2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)

Knowledge Extraction and Integration for Information Gathering in Penetration Testing

Anis Kothia, Bobby Swar, Fehmi Jaafar
Department of Information Systems Security and Assurance Management
Concordia University of Edmonton, Alberta, Canada
akothiac@student.concordia.ab.ca, {bobby.swar, fehmi.jaafar}@concordia.ab.ca

***Abstract*—Assets identification is an important aspect of penetration test on which security practitioner develop their defense mechanism. In addition, assets identification is an essential piece of information for penetration testers to find a weakness in the targeted organization. Information gathering is the process of extracting knowledge to recognize the organizations' assets available on the internet. There are many open source tools available for information gathering. However, penetration tester needs to put manual effort (during several hours to multiple days) to extract useful knowledge from the output of one tool and integrate that knowledge in another tool. Penetration tester can increase speed and accuracy of the overall information gathering process by automating the knowledge extraction and integration. This paper review and identify open source subdomain enumeration and service scanning tools and present an approach to integrate and automate identified tools. The result reveals that there is a significant improvement of the information gathering process by using our approach due to the reduction of manual tasks.**

***Index Terms*—penetration testing, information gathering, domain, subdomain, services, port, enumeration, integration, automation.**

I. Introduction

A penetration test is a proactive and authorized exercise to break through the security of an IT system. A penetration test contains typically the use of attacking approach performed by a trusted person that is equivalently used by hostile intruders or hackers [1]. Penetration testing defends the organization against failures through proving due diligence and compliance to regulators, stakeholders, and clients [2]. It provides a proof of issue and a solid business case for the suggestion of investment to senior management to maintain the cybersecurity posture of an organization. Hence, a penetration testing should be regularly conducted for an organization to identify the set of vulnerabilities and critical risks in people, process, and technology [2].

Information gathering is one of the most important processes of penetration testing, as it is the first stage in which direct actions against the target are taken. Information gathering requires a scientific approach to find information and it requires penetration tester creativity to explore new arena where more information can be available [3]. Hence, during this stage, the penetration tester investigates each conceivable road to acquiring comprehension of attack surface area and its assets [4].

Searching for a target organization manually over the internet is cumbersome. Thus, there is an enormous demand for doing information gathering using tools. There are many open source tools developed in recent years to search for a specific type of information accurately and faster. However, those tools are not interoperable and require manual effort to manage and use their findings.

The main objective of the paper is to identify, review and evaluate subdomain enumeration and service scanning tools. In addition, this paper proposed the integration of these tools and the automation of the knowledge communications between them in order to improve the efficiency and effectiveness of the information gathering process. Concretely, we are reviewing and identifying the subdomain and service scanning tools that are most efficient and effective in identifying the subdomain and open ports for information systems. Then we integrate these tools using a shell script. We compare the results of the integrated tools with independent tool to evaluate the efficiency and effectiveness of our information gathering approach. We conducted an empirical study that shows that the integration and automation of open source information gathering tools improve the information gathering process by eliminating the manual effort.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents our approach including its components along with the experimental results of a comparative study. We discuss the results of our approach is Section IV. Section V concludes the paper with our future work.

II. Related work

We are reviewing in this section the set of methodologies that focuses on publicly available information and network information gathering. Then we

978-1-7281-3925-8/19/$31.00 ©2019 IEEE
DOI 10.1109/QRS-C.2019.00068

are presenting the different types of penetration testing. Finally, we are discussing the researchers' effort on developing tools used in information gathering.

A. Information gathering methodologies

There are several methodologies to gather information for target organization. The FedRAMP and NIST 800-115 guideline emphasis on publicly available information and network information gathering. The FedRAMP suggests that penetration tester should use Open Source Intelligence (OSINT) Gathering Activities at first stage and enumerated on active network endpoints. Open source intelligence gathering activities used various internet resources to search organization presence on the internet [5]. NIST 800-115 technical guide for information security testing and assessment suggested the use of network discovery with many approaches to detect active and lively hosts on a network, identify shortcomings, and gain knowledge of how the network behaves and operates. Both active and passive methods exist for identifying and discovering machines on a network [6]. Network port and service detection mostly use the IP address outcome of network information gathering as the devices and machine to check and scan. Port discovery scans can be executed separately for a full range of IP addresses. The output of network scan and network service and port discovery is an inventory of all active and lively machine and devices running in the address range that answered to port discovery tool.

Looking for a public profile of the target organization using internet resources is a time-consuming and cumbersome task. For example, searching for target.com subdomain on Google, Shodan, Baidu (Chinese search engine), Virus total, and Netcraft to gather subdomain related information for target organization will take a significant amount of time. A penetration tester must manually test and record the result for each tool. For example, the discovery of the set of IP addresses with specific criteria is a tedious task. For instance, which IP address hosts most number of application and which IP addresses are used as development or test assets. This research paper focuses on identifying tools that can gather information in order to integrate and interlink those tools so that the combined output of these tools can be managed, organized and analyzed easily.

B. Types of penetration testing

As is defined in the Open Source Security Testing Methodology Manual [7], there are six type of penetration testing called as blind, double-blind, grey box, double grey box, tandem, and reversal. In the grey box, double grey box, tandem, and reversal penetrating types, the organization shares the full or some information about its information system with a penetration tester or an ethical hacker. Thus, the penetration testing project contract declares a range of assets that could be in scope. Hence, this research is not applicable to these penetration-testing types. This paper is applicable and useful while performing blind and double-blind penetration testing type as in such types the penetration tester has zero knowledge of the target. Indeed, a penetration tester must spend the maximum time to gather information during the blind and double-blind type of penetration testing.

C. Information gathering tools

There are many ways to gather information for any organization. I addition, there are many open source tools and proprietary tools developed to gather each unique type of information about the organization.

Net-Nirikshak, which is developed by Sugandh Shah [8], works in five stages. Information Gathering, Scanning, Vulnerability Detection, Exploitation, Report Generation. The data flow of Net-Nirikshak identifies valuable knowledge for the information gathering and forward it to scanning phase. This tool uses *Whois* query and banner grabbing for information gathering. The *Whois* is a query and response protocol used for querying server that store registered assignee of internet resources such as domain name, IP address block and other information.

Nikita Jhala et al. [9] reviewed the *dnsenum, dnsmap, dnsrecon* and fierce subdomain enumeration tools. The DNS information provides many insights into the target organization [11]. For example, if the subdomain "dev-webconf.target.com" exists, then it will provide information that this application or server is used as a development server for the target organization. In fact, these DNS enumeration tools used to gather subdomain information using brute force method. These tools used a default wordlist (a set of 5000 words used in brute force). Hence, using a default wordlist of 5000 words does not help in identifying all the assets of an organization that are available on the internet. In addition, DNS enumeration tool using brute force method cannot identify all subdomain of the organization in short time duration if large wordlist used. This restriction limits the attack surface area and hence reduce the chance of successful an attack.

Search engine used to identify assets available on the internet, checking DNS records to identify assets and IP address range can also be used to identify the assets. However, with only the search engine results, the penetration testers devote a meaningful amount of time in manual information gathering tasks. Very few papers discuss integrating various information gathering tools for penetration testing. In [10], the author listed down the drawbacks of manual effort to perform information gathering such as the high investment in human resources, the high time consumption, the lack of reliability, etc.

III. AN INTEGRATED APPROACH

We propose a modular integrated approach for information gathering that enables the adding of more tools in future. Information gathering is never ending process hence it is crucial to understand its requirement, use, and purpose. This research is limited to the applicable item listed in Table I.

TABLE I: RESEARCH SCOPE

Scope Type	Applicable	Partial applicable	Not applicable
Information related	Organization	NA	Person
Type of tools	Open source, Freeware	NA	Commercial, Expensive tool
Penetration testing type	Blind, Double Blind	Double Gray Box, Gray Box	Tandem, Reversal
Information type	Domain identification, Open services	NA	OS discovery, Web server discovery
Target organization capacity	Large and medium size	Small	NA

Our proposed approach follows the following steps:

A. Identification of subdomain enumeration tools and service scanning tool to implement our approach

1) *Subdomain enumeration using internet resources:* In this step, we identify are using subdomain enumeration tools: *Sublist3r* and *enumall* [12] [13]. These two tools are open source and use the set of public resources mentioned in Table II.

TABLE II
INTERNET RESOURCES USED BY SUBLIST3R AND ENUMALL [12] [13]

Sublist3r	Use by both	enumall
Ask	Google	Shodan
Virustotal	Yahoo	Hackertarget API
DNSdumpster	Bing	SSLtools API
ReverseDNS	Baidu	
PTArchieve.com	Netcraft	
	Threatcrowd	

One the one hand, Ask, Google, Yahoo, Bing, and Baidu are a search engine tools. On the other hand, DNSdumpster, ReverseDNS, PTArchieve.com, Netcraft, ThreatCrowd, Shodan, Hackertarget and SSLTOOLS are security-focused organizations that collect, record, and provide information regarding organization subdomains. *Sublist3r* and *enumall* use Google, Yahoo, Bing, Baidu, Netcraft, and ThreatCrowd API to discover subdomain information for an organization. Additionally, Sublist3r uses Ask, Virustotal, DNSdumpster, and ReverseDNS to identify subdomain information available publicly. Similarly, enumall looks for subdomain information from Shodan, Hackertarget, and Ssltools [13].

2) *Subdomain enumeration using DNS query:* In this step, we use the Massdns tool that implements dictionary approach to verify probable subdomain for the target organization. Subdomain enumeration using DNS query tool has been shortlisted based on how quickly the tool resolves the DNS subdomain. Massdns tool was capable of resolving 350,000 per second using publicly available DNS resolver [15]. It is the fastest DNS subdomain resolver as it is able to resolve a massive amount of domain names in the order of millions or even billions [16].
3) *Service scanning:* In this step, we use the Masscan tool that can scan open ports for the more substantial IP address list. To scan one port for 100 of IP addresses takes minutes or hours of time using off-the-shelf tools. Hence, it will takes days or weeks to scan hundreds of ports for 100 IP addresses using off-the-self tools. Masscan puts emphasis on high-performance services scanning tool using gigabit internet speed. It is the fastest port-scanning tool with its capacity to send 10 million packets per second [19].
4) *Integration and automation of tools:* In this step, we developed a shell script to manipulate the input and the output of identified tools. The automation task consists of the sequencing execution of tools. The shell script performs the configuration of the selected tools by executing the transformation, mapping, and cleansing of data generated by the different selected tools. This enable us to connect different information gathering tools and to exchange data through them and across varying formats.

B. Identification of comparison criteria

In this step, we adopted the comparison criteria from Halfond et al. [20]. Yet, we focused on four primary quantitative criteria as below:

1) *Practicality:* It is considered based on the time taken to retrieve the information. This presents the cumulative time for execution of each tool plus manual effort required to extract useful information for another tool. The manual effort that required is extracting subdomain, IP addresses, removing duplicates, and identifying and removing false positive from the output of subdomain enumeration tools.
2) *Extensiveness:* It is considered based on input vector identified by each tool. That led to selecting a number of total subdomains, IP address and port identified to evaluate extensiveness of our approach.
3) *Effectiveness:* It is considered based on the number of subdomains entries that are unnecessary and removed from the original output. The removed entries contain false positive records where subdomains are found, but they are not live/active and subdomain that does not belong to target organization. This led to selecting a number of valid subdomains and number of the invalid subdomains to evaluate the effectiveness of approach.

4) *Usability:* We will evaluate if the output provided by tools provide the capability of filtering or sorting the records.

HackerOne was started by hackers and security leaders who are driven by a passion for making the internet safer. This platform is the industry standard for hacker-powered security. It collaborates with the global hacker community to surface the most relevant security issues of their customers before criminals can exploit it [21]. The organization, which registers with HackerOne are looking to find a weakness in their organization information system, and they allow the registered ethical security researcher and hacker to perform penetration-testing using, provided scope on HackerOne. Hence, organizations registered with HackerOne considered for performing passive and active information gathering.

In this paper, the name of the target organization has been changed to ”org1” to keep confidentiality for the organization. However, the results in the paper are from the real-time experiment.

C. Installation, configuration, and execution of identified tools

```
430 web1000.org1.com.   300 IN  A   142.54.173.92
431 s4155a.org1.com.    300 IN  A   142.54.173.92
432 web637.org1.com.    300 IN  A   142.54.173.92
433
```

Fig. 1. The Output from the brutesubs tool

The installation and the configuration of brutesubs, massdns, and masscan have been implemented on a virtual private server. The Google and Shodan API keys are added in the configuration file for the brutesubs. The wordlist used while searching for subdomain name contains 110000 words. This wordlist is compiled by various security researcher over the years based on words that are commonly used while setting up their DNS records [22].

The empirical study has been conducted in two parts. In the first part, a set of information gathering tools has been executed for a target organization. In the second part, we applied our approach to integrate the identified tools. The brutesubs tool execution time was recorded to be 31 min. It has discovered and recorded 230 subdomains for the targeted organization in a text file. Figure 1 showed the brutesubs tool output. The massdns execution time was recorded to be 1 min 30 second for 110000 words from subdomain wordlist. It has discovered and recorded 432 subdomains for the targeted organization in a text file. The massdns tool output can be referred in Figure 2. The manual effort in the empirical study has been calculated based on the average amount of time taken to identify false positive or duplicate subdomain from brutesubs and massdns output file. The average time for each subdomain was 30 seconds.

The maximum number of subdomain identified in the empirical study result is 432 subdomain. Thus, it can take around 12960 (432*30) seconds or rround 216 minutes to check for each subdomain if it is duplicate, false positive or a valid subdomain.

```
401 specialforces.org1.com. 720 IN  CNAME
    orgllb-specialforces-656516969.us-west-1.elb.amazonaws.com.
402 orgllb-specialforces-656516969.us-west-1.elb.amazonaws.com. 60  IN  A
    184.169.151.208
403 orgllb-specialforces-656516969.us-west-1.elb.amazonaws.com. 60  IN  A
    50.18.125.216
404 benzy.org1.com. 300 IN  A   142.54.173.92
405 13abundance-com.org1.com.   300 IN  A   142.54.173.92
406 svr2.org1.com.  720 IN  A   50.18.47.153
407 pjocuri.org1.com.   300 IN  A   142.54.173.92
408 hanacupid-plus-net.org1.com.    300 IN  A   142.54.173.92
409 hishi-ki-cojp.org1.com. 300 IN  A   142.54.173.92
410 secureportal.org1.com.  300 IN  A   142.54.173.92
411 s-gcglobaladmin.org1.com.   300 IN  A   142.54.173.92
412 picocraft-xsrvjp.org1.com.  300 IN  A   142.54.173.92
413 webdisk.www1.org1.com.  300 IN  A   142.54.173.92
414 hiserve-cojp.org1.com.  300 IN  A   142.54.173.92
415 fogbugz.org1.com.   300 IN  A   142.54.173.92
416 kinkipesodan-xsrvjp.org1.com.   300 IN  A   142.54.173.92
417 pier-s-com.org1.com.    300 IN  A   142.54.173.92
418 trust-cars-com.org1.com.    300 IN  A   142.54.173.92
419 web1078.org1.com.   300 IN  A   142.54.173.92
420 s823z.org1.com. 300 IN  A   142.54.173.92
421 ashula-king-com.org1.com.   300 IN  A   142.54.173.92
422 a88.org1.com.   300 IN  A   142.54.173.92
423 wscr.org1.com.  300 IN  A   142.54.173.92
424 skyofking.org1.com. 300 IN  A   142.54.173.92
425 www.ultimate.org1.com.  300 IN  A   142.54.173.92
426 cssource.org1.com.  300 IN  A   142.54.173.92
427 www.msu.org1.com.   300 IN  A   142.54.173.92
428 nvg.org1.com.   300 IN  A   142.54.173.92
429 distrib.org1.com.   720 IN  A   74.54.23.227
430 web1000.org1.com.   300 IN  A   142.54.173.92
431 s4155a.org1.com.    300 IN  A   142.54.173.92
432 web637.org1.com.    300 IN  A   142.54.173.92
433
```

Fig. 2. The Output from the massdns tool

Figure 3 highlights the set of false positives collected from the massdns tool output that provides invalid subdomains and IP address which does not belong to target organization.

```
197 svr3.org1.com
198 team.org1.com
199 th.org1.com
200 themes.org1.com
201 themes.org1.com.herokudns.com
202 titan.org1.com
203 tp.betazvault.org1.com
204 tp.zvault.org1.com
205 tw.org1.com
206 uk.org1.com
207 us.org1.com
208 v5-dev.org1.com
209 v5-sandbox.org1.com
210 v5-stage.org1.com
211 view.connect.org1.com
212 vn.org1.com
213 wa.betazvault.org1.com
214 wa.zvault.org1.com
215 web1.org1.com
216 web2.org1.com
217 webmail2.org1.com
218 webmail3.org1.com
219 wiki.org1.com
220 wot.org1.com
221 ww2.org1.com
222 www-dev.org1.com
223 www.betazvault.org1.com
224 www.org1.com
225 www2.org1.com
226 www3.org1.com
227 x5.org1.com
228 zd.zvault.org1.com
229 zv.org1.com
230 zvault.org1.com
231
```

Fig. 3. The false positve output from massdns tool output

Figure 4 highlighted the CNAME records that belong to a cloud organization for one subdomain. This CNAME record needs to be removed to identify the unique subdomain found for the target organization.

```
email.org1.com. 600 IN  A   8.23.247.216
email.org1.com. 600 IN  A   209.3.218.216
wiki.org1.com.  720 IN  A   50.17.217.102
ns5.org1.com.   720 IN  A   208.94.149.4
web2.org1.com.  720 IN  A   50.18.121.182
ask.org1.com.   720 IN  CNAME   org1lb-237370467.us-west-1.elb.amazonaws.
org1lb-237370467.us-west-1.elb.amazonaws.com.   60  IN  A   50.18.116.72
org1lb-237370467.us-west-1.elb.amazonaws.com.   60  IN  A   184.72.62.55
org1lb-237370467.us-west-1.elb.amazonaws.com.   60  IN  A   50.18.151.173
org1lb-237370467.us-west-1.elb.amazonaws.com.   60  IN  A   50.18.107.54
```

Fig. 4. The CNAME record that should be removed to identify unique assets

Figure 5 shows a set of false positives from the brutesubs output. A false positive subdomain exists in DNS record but it is inaccessible from the internet. Hence, this subdomain record needs to be skipped.

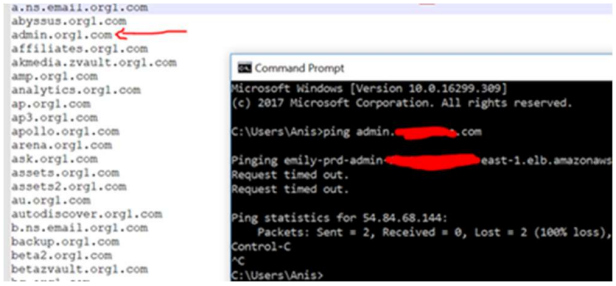


Fig. 5. False positive from brutesubs

Figure 6 shows the duplicated records between brutesubs and massdns output. The manual effort to identify unique IP addresses from brutesubs finding took over 30 min. The masscan has identified 577 open services for the target organization

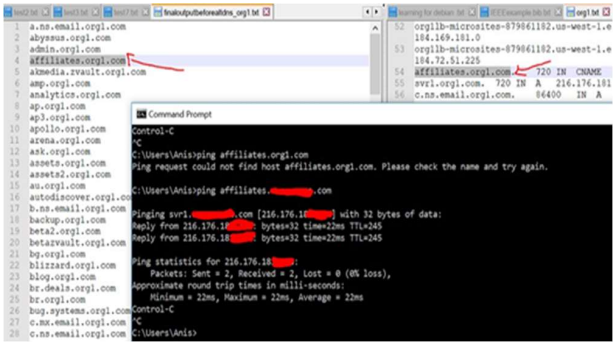
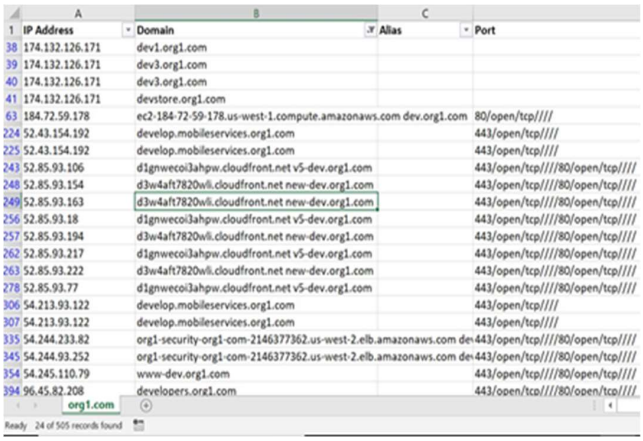


Fig. 6. Duplicate records from brutesubs and massdns tool output

The integrated tool execution time was 45 minutes for the 110000 words from the subdomain wordlist. It has identified 178 subdomains for the targeted organization.

The integrated tool executed masscan tool automatically after identifying the unique IP addresses to scan. It also provided 577 open services for the target organization. This saved 30 minutes of manual effort. The output of each independent tool is in a text file as shown in Figure 1 and Figure 2. A Penetration tester has to put manual effort to identify the relation between individual tools output. For example, to identify unique development server with different IP addresses, penetration tester has to manually go through each file, get all development server records and put in another file. For our empirical study, the penetration tester wanted to check all development environments that can be an easy target to find vulnerabilities.



	A	B	C	
1	IP Address	Domain	Alias	Port
38	174.132.126.171	dev1.org1.com		
39	174.132.126.171	dev3.org1.com		
40	174.132.126.171	dev3.org1.com		
41	174.132.126.171	devstore.org1.com		
63	184.72.59.178	ec2-184-72-59-178.us-west-1.compute.amazonaws.com	dev.org1.com	80/open/tcp////
224	52.43.154.192	develop.mobileservices.org1.com		443/open/tcp////
225	52.43.154.192	develop.mobileservices.org1.com		443/open/tcp////
243	52.85.93.106	d1gnwecoi3ahpw.cloudfront.net v5-dev.org1.com		443/open/tcp////80/open/tcp////
248	52.85.93.154	d3w4aft7820wli.cloudfront.net new-dev.org1.com		443/open/tcp////80/open/tcp////
249	52.85.93.163	d3w4aft7820wli.cloudfront.net new-dev.org1.com		443/open/tcp////80/open/tcp////
256	52.85.93.18	d1gnwecoi3ahpw.cloudfront.net v5-dev.org1.com		443/open/tcp////80/open/tcp////
257	52.85.93.194	d3w4aft7820wli.cloudfront.net new-dev.org1.com		443/open/tcp////80/open/tcp////
262	52.85.93.217	d1gnwecoi3ahpw.cloudfront.net v5-dev.org1.com		443/open/tcp////80/open/tcp////
263	52.85.93.222	d3w4aft7820wli.cloudfront.net new-dev.org1.com		443/open/tcp////80/open/tcp////
278	52.85.93.77	d1gnwecoi3ahpw.cloudfront.net v5-dev.org1.com		443/open/tcp////80/open/tcp////
306	54.213.93.122	develop.mobileservices.org1.com		443/open/tcp////
307	54.213.93.122	develop.mobileservices.org1.com		443/open/tcp////
335	54.244.233.82	org1-security-org1-com-2146377362.us-west-2.elb.amazonaws.com	de	443/open/tcp////80/open/tcp////
345	54.244.93.252	org1-security-org1-com-2146377362.us-west-2.elb.amazonaws.com	de	443/open/tcp////80/open/tcp////
354	54.245.110.79	www-dev.org1.com		443/open/tcp////80/open/tcp////
394	96.45.82.208	developers.org1.com		443/open/tcp////80/open/tcp////

Fig. 7. Integrated Tool Output File Format

Based on the experiment, the recorded result for each identified tool and the integrated tool is shown in Table III.

TABLE III:COMPARISON CRITERIA EVALUATION

Criteria	brutesubs	massdns	masscan	Our tool
Execution time	31 min	1 min 30 sec	15 min	45 min
Manual time	75 min	190 min	15 min	0
Total number of subdomain	230	432	NA	178
Number of invalid subdomains	133	351	NA	0
Number of valid subdomains	97	81	NA	178
Number of open services	NA	NA	577	577
Output format	Text	Text	Text	CSV

IV. Discussion

The manual efforts taken to remove duplicates and false positive from brutesubs and massdns were in order of 60 minutes and 180 minutes respectively. The manual effort to identify unique IP addresses to scan for services was in order of 15 min for each tool. Hence, the total manual efforts for brutesubs and massdns were 75 minutes and 195 minutes respectively. Thus, using three tools requires 270 minutes of manual effort apart from executing the tools. Penetration tester uses five to seven tools on average without any manual effort.

The integrated tool result shows that it saves more than 250 minutes of manual effort of the overall information gathering process. The integrated tool automatically checks for false positives and removes duplicates between various tool outputs. Thus, it improves the effectiveness of the overall information gathering process. The integrated tool provides interactive output format that elevates usability of the overall information gathering process.

.

V. Conclusion

A quite number of open source information gathering tools are available. Many of them are providing different type of results, and thus, penetration testers need to use many of the information gathering tools together frequently. This paper presents an insight into information gathering tools and explores a new avenue of gathering information for penetration testing. Furthermore, we presented an approach that integrates independent open source tools to improve the effectiveness and efficiency of the information gathering process. We observed that a penetration tester have to execute a significant manual effort while performing penetration testing. We performed an empirical study that shows that the integration and automation of open source information gathering tools improve the information gathering process by eliminating the manual effort. In future work, we are exploring the use of our approach to extract tactic knowledge for penetration testing.

ACKNOWLEDGMENT

This paper has been partly funded by Concordia University of Edmonton, an NSERC Discovery Grant, and by the Computer Research Institute of Montréal (CRIM). We gratefully thank the Ministère de l'Économie et Innovation du Québec (MEI) for its generous support.

References

[1] C. Wai, "Conducting a penetration test on an organization," *SANS Institute InfoSec Reading Room*, vol. 1, pp. 1–14, 2002.

[2] A. G. Bacudio, X. Yuan, B.-T. B. Chu, and M. Jones, "An overview of penetration testing," *International Journal of Network Security & Its Applications*, vol. 3, no. 6, p. 19, 2011.

[3] D. Geer and J. Harthorne, "Penetration testing: A duet," in *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*. IEEE, 2002, pp. 185–195.

[4] A. Stoica, "A study on the information gathering method for penetration testing," *Security Engineering*, vol. 5, no. 5, pp. 411–418, 2008.

[5] P. FedRAMP, "Fedramp penetration test guidance," *The Federal Risk and Authorization Program*, 2015.

[6] K. Stouffer, J. Falco, and K. Scarfone, "Nist sp 800-115: technical guide to information security testing and assessment," *National Institute of Standards and Technology*, 2008.

[7] P. Herzog, "Open-source security testing methodology manual," *Institute for Security and Open Methodologies (ISECOM)*, vol. 3, 2010.

[8] S. Shah and B. Mehtre, "An automated approach to vulnerability assessment and penetration testing using net-nirikshak 1.0," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*. IEEE, 2014, pp. 707–712.

[9] N. Y. Jhala, "Network scanning & vulnerability assessment with report generation," Master's thesis, Department of Computer Science and Engineering,Nirma University, Gujarat,India, 2014.

[10] R. Sharma, "Quantitative analysis of automation and manual testing," *International Journal of Engineering and Innovative Technology*, vol. 4, no. 1, 2014.

[11] D. K. Pon, E. Manousos, C. Kiernan, B. Adams, M. Chiu, and J. Edgeworth, "System and method of identifying internet-facing assets," Apr. 21 2016, uS Patent App. 14/520,029.

[12] A. Aboul-Ela, "Sublist3r," https://github.com/aboul3la/Sublist3r, 2017.

[13] J. Haddix, "enumall," https://github.com/jhaddix/domain/blob/ master/enumall.py, 2017.

[14] A. Bhartiya, "brutesubs," https://github.com/anshumanbh/brutesubs, 2017.

[15] N. Walsh. Use massdns. [Online]. Available: http://offsecbyautomation.com/Use-MassDNS/

[16] B. Blechschmidt, "massdns," https://github.com/blechschmidt/massdns, 2016.

[17] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[18] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet- wide scanning and its security applications." in *USENIX Security Sym- posium*, vol. 8, 2013, pp. 47–53.

[19] R. D. Graham, "masscan," https://github.com/robertdavidgraham/masscan, 2014.

[20] W. G. Halfond, S. R. Choudhary, and A. Orso, "Improving penetration testing through static and dynamic analysis," *Software Testing, Verifica- tion and Reliability*, vol. 21, no. 3, pp. 195–214, 2011.

[21] HackerOne. About hackerone. [Online]. Available: https://www.hackerone.com/about

[22] D. Miessler, "Seclists," https://github.com/danielmiessler/SecLists /blob/master/Discovery/DNS/subdomains-top1mil-110000.txt, 2018.

[23] L. A. Meyerovich and A. S. Rabkin, "Empirical analysis of programming language adoption," *ACM SIGPLAN Notices*, vol. 48, no. 10, pp. 1–18, 2013.