 %	26.8 %
threatcrowd	62	42	67.7 %	10.8 %

TABLE II: Comparison Table among different tools after testing on "vit.ac.in"

ACKNOWLEDGMENT

We would like to say that this project would not have been feasible without the guidance and support of Prof. Jayakumar K. He taught us the subject "Technical Answers for Real World Problems" and went out of his way to encourage us to look for problems and shortcomings we found in our day-to-day activity and try to come up with practical and innovative solutions.

REFERENCES

[1] Le, Ha and Loh, Peter and Lau, Chiew. (2016). Performance evaluation of cyber reconnaissance tools. International Journal of Information Privacy, Security and Integrity. 2. 177. 10.1504/IJIPSI.2016.078589

[2] A. J. Beecroft and J. B. Michael, "Passive Fingerprinting of Network Reconnaissance Tools," in Computer, vol. 42, no. 12, pp. 91-93, Dec. 2009, doi: 10.1109/MC.2009.405.

[3] S. N. Hidayah Zulkiffli, M. N. Ahmad Zawawi and F. A. Rahim, "Passive and Active Reconnaissance: A Social Engineering Case Study," 2020 8th International Conference on Information Technology and Multimedia (ICIMU), 2020, pp. 138-143, doi: 10.1109/ICIMU49871.2020.9243402.

[4] Ahmed, Sheeraz and Khan, Hamayun and Saeed, Khalid. (2019). Penetration Testing Active Reconnaissance Phase -Optimized Port Scanning With Nmap Tool. 10.1109/ICOMET.2019.8673520.

[5] S. Patil, A. Jangra, M. Bhale, A. Raina and P. Kulkarni, "Ethical hacking: The need for cyber security," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 1602-1606, doi: 10.1109/ICPCSI.2017.8391982.

[6] Y. Wang and J. Yang, "Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool," 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2017, pp. 110-113, doi: 10.1109/WAINA.2017.39.

[7] R. S. Devi and M. M. Kumar, "Testing for Security Weakness of Web Applications using Ethical Hacking," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 354-361, doi: 10.1109/ICOEI48184.2020.9143018.

[8] Dar, Usman and Iqbal, Arsalan. (2018). The Silent Art of Reconnaissance: The Other Side of the Hill. VOL. 6, NO. 12. 250-263.

[9] Sanghvi, H. and Dahiya, Ms. (2013). Cyber Reconnaissance: An Alarm before Cyber Attack. International Journal of Computer Applications. 63. 36-38. 10.5120/10472-5202.

[10] Navjot Kaur and Ms. Gurline Kaur, "Penetration Testing Reconnaissance with Nmap Tool", IJARCS, Volume 8 Issue 3, March-April 2017.

[11] A. Roy, L. Mejia, P. Helling and A. Olmsted, "Automation of cyber-reconnaissance: A Java-based open source tool for information gathering," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017, pp. 424-426, doi: 10.23919/ICITST.2017.8356437.

[12] Garg, Anushka and Dubey, Vandana. (2020). Basics for the Process and Requirements of Ethical Hackers: A Study. 10.1007 / 978-981-15-3369-3_59

[13] Mandal, Nabanita and Jadhav, Sonali. (2016). A survey on network security tools for open source. 1-6. 10.1109/ICCTAC.2016.7567330.

[14] F. Holik and S. Neradova, "Vulnerabilities of modern web applications," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1256-1261, doi: 10.23919/MIPRO.2017.7973616.

[15] F. S. L. G. Duarte, F. Sikansi, F. M. Fatore, S. G. Fadel and F. V. Paulovich, "Nmap: A Novel Neighborhood Preservation Space-filling Algorithm," in IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 12, pp. 2063-2071, 31 Dec. 2014, doi: 10.1109/TVCG.2014.2346276.

[16] U.S. Department of the Air Force, Air and Space Power Journal, "For and from Cyberspace: Conceptualizing Cyber Intelligence, Surveillance, and Reconnaissance", By Hurley, Matthew M., Air University (U.S.). Press, Air and Space Power Journal November-December 2012), v.26 no.6, p.12-33, November–December 2012

[17] Ndatinya, Vivens and Xiao, Zhifeng and Manepalli, Vasudeva and Meng, Ke and Xiao, Yang. (2015). Network forensics analysis using Wireshark. International Journal of Security and Networks. 10. 91. 10.1504/IJSN.2015.070421.

[18] Medeiros J.P.S., Brito A.M., Pires P.S.M. (2009) A Data Mining Based Analysis of Nmap Operating System Fingerprint Database. In: Herrero Á., Gastaldo P., Zunino R., Corchado E. (eds) Computational Intelligence in Security for Information Systems. Advances in Intelligent and Soft Computing, vol 63. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04091-7_1

[19] Standard, S., Greenlaw, R., Phillips, A., Stahl, D. and Schultz, J., 2013. Network reconnaissance, attack, and defense laboratories for an introductory cyber-security course. ACM Inroads, 4(3), pp.52-64.

[20] Caviglione, Luca. “Cyber Reconnaissance Techniques.” Communications of the ACM, 2021. doi:10.1145/3418293.
[21] R. Masood, Um-e-Ghazia and Z. Anwar, ”SWAM: Stuxnet Worm Analysis in Metasploit,” 2011 Frontiers of Information Technology, 2011, pp. 142-147, doi: 10.1109/FIT.2011.34.