Identifying Key Strategies for Reconnaissance in Cybersecurity

V. Vishnu and K. Praveen

1 Introduction

The Internet creates an opportunity for offenders to carry out different types of attacks. Public service transparency and anonymous internet connectivity are improving those attacks. Cyber-attacks cost US$ 114 billion per year, according to the Symantec cybercrime study released in April 2012. If indeed the time lost by companies attempting to recover from cyber-attacks is counted, the overall cost of cyber-attacks will exceed US$385 billion [1]. Cyber-attack victims are also rising significantly. Symantec's study, in which 20,000 people in 24 countries were interviewed, revealed that 69% were the victim of a cyber-attack in their lives. Symantec also estimated that 14 people are the targets of cyber-attacks every second [1]. How do cyber threats thrive? It's because cyber-attacks are less costly, easier and involve lesser risk than physical attacks. Cyber attackers only need a few resources besides a computer and an Internet connection. They are unaffected by geography and space. They are difficult to track and prosecute because of the open and anonymous nature of the Internet. Considering that attacks on IT infrastructure are very enticing, it is expected that the number and severity of cyber-attacks will continue to increase.

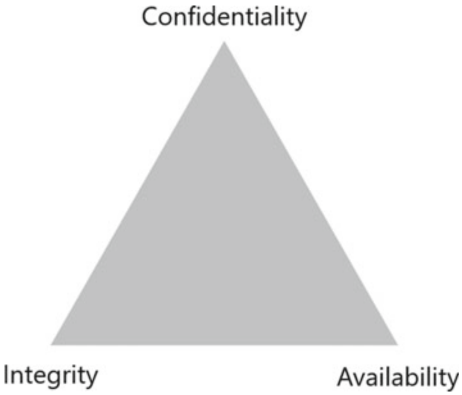
Cybersecurity is built to protect the digital assets of the company from ever-growing cyber-attacks. Cybersecurity can be achieved by the introduction of adequate security controls that include many security features, such as cybercrime deterrence, prevention and identification. Cybersecurity's primary purpose is to provide data and

V. Vishnu (✉) · K. Praveen
TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: cb.en.p2cys18029@cb.students.amrita.edu

K. Praveen
e-mail: k_praveen@cb.amrita.edu

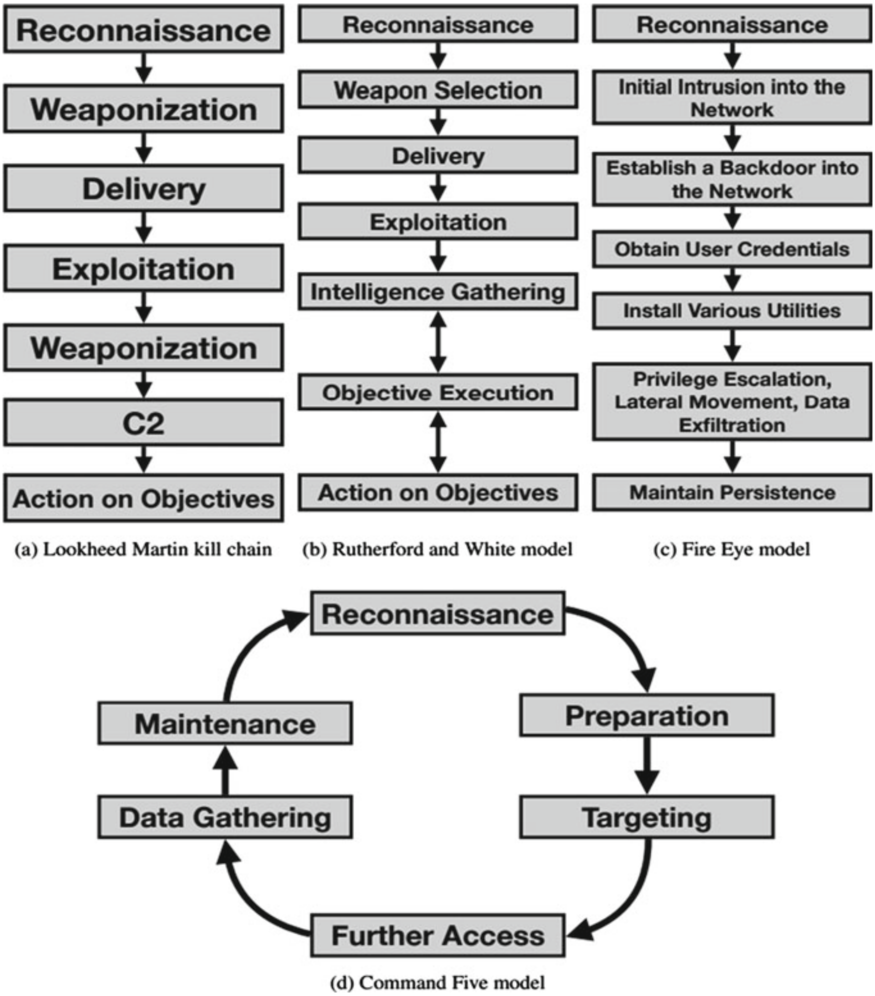
R. Agrawal et al. (eds.), *Cyber Security in Intelligent Computing and Communications*,
Studies in Computational Intelligence 1007,
https://doi.org/10.1007/978-981-16-8012-0_3

Fig. 1 The CIA triad



services with confidentiality, integrity and availability (CIA), sometimes referred to as the CIA triad (Fig. 1).

Attackers begin by gathering all types of data for a successful attack. Reconnaissance is one of the very first steps that attackers use to obtain information. Advanced port scanning techniques are accessible via free open-source software that provides a wealth of network information. This useful information includes port status, device types and variants, operating systems and service packages. Attackers are likely to succeed in combining information about vulnerable facilities with reconnaissance data. Social engineering is a technique of deception that uses human error to gain sensitive information, access or useful Intel. These “human hackers” scams in cybercrimes appear to attract unsuspecting users to reveal data, to spread malware or to give them access to restricted systems. Attacks can occur online, in-person and through other interactions. Social engineering schemes are based on how people think and behave. Once an attacker knows what motivates a user’s actions, it can successfully trick and exploit the user. Unpatched software is terminology for defining a computer code with known vulnerabilities. Once security bugs occur in the computer code, software developers write add-ons to the code known as “patches” to fix the security gaps in the code. Running unpatched software is dangerous because hackers are well aware of vulnerabilities until they appear. Advanced Persistent Threats (APT’s) typically involve multiple stages, including network hacking, the avoidance of tracking, the development of a strategy to target and monitor company data, the collections and exfiltration of confidential company data, to decide when the required information is most available. Advanced persistent threats have resulted in many massive, expensive privacy violations and are proven to be unnoticeable by traditional security assessments. Advanced persistent attacks are now becoming more and more common as cybercriminals try advanced techniques to accomplish their objectives. There have been a number of qualitative efforts to model a cyber-attack. As highlighted in Fig. 2, the most prominent of them is Lockheed Martin Model, which introduces the notion of a cyber-attack kill chain, describing sophisticated attacks, such as Advanced Persistent Threats (APTs), in seven steps: reconnaissance, weaponization, delivery,



(a) Lookheed Martin kill chain (b) Rutherford and White model (c) Fire Eye model

(d) Command Five model

Fig. 2 Models for describing cyber-attacks: **a** Lockheed Martin cyber kill chain; **b** the Rutherford and White variant of the cyber kill chain model; **c** Fire Eye model; and **d** the Command Five Pty Ltd attack model [2]

exploitation, installation, command and control and act on the objective. Rutherford and White improved the model by adding intelligence gathering. Researchers at FireEye and Command Five introduced their own models to describe and model a cyber-attack.

2 Background

Confidentiality is securing resources from unwanted or illegal access, integrity is ensuring that data remain unchanged, availability ensures that data are available without any drop in access, non-repudiation ensures no one can deny another person any service and authentication ensures people with proper access is only allowed access. These features are the cornerstone of any security system and are better known as the five pillars of information security. We can consider a cyber-attack as an assault that breaches any of the pillars. The attacker, thus, tries to breach one or more of the pillars or to attack a targeted system as successfully as possible. The objective might be a host, a network or a major infrastructure of information systems [3].

2.1 Anatomy of a Typical Cyber-Attack

Reconnaissance. The first step where the attacker begins by learning and understanding the details of an organization and network and to identify online behavior of key people of the organization, typically the system architect and network administrator.

Information Enumeration. Enumeration is characterized as the process of listing of usernames, hostnames and IP addresses, network resources and devices used, the operating systems and network topology. The attacker creates an active connection and conducts guided queries to get to know the target and collect more information. These collected data will contain all the security vulnerabilities or weaknesses present in network and attempts to exploit them during the exploitation point.

Network and System Penetration. The attacker is now trying to penetrate the network and the infrastructures once they have access to the network. This can occur in several ways. The attacker can record keystrokes using a malware called keylogger to capture important information entered by a user like a password, or the attacker can plant a worm and leave. The worm will be able to spot vulnerable systems and start spreading across the entire network. They expect the attack to continue as long as the attacker wants to until detected. It could be months or years. A modern vector for attackers is cloud-based data storage. With large-scale adoption of this technology, there has been an increased lack of security features [4]. Basic Role-Based-Encryption schemes have not been implemented. This gives an attacker more data to play with.

Data Exfiltration. Once the attacker has penetrated the system, he has to transfer the data to his system. Any unauthorized transition of data is data exfiltration. Whether data are stolen from a printer or a thumb drive, it's a rather real threat to corporations. Attacks can be performed manually by an authorized person who has access to enterprise systems or by malicious external actors who have access to them. Exfiltration of data can be a big business for cybercriminals and a big problem for any organization that is attacked.

Attack Sanitation. This is the last process of an attack where an attacker cleans up. All evidence of the attackers' presence is removed from the network and host systems to look like nothing has happened.

2.2 Cyber Kill Chain

Lockheed Martin, an American defense company initially authored the Cyber Kill-Chain Framework for Cyber Intrusion Identification and Prevention as part of its information and intelligence-driven defense and security model. The model defines what adversaries need to accomplish in pursuit of that goal by targeting the network, exfiltration of data and maintaining persistence within the organization. This model essentially implies that stopping adversaries at any step in the kill chain breaks the sequence of attack. Attackers must make full headway through all steps of the kill chain to complete a successful attack. The outcome is an inescapable phase where all attacks last, and thus stopping them at this stage massively improves the potential of busting any cyber-attack. When stopped in the early stages of the chain, the speed of success will be enhanced. Moreover, any threat, and therefore its traces, could be an opportunity to learn more about the attackers, their methodology and use their resistance to improve security systems. A far more realistic configuration of defenses results in a greater understanding of enemies and their directions [5]. The Cyber-Kill Chain proposes that the attackers will take six specific steps to try their attacks: Reconnaissance: This stage may where attacker employs the process of target selection, the classification of organizational information, organizational rules and laws, information on the organization's technological preference and topography, social network behavior of the employees or mailing lists available on a public repository. Weaponization and Packaging: In this stage, the attacker employs several forms: internet-facing application and thin client penetration, licensed and publicly available resources or modified software (licensed by the organization or modified to suit their purpose), enterprise software bugs like an XML parser or a word processor spreadsheets and presentation software which makes use of macros, target the most common sites used by the employees and set up phishing sites. Usually, they are equipped with information on duplicitous or accurate objectives. Delivery: The payload is supplied either by the victim (for example the browsing of a malicious site that leads to a malware deployment). The supply of ransom software, or the opening of fake PDF documents, or an intrusion (SQL or network access exploit) is triggered. Delivery: After delivery to the user, computer or device, the malicious payload will compromise the targeted asset and therefore the environment will become an initial foothold. This will be generally achieved by exploiting a security flaw that already has a patch. Although zero-days are exploited, in most cases, adversaries do not need to perform them as most of the targets will already be affected.

3 Tool Selection

The goal of this research is to assess the features of free/open source reconnaissance tools developed by independent researchers. Open-source software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software to anyone and for any purpose [3]. The following research process was adopted for the assessment:

- The selection of the most used reconnaissance tools.
- The assessment and evaluation criteria are defined solely based on the ease of use, availability of a graphical interface and the number of features it provides.
- The execution of the tool.
- The analysis of the output generated by each tool.

3.1 Reconnaissance Frameworks: A Comparison and Study of Open-Source Automated Recon Tools

We searched the GitHub repository, one of the biggest sources for open source projects, with the goal of selecting the five most common free/open source tools to be evaluated. We used keywords such as "Automated", "Reconnaissance" and "Cyber". We further filtered the search results using the following filters:

- The last update or commit is after 2016.
- OS independent, i.e. compatible with both Windows and Linux
- Makes use of OSINT for data collection.

In total, there were 264 public repositories on GitHub matching the above keywords. Using the inclusion filters and sorting by the most downloaded, we shortlisted the following five tools:

- Osmedeus
- AutoRecon
- AutoReconR
- InstaRecon
- InstaRecon.

For the assessment, only key functionalities of each framework are addressed, and no additional plug-ins are considered.

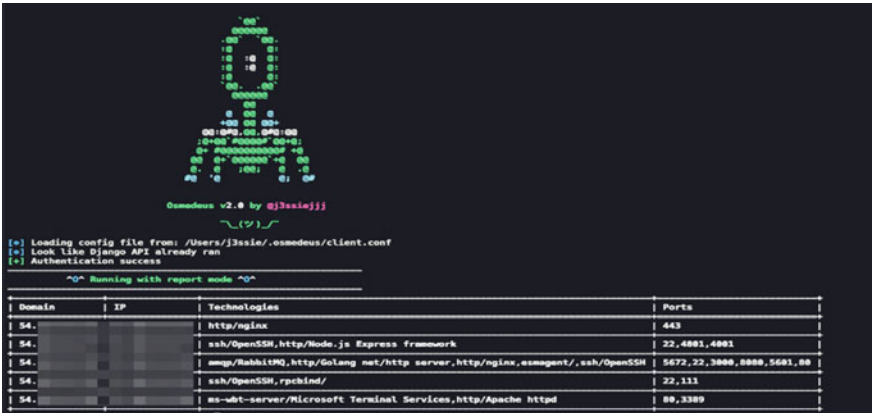
3.2 Evaluation Criteria

To evaluate the tools, we consider the following features.

- IP Recon: These tools are used to perform reconnaissance and gather information about the network the host is connected to.
- Social Engineering: This technique involves looking for reasoning to gain sensitive information or text by stimulating an individual mind or sense of social norms.
- Threat Intel: Cyber threat intelligence is information about risks and threat actors helping to prevent dangerous cyberspace incidents. Information outlets for cyber threats include open source information, social networking intelligence, human intelligence, digital intelligence, or deep and dark web intelligence.
- Forensic Tools: Tools used to investigate and gather evidence from various sources like an image file, audio file or even a sound file.

3.3 Evaluation Results

- Osmedeus is an open-source tool developed by a security researcher called @j3ssie. It is developed using a python client and Django API server. It can be used to scan the vulnerabilities of the target network and server. It features an impressive collection of tools such as web technology detection, IP discovery, and machine discovery. You can split the workspace to store all the scan data and logging information. Finally, it facilitates continuous scanning and allows you to access the scan report from the command line [6]. It is also loaded with web-based technology tracking, IP discovery, and system discovery backtrack features. The framework will split the workspace to store all scan data and log information. Eventually, it can support continuous scanning and allow you to view the scan report from the command line (Fig. 3).
- AutoRecon, developed by Tib3rius is an automated recon tool, which is capable of running multiple threads. This means that it can perform simultaneous network

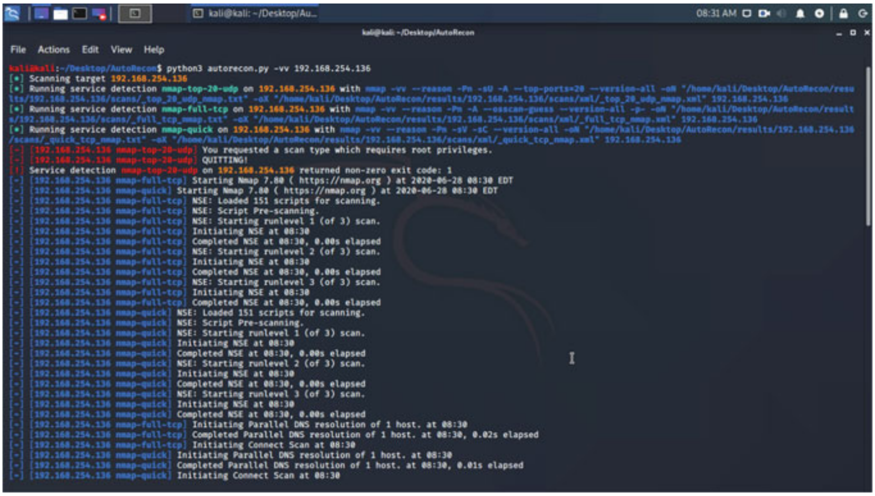


```
                    Osmedeus v2.0 by @j3ssiejjj

                            ¯\_(ツ)_/¯

[*] Loading config file from: /Users/j3ssie/.osmedeus/client.conf
[*] Look like Django API already ran
[+] Authentication success
--------------------------------------------------------------------------
          ^0^ Running with report mode ^0^
--------------------------------------------------------------------------
+------------+----------+----------------------------------------------------------------+------------------------------------+
| Domain     | IP       | Technologies                                                   | Ports                              |
+------------+----------+----------------------------------------------------------------+------------------------------------+
| 54.[redacted] | [redacted] | http/nginx                                               | 443                                |
+------------+----------+----------------------------------------------------------------+------------------------------------+
| 54.[redacted] | [redacted] | ssh/OpenSSH,http/Node.js Express framework               | 22,4001,4001                       |
+------------+----------+----------------------------------------------------------------+------------------------------------+
| 54.[redacted] | [redacted] | amqp/RabbitMQ,http/Golang net/http server,http/nginx,esmagent/,ssh/OpenSSH | 5672,22,3000,8080,5601,80 |
+------------+----------+----------------------------------------------------------------+------------------------------------+
| 54.[redacted] | [redacted] | ssh/OpenSSH,rpcbind/                                     | 22,111                             |
+------------+----------+----------------------------------------------------------------+------------------------------------+
| 54.[redacted] | [redacted] | ms-wbt-server/Microsoft Terminal Services,http/Apache httpd | 80,3389                         |
+------------+----------+----------------------------------------------------------------+------------------------------------+
```

Fig. 3 Osemedus [7]



```
kali@kali:~/Desktop/AutoRecon$ python3 autorecon.py -vv 192.168.254.136
[*] Scanning target 192.168.254.136
[*] Running service detection nmap-top-20-udp on 192.168.254.136 with nmap -vv --reason -Pn -sU -A --top-ports=20 --version-all -oN "/home/kali/Desktop/AutoRecon/resu
lts/192.168.254.136/scans/_top_20_udp_nmap.txt" -oX "/home/kali/Desktop/AutoRecon/results/192.168.254.136/scans/xml/_top_20_udp_nmap.xml" 192.168.254.136
[*] Running service detection nmap-full-tcp on 192.168.254.136 with nmap -vv --reason -Pn -A --osscan-guess --version-all -p- -oN "/home/kali/Desktop/AutoRecon/result
s/192.168.254.136/scans/_full_tcp_nmap.txt" -oX "/home/kali/Desktop/AutoRecon/results/192.168.254.136/scans/xml/_full_tcp_nmap.xml" 192.168.254.136
[*] Running service detection nmap-quick on 192.168.254.136 with nmap -vv --reason -Pn -sV -sC --version-all -oN "/home/kali/Desktop/AutoRecon/results/192.168.254.136
/scans/_quick_tcp_nmap.txt" -oX "/home/kali/Desktop/AutoRecon/results/192.168.254.136/scans/xml/_quick_tcp_nmap.xml" 192.168.254.136
[-] [192.168.254.136 nmap-top-20-udp] You requested a scan type which requires root privileges.
[-] [192.168.254.136 nmap-top-20-udp] QUITTING!
[!] Service detection nmap-top-20-udp on 192.168.254.136 returned non-zero exit code: 1
[-] [192.168.254.136 nmap-full-tcp] Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-28 08:30 EDT
[-] [192.168.254.136 nmap-quick] Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-28 08:30 EDT
[-] [192.168.254.136 nmap-full-tcp] NSE: Loaded 151 scripts for scanning.
[-] [192.168.254.136 nmap-full-tcp] NSE: Script Pre-scanning.
[-] [192.168.254.136 nmap-full-tcp] NSE: Starting runlevel 1 (of 3) scan.
[-] [192.168.254.136 nmap-full-tcp] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-full-tcp] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-full-tcp] NSE: Starting runlevel 2 (of 3) scan.
[-] [192.168.254.136 nmap-full-tcp] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-full-tcp] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-full-tcp] NSE: Starting runlevel 3 (of 3) scan.
[-] [192.168.254.136 nmap-full-tcp] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-full-tcp] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-quick] NSE: Loaded 151 scripts for scanning.
[-] [192.168.254.136 nmap-quick] NSE: Script Pre-scanning.
[-] [192.168.254.136 nmap-quick] NSE: Starting runlevel 1 (of 3) scan.
[-] [192.168.254.136 nmap-quick] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-quick] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-quick] NSE: Starting runlevel 2 (of 3) scan.
[-] [192.168.254.136 nmap-quick] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-quick] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-quick] NSE: Starting runlevel 3 (of 3) scan.
[-] [192.168.254.136 nmap-quick] Initiating NSE at 08:30
[-] [192.168.254.136 nmap-quick] Completed NSE at 08:30, 0.00s elapsed
[-] [192.168.254.136 nmap-full-tcp] Initiating Parallel DNS resolution of 1 host. at 08:30
[-] [192.168.254.136 nmap-full-tcp] Completed Parallel DNS resolution of 1 host. at 08:30, 0.02s elapsed
[-] [192.168.254.136 nmap-full-tcp] Initiating Connect Scan at 08:30
[-] [192.168.254.136 nmap-quick] Initiating Parallel DNS resolution of 1 host. at 08:30
[-] [192.168.254.136 nmap-quick] Completed Parallel DNS resolution of 1 host. at 08:30, 0.01s elapsed
[-] [192.168.254.136 nmap-quick] Initiating Connect Scan at 08:30
```

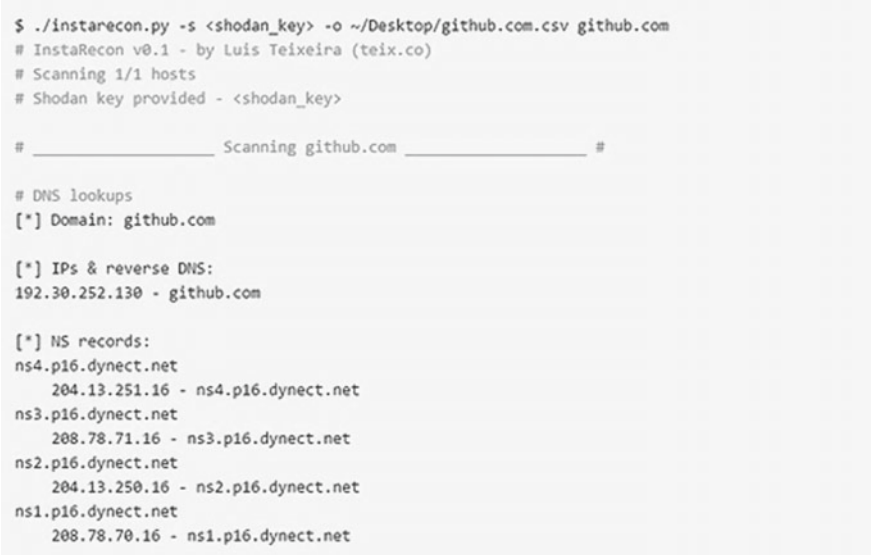
Fig. 4 AutoRecon [8]

scan at a time. The tool is highly flexible and is written in python. It uses modules from standalone recon apps [7]. The initial vector for the tool is a network scan which includes both port scan as well. The results from these scans are used for performing further enumeration of the network. The IP addresses of the targets can be provided as a list. Since multi-threading is enabled, multiple hosts can be scanned simultaneously. Port scanning can be customized to perform scans on well-known ports, top 1000 ports or all ports, suggests recommended commands to be performed after the initial scan. Proper directories are maintained. Logs and scan results are saved in a structural format. Vulnerable components are highlighted to help the user to target the system (Fig. 4).

- AutoReconR developed by Stefan Voemel is an improvement on AutoRecon with various added features. It attempts to automate parts of the identification and listing process of the network. Respective findings are described and summarized in an automatic report. As such, AutoReconR can make it easier to identify potential weaknesses in target systems more quickly and to find an entry point [8]. The tool is designed to run in the background while the tester can focus on other tasks in parallel. For example, in the laboratory environments provided by Offensive Security or during security tests such as OSCP, the tester can start writing exploits while AutoReconR scans the remaining targets and performs automatic service listings. Some of the features are: the possibility of specifying targets either via a command line or an input file. Define scanning and service listing profiles in custom configuration files. Automatically store scan results in a folder structure defined by service name. Launch additional actions based on identified services and service patterns. Summarize the results in a corresponding (at present very basic) PDF study combine software runtime and scan depth with the aid of complexity levels.

- InstaRecon developed by Luis Teixeira is an automated digital reconnaissance used to get the initial footprint of the system. It helps the attacker to obtain information about the target device or the network. This knowledge can be used to strike the device. That is why it can be called a Pre-Attack because all the information is checked to ensure a complete and effective resolution of the attack [9]. The features of InstaRecon are IP information and domain name information using whois and dig, Obtain IP address of domain name by using whatsmyip, Search engine lookup using Shodan, Domain Name Lookup using MX records, subdomain enumeration by using Google website reverse lookup (Fig. 5).

From Table 1, it is evident that there is a significant gap in the current open source tools. With the ever-changing threat landscape, the need for integrating social engineering and forensic tools along with a threat intelligence model has become paramount.



```
$ ./instarecon.py -s <shodan_key> -o ~/Desktop/github.com.csv github.com
# InstaRecon v0.1 - by Luis Teixeira (teix.co)
# Scanning 1/1 hosts
# Shodan key provided - <shodan_key>

# ____________________ Scanning github.com ____________________ #

# DNS lookups
[*] Domain: github.com

[*] IPs & reverse DNS:
192.30.252.130 - github.com

[*] NS records:
ns4.p16.dynect.net
    204.13.251.16 - ns4.p16.dynect.net
ns3.p16.dynect.net
    208.78.71.16 - ns3.p16.dynect.net
ns2.p16.dynect.net
    204.13.250.16 - ns2.p16.dynect.net
ns1.p16.dynect.net
    208.78.70.16 - ns1.p16.dynect.net
```

Fig. 5 InstaRecon [9]

Table 1 Comparison between the tools

Tool	IP Recon	Social engineering tool	Threat Intel	Automated workflow	Forensic tool
Osmedeus	Yes	No	No	Yes	No
AutoRecon	Yes	No	No	Yes	No
AutoreconR	Yes	No	No	Yes	No
Instarecon	Yes	No	No	Yes	No

4 Literature Survey/Related Work

To develop a reconnaissance tool, we need to analyze and characterize the relationship between an attacker and a victim. To systematically understand and characterize the behavior of cyber-attack reconnaissance behavior, Richard B. Garcia-LeBron (2018) proposes a systematic framework to study an attacker's behavior [10]. The system is composed of three abstraction levels: macroscopic, mesoscopic and microscopic as seen in Fig. 6.

On the macroscopic level, the graph of the time series of attacks in victim relations of the perception and understanding of cyber-attacker activities is studied. This graph-theoretical abstraction allows various information to be obtained by the use of a set of existing capabilities. To define the graphs of the attacker/victim relationship, the features that display these graphs were used which implements the comparisons that match different times windows between these graphs [12]. Additionally, the notions of efficient characteristics (i.e. features that may or may not characterize the creation of bipartite attacker-victim graphs) and robust characteristics (i.e. characteristics that are efficient in time resolutions) are described [13]. Finally, a dataset that was collected from a honeypot and used to conduct a case study to investigate the time resolutions that need to be considered to characterize as comprehensive as possible the evolution of the attacker-victim bipartite graphs [11]. Therefore to represent the behavior, only a small number of time resolutions have to be considered. At the mesoscopic level, the study of clustering cyber attackers using their identification behaviors, modeled as time series of reconnaissance activities is initiated [14]. A two-resolution methodology is used to characterize cyber-attack reconnaissance behaviors. At the microscopic level, a system to classify and organize temporal-spatial behaviors is proposed. The system provides a novel concept for the identification actions of the attackers dubbed: attacker identification trajectories. It also provides the cluster of attacker reconnaissance trajectories called a visual representation: Attacker reconnaissance trajectory hierarchy trees. A target reconnaissance model was proposed by Hung T. Nguyen et al. (2016) using probabilistic

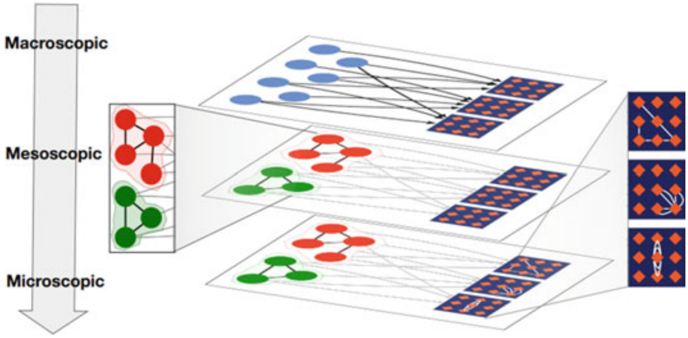


Fig. 6 The three levels of abstraction proposed by Richard B. Garcia-Lebron [11]

graphs on Facebook. Michael Glassman et al. (2012) present the idea of OSINT as an essential component in understanding the resolution of human problems in the twenty-first century. OSINT is a result of the emerging ties of human intelligence, which occur off the emergence of the Internet and World Wide Web expansionism of everyday life in various ways. Abel Yeboah-Ofori et al. (2016) propose a framework to thoroughly analyze and review existing study results on cyber intelligence and open-source information analysis and recognize all the risks and weaknesses of social networks online for mitigating purposes.

5 Design Considerations for a New Open Source Tool

Reconnaissance can take place vertically and horizontally. First, an intruder needs to obtain all possible subnet information from hosts and ports. This allows for the development of a targeted network map. Second, the attacker tries to evaluate the possible vulnerabilities of particular services. Cyber Reconnaissance is critical in today's world due to an increase in cyber warfare involving actions by a political entity or nation-states to attack and attempt to damage other nations' computers or information networks through, for example, computer viruses or denial of service attacks. Cyber reconnaissance is designed to level the playing field by providing organizations with a high-resolution picture of their cyber landscape from an adversary's perspective. While cybersecurity uses the implementation of technical means to protect the critical infrastructure or assets of an organization, it is also important that data collected during the reconnaissance phase is converted into intelligence, also known as cyber threat intelligence (CTI). The CTI is based on the collection of intelligence using open-source intelligence (OSINT), social media intelligence (SOCMINT), human intelligence (HUMINT), or insight from the deep and dark web.

5.1 Techniques Involved

Network scanning and port scanning. Processes to learn about a network's structure and actions—not hostile intrinsically, but often used by malicious individuals to conduct recon before trying to penetrate a network. Network Scan detects and maps all active hosts to their IP addresses within a network. Port Scan scanning refers to the process of sending packets to various host ports and analyzing responses to gain information about their operating system and running services. The first step in aggressive network scanning is often the host discovery process of deciding which devices on a network are down and up. For host discovery, two protocols are used most regularly: Address Resolution Protocol (ARP) scan and various forms of ICMP (Internet Control Message Protocol) scan. Because all individual ARP requests are required to map IP addresses to MAC addresses on a local subnet, ARP requests can be forwarded to multiple IP addresses on a local area network (LAN) based on the ARP response. Port scanning may be used to identify service uses in particular ports once the hosts that are available on the network have been scanned by network scanning. Typically, port scanning attempts to classify port as one of three names, Open: the destination answers with a packet that indicates the port that listens to it, meaning that any service used (usually TCP or UDP) for scanning is also in use. Closed: The request packet has received the destination but replies that there is no port listening service. Filtered: the port may be open, but the packet was filtered out of the firewall and excluded, so no response was received.

OS Fingerprinting. To detect OS, networks, utilities, and program names and numbers, attackers can send custom packets to the target. These packets will receive the victim's response in the form of a digital signature. This signature is one of the keys to recognize which applications, protocols and OS is running the target system. If the attackers have the right details, they will know your scenario and will be able to build a complete infrastructure map of all your services and potential network topology to fine-tune their digital assault. OS Fingerprinting is a method for determining which operating system the remote computer is running. OS Fingerprinting is often used for cyber reconnaissance because most exploitable bugs are unique to the operating system.

6 Conclusion and Future Work

The importance of reconnaissance has been discussed in-depth in the paper. Global cyber threat landscape mandates organizations to perform penetration test exercises often to secure their systems from the ever-evolving attacker landscape. From the cyber-kill chain, it is obvious that reconnaissance is the first step in any attack. Therefore, the need for an automated tool is necessary to streamline the process. The tools mentioned in the paper are the first step toward that and are not the finished product. We can further improve this software by including the following modules: Forensic Module: To obtain metadata from video and audio files, documents, etc. Automated Workflows: To directly provide input for reverse DNS lookup from the Nmap scan result page. CVE details: To obtain public vulnerabilities of the technology used Since OSINT is an ocean of data, there have to be two key components to any software, Fluidity and Rigor. Therefore, the new proposed system would be designed based on the experimental results of the attacker victim relationship. This helps us to model our software on current attacker trends. Along with this, a study on alert correlation systems will help us in reducing the number of alerts generated by an IDS or IPS when obtaining dataset to generate an attacker model [15]. A major issue when taking datasets generated from these systems is the false positive. Negating this will help us better in identifying actual threats. This will help in timely insight, which will enable to organize a better defense from this ever-changing threat.

References

1. Internet Security Threats Report. https://www.symantec.com/threatreport/
2. J.R. Rutherford, G.B. White, Using an improved cybersecurity kill chain to develop an improved honey community, in *Hawaii International Conference on System Sciences (HICSS)* (2016), pp. 2624–2632
3. E. Alata, M. Dacier, Y. Deswarte, M. Kaaniche, K. Kortchinsky, V. Nicomette, V.-H. Pham, F. Pouget, Collection and analysis of attack data based on honeypots deployed on the internet, in *Quality of Protection* (2006), pp. 79–91
4. D. Nidhin, I. Praveen, K. Praveen, Role-based access control for encrypted data using vector decomposition, in *Proceedings of the International Conference on Soft Computing Systems* (2016), pp. 123–131
5. D. Kiwia, A. Dehghantanha, K.-K. Choo, J. Slaughter, A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. J. Comput. Sci. **2**(5), 394–409 (2018)
6. Osemedeus, https://github.com/j3ssie/Osmedeus.git
7. AutoRecon, https://github.com/Tib3rius/AutoRecon.git
8. AutoReconR, https://github.com/svo80/AutoReconR.git
9. InstaRecon, https://github.com/vergl4s/instarecon.git
10. R.B. Garcia-Lebron, K.M. Schweitzer, R.M. Bateman, S. Xu, A framework for characterizing the evolution of cyber attacker-victim relation graphs, in *JMILCOM 2018–2018 IEEE Military Communications Conference (MILCOM)* (2018), pp. 70–75
11. Z. Zhan, M. Xu, S. Xu, Characterizing honeypotcaptured cyber-attacks: statistical framework and case study. IEEE Trans. Inf. Forensics Secur. 1775–1789 (2013)
12. X. Shouhuai, Cybersecurity dynamics: a foundation for the science of cybersecurity, in *Proactive and Dynamic Network Defense* (Springer, New York, 2019)
13. S. Banerjee, M. Jenamani, D.K. Pratihar, Properties of a projected network of a bipartite network, in *2017 International Conference on Communication and Signal Processing (ICCSP)* (2017), pp. 0143–0147
14. D. Koutra, J.T. Vogelstein, C. Faloutsos, DELTACON: a principled massive-graph similarity function, in *Proceedings of the 2013 SIAM International Conference on Data Mining* (2013), pp. 162–170
15. S. Mallissery, K. Praveen, S. Sathar, Correlation of alerts using prerequisites and consequences for intrusion detection, in *International Conference on Computational Intelligence and Information Technology* (2011), pp. 662–666