Survey and Taxonomy of Adversarial Reconnaissance Techniques

SHANTO ROY, University of Houston, USA
NAZIA SHARMIN, University of Texas at El Paso, USA
JAIME C. ACOSTA, DEVCOM Army Research Laboratory, USA
CHRISTOPHER KIEKINTVELD, University of Texas at El Paso, USA
ARON LASZKA, University of Houston, USA

Adversaries are often able to penetrate networks and compromise systems by exploiting vulnerabilities in people and systems. The key to the success of these attacks is information that adversaries collect throughout the phases of the cyber kill chain. We summarize and analyze the methods, tactics, and tools that adversaries use to conduct reconnaissance activities throughout the attack process. First, we discuss what types of information adversaries seek and how and when they can obtain this information. Then, we provide a taxonomy and detailed overview of adversarial reconnaissance techniques. The taxonomy introduces a categorization of reconnaissance techniques based on the source as third-party and human-, and system-based information gathering. This article provides a comprehensive view of adversarial reconnaissance that can help in understanding and modeling this complex but vital aspect of cyber attacks as well as insights that can improve defensive strategies, such as cyber deception.

CCS Concepts: • **Security and privacy**;

Additional Key Words and Phrases: Cybersecurity, cyber kill chain, adversarial reconnaissance, cyber reconnaissance, footprinting, open source intelligence, information gathering, social engineering, network scanning, localhost discovery, sniffing, side-channel attacks, cyber deception

ACM Reference format:
Shanto Roy, Nazia Sharmin, Jaime C. Acosta, Christopher Kiekintveld, and Aron Laszka. 2022. Survey and Taxonomy of Adversarial Reconnaissance Techniques. *ACM Comput. Surv.* 55, 6, Article 112 (December 2022), 38 pages.
https://doi.org/10.1145/3538704

1 INTRODUCTION

Businesses and governments must develop novel capabilities and technologies to stay competitive, but this innovation also leads to new cybersecurity challenges. As new security measures are

This material is based upon work supported by the National Science Foundation under Grant CNS-1850510 and by the Army Research Office under Award W911NF-17-1-0370. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Army Research Office.
Authors' addresses: S. Roy, University of Houston, 3551 Cullen Blvd., Room 309, Houston, TX 77204, USA; email: sroy10@uh.edu; N. Sharmin and C. Kiekintveld, University of Texas at El Paso, Computer Science Bldg (CCSB), 500 W University Ave, TX 79968, USA; emails: nsharmin@miners.utep.edu, cdkiekintveld@utep.edu; J. C. Acosta, DEVCOM Army Research Laboratory, 2800 Powder Mill Rd, Adelphi, MD 20783, USA; email: jaime.c.acosta.civ@army.mil; A. Laszka, University of Houston, 3551 Cullen Blvd., Room 501, Houston, TX 77204, USA; email: alaszka@uh.edu.

0360-0300/2022/12-ART112 $15.00
https://doi.org/10.1145/3538704

developed and implemented, adversaries continuously evolve new tactics and identify new vulnerabilities to conduct attacks. A 2018 Gartner report estimated that security expenses would grow to $124 billion in total in 2019 [7]. Another report published by Verizon reveals that 33% of data breaches involved social engineering and 28% involved malware [11]. The report also shows that in 56% of the reported breaches, it took months or longer to discover the attack. Reconnaissance activitiesare one of the key stages in conducting successful attacks.

Reconnaissance (or *recon*) in cybersecurity refers to the ongoing process used by attackers to gather as much information as possible about target systems or networks that can be used to conduct various types of malicious activity, such as gaining unauthorized access or denial of service. This is a crucial aspect of a successful attack, since the gaps in the security of a well-managed network may be small, and attackers may need to chain together exploits of multiple vulnerabilities to execute highly effective attacks. Better understanding of how attackers go about gaining this information can help us to model this key aspect of attacker behavior and to build better, more targeted defensive strategies for preventing attackers from easily gaining the valuable information that they need for planning attacks.

Definition 1. Reconnaissance is a process (sequence of actions) performed by adversaries to gather information about target networks and systems that is necessary for successfully exploiting vulnerabilities and furthering the adversaries' goals.

Reconnaissance plays a crucial role throughout the cyber kill chain.[1] Adversaries collect information about targets using different **tactics, techniques, and procedures (TTP)**. Adversaries can gather information sitting outside or inside target networks for months or even years. Sophisticated attackers can utilize multiple reconnaissance techniques while remaining undetected, making it more difficult for defenders to realize when a system is under attack. Several companies and organizations, including FireEye,[2] Cisco,[3] Symantec,[4] McAfee,[5] Microsoft,[6] Malwarebytes,[7] Bitdefender,[8] Kaspersky,[9] Fortinet,[10] ThreatTrack (now Vipre),[11] ISACA,[12] and CIS[13] are involved in investigating and analyzing the TTPs of attackers. However, despite the importance of reconnaissance in understanding attacker behavior, there is relatively little comprehensive academic research published on the reconnaissance process and TTPs. In particular, there is a gap in the literature on surveying, categorizing, and understanding the overall attacker reconnaissance process. We bridge this gap by collecting and analyzing a broad range of work on adversarial reconnaissance and building a taxonomy of reconnaissance activities and techniques that addresses the following main questions:

Q1. Reconnaissance Target Information: What types of information do adversaries seek through reconnaissance?

[1]The cyber kill chain describes different stages of a cyber attack. More details are provided in Section 5.
[2]https://www.fireeye.com/.
[3]https://tools.cisco.com/.
[4]https://www.symantec.com/.
[5]https://www.mcafee.com/.
[6]https://www.microsoft.com/.
[7]https://www.malwarebytes.com/.
[8]https://www.bitdefender.com/.
[9]https://usa.kaspersky.com/.
[10]https://www.fortinet.com/.
[11]https://www.vipre.com/.
[12]https://www.isaca.org/.
[13]https://www.cisecurity.org/.

Q2. Reconnaissance Phases: When do adversaries perform reconnaissance?

Q3. Taxonomy of Reconnaissance Techniques: What are the main categories of reconnaissance techniques and how do adversaries apply these techniques? What are the characteristics of these techniques in terms of what information is obtained and when/how they are utilized?

We answer the first question with an analysis of the different types of information commonly collected by attackers in the reconnaissance process (Section 4). Initially, we categorize the target information in terms of *non-technical* and *technical* information. The non-technical (or social) category includes *organization details* and *people information*. Technical information consists of *network*, *host machine*, *application*, and *user*-level information.

We answer the second question by considering reconnaissance activities in two main parts of the *cyber kill chain*. Here, we divide reconnaissance activities into *external* and *internal* reconnaissance. External recon is performed from outside the organization's network while internal recon is performed after gaining access to the target network. Internal recon is comparatively more effective in terms of gathering detailed information; however, external recon process has less chance to be identified by the defender.

We answer the third question by developing a taxonomy of reconnaissance techniques (Section 6). We categorize different reconnaissance techniques and map the techniques with the target data (Section 4) and phase (Section 5). We categorize recon techniques initially based on source: *third-party source*, *human*, and *system*. Third-party source-based target footprinting includes mostly passive techniques that are performed by tracking down the online (Internet) or offline (documents) footprints of targets that can be obtained from third parties (e.g., public third-party websites, or the dark web). Human-based recon techniques, or social engineering, involves active techniques intending to fool people into giving away confidential details or access information. System-based recon techniques are used to obtain information by observing or interacting with the target system locally (e.g., localhost discovery) or remotely (e.g., network scanning and sniffing).

Scope. Reconnaissance is performed not only by adversaries (black/grey hat hackers) but also by security researchers (white hat hackers, blue teams, etc.) for security testing purposes. We specifically discuss reconnaissance from the adversary's perspective, broadly focusing on targeted attack scenarios (both large-scale and small-scale attacks), including advanced persistent threats.

We discuss and elaborate the taxonomy based solely on reconnaissance procedures; the taxonomy does not cover other steps, techniques, or phases included in a threat model or the cyber kill chain. For example, we do not cover what procedures adversaries follow to compromise a host or to install a malware on a system. Nonetheless, we cover different recon techniques (e.g., social engineering, scanning, etc.) that are used to collect technical or non-technical information at both the external and internal recon phases.

Organization. The rest of this article is organized as follows: Section 2 presents case studies of previous cyber attacks and how adversarial reconnaissance played an important role in determining attack strategies. Section 3 presents related surveys and case studies regarding general or specific reconnaissance techniques and tools. Then, Section 4 provides insight into what information adversaries look for, and Section 5 discusses when they apply recon techniques during an attack. Section 6 categorizes and discusses different reconnaissance techniques used in both external and internal phases. Finally, Section 7 concludes the article by summarizing our findings and highlighting research gaps and opportunities in modeling reconnaissance and developing countermeasures.

2 CASE STUDIES OF REAL-WORLD CYBERATTACKS

Reconnaissance enables attackers to understand system configurations and to find alternative ways to exploit a system. To illustrate this, we present an example of an advanced persistent threat, called *APT41*, which has been responsible for several cyberattacks since 2014. Then, we discuss two additional well-documented cases that caused tremendous losses as examples of how reconnaissance is important to launching a successful attack and the types of methods and information involved.

2.1 APT41: Advanced Persistent Threats Analysis

An **advanced persistent threat (APT)** is a stealthy computer network threat actor that uses clandestine, evasive, continuous, and sophisticated cyberattacks to gain and maintain unauthorized access to a system for a prolonged period without getting detected [55]. Usually, the purpose of an APT is to steal sensitive information by monitoring, intercepting, and relaying it rather than causing network outage, denial of service, or infecting systems with malware. What differentiates APTs from other attacks are the TTPs that they employ and how the illegally obtained information is used to satisfy the ulterior motives of the threat actors. For example, APT41 [12], a Chinese espionage operator, targets healthcare, technology, telecommunications, travel services, news, and media firms. These sectors play crucial roles in China's five-year economic development plan [97]. The group injects malicious code into files then signs them with stolen legitimate code-signing certificates. This kind of attack affects a large number of hosts across the world after the distribution of the package. Using a technique called Execution Guardrail, information collected from a host (OS version, IP address, Active Directory name, shared network name, etc.) can be used to limit the activation of malware. Once a host has been compromised, the group used techniques like automated collection, data from information repositories, data from local systems, input captures, and screen captures to collect surveillance data that serves their interests.

2.2 Cyberattack on the Ukrainian Power Grid

On December 23, 2015, a power grid in Ukraine was compromised by a cyberattack, causing a service outage to the customers. The duration of the outage was only 6 hours. However, it took months to recover from the attack as most of the device firmware was overwritten with malware. The reconnaissance phase started much earlier, and the group of attackers initially utilized spear-phishing (water-hole attack) and email spoofing attacks to send emails to company workers with a malicious document attached. When a user opened the document, a pop-up menu appeared asking if the user would like to enable a macro; if the user agreed the macro installed a backdoor. The attackers potentially gained user credentials to log in to the system remotely. As there was no two-factor authentication, it was easy for the attackers to gain access as regular workers. Thereafter, they studied the whole network using an *internal reconnaissance* process for six consecutive months before launching the attack on December 2015 [48]. The reconnaissance included mostly network and system scanning [2, 143] and discovered field devices including serial-to-Ethernet devices that helped to interpret commands from the deployed SCADA network to the substation control systems.

2.3 Cyber Heist at Bangladesh Bank

There are also threat groups who are financially motivated to perform cyber heists. For example, the Bangladesh Bank cyber heist caused $81 million in losses [80]. *APT38* was the threat actor behind this heist [6], which was well planned to mitigate risks. It has performed some of the biggest cyber heists in the history of cyber crime [9]. Before an attack campaign, the group conducted an

extensive level of reconnaissance on the target system's personnel for watering hole attacks [25]. In one instance, the group targeted a manager's mailbox to learn about employees who have access to **Society for Worldwide Interbank Financial Telecommunication (SWIFT)** servers. SWIFT enables secure transactions among financial institutions. The group performed reconnaissance on a bank's remote connection to a third-party vendor with access to the SWIFT servers, which the group later utilized to build their malware. The group also performed prolonged reconnaissance of network activity and collected user and system information. In one case, it sent LinkedIn invitations to employees who were later targeted in watering hole attacks. Once the group had a foothold inside a network, it spent a prolonged period performing reconnaissance over the network—in some cases for two years—before starting fraudulent SWIFT transactions. The group exploited persistent access for as long as it took to learn network topology, permissions, monitoring software, and SWIFT systems. They also took control of *sysmon* and *sysinternal* utilities for internal monitoring.

Lessons Learned. By analyzing real-world attack scenarios, we see that both external (spearphishing or water-hole attacks in the Ukrainian power-grid cyberattack and the Bangladesh Bank cyber heist) and internal reconnaissance (Execution Guardrail techniques of APT41, system scanning in the Ukrainian power-grid cyberattack) play a significant role in a successful attack. The threat groups initially collect publicly available information, extract necessary details, and plan accordingly. Then, they gain access by breaching internal systems and use malware to gain access in the internal network, followed by obtaining system details. Advanced persistent threats can perform internal reconnaissance for a long time (six months during the Ukrainian power-grid cyberattack) without being detected; and in the meantime, the adversaries keep finding loopholes to improve their attack plan. Based on these case studies, we can see that many attacks are well planned and cause tremendous loss to the target organizations.

3 RELATED LITERATURE SURVEYS

We now discuss previous survey papers that discuss different aspects of cyber reconnaissance in terms of techniques and tools. The number of reviews that focus specifically on reconnaissance is relatively low. Some studies have surveyed and discussed different reconnaissance techniques, methodologies, and approaches (e.g., References [39, 56, 75, 108, 129, 139, 139]). Other works have evaluated the performance of publicly available reconnaissance tools (e.g., References [52, 81, 163]).

3.1 Surveys of Reconnaissance Techniques

A few previous papers [56, 108, 135] have attempted to present adversarial reconnaissance techniques comprehensively. For example, Mazurczyk et al. classified the evolution of cyber reconnaissance into four categories: internet intelligence, network information gathering, side-channel attacks, and social engineering [108]. They also provide examples of specific techniques based on the level of interaction and evolution over time (older vs. newer techniques). However, the categorization is not comprehensive (not all types of recon techniques are mentioned, e.g., sniffing and localhost discoveries) and did not provide a clear and concise taxonomy. The authors also discussed human-based countermeasures (awareness), reactive countermeasures (sniffing and side-channel prevention), and proactive countermeasures (cyber deception and moving target defense) to mitigate reconnaissance. The two other papers focused on network-based reconnaissance techniques and did not include other reconnaissance techniques. Next, we discuss survey papers that focus on and categorize specific types of reconnaissance techniques.

Open Source Intelligence. There are few works [68, 75, 93, 129, 152] that survey different techniques in **open source intelligence (OSINT)** from the perspective of cyber security. Glassman

et al. discussed how the world wide web provides access to immense information that can be potentially used for decision making and problem solving [68]. Tabatabaei et al. listed several tools that can be used for the collection, storage, and classification of open source data [152] in the context of security. Some other papers discussed OSINT for a specific purpose such as reliable web searching [129] or password cracking [93].

Social Engineering. Several papers have presented taxonomies of different **social engineering (SE)** attacks [23, 50, 72, 78, 133]. Alharthi et al. categorized the techniques in two types: technical, where the attacker uses media (e.g., mobile text and phishing site) to manipulate the user to reveal sensitive data, and non-technical, where the attacker directly interacts with the target. Salahdine et al. also classified the SE attacks as *human based* and *computer based* [133]. Heartfield et al. presented a taxonomy of semantic attack mechanisms of different social engineering attacks where they defined attack characteristics in three stages: orchestration, exploitation, and execution [78]. Other works surveyed specific SE techniques such as phishing [50, 72].

Cyber Scanning. There are a number of works that discussed cyber scanning techniques [33, 39, 42, 51, 57, 139]. Shaikh et al. provided a general overview of reconnaissance techniques that focuses on the classification of probes and discusses methods of reconnaissance including surveillance, eavesdropping, and intercepting communications [139]. The authors classified probes into three groups: host detection, port enumeration, and vulnerability assessment. They also highlighted several detection challenges and approaches for counter-probing activities. Some studies have focused on scanning techniques, such as Arkin's work, which reviewed scanning techniques concentrating on ping sweeps, port scans, and operating system identification [29]. Meanwhile, Claypool conducted a study on stealthy port scanning methods [51]. He discussed half-open scan, Xmas tree scan, UDP scan, Null scan, Fragmentation, Decoying, and Spoofing, which are popular forms of stealthy scanning techniques. Vivo et al. reviewed TCP port scanners, several scanning techniques developed to bypass firewalls analysis and filtering, stealth scanning, basics of UDP scanning, and scanning related to specific application-level protocols [57]. Bhuyan et al. surveyed and discussed the effects of frequent port scan attacks [39]. A comparison of port scan methods based on type, mode of detection, mechanisms used for detection, and other characteristics were discussed in detail.

Side-channel Attacks. Surveys of side-channel attacks [79, 106, 136, 145] have categorized these techniques based on different dimensions: active vs. passive, logical properties vs. physical properties, and local vs. vicinity vs. remote [145]. All of these categorizations have overlapping techniques and there is no clearly preferable way to categorize different side-channel attacks. Some works surveyed different side-channel attacks for specific techniques such as cache [106] and electromagnetic emission [136] or applications of different machine learning techniques in side-channel attacks [79].

Summary. We have described several survey papers that focus on specific types of reconnaissance techniques (e.g., open source intelligence, social engineering, scanning, sniffing, and side-channel attacks). However, to the best of our knowledge, no comprehensive survey work [56, 108, 135] categorizes and presents a taxonomy of all types of reconnaissance techniques, with clear categorization and distinction. Table 1 presents a comparison between our work and other general reconnaissance surveys in terms of which techniques are discussed and which are not. Our main objective in this article is to comprehensively describe what target data adversaries are looking for, when and how they perform recon, and to provide a clear taxonomy of recon techniques.

Table 1. Comparison of Existing Reconnaissance Surveys

Techniques / Works	Open Source Intelligence	Social Engineering	Cyber Scanning	Sniffing	Host Discovery	Side-Channel Attacks
Sangvi et al. [2013]	X	X	✓	X	X	X
Dar et al. [2018]	X	X	✓	X	X	X
Mazurczyk et al. [2021]	✓	✓	✓	X	X	✓
Our Work	✓	✓	✓	✓	✓	✓

3.2 Surveys of Reconnaissance Tools

Tundis et al. discussed varieties of vulnerability analysis tools [157] and provided corresponding qualitative analysis including the advantages and disadvantages of these tools. The work was not limited to finding available tools and procedures to recon a typical system, but it also provided a comprehensive overview of how adversaries collect information about various networks. Wang et al. reviewed the state of the art of open source vulnerability scanning tools [163]. The authors built a virtual lab environment and analyzed the virtual network with Nmap, Nessus,[14] Retina CS Community,[15] OpenVAS,[16] Microsoft Baseline Security Analyzer, and Nexpose Community Edition.[17] Dar et al. investigated several tools, e.g., DNSEnum,[18] NMap,[19] ZENMap,[20] DNSstuff,[21] and MxToolbox[22] and experimented on a variety of operating systems. They concluded that DNSEnum, NMap, and ZENMap performed well in active reconnaissance and DNSstuff and MxToolbox in passive reconnaissance if adversaries want to keep their identity hidden.

Holm et al. analyzed the performance of seven popular scanners AVDS,[23] McAfee, Nessus, NexPose, Patchlink,[24] QualysGuard,[25] and SAINT[26] [81] on a network consisting of 20 physical servers running a total of 28 virtual machines with various operating systems and versions. In another work, Coffey et al. analyzed various network scanning tools against SCADA equipment to examine the differences between issue identification and asset discovery [52]. The authors experimented on ICS and SCADA systems by finding vulnerabilities using the same scanning tools that are employed on conventional IP networks and suggested developing a network scanner that is capable of obtaining information from both serial and Ethernet devices at the same time.

Summary. We find several research works that survey and discuss different aspects of reconnaissance. Some of these categorize techniques while others evaluate related tools in terms of effectiveness. However, none of these existing works provide a comprehensive overview of the entire reconnaissance process; rather, they each focus on certain parts of the process. Therefore, in this article, we address this gap by providing a comprehensive survey and taxonomy of reconnaissance techniques, which considers information gathering procedures throughout the entire cyber attack process.

[14] https://www.tenable.com/products/nessus.
[15] https://sourceforge.net/projects/retinacommunity/.
[16] https://www.openvas.org/.
[17] https://www.rapid7.com/products/nexpose/.
[18] https://github.com/fwaeytens/dnsenum.
[19] https://nmap.org/.
[20] https://nmap.org/zenmap/.
[21] https://www.dnsstuff.com/.
[22] https://mxtoolbox.com/.
[23] https://beyondsecurity.com/avds.html.
[24] https://www.ivanti.com/solutions/security.
[25] https://www.qualys.com/qualysguard/.
[26] https://www.carson-saint.com/.

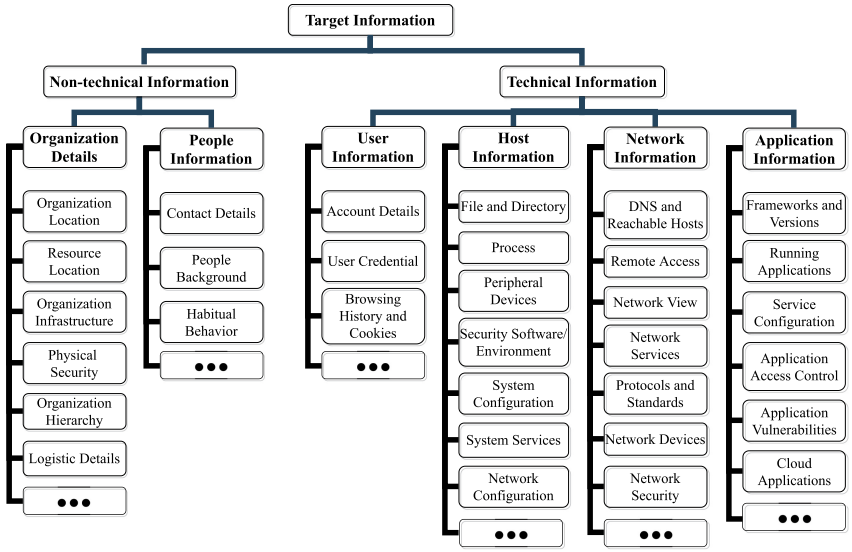

Fig. 1. Categories of target information for reconnaissance.

4 RECONNAISSANCE TARGET INFORMATION

Adversaries look for different types of information throughout the attack process. Target information is highly interconnected, and adversaries may need to acquire it sequentially. Furthermore, the type of information the attacker needs also depends largely on the adversary's objectives and capabilities. We categorize the information that adversaries look for during a large-scale network breach. We consider primarily large-scale attacks, since they cover an expansive scope of the information that may be acquired by sophisticated adversaries. Figure 1 presents our categorization of the types of information that adversaries look for while performing an attack. We divide the adversary's target information in two main types: *Non-technical (Social) Information* and *Technical Information*. The reasoning for this high-level categorization is that adversaries use these types of information for different types of attacks. Non-technical information (e.g., people contact details, physical security, etc.) is often most useful for performing social engineering and initial access planning. Technical information (e.g., host or network configurations) is helpful for adversaries to find vulnerabilities to compromise specific systems, escalate privileges, establish durable footholds, move laterally in networks, and achieve specific objectives.

4.1 Non-Technical Information

Non-technical or social information includes details about the target organization, its physical infrastructure, business processes, logistic details, and most importantly, potential vulnerabilities (e.g., flaws in physical security systems or building access control). Information regarding people who are employees or members of the target organization is another crucial element, since adversaries can use this information to trick people into giving away confidential information or granting access to resources [98].

4.1.1 Organization Background and Details. Organization information includes the organization's background, resources, employee contacts and work details, physical access and security policies, and so on [152]. Whether adversaries target a particular organization depends primarily on the organization's resources and if those resources are valuable, vulnerable, and accessible at

the same time. The security of technical assets depends in part on physical security mechanisms, since gaining physical access can be a viable approach for compromising security [125]. Publicly available resources are one of the primary initial data sources for adversaries.

- **Physical Attributes:** Adversaries can attempt to discover physical attributes of an organization such as location, physical infrastructure, physical security systems, physical resource locations and organization, and resource accessibility [94]. Adequate information can lead to effective social engineering attacks, such as gaining physical access using reverse social engineering [48].
- **Logistics Details:** Adversaries can look for logistics information such as financial and business processes or intelligence, employee and management hierarchy, resource arrangement, and other activities [152]. Supply chain management is also important, since it may leak important data regarding the organization [44]. Adversaries have also been reported to inspect and steal data from the third parties [90].

Much of the organizational information is available online (e.g., on business websites). News and blogs are also considered reliable sources for providing an outline and profile of an organization [152]. With increasing communication through social media, obtaining organizational information has become easier. Adversaries can also join the organization and access confidential information as an insider [120].

4.1.2 Personal Information. Personal information about people, such as contact details, technical or financial background, habits, and behavioral traits, are information that adversaries attempt to collect to analyze people's weaknesses. Finding these weaknesses is useful for applying social engineering techniques to gain remote access to the victims' machines or online accounts [22].

- **Contact Details:** Adversaries can collect contact details such as email addresses, phone numbers, identity information, and so on. For example, theHarvester[27] is an open source tool that can collect email addresses given a domain name. A user's contact addresses may also be found in social media and personal or organization websites [85, 98].
- **Personal Background:** Information related to the technical or financial background of people is also useful to adversaries for crafting social engineering attacks [85, 98]. The technical background of a person reveals what information and organization resources they may have access to. Technical background can be found on the organization website, an employee's LinkedIn[28] profile, BeenVerified[29] report, or a curriculum vitae.
- **Habitual Behavior:** Adversaries can also attempt to learn their targets' habits to perform social engineering (e.g., phishing) attacks [22]. Sophisticated adversaries can even track habitual social media usage to deceive people, for example based on Facebook usage [160].
- **Emotional States and Blackmail:** Adversaries can also try to observe people's emotional states. For instance, adversaries have been reported to spy on people through webcams in compromised hosts [45]. Further, adversaries may take photos of victims to blackmail them [101].

4.2 Technical Information

Technical details include diverse information about networks, hosts, applications, and users. Technical details are especially useful once adversaries have access to the target organization's internal

[27] https://github.com/laramies/theHarvester.
[28] https://www.linkedin.com.
[29] https://www.beenverified.com/.

network. Basic technical information can be obtained from an external network, but adversaries usually need to breach the target network or system to be able to gather more accurate details.

4.2.1 Network-Level Information. Adversaries look for network-level information, such as the network topology, network protocols, devices, and services, to understand the local network [33]. Scanning and sniffing are highly effective approaches for obtaining target information at the network level. Adversaries can also look for network security measures, such as the presence of firewalls or intrusion detection systems. Here, we discuss the most common network-level information that adversaries attempt to obtain.

- **Domain Names:** Domain and hostnames are identifiers that adversaries can use to tell which hosts belong to a particular domain. For example, hosts within the domain "`example.com`" may have hostnames "`host1.example.com`", "`host2.example.com`", and so on. Adversaries can use domain names associated with a particular organization to find extensive technical (e.g., subdomains, standard records such as SOA, MX, etc.) and personal details (e.g., admin contacts) [76].
- **Remote Hosts and Network Topology:** Adversaries may try to obtain the reachable IP addresses of either the external (public-facing) or the internal network. Reachable IP addresses can be identified through **Internet Control Message Protocol (ICMP)** or communication protocols such as TCP or UDP (e.g., APT: OSinfo [151]). Sometimes, the list of reachable IP addresses helps adversaries to map the whole network view (hosts, routers, switches, firewalls, and other network devices).
- **Network Protocols and Services:** A server can run a wide range of network protocols and services. For example, a server may provide web service (e.g., HTTP), file transfer service (e.g., FTP), name service (e.g., DNS), mail service (e.g., SMTP), and so on. For public-facing servers, running services can be identified from outside the organization's network by interacting with the server or sniffing packets. The same objective is achievable for internal servers if adversaries have access to compromised hosts on the internal network (e.g., APT: FIN6 [4]).
- **Network Devices:** Adversaries can look for network device information such as hardware device manufacturer or vendor, operating systems and version, manufacturer settings, networking configurations, and so on [40, 122, 139, 140]. Device information is useful for adversaries when employing exploits that target known vulnerabilities. Numerous tools are available (e.g., Network Watcher[30]) for identifying device information, such as the manufacturing company.
- **Network Security:** Organizations often implement network security measures including firewalls, intrusion detection systems, network zone isolation, honeypots, and so on, to prevent, detect, and mitigate attacks. Firewall rules define the filtering of inbound and outbound packets for a node or network; zone isolation is a layerwise security measure; honeypots can identify the presence of intruders by analyzing network traffic or resource request behavior. Adversaries can avoid signature-based detection (recognizing the signatures of known malware) using zero-day exploits and try to avoid anomaly detection (detecting a deviation from normal system or network behavior) using stealthier techniques, such as slower scanning for reconnaissance [51].

Network-level information is essential for planning remote attacks to penetrate an organization's network and for lateral movement and avoiding detection once an internal network is

[30] http://www.nirsoft.net/utils/wireless_network_watcher.html.

breached. Modern botnet-based attacks typically compromise systems remotely and then maintain command channels to execute commands on the compromised systems [86]. Channels include various protocols (e.g., Telnet, SSH) used by the remote shell client software. Adversaries can obtain network information using network or Internet footprinting (Section 6.1), scanning or fingerprinting techniques (Section 6.3.1), and social engineering (Section 6.2).

4.2.2 Host-Level Information. Host-level information (such as software configurations, running processes, files and directories, and security environments) is very useful to adversaries for performing the next stages of attacks. Specific details can be obtained once a host machine is compromised. Here, we list the most common host-level information that adversaries look for.

- **System Processes:** Information regarding details of installed software, presence of security software or environments, development frameworks, resource location, hardware and software configurations, application setup environment, and so on (e.g., APT: APT1 [1], OilRig [64]) can be obtained by monitoring and enumerating running processes. Process discovery on a compromised machine reveals the list of running processes and services of the system (e.g., APT: GravityRAT [111]).
- **System Platform:** The type of operating system and its version are crucial factors in security; using old versions creates more opportunities for attackers to utilize known tools to exploit. Apart from version identification, adversaries are able to collect OS build type, serial number and installation date (e.g., APT: PowerDuke [18]), and BIOS information (e.g., APT: BlackEnergy [36]).
- **System Configuration:** System configuration includes a wide range of settings from system services to hardware settings. Adversaries can gather information from the Windows registry system using remote access tools and can learn about running programs, their configurations, presence of antivirus or sandbox, and so on. Adversaries can collect hardware information such as CPU speed from a particular registry value (e.g., Trojan: Trojan.Hydraq [131]) and system manufacturer's value from the registry to identify the type of the machine (e.g., Group: Group 123 [111]) as well.
- **System Hardware and Peripheral Devices:** Hardware details, including CPU, primary memory, secondary storage, network card, video card, and peripheral devices (e.g., USB or Bluetooth devices), and so on, may constitute useful information for learning about the vendors, virtual machines, and forensic setups. Device information helps adversaries to identify known vulnerabilities in a vendor's product and thus to devise exploitation strategies. Adversaries can collect more information about particular hardware such as processor (e.g., APT: FALLCHILL [5]), processor architecture (e.g., APT: DarkHotel [70]), motherboard (e.g., APT: BlackEnergy [36], OopsIE [65]), primary memory (e.g., APT: DarkComet [3]) and drive/volume information (e.g., APT: RunningRAT [142]), video cards (e.g., APT: Agent Tesla [171]), and peripheral devices like keyboards (e.g., APT: SynAck [84]).
- **Security Environment:** Adversaries can learn about security environments (e.g., virtualization or sandbox) by querying registry values, system services, BIOS information, process list, and system information, such as hardware configuration. Usually, malware is executed after sandbox/VM evasion techniques. Security information includes firewall rules, presence of antivirus, honeypot or sandbox setup, virtualization environment, and so on (e.g., APT: DarkHotel [70]).
- **Files and Directories:** Adversaries can look for directory contents and file lists (e.g., APT: Brave Prince [142]). Particular directories containing configuration information or files with specific extensions (e.g., APT: Microspia [156]) can be useful for extracting information about user accounts, password management, software or application configurations, network

configurations, and so on. Adversaries can also look for users' personal files, financial reports, and proprietary data.

4.2.3 Application-Level Information. Security vulnerabilities at the application level depend largely on three factors: exploitability, detectability, and impact of damage. To exploit application-level vulnerabilities (e.g., SQL injection, cross-site scripting, broken access control, etc.), adversaries collect application-level information from a system or a network. Here, we list the most common application-level information adversaries look for.

- **Frameworks and Environments:** Hosts run various development frameworks (e.g., web-based frameworks such as Laravel[31] or Django[32]) and environments (e.g., application runtime environments such as Java VM), which may have vulnerabilities. Misconfiguration is another possible weakness that creates loopholes and attract adversaries [62]. Therefore, adversaries may attempt to collect the names, version, and runtime configuration information of frameworks that are installed on a system.
- **Security Tools and Applications:** Presence of anti-malware and forensic tools may be identified by querying the default software installation directory (e.g., "Program Files" on a Windows system (e.g., APT: Astaroth [59])) or by querying registry (e.g., APT: FIN8 [10]), and running processes (e.g., APT: Darkhotel [70]).
- **Application or Package Configuration:** Adversaries may also be interested in learning about the configuration of installed software and applications on a host [35]. Depending on the obtained information (e.g., versions), adversaries may utilize a database of existing exploits available on the dark web or develop exploits themselves. Application configuration information can also reveal access tokens and user credentials.
- **Cloud Dashboard and API:** Adversaries can gather information about virtual machines, cloud tools, services, and other cloud assets that are accessible from the compromised host [35]. Information related to Amazon AWS, Google Cloud Platform, Microsoft Azure, and other popular cloud service configurations can be queried or accessed using dashboards API and command-line interfaces.[33,34]
- **Databases:** Database systems are prone to have misconfiguration and human errors that leave systems vulnerable to attacks [58]. Adversaries can fingerprint versions of MySQL, PostgreSQL, Microsoft SQL Server, and Oracle Database by performing advanced queries [37]. Advanced remote attackers can also identify the state of an application database, e.g., they can check if the target machine's antivirus signature is updated [20].
- **GUIs:** Apart from this information, adversaries can also obtain data from the GUI windows of running applications. For example, they can collect window titles (e.g., APT: Remexi [100]) or text content (e.g., APT: PowerDuke [18]). Adversaries are also capable of enumerating application windows (e.g., APT: SOUNDBITE [47]) and capturing screenshots of them (e.g., APT: Catchamas [32]).

4.2.4 User-Level Information. User-level information such as account details and access credentials are useful for everything from gaining initial foothold in an internal network to privilege escalation on a compromised host. Often, adversaries collect information about user accounts and then try brute-force or dictionary-based attacks to gain access [117].

[31] https://laravel.com/.
[32] https://www.django-cms.org/en/.
[33] https://github.com/RhinoSecurityLabs/pacu.
[34] https://cloud.google.com/security-command-center/docs/quickstart-scc-dashboard.

- **Account Details:** User and group information includes the list of users and groups, their login types, access control policies, group permissions, and so on. APTs can gather information about domain and account information (e.g., account ID, token information, etc.) by observing the list of running processes [141]. Some APTs are also capable of querying information from account associated directories and enumerating local and domain users [151].
- **User Credentials:** Some of the most common practices of obtaining user credentials are performing social engineering attacks (e.g., phishing) against target users and installing keyloggers on the users' machines [38] or utilizing spyware to collect user profile data or login information stored in a browser cache [137] (e.g., APT: Machete [61]). Adversaries can also take advantage of web browser vulnerabilities to collect user-level information, e.g., by installing a malware extension [154] and stealing sensitive information when the user fills out a web form or from a browser cache [172]. This has the potential for compromising other services, since users often use the same passwords for multiple accounts.

5 RECONNAISSANCE PHASES

Reconnaissance is present in different forms throughout the attack process and provides key information that is needed to execute subsequent phases. Typically, an adversary first selects the target organization and then collects as much information as possible regarding the technical and non-technical features of the target organization using externally available sources to create an effective plan for initial access [76]. Once adversaries have access to the internal network, they seek more information about the network to engage in lateral movement and compromise other resources [166]. Sophisticated APTs are capable of staying inside their target networks for extended periods of time [48]. Ongoing lateral movement with internal discoveries results in continuously expanding access and capability to affect the target.

To understand the adversary's TTP, Lockheed Martin has developed a model called the Cyber Kill Chain,[35] which describes the technical aspects and a sequential step-by-step model to understand the movement of APTs [166]. However, the basic model does not provide much detailed insight about the reconnaissance (e.g., internal scanning and discovery) processes throughout the chain. Therefore, we introduce the concept of reconnaissance in two phases: *external reconnaissance*, which is performed to collect technical or non-technical information before gaining access to an internal asset, and *internal reconnaissance*, which is performed to obtain system information from the internal network.

Figure 2 shows a detailed view of external and internal reconnaissance phases in the cyber kill chain model, focusing on large-scale attacks that are carried out by APTs. The attack process starts with target selection and planning. The adversary begins collecting information about the target organization utilizing various footprinting, scanning, and social engineering techniques. Next, the adversary attempts to gain an initial foothold by compromising the target and installing malware or establishing command-and-control (C2) through other means. Then, the adversary can perform internal reconnaissance utilizing various scanning (e.g., active host or port scan) and localhost discovery (e.g., process discovery on host) techniques.

We could also categorize reconnaissance from the standpoint of the access level required, ranging from *outsider* through *nearsider* to *insider*. As an outsider, adversaries perform external reconnaissance and can collect publicly available information (e.g., organization or people information) and limited scan results (e.g., public-facing systems, web server version). As a nearsider, adversaries can plant rogue routers to collect network information and to compromise user machines (e.g., employees' portable devices or computers) [168]. Sometimes adversaries manage to gain physical

[35] https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

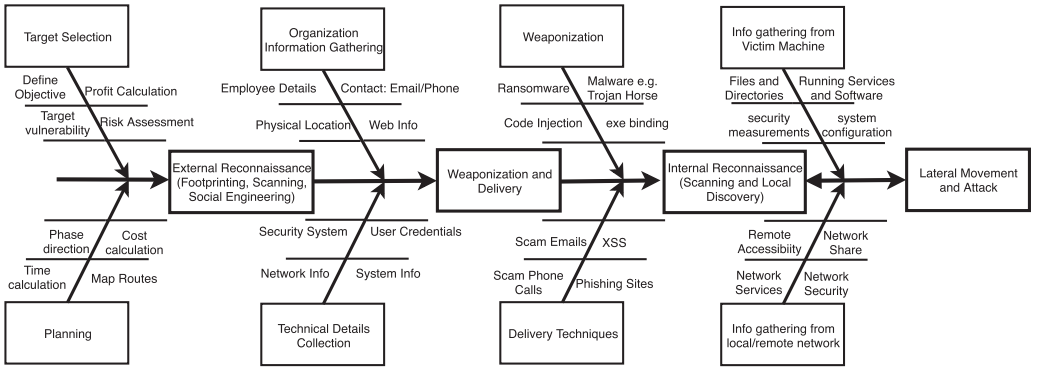

Fig. 2. External and internal reconnaissance.

access to the assets of the target organization (e.g., posing as an employee or a serviceman using social engineering) or compromise and take control of them remotely, which enables adversaries to act as malicious insiders and perform internal reconnaissance [91].

5.1 External Reconnaissance

External reconnaissance refers to activities before adversaries gain access to the internal network. Adversaries can obtain crucial information from public-facing nodes, online footprints, and people, which helps them to prioritize objectives and plan attacks. OSINT is one of the primary approaches for performing external reconnaissance. Technical, organizational, and personal weaknesses may be identified by analyzing public sources of information [74, 152], while remaining undetected.

Adversaries often start by collecting organization information and people's contact details. They can learn different technical details using Internet footprinting, which requires passive techniques with little threat of detection [76, 107]. However, Internet footprinting tends to provide limited information, so adversaries may also use social engineering techniques to manipulate people into providing additional information [22, 98, 118]. Attackers next move their attention to designing attacks and malware and try to compromise at least one internal host. After they have succeeded, adversaries can stay inside the network for months, performing internal reconnaissance and escalating their attacks until they reach their targets [166].

5.2 Internal Reconnaissance

Once an attacker has compromised at least one host inside the target network or has established insider access, they may create a secure channel between an installed backdoor and a command and control server [170]. The next steps and objectives depend on the information that the adversary can gain from the compromised host. Initially, adversaries can look for user and host-level information. Running processes and configurations expose the list of installed software and applications used by the victim host and other hosts [34]. They can use system commands and custom tools to collect user, host, network, and application-level information. Sometimes, adversaries wait and utilize passive scanning techniques such as sniffing packets to obtain a network view and discover system architectures, protocol mappings, and exploitable vulnerabilities [82]. Passive scanning helps adversaries to remain undetected for extended periods of time. Adversaries can exploit vulnerabilities using the collected information to compromise other hosts to get closer to the target resources [82].

6 RECONNAISSANCE TECHNIQUES

We now categorize the most common techniques used for gathering information. These techniques are are either *active* or *passive* in nature and used to collect *organization*, *user*, *host*, *network* or *application*-level information. They can be used in either the *external* or *internal* recon phase.

Reconnaissance techniques are always evolving, with varying intentions and technical approaches. Generally, reconnaissance can be performed to collect data from different types of sources. For example, information can be obtained from third-party sources at an initial stage, by fooling a target human, or directly from the system resources. In this section, we categorize various reconnaissance techniques and discuss them in the context of the questions that we explored previously: what target information attackers aim to collect (Section 4) and when they apply the recon techniques (Section 5).

In some cases, adversaries need to interact with the target to obtain information. In other cases, they can obtain information through passive observation or indirect interaction, which is more stealthy. Therefore, many works categorize reconnaissance techniques as either *active* and *passive*. However, while it is possible to categorize some techniques (social engineering, scanning, or side channels) as either active or passive, there is not always a sharp distinction; so it is not an ideal basis for a comprehensive taxonomy. Instead, we categorize reconnaissance techniques primarily based on the source of the information: third-party-based reconnaissance techniques, human-based reconnaissance techniques, and system-based reconnaissance techniques. Figure 3 lists examples of techniques for each type.

- **Third-party source-based reconnaissance techniques:** Extracting information from third parties (e.g., third-party websites and services, dark web).
- **Human-based reconnaissance techniques:** Gathering information from humans by focusing on persons at the target organization.
- **System-based reconnaissance techniques:** Collecting information from computer systems (hardware or software) at the target either by exploiting weaknesses or using standard interfaces.

Third-party source-based and human-based reconnaissance techniques are usually performed in the external phase, when adversaries look for information about targets prior to launching attacks. System-based reconnaissance techniques can be applied both externally and internally. For example, external scanning gathers information necessary for the initial compromise of the target organization's network. Internal scanning extracts more detailed information regarding the target organization's hosts, networks, services, and applications. However, internal scanning techniques can be riskier for the adversary due to the higher chance of being detected by intrusion detection system [42]. Nonetheless, sophisticated APTs can stay hidden inside compromised networks for months to years and perform extensive internal discovery (e.g., Ukraine Power Grid Attack [48]). In this section, we discuss available tools, outcomes, types of actions (active or passive), and reconnaissance phases (external or internal) for each of the four types. Table 2 shows techniques, types, target information, tools, and publicly available tools for third-party source-based target footprinting.

6.1 Third-party Source-based Reconnaissance

Third-party source-based target footprinting techniques are typically performed during the early stages of an attack to collect useful information about the organization, personnel, and resources. Third parties include websites, search engines, dark web, or personnel who are not involved with the target organization. We discuss the most common third-party source-based footprinting

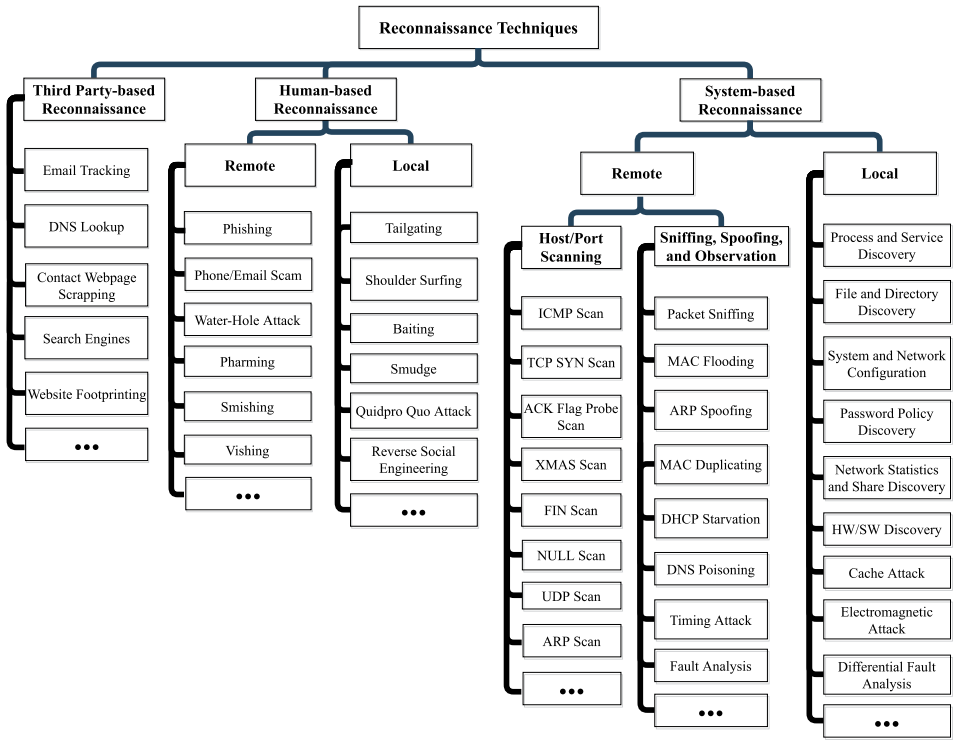

Fig. 3. Taxonomy of reconnaissance techniques.

techniques here. Table 2 shows techniques, types, target information, tools, and publicly available tools for third-party source-based target footprinting.

- **Internet Footprinting:** Adversaries can use tools such as website downloaders, data scrapers, and custom-made scripts to perform Internet footprinting manually. Adversaries often start collecting publicly available technical details and then identify underlying technologies [75]. For example, an online tool like NetCraft[36] is capable of exposing the software and platform behind a website. Site reports contain IP addresses, OS, web server software (e.g., Apache[37] and IIS[38]), nameserver, DNS admin, resource specified rules, and site technologies.
- **Whois Lookup:** A WHOIS record contains details about the owner of a domain, physical addresses, contact addresses (e.g., telephone numbers and email addresses), and other related information [165]. WHOIS information is usually stored in WHOIS databases and is maintained by regional Internet registries. The domain registration processes usually require a new domain owner to register with verifiable current contact details. Adversaries can perform WHOIS lookup to find administrative information, including domain name details, the contact information of the owner, name servers, and so on. After extracting administrative information, adversaries can perform social engineering attacks to obtain further information about the target.

[36] https://www.netcraft.com/.
[37] https://www.apache.org/.
[38] https://www.iis.net/.

- **DNS Interrogation:** DNS interrogation tools are used to search for hosts in a network to obtain an interal view of the network. Several online tools leverage the opportunity to perform a lookup to find additional hosts inside the network. Adversaries can find potential targets by obtaining records of CNAME, PTR, MX, HINFO, and AXFR if misconfigured by administrators [165]. NSLookup[39] is the most common tool for DNS interrogation.
- **Website Footprinting:** Adversaries can extract typical information such as server and application versions, files, contact details, and so on, using website footprinting [107]. Footprinting websites is relatively easy, since there are many tools available for scanning websites and extracting information (e.g., identifying underlying technology using "`builtwith`"[40] and web crawling using "`HTTrack`"[41]). Tools like WebExtractor[42] can collect contact information such as phone numbers, email addresses, and fax numbers. Other tools, such as Website Watcher,[43] are capable of monitoring web updates. Backdated site information can also be obtained from the Internet Archive.[44]
- **Social Media Tracking:** Personal information can be obtained through search engines and social media including Facebook, Twitter, and LinkedIn. LinkedIn and other job sites can reveal a person's technical background and responsibility within an organization [21]. Adversaries can follow the online activities of a person and learn about the person's habits, psychological state, and preferences [77] for use in social engineering attacks.
- **Email Tracking:** Email tracking can include monitoring a user's time and frequency of opening and reading emails using publicly available email trackers (e.g., browser extensions such as Streak[45]). This enables adversaries to learn about their targets' email reading times and associated habits [60], which they can exploit in social engineering. Initially, they can collect users' email addresses through website footprinting and scraping (e.g., finding contact information on personal websites) or social media scraping (e.g., harvesting user account details from social-media sites). Adversaries can then send malicious links and track if an email was read and if a target followed a link [98].
- **Search Engines and Google Hacking:** Search engines (such as Google, Yahoo, and Bing) can find background information (e.g., financial, technical, or business process reports) about an organization[152]. **Google hacking database (GHDB)**[46] and advanced search queries[47] can help adversaries use advanced features of Google search to find more details (e.g., "`filetype`" can be used to search specific files). In some cases, confidential information including user credentials, vulnerabilities, weaknesses, specific files, and so on, can be found in GHDB. Alert services such as Google and Yahoo alerts can track updates of a target website, blog, or media.

6.2 Human-based Reconnaissance

Human-based reconnaissance, or SE, attacks represent some of the most powerful information-gathering techniques according to Kevin D. Mitnick [113]. Typically, fooling a human is significantly easier than fooling firewalls, honeypots, or intrusion detection/prevention systems. Social

[39] https://linux.die.net/man/1/nslookup.
[40] https://builtwith.com/.
[41] https://www.httrack.com/.
[42] http://www.webextractor.com/.
[43] https://www.aignes.com/.
[44] https://archive.org/.
[45] https://www.streak.com/for/email-tracking-in-gmail.
[46] https://www.exploit-db.com/google-hacking-database.
[47] http://www.googleguide.com/advanced_operators_reference.html.

Table 2. Third-party Footprinting Techniques and Tools

Techniques	Type	Target Information	Phase	Publicly Available Tools
Internet Footprinting	Passive	Organization Details, People Information	External	Web tools (e.g., spiderfoot[1]), search engines (e.g., Google), location (e.g., Google Earth), people (e.g., pipl[2])
Whois Lookup	Passive	User Account Details, DNS and Reachable Hosts	External	Online tools (e.g., whois [103])
DNS Lookup	Passive	DNS and Reachable Hosts, Network View, Network Devices	External	Online tools (e.g., NSLookup, DNSLookUp [73])
Network Footprinting	Active/ Passive	Network View	External	Traceroute, ARIN DB,[3] LoriotPro,[4] RIPE,[5] LACNIC,[6] APNIC,[7] and other online tools
Website Footprinting	Passive	Organization Details	External	archive.org, website mirroring tools (e.g., NCollector Studio[8])
Email Tracking	Passive	Contact Details, Account Details	External	Tracking tools (e.g., VisualRoute[9] and GeoSpider[10])
Google Hacking	Passive	Non-/Technical Details	External	Google advanced search operators

[1]https://www.spiderfoot.net/; [2]https://pipl.com/; [3]https://www.arin.net/resources/guide/account/database/;
[4]https://www.loriotpro.com/; [5]https://www.ripe.net/; [6]https://www.lacnic.net/; [7]https://www.apnic.net/;
[8]http://www.calluna-software.com/; [9]http://www.visualroute.com/;
[10]http://www.oreware.com/viewprogram.php?prog=22.

engineering has been recognized as one of the most common techniques employed by cyber attacks that result in high-profile data breaches (e.g., RSA's SecurID system compromise in 2011 and the New York Times network breach in 2013) [98]. Social engineering is based on using deception to gain information through methods like baiting, pretexting, phishing, and spear-phishing. We now discuss some of the most common social engineering techniques.

Existing works categorize social engineering as non-technical vs. technical [23] or human-based vs. computer-based [133]. We categorize based on a similar concept of local vs. remote social engineering techniques. Local SE techniques (e.g., baiting, tailgating, shoulder surfing, etc.) require direct in-person involvement, and remote SE techniques (e.g., phishing, vishing, pharming, malware, etc.) can be performed remotely via web or mobile media.

6.2.1 Remote SE Techniques. Remote SE techniques are performed remotely using media channels such as mobile, fake websites, spam messages or emails, and malware (e.g., trojan horses or ransomware). These techniques are more common than local SE techniques.

- **Phishing:** Phishing has proven to be a very effective technique for stealing user credentials [78]. In a recent paper, Chiew et al. presents linkages between media, vectors, and technical approaches of phishing techniques that provide a better understanding of why phishing has been so successful over the years [50]. The authors note the Internet, **short messaging service (SMS)**, eFax, instant messaging, social networking, and telephone services as the primary media of phishing. Adversaries can also utilize *evil-twin* attacks, where they lure a target user to connect to a fake wireless access point and authenticate to a forged server so that adversaries obtain the user credentials [168].
- **Watering Hole:** A watering hole attack typically compromises a victim's machine by installing malicious code from a malicious website [25]. Adversaries start by profiling a target user or group to learn their habits, such as visits to popular websites. Then, the adversaries

exploit vulnerabilities in those websites or place links that redirect the users to a malicious site. Since users trust these websites, they may fall victim by accepting downloads or by following malicious hyperlinks allowing attackers to gain access to the victims' machines. Once an adversary has compromised the host (e.g., by installing Trojan horse malware), they can collect user, host, network, and application-level information.

- **Pretexting and Vishing:** Pretexting and vishing refer to impersonation through text messages or voice calls (vishing) and convincing targets to give access to particular resources [104, 169]. For example, an adversary can call a bank pretending to be a trusted person, and convince the official to grant access or to disclose usernames and passwords. Adversaries may require some confidential information to perform this type of attack convincingly [162].
- **Pharming:** Pharming is similar to phishing in terms of tempting a target user to visit a fake webpage, but it is more sophisticated technologically, since it typically involves secretly installing malicious software on the victim's computer [46]. Pharming is often performed through DNS poisoning, which enables redirecting victim users to malicious sites even if they attempt to visit only legitimate sites [146]. Therefore, regardless of the security measures taken by a user they may still fall victim to visiting malicious content.
- **Smishing:** Smishing (a combination of the words "SMS" and "phishing.") is a form of phishing in which a victim receives a malicious link in an SMS message [92]. The victim is tempted to download and install a Trojan horse, keylogger, or some other malware on the victim's mobile phone by following the link in the received message. Several Trojan horses feature keyloggers, which record every keystroke and send the records back to adversaries when the device is connected to the Internet. Account information, credentials, search habits, and so on, can be obtained from a keylogger-infected machine [38].

6.2.2 Local SE Techniques. Local SE techniques involve in-person direct or indirect interaction, such as talking face-to-face, following a person to access a building, or fooling the target by impersonating an authorized person.

- **Tailgating:** A tailgating attack is effective for attackers to have physical access to an organization or a resource. For example, an attacker can pretend to forget to bring his card and manipulate the target to give him access to a building or secure zone [23, 133]. RFID card attacks are also common now, since many organizations use these as an access token due to low cost and good user experience. However, an attacker can manipulate the RFID network and gain access to the target secure zone [133].
- **Shoulder Surfing and Smudge:** An attacker can watch the target person entering a username, passwords, credit information, or other sensitive information by standing near them [23, 133]. The attacker can also retrieve user input from touch screen devices in the absence of the target person in a *Smudge attack* [23].
- **Baiting:** Baiting is an effective technique for obtaining information by spreading Trojan horses using physical media such as flash drives, CD/DVD-ROMs, memory cards, or other portable devices [98]. Usually, the infected media are left in places where target users can find them. If they insert the media into their machines due to curiosity or the intention to return the media, then this can result in infecting the victims' machines and creating backdoors for adversaries. Using a keylogger and reverse shell, adversaries can obtain sensitive information from an infected host. In most scenarios, adversaries combine exploits with regular files, so that a victim does not suspect the bait [147].
- **Reverse Social Engineering and Quid pro Quo:** Reverse social engineering is another way of manipulating victims to give away confidential information or to let the adversaries

Table 3. Social Engineering Techniques

Techniques	Approach	Target Information	Phase	Type
Phishing, Whaling Attack	Active/ Passive	User Credentials, Contact/Account Details	External	Remote
Watering hole Attack	Active	User/Host/Network/Application Information	External	Remote
Pretexting and Vishing	Active	User credentials, Organization Infrastructure, Physical Security	External	Remote
Baiting and Quid Pro Quo attacks	Active/Passive	User/Host/Network/Application Information	External/ Internal	Local
Tailgating	Active	Organization Infrastructure, Physical Security	External	Local
Reverse Social Engineering	Active/Passive	User Credentials	External	Local

gain access. Rick Nelson describes three parts of reverse social engineering: sabotage, advertising, and assisting [118]. Adversaries initiate the process by corrupting or damaging a particular device or workstation. Then, they show advertisements saying that they are capable of fixing it; when the victim asks for help, they extract target information during repair. Quid pro quo attacks are a form of reverse social engineering where adversaries call or send messages to random people at the target organization, asking if they requested technical support in the hope that they will eventually contact a person who did [85].

Table 3 shows approaches, typical target information, typical phases, and types of common social engineering techniques. Approach refers to whether a particular social engineering technique is active or passive. Social engineering techniques are usually utilized in the external phase and are quite effective in terms of collecting confidential user credentials or other sensitive information: around 85% of organizations have faced phishing or other social engineering attacks in 2019, which is 16% higher than in the previous year [138].

6.3 System-based Reconnaissance

System-based recon techniques can be categorized into *remote* and *local* information gathering techniques. Adversaries can perform scanning (e.g., TCP, UDP, or ICMP scans) and sniffing (often with the help of, e.g., MAC flooding or ARP spoofing) techniques in a network remotely. Local recon techniques, however, include discoveries within a compromised host by reading file contents or using operating system commands to explore configurations.

System-based recon techniques can involve gathering information by directly or indirectly interacting with a system. For example, an attacker can directly scan active hosts by interacting with the hosts (e.g., sending TCP SYN packets to them). The attacker can also gather information indirectly, without interacting with the target (e.g., observing or monitoring leaked information).

6.3.1 Remote System-based Reconnaissance Techniques. Adversaries can perform remote reconnaissance techniques from a remote location to gather information using direct or indirect interaction with a system. Network scanning and sniffing are performed to discover active network resources from an external network or within an internal network. Effective scanning techniques often enable adversaries to find vulnerabilities and to compromise IT assets [33]. This information can then be mapped to, e.g., a **Common Vulnerability and Exposure (CVE)** database, which provides detailed information about publicly known vulnerabilities. Databases and categorization

of CVEs are available at MITRE,[48] the National Vulnerability Database,[49] CVE Details,[50] and so on. Sniffing techniques are primarily used to capture network packets that reveal sensitive information such as user credentials and protocols being used in the network. One significant distinction between scanning and sniffing is that scanning techniques require direct interaction with the target system, while sniffing uses indirect interaction.

Scanning Techniques. Achleitner et al. categorized malicious network scanning based on the process of selecting addresses from a scanning space (e.g., IP address space) [16]. According to the authors, network scanning includes uniform scanning (probing random hosts within a IP range), local-preference (preferring a particular region), preference-sequential (probing IP addresses sequentially), non-preference sequential (selecting random IP ranges), and preference-parallel (performing parallel scans).

Scanning techniques can be categorized as *stealthy* or *non-stealthy* scanning. With stealthy scanning techniques, adversaries leave minimal trace of the scan and its origin, which makes stealthy scanning difficult to detect using conventional security measures. Non-stealthy scans are more "aggressive," and there is greater chance of being detected by an IDS. Stealthy scanning by bots is one of the most sophisticated techniques to efficiently gather information about a network [54]. Botnets can be configured to perform a variety of scan types, including uniform scanning where every host is scanned with equal probability [17], sequential scanning that systematically explores a space of IP addresses and/or ports [17], and preferential scanning that uses additional information to bias the search to specific parts of the network, types of hosts, or ports [15]. Botnet-based stealthy scanning is useful for discovering and compromising network infrastructure while minimizing detection by scanning from many hosts over multiple days [54].

Scanning techniques can also be categorized as *horizontal scans*, *vertical scans*, and *coordinated/distributed scans* [33]. If an adversary targets multiple ports on a single IP address, then the scan is vertical. A horizontal scan involves targeting a specific port on multiple IP addresses. A coordinated or distributed scan is a combination of both horizontal and vertical scans and can be launched from multiple scanning hosts (e.g., botnet-based scanning).

First, we discuss some of the most common *low-level (i.e., network or transport layer) scanning techniques*, emphasizing the network packet attributes.

- **TCP Scan with SYN/ACK Flag:** There are several TCP scanning techniques that use SYN or ACK flags to scan a network. TCP SYN scan is a widely used scanning technique; it does not establish a full connection, which makes it relatively stealthy and fast. Adversaries can use the ACK flag to identify open ports as well.
  - TCP Connect: TCP connect scan establishes a full three-way handshake with hosts within the target IP range [39]. It starts by sending a SYN packet from a client to the target host. The server responds with a SYN|ACK packet (RST packet is sent if the port is closed). Finally, the client sends an ACK in return, establishing the full connection. TCP connect is the simplest scanning technique, and it can be performed without admin privileges, since it scans active ports, which does not require any special flag settings. However, this scan increases the chance of being detected by an IDS due to establishing an active session [39].
  - TCP SYN Scan: SYN scan is a common scanning technique for identifying open and closed ports. SYN scan is also called a *half-open* scanning technique, since it does not establish a full TCP connection [105]. A SYN scan can be performed quickly within a given range of

[48] https://cve.mitre.org/.
[49] https://nvd.nist.gov/.
[50] https://www.cvedetails.com/.

ports, and it is a relatively stealthy technique. To perform this scan, adversaries send a SYN packet to the target host, and wait to receive the response. If a *SYN* or *ACK* is received, then the port is open. If the response is *RST (reset)*, then the port is closed.
  - ACK Flag Probe Scan: This scanning technique sets the *ACK* flag instead of the *SYN* flag and determines if a port is open, closed, or unfiltered by analyzing the **Time-To-Live (TTL)** and window fields within the RST packet header [26]. The target port is open if the *TTL* value is less than 64 or if the window value is not 0. Further, an ACK flag probe may also be able to differentiate between the presence of a stateful or stateless firewall and filtering rules by checking the response or error message (e.g., destination unreachable) [105].
- **TCP Scan based on RST Response:** Adversaries can set or unset several flags (e.g., FIN, PSH, URG) to perform stealthy scanning. Receiving a packet with *RST* means the port is closed; otherwise, it is open. A popular example of setting the flags is *XMAS* Scan. An inverse TCP scan sets either one flag or none in a TCP packet and is similar to XMAS Scan in terms of detecting open or closed ports.
  - XMAS Scan: XMAS scan is used to identify ports with the status open and closed [43]. The scan involves manipulating the *PSH*, *URG*, and *FIN* flags of a TCP header in crafted packets. An XMAS scan may bypass firewall and ACL filters, and it is fast as well [105]. It is called "XMAS scan," because if the packet is viewed within *Wireshark*, then the enabled alternating bits look like a XMAS-tree.
  - FIN Scan: FIN scan is also a stealthy scanning technique, similar to the XMAS scan. However, only the *FIN* flag is set [57].
  - NULL Scan: NULL scan is a stealthy technique similar to XMAS and FIN scanning techniques, but no flag is set in the packet [57]. The result is the same: ignored packet means open ports, while an *RST* response indicates that the corresponding port is closed.
- **UDP Scan:** UDP is simpler than TCP and does not provide the same variety of flag modification schemes as TCP does. However, a UDP scan can still be used to scan open UDP ports that provide a running service. In a UDP scan, a response is typically received if the port is closed. Typical open services such as DNS, VPN, SNMP, NTP, and so on, can be determined using UDP port scan [105]. In some cases, it is possible to detect versions of services and operating systems as well [105]. *Listing scanning* is another form of UDP scan that lists IP addresses and names by discovering hosts indirectly [26]. The technique involves performing a reverse DNS resolution to determine hostnames.
- **ICMP Scan:** A simple ICMP scan is performed to identify an active network device given a particular IP address [30]. An "ICMP Covering Ping Sweep" can discover active hosts within a range of IP addresses and can list active nodes based on the subnets [29].
- **ARP Scan:** ARP scanning is a network discovery technique that works by broadcasting an ARP packet in the network and checking which hosts respond [114]. Hosts that respond to the broadcast message are active hosts. The ARP scan is a low-level scanning technique that works in local area networks and is usually used to obtain both physical (MAC address) and logical (IPv4/6) addresses of active hosts.

Adversaries may be able to perform TCP, UDP, and ICMP scans from an external network, since all of these techniques are routable. Since ARP scan is non-routable, adversaries can perform it only in a local area network. Adversaries can start scanning hosts and ports locally once they have at least one compromised host in the target network.

Adversaries can also vary the attributes of network scans, including the speed, distribution, and destination of scanning [33]. Depending on their motivations and on the defenses of the networks,

Table 4. Network/Transport Layer Scanning Techniques and Tools

Techniques	Approach	Target Information	Phase	Tools
ICMP/TCP/UDP Scanning	Active	Network View/Security	Internal/ External	NMap[1]
Ping Sweep	Active	Network View	Internal/ External	NMap, Angry IP scanner[2], Solarwinds tools[3]
ARP Scanning	Active	Network View	Internal	ARP Ping[4]
Custom packets using TCP flags (SYN/ACK/FIN scan, XMAS scan, NULL scan)	Active	System Services, Network Security	Internal/ External	NMap, Hping2/ Hping3[6], Amap[7], SuperScan[8]
UDP Scan	Active	DNS, Network View, System Services	Internal/ External	Nmap

[1]https://nmap.org/; [2]https://angryip.org/;
[3]https://www.solarwinds.com/engineers-toolset/use-cases/network-monitoring-tools;
[4]https://www.netscantools.com/nstpro_arpping.html; [5]https://www.ettercap-project.org/; [6]http://www.hping.org/;
[7]https://tools.kali.org/information-gathering/amap; [8]https://sectools.org/tool/superscan/.

adversaries may prefer a *slow scan* approach to avoid detection [51]. For example, if a port scanner is scanning a host with ports ranging from 1 to 1024 and with a time interval of 5 minutes between each port, then performing the scan will take approximately 85 hours. It is harder for defenders to match and trace these suspicious packets in a vast dataset of traffic over a longer period in a large enterprise system.

Table 4 shows the approach, target information, phases, and examples of publicly available tools for scanning techniques. Scanning techniques include ICMP, UDP, ARP, or TCP scanning techniques. *Type* refers to whether the techniques are active or passive. *Target information* is what adversaries are looking for using these techniques. *Phase* denotes if a particular technique is utilized in external or internal phase. Finally, we include publicly available *tools* that are used by security researchers as references. However, adversaries may use more sophisticated techniques, such as exploiting services or software vulnerabilities without crashing, performing reconnaissance as regular users, and so on, to avoid detection [148].

Attackers can also perform *application-level scanning techniques*, such as banner grabbing, operating system and application fingerprinting. Here, we discuss some of the common techniques. Table 5 presents the approach, target information, phases, and examples of publicly available tools for different application-level scanning techniques.

- **Banner Grabbing:** Banner grabbing is a vulnerability scanning techniques that uses application banner information, including name and version [140]. There are two types of banner grabbing: active and passive. Active banner grabbing requires establishing TCP connections with a remote host to send crafted packets. Adversaries then receive and process the response. Passive banner grabbing involves passive sniffing techniques to capture and analyze network packets. Active banner grabbing techniques are more prone to detection by the defender. Adversaries usually target service ports, such as HTTP, FTP, and SMTP services (ports 80, 21, and 25, respectively). Using banner grabbing techniques adversaries can potentially map an entire network [31].
- **Fingerprinting:** Fingerprinting is a method of analyzing response packets to determine the operating system, application version (e.g., web server), or network protocol (e.g., SNMP). Often, the operating system and/or the application reply with packets that expose the

Table 5. Application-level Scanning Techniques and Tools

Techniques	Approach	Target Information	Phase	Tools
Banner grabbing and OS fingerprinting by sending crafted packets and analyzing responses	Active	System/Service Configurations, Applications Versions	Internal/ External	Telnet, NetCraft, IDServe,[1] Nmap, Winfingerprint,[2] Xprobe2[3]
Fingerprinting and patch-level assessment	Active/ Passive	Host/Network/ Application Vulnerabilities	Internal/ External	Nessus, Saint,[9] Cisco-Torch[10] and other vulnerability scanning tools

[1]https://www.grc.com/id/idserve.htm; [2]https://securiteam.com/tools/5HP0A1P2LK/;
[3]https://github.com/binarytrails/xprobe2; [4]http://lcamtuf.coredump.cx/p0f3/#;
[5]https://www.netresec.com/?page=networkminer;
[6]https://www.securitywizardry.com/products/scanning-products/wireless-tools/netsleuth;
[7]https://github.com/gamelinux/prads; [8]https://github.com/xnih/satori;
[9]https://www.saintcorporation.com/products/penetration-testing/;
[10]https://tools.kali.org/information-gathering/cisco-torch.

platform and version in the packet header. Adversaries can analyze the response packets, compare the values against a dataset of various operating systems and versions, and identify the OS version (e.g., APT32 [53]). Information can also be obtained by examining error-message responses.

Sniffing Techniques. Adversaries can perform sniffing to capture and analyze unencrypted network packets [49] to collect information like user credentials, e.g., usernames and passwords sent in plaintext. Network packets may also contain information about installed operating systems, applications, protocol versions, source, and destination ports, packet and frame sequences, and so on. By analyzing packets frame by frame, adversaries may be able to find misconfigurations and vulnerabilities in services. Some protocols are particularly vulnerable to sniffing; for example, Telnet can expose keystrokes (names and passwords), HTTP can reveal data sent in clear texts, SMTP/NMTP/POP/FTP/IMAP can reveal passwords or data sent in cleartext.

Sniffers usually operate in the data link layer of the OSI model. The objective is to compromise the communication channel before the defender in the upper layers is aware and prevents attacks. Attackers often place physical hardware sniffers or network analyzers if they can manage physical access (or a malicious insider) to an organization network (e.g., connect to the SPAN port of a switch that broadcasts all incoming or outgoing traffic).

Passive sniffing or directly capturing packets is performed for discovering network protocols and services, as well as active hosts and ports [49]. Many packet capturing and analysis tools are available on the market; for example, SolarWinds Network Performance Monitor,[51] ManageEngine NetFlow Analyzer,[52] tcpdump,[53] WinDump,[54] and Wireshark.[55] These are publicly available tools marketed to network admins, but may be used by adversaries as well. Adversaries can also perform scans using tools and scripts that are customized for a particular vulnerability to remain undetected for a longer period [158].

[51]https://www.solarwinds.com/network-performance-monitor.
[52]https://www.manageengine.com/products/netflow/.
[53]https://www.tcpdump.org/.
[54]https://www.winpcap.org/windump/.
[55]https://www.wireshark.org/.

Active sniffing involves traffic flooding or spoofing attacks to capture traffic or redirect the traffic towards a host controlled by the attacker. Active sniffing is usually performed in a switched network where the attacker might need to use these techniques to capture network traffic.

- **MAC Flooding:** MAC flooding involves flooding a switch with abundant mapping requests so that the switch overflows at some point [121]. Eventually, the switch acts as a hub and starts broadcasting all packets, making it easy for the attacker to capture packets.
- **ARP Spoofing:** In this techniques, the attacker usually generates a lot of forged ARP requests and reply packets to flood a switch. When flooded with spoofed ARP requests, the switch is set to "forwarding mode" and it is easier for the attacker to capture packets. The attacker can also try to poison the target's ARP table with forged entries that eventually lead to sophisticated attacks like Denial-of-Service and man-in-the-middle (MITM) [130].
- **MAC Duplicating/Spoofing:** The attacker can spoof the MAC address of an active target [27]. By duplicating the MAC address, the attacker can take over someone's identity. The technique is useful to gain access to the network if the target MAC address is used to authorize network access. However, this attack is easily detectable by the defender.
- **DHCP Starvation:** In this technique, the attacker sends "DHCP discovery" to the routers and attempts to lease all the available IP addresses [115]. DHCP starvation is sort of a Denial-of-Service (DoS) attack using DHCP requests. The primary reason for using this technique is to set up a rogue DHCP server that provides IP addresses to others joining the network. Then the attacker can establish the wrong IP, gateway, or DNS servers; used to capture packets.
- **DNS Poisoning:** DNS poisoning is performed by tricking a DNS server into believing the attacker has authentic information that allows the attacker to replace valid IP address entries with fake entries [27]. For example, the attacker can replace a valid IP entry with the IP of a fraud or a phishing site for social engineering or stealing information. The attacker can perform a DNS poisoning attack in two ways: within an internal network, or intranet (LAN), or replace entries stored in a proxy server. DNS poisoning helps the attacker to bypass security toolbars and phishing filters [14].

Table 6 presents the approach, target information, phases, and examples of publicly available tools for different sniffing techniques. Passive sniffing refers to listening to the network traffic where the active sniffing techniques are used to enable attacker capture packets in a switched network. Some of these techniques can be performed both externally or internally; other techniques are used within the local area network. Some remote side-channel attacks (e.g., timing or fault analysis) are used to reveal information by sending payloads and then analyzing the responses.

- **Timing Attack:** Leaked timing information from the CPU or memory can be utilized to determine the secret key of a crypto-system or algorithm (e.g., elliptic curve scalar multiplication algorithms). The time samples are gathered using various inputs and placed into a statistical model that predicts the key with a high degree of certainty [95, 123].
- **Differential Fault Analysis (DFA):** DFA is used primarily for performing cryptanalysis on several cryptographic algorithms (e.g., DES). To compute the amount of leaked information in a practical DFA attack, the attacker must first analyze the distribution of the leaked information and restrict the keyspace. The secret key can be discovered by using appropriate information estimate modeling [126].

6.3.2 Local System-based Reconnaissance. Once adversaries have compromised at least one asset in the target organization, they can start collecting local system information. For example, they can install rootkits, Trojan horses, or other malware that connect back to the command and

Table 6. Sniffing Techniques and Tools

Techniques	Approach	Target Information	Phase	Tools
MAC Flooding	Active	Running Protocols and Services, User Data, User Credentials	Internal	macof[1], yersinia[2]
ARP Spoofing	Active		Internal	WinARPAttacker,[4] Cain & Abel,[3] UfaSoft Snif[5]
MAC Duplicating	Active		Internal	macchanger[6]
Network Traffic Sniffing	Passive		Internal/ External	Wireshark, Ettercap, TCPdump, Windump
DHCP Starvation	Active		Internal	Gobbler[7]
DNS Poisoning	Active		Internal/ External	Ettercap

[1]https://github.com/WhiteWinterWolf/macof.py; [2]https://github.com/tomac/yersinia;
[3]https://github.com/xchwarze/Cain; [4]http://www.hacker-soft.net/Soft/Soft_2641.htm; [5]https://ufasoft.com/sniffer/;
[6]https://github.com/alobbs/macchanger; [7]http://gobbler.sourceforge.net/.

control servers established by adversaries beforehand [55]. Adversaries can then remotely execute commands or use additional exploits.

- **User and Group Discovery:** Adversaries can look for system and domain account information to learn about user and group credentials, which they may then use for privilege escalation. On the Windows platform, commands such as "`net user`", "`net group`", and "`net localgroup`" can be used for querying user or group information (e.g., APT1 [1]). On Unix-based systems, "`/etc/passwd`" and "`/etc/groups`" files are available for querying user and group information.
- **Process Discovery:** On most platforms there are several built-in command tools that can discover running processes on a system. For example, on the Windows platform, a built-in tool named "`tasklist`" is available for performing process and security system queries (e.g., APT: navRAT [111]). On Unix-based systems, the built-in command "`ps`" is available for checking running processes (e.g., APT: XAgentOSX [63]).
- **Service Discovery:** Adversaries can collect information about running services on the Windows platform using system commands like "`net start`" (e.g., APT: Sykipot [41]), "`tasklist`" (e.g., APT: Kwampirs [8]), or "`sc query`" (e.g., APT: OilRig [64]). On Unix-based systems, they can run system commands like "`service`", "`chkconfig`", or "`netstat`" to obtain service-oriented information.
- **Network Configuration Discovery:** Adversaries can look for basic network configuration information such as IP and MAC addresses, network adapters or interface, and so on, using the commands "`ipconfig`" (e.g., APT: BabyShark [13]) and "`ifconfig`" (e.g., APT: Calisto [99]) on Windows and Unix-based systems. They can then look for more details including the default gateway, primary and secondary WINS, DHCP configuration, and DNS server details. A number of APTs use "`nbtstat`" (e.g., APT: Epic [69]) or "`nbtscan`" (e.g., APT: Soft Cell [119]) to query NetBIOS name resolution information and to find vulnerabilities (e.g., APT: Turla [83]). ARP information can be obtained using the command "`arp -a`" (e.g., APT: Kwampirs [8]). Some APTs can perform query and enumeration over the ARP cache or table (e.g., APT: Olympic Destroyer [110]).
- **File and Directory Discovery:** Adversaries can list directory items on a Windows-based system by running "`dir`" or "`tree`" command (e.g., APT: BabyShark [13]). Adversaries have been reported to go through both system configuration files and user-created files [69]. On Unix systems, configuration files can typically be accessed from the "`/etc`" directory. Basic commands like "`ls`", "`find`", "`locate`" and so on, are available to search and explore files

on Unix systems. On Windows, software information is available in the "`Program Files`" directory. Adversaries can use custom scripts that can search for specific files with particular extensions (e.g., APT: Micropsia [156]).

- **Password Policy Discovery:** Adversaries can also learn information about the password policies enforced on a system. This is helpful for planning brute-forcing attacks or designing custom password dictionaries. Details such as user password age, password type, or hints can be obtained using user commands, e.g., "`chage -l $USER`" on Unix or Linux platforms. For the Windows platform, "`net accounts`" command provides account password policies (e.g., APT: OilRig [144]); while for macOS, user command "`pwpolicy getaccountpolicies`" can be used. On Linux systems, the policies are available in the "`/etc/pam.d/common-password`" file.
- **Network Statistics Discovery:** If adversaries intend to perform detailed internal scanning later, then they may initially want to learn network statistics, e.g., local TCP and UDP connections, routing tables, lists of network interfaces, and so on, using the command line tools "`netstat`" (e.g., APT: BlackEnergy [2, 36]), "`net use`" (e.g., APT: APT1 [1]), and "`net session`" (e.g., APT: Epic [69]). "`netstat -aon`" is a common command to gather network connection information; it reveals network connections and can search a specific IP range in a network.
- **Network Share Discovery:** Shared directories and files across the network provide access may also contain valuable information. Some APTs are also able to perform enumeration of network shares [153], which results in gathering potential attack vectors for other systems. "`net view`" or "`net share`" is used to collect SMB information across Windows platform-based networks (e.g., APT: APT41 [12]). Linux supports both NFS and SMB. "`smbclient`", "`nfsstat -m`", and "`df -aH`" commands can be used to explore if a network share is available on the compromised machine.
- **Keylogging and Screen Capture:** Adversaries can use keyloggers to collect users' keystrokes and information, such as passwords, habits, or financial information [38]. For example, terminal commands or application names typed in by a user can reveal further details of a system, used applications, and services. Keylogging helps adversaries to monitor host activities passively. Several keyloggers are also capable of recording desktop screens [38].

Adversaries utilize local system-based (host) reconnaissance techniques to determine installed software, applications, packages, and frameworks. Configurations and environment variables are relatively easy to discover in a compromised host. Files and directories may contain important and confidential information for further compromise or exhilaration. Internal host discovery can directly or indirectly lead to further exploitation, lateral movement, escalation, or data that are the ultimate target of the attacker. Table 7 presents examples of local discovery techniques, approach, target information, phases, and command-line tools with examples of APTs that use these tools.

There are several other local system-based reconnaissance techniques that require the attacker to have physical access to the system to observe characteristics of the system. For example, numerous side-channel attacks including power [128], **electromagnetic (EM)** emanation [95], and acoustical [66, 67] analyses require physical access to the devices.

- **Cache Attack:** A cache attack can be triggered by eavesdropping on keyboard timings, for example, an attack in an address in the GTK Library while processing keystrokes [71]. The attack is executed as a program that flushes the address and identifies when a keystroke occurred based on memory access times or the clflush instruction's execution time. Two major CPU vulnerabilities (Meltdown [102] and Spectre [96]) can be used to perform cache-based side-channel attacks by leaking sensitive information from the memory.

Table 7. Local System-based (Host) Reconnaissance Techniques

Techniques	Approach	Target Information	Phase	Commands-line Tools	Example APTs
User and Group Discovery	Active	Account Information	Internal	*Windows:* "`net user`", "`net group`", "`net localgroup`"	Ke3chang [159], OilRig [64]
Process Discovery	Active	Process	Internal	*Windows:* "`tasklist`"; *Unix:* "`ps`"	APT1 [1], BabyShark [13], ZxShell [24]
Service Discovery	Active	System Services, Service Configuration	Internal	*Windows:* "`net start`", "`tasklist`", "`sc query`"; *Unix:* "`service`", "`chkconfig`", "`netstat`"	APT1 [1], Epic [69], Ke3chang [159], OilRig [64], Turla [83]
Network Configuration Discovery	Active	Network View, Peripheral Devices	Internal	*Windows:* "`ipconfig`", "`nbtstat`", "`nbtscan`"; *Unix:* "`ifconfig`"	APT1 [1], APT32 [53], Epic [69], Turla [83]
File and Directory Discovery	Active	Files and Directories	Internal	*Windows:* "`dir`", "`tree`"; *Unix:* "`ls`", "`find`", "`locate`"	admin@338, BabyShark [13], Elise
Password Policy Discovery	Active	System Configuration (Password Policy)	Internal	*Windows:* "`net accounts`"; *Unix:* "`chage -l $USER`"; *macOS:* "`pwpolicy getaccountpolicies`"	Kwampirs [8], OilRig [64]
Network Statistics Discovery	Active	Network Traffics, Host Peripheral Devices	Internal	*Windows:* "`netstat`", "`net use`", "`net session`"; *Unix:* "`netstat`"	APT1 [1], APT32 [53], APT41 [12], Epic [69], Oilrig [64], Turla [83]
Network Share Discovery	Active	Shared Files and Directories	Internal	*Windows:* "`net view`", "`net share`"; *Unix:* "`smbclient`", "`nfsstat`", "`df`"	APT41 [12], Kwampirs [8]
Keylogging and Screen Capturing	Passive	Host I/O Interfacing	Internal	N/A	APT41 [12], Oilrig [64], Turla [83]

- **Electromagnetic Attack:** An electromagnetic signal carries information such as power, time, and so on. Leakage of this information can aid attackers to break into the security system to find out secret keys [95]. The leakage of compromising information via EM emanations from CMOS devices can lead to attacks on cryptographic devices where the power side channel is unavailable. Signal Detection and Estimation Theory techniques can be used to combine leakages from several EM channels, resulting in powerful attacks [19].
- **Acoustic Cryptanalysis:** Eavesdropping on acoustic emanations can be used to listen in on slow electromechanical components like keyboards and printers to reveal additional data for side channel attacks. Compromised mobile device eavesdropping, eavesdropping bugs, and auditory eavesdropping can all be used to carry out similar attacks [67]. For example, this data can be used to extract information about the CPU operations of laptop computers. An attacker can even discover the commands that the target computer executes by eavesdropping on acoustic emanations with a microphone [66].
- **Additional Methods:** In addition to the ones listed above there are many other local side-channel attacks such as NAND mirroring, clock or power glitches, temperature variation, smudges, differential computation analysis, and so on [145]. Nearly any feature of a local system is potentially a useful source of side-channel information.

7 CONCLUSION AND FUTURE WORK

Gaining a clearer understanding of how and why cyber adversaries conduct reconnaissance activities is a critical area of research for cyber defense, since successful attacks depend so heavily

on effective reconnaissance. However, we find that there is little comprehensive research on this topic to establish a big picture view of how reconnaissance works, including the large variety of methodologies and tools used to conduct reconnaissance. Our first research goal was to establish a broad picture of what types of information adversaries seek. Next, we consider when attackers conduct reconnaissance in the standard kill chain model, as well as from what perspective (internal vs. external). Our third goal focuses on understanding the wide variety of specific techniques and tools adversaries use to gather this information. We developed taxonomies for both information types and the techniques to help organize these into useful categories. While we draw inspiration from distinctions previously drawn in the literature, in some cases we find that in some cases these distinctions were too vague or limited to be used for a general taxonomy, so we adopted new dimensions that are clearer and more useful to practitioners.

One of the main lessons from our survey is the overall scope and diversity of adversarial reconnaissance in cybersecurity. The variety of types of information that could potentially be useful to an attacker is vast, as is the number of tools and specific techniques for obtaining it. This is also a moving target, since the types of information that are relevant and the tools will naturally evolve over time with technology. Nevertheless, we were able to capture some common distinctions in our taxonomy as well as the analysis and clustering of more specific techniques. For example, it is important to think broadly about the types of the information that is being collected and the different places it is collected from, including what is publicly accessible and what is not. The techniques used are quite different depending on the source of the information (including the key distinction between technological and social methods), and therefore the relevant defenses are also quite different. Some common features of the techniques such as the spectrum from active to passive are also very important, since these have a direct correlation with the likelihood of detection and when in the attack cycle they are most likely to be deployed. We also observe that the type and objectives of the adversary may have a great impact on how they conduct reconnaissance activities. We now go into greater detail on some of the specific observations from our study that can lead to areas for improvement and future research in adversarial reconnaissance as well as potential counter-measures.

7.1 Improving Adversarial Reconnaissance Models

Our survey provides a comprehensive overview of reconnaissance activities; however, we still have a limited understanding of how to model the details of the reconnaissance process of different types of attackers. This includes how they make decisions about what types of reconnaissance to conduct, how to prioritize different types of information, and how they form detailed beliefs about systems and defenses based on limited and uncertain information. There is also a limited understanding of how attackers make key tradeoffs such as utilizing stealthy vs. non-stealthy methods.

While there are many case studies and examples, we lack a general framework and data to model typical reconnaissance activities. The formal models that do exist to date (e.g., in the literature on cyber deception) are typically quite simplistic and/or limited in scope (e.g., specifying just one type of scanning procedure and assuming attackers collect perfect information. There are some recent studies that have considered evaluating the efficiency of deception in the reconnaissance phase, including models that incorporate Bayesian updating of beliefs [149]. Another study developed a model of the reconnaissance capabilities of persistent, stealthy adversaries and demonstrated that these adversaries are capable of conducting effective network reconnaissance passively; this offers a method for defining cost and reward criteria that adversaries use to determine which targets to pursue when moving laterally across the network [124]. Another line of work considers modeling adversary knowledge using a set of logic formulas with probabilities [87].

Table 8. Defensive Measures against Reconnaissance Techniques

Measures \ Techniques	Third-party source-based	Human-based	System-based
Reconnaissance Detection	X	✓	✓
Cyber Deception/MTD	X	X	✓
Security Awareness and Best Practices	✓	✓	✓

7.2 Empirical Studies of Reconnaissance

A related issue is the general need for more empirical research to answer basic questions including what the most common types of reconnaissance activities are, how these activities vary across different types of attackers, how attackers make decisions about conducting reconnaissance, how they use this information in attack planning. Such studies are naturally difficult to conduct, since attackers actively try to hide much of this information, but there is still a notable lack of good, high-quality data and empirical work to study both the prevalence and effectiveness of different types of reconnaissance approaches in both controlled and uncontrolled environments. There is also a lack of good metrics and empirical evidence regarding the effectiveness of different defensive mitigation strategies at limited or obfuscating the information attackers can gather, as well as the ability to detect different types of reconnaissance activities. More effective models of the process and beliefs of adversaries will help to scope this type of evaluation, but we also need better sources of real world data and experimental designs to understand how attackers gather information in the real world.

7.3 Reconnaissance Countermeasures

Another useful outcome of our survey is to contribute to developing countermeasures that can hinder the ability of adversaries to obtain key information that would enable successful attacks. Network and host-based intrusion detection systems typically monitor for scanning and other known/obvious adversarial reconnaissance activities. Many techniques have been proposed in the literature to use deception and information hiding to mitigate reconnaissance, including honeypots, honey tokens, honey passwords, honey permissions and parameters, and so on [88, 89, 164]. **Moving Target Defense (MTD)** can also increase the complexity, diversity, and randomness of the cyber systems for an attacker doing reconnaissance [161]. Techniques such as dynamic host address translation, route alteration, and IP randomization can lower the success of passive reconnaissance [16, 17]. Additional methods including database decoys, OS obfuscation, source code decoys, forging fake traffic, topology deception, hyperlinks decoys, simulation deception, and code embedding deception [132, 134, 150] can mitigate both passive and active reconnaissance. Employee training, security awareness and best practices can mitigate social engineering tactics to some extent. Table 8 presents an overview of which types of defensive measures can counter particular reconnaissance techniques.

Future work could elaborate on reconnaissance countermeasures based on each category in the taxonomy. Defenders can detect some techniques more easily; for example, scanning and sniffing in an internal network can be detected using an intrusion detection system. There is a need to evaluate the effectiveness of different strategies to detect the different types of recon techniques. Some techniques (e.g., side-channel attacks) are challenging to detect, and other techniques (e.g., third-party source-based recon) cannot be identified. Therefore, other types of mitigation strategies are necessary, and must also be evaluated. Better models (as discussed above) can help to formalize many of these questions and provide useful evaluation measures, and out taxonomy can help to ensure comprehensive coverage and identify areas with limited mitigation options.

7.4 Evolving Forms of Reconnaissance Techniques

As technology changes, the nature of reconnaissance also changes due to new types of information becoming relevant as well as new techniques being developed to extract useful information. While we have focused mostly on common current techniques, we note some evolving trends that will affect adversary reconnaissance in the future. One is the rise of disruptive technologies such as virtualization, cloud, fog, and mobile or edge computing, and containerization. One of the effects of this is that organizations often do not control all of their own computing resources and data locally, but outsource them to other vendors who operate cloud resources. This presents new opportunities for social engineering attacks, as well as new types of side channel attacks, for example, cross-VM cache-based [28], GPU-based [116], or directory-based [167], and so on, that have only recently begun to be recognized and considered in the literature [106]. The increasing complexity for organizations that must operate or source software and hardware across national boundaries and different regulatory jurisdictions is also an issue that will need further consideration.

Another rapidly evolving area is the use of artificial intelligence and machine learning methods both in business processes as well as network management and cyber defense. These autonomous agents present a new target for attackers. One example of this is accessing valuable data, such as by stealing machine learning models that have been trained on highly valuable data sets [109, 112, 155]. Another problems is the potential for attackers to fool AI system (including authentication and intrusion detection systems) into making erroneous decisions and providing additional vulnerabilities for system access [127]. The means for attackers to learn about these automated systems are only just starting to be explored, and while impressive proofs of concept of the vulnerabilities of these systems have been demonstrated there is limited work done so far to understand the extent, prevalence, and exploitability of these systems in the real world, or to address how attackers can systematically gather reconnaissance information about AI systems to use in attacks.

REFERENCES

[1] 2013. APT1: Exposing One of China's Cyber Espionage Units. Mandiant, FireEye. Retrieved from https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf.

[2] 2014. BlackEnergy & Quedagh: The Convergence of Crimeware and APT Attacks. F-Secure Labs. Retrieved from https://www.f-secure.com/documents/996508/1030745/blackenergy_whitepaper.pdf.

[3] 2014. DARKCOMET–Threat Encyclopedia. Trend Micro. Retrieved from https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/DARKCOMET.

[4] 2016. Follow The Money: Dissecting The Operations Of The Cyber Crime Group FIN6. FireEye Inc. Retrieved from https://www2.fireeye.com/rs/848-DID-242/images/rpt-fin6.pdf.

[5] 2017. HIDDEN COBRA–North Korean Remote Administration Tool: FALLCHILL. CISA. Retrieved from https://www.us-cert.gov/ncas/alerts/TA17-318A.

[6] 2018. APT38: Un-usual Suspects. FireEye Inc. Retrieved from https://content.fireeye.com/apt/rpt-apt38.

[7] 2018. Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019. Retrieved from https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-billion-in-2019.

[8] 2018. New Orangeworm Attack Group Targets the Healthcare Sector in the U.S., Europe, and Asia. Symantec. Retrieved from https://www.symantec.com/blogs/threat-intelligence/orangeworm-targets-healthcare-us-europe-asia.

[9] 2018. North Korean Regime-backed Programmer Charged with Conspiracy to Conduct Multiple Cyber Attacks and Intrusions. OPA–Department of Justice. Retrieved from https://www.justice.gov/opa/pr/north-korean-regime-backed-programmer-charged-conspiracy-conduct-multiple-cyber-attacks-and.

[10] 2018. Spear Phishing Attacks: Why They Are Successful and How to Stop Them. FireEye Inc. Retrieved from https://www.fireeye.com/solutions/ex-email-security-products/wp-spearphishing-attacks.html.

[11] 2019. 2019 Data Breach Investigations Report. Verizon. Retrieved from https://enterprise.verizon.com/resources/reports/2019-data-breach-investigations-report-emea.pdf.

[12] 2019. Double Dragon: APT41, a Dual Espionage and Cyber Crime Operation. FireEye Inc. Retrieved from https://content.fireeye.com/apt-41/rpt-apt41.
[13] Unit 42. 2019. New BabyShark Malware Targets U.S. National Security Think Tanks. Retrieved from https://unit42.paloaltonetworks.com/new-babyshark-malware-targets-u-s-national-security-think-tanks/.
[14] Saeed Abu-Nimeh and Suku Nair. 2008. Bypassing security toolbars and phishing filters via dns poisoning. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'08)*. IEEE, 1–6.
[15] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. 2006. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. 41–52.
[16] Stefan Achleitner, Thomas La Porta, Patrick McDaniel, Shridatt Sugrim, Srikanth V. Krishnamurthy, and Ritu Chadha. 2016. Cyber deception: Virtual networks to defend insider reconnaissance. In *Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats*. ACM, 57–68.
[17] Stefan Achleitner, Thomas F. La Porta, Patrick McDaniel, Shridatt Sugrim, Srikanth V. Krishnamurthy, and Ritu Chadha. 2017. Deceiving network reconnaissance using SDN-based virtual topologies. *IEEE Trans. Netw. Serv. Manage.* 14, 4 (2017), 1098–1112.
[18] Steven Adair. 2016. PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs. Volexity. Retrieved from https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/.
[19] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. 2002. The EM side-channel (s). In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 29–45.
[20] Mohammed I. Al-Saleh and Jedidiah R. Crandall. 2011. Application-level reconnaissance: Timing channel attacks against antivirus software. In *Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'11)*.
[21] Safwan Alam and Khalil El-Khatib. 2016. Phishing susceptibility detection through social media analytics. In *Proceedings of the 9th International Conference on Security of Information and Networks (SINCONF'16)*. 61–64.
[22] Samar Muslah Albladi and George R. S. Weir. 2018. User characteristics that influence judgment of social engineering attacks in social networks. *Hum.-centr. Comput. Inf. Sci.* 8, 1 (2018), 5.
[23] Dalal N. Alharthi, Mahmoud M. Hammad, and Amelia C. Regan. 2020. A taxonomy of social engineering defense mechanisms. In *Future of Information and Communication Conference*. Springer, 27–41.
[24] Andrea Allievi, Douglas Goddard, Shaun Hurley, and Alain Zidouemba. 2014. Threat Spotlight: Group 72, Opening the ZxShell. Cisco. Retrieved from https://blogs.cisco.com/security/talos/opening-zxshell.
[25] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Xiaojing Liao, Alina Oprea, XiaoFeng Wang, and Zhou Li. 2016. Catching predators at watering holes: Finding and understanding strategically compromised websites. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 153–166.
[26] Amir. 2019. Network Scanning Techniques. DevQA. Retrieved from https://devqa.io/network-scanning-techniques/.
[27] P. Anu and S. Vimala. 2017. A survey on sniffing attacks on computer networks. In *Proceedings of the International Conference on Intelligent Computing and Control (I2C2'17)*. IEEE, 1–5.
[28] Shahid Anwar, Zakira Inayat, Mohamad Fadli Zolkipli, Jasni Mohamad Zain, Abdullah Gani, Nor Badrul Anuar, Muhammad Khurram Khan, and Victor Chang. 2017. Cross-VM cache-based side channel attacks and proposed prevention mechanisms: A survey. *J. Netw. Comput. Appl.* 93 (2017), 259–279. https://www.sciencedirect.com/science/article/pii/S1084804517302205.
[29] Ofir Arkin. 1999. Network Scanning Techniques. Publicom Communications Solutions (1999).
[30] Ofir Arkin et al. 2001. ICMP Usage in Scanning. The Complete Know-How (2001). The Sys-Security Group.
[31] Pranshu Bajpai, Aditya K. Sood, and Richard J. Enbody. 2018. The art of mapping IoT devices in networks. *Netw. Secur.* 2018, 4 (2018), 8–15.
[32] Mark Anthony Balanza. 2018. Infostealer.Catchamas. Symantec. Retrieved from https://www-west.symantec.com/content/symantec/english/en/security-center/writeup.html/2018-040209-1742-99.
[33] Richard J. Barnett and Barry Irwin. 2008. Towards a taxonomy of network scanning techniques. In *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*. 1–7.
[34] Genevieve Bartlett, John Heidemann, and Christos Papadopoulos. 2007. Understanding passive and active service discovery. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. 57–70.
[35] Salman Baset, Sahil Suneja, Nilton Bila, Ozan Tuncer, and Canturk Isci. 2017. Usable declarative configuration specification and validation for applications, systems, and cloud. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Industrial Track*. 29–35.
[36] Kurt Baumgartner and Maria Garnaeva. 2014. BE2 Custom Plugins, Router Abuse, and Target Profiles—New observations on BlackEnergy2 APT activity. SecureList. Retrieved from https://securelist.com/be2-custom-plugins-router-abuse-and-target-profiles/67353/.

[37] Elisa Bertino, Ashish Kamra, and James P. Early. 2007. Profiling database application to detect SQL injection attacks. In *Proceedings of the IEEE International Performance, Computing, and Communications Conference*. IEEE, 449–458.
[38] Akashdeep Bhardwaj and Sam Goundar. 2020. Keyloggers: Silent cyber security weapons. *Netw. Secur.* 2 (2020), 14–19.
[39] Monowar H. Bhuyan, Dhruba Kr Bhattacharyya, and Jugal K. Kalita. 2011. Surveying port scans and their detection methodologies. *Comput. J.* 54, 10 (2011), 1565–1581.
[40] Roberto Bifulco, Heng Cui, Ghassan O. Karame, and Felix Klaedtke. 2015. Fingerprinting software-defined networks. In *Proceedings of the IEEE 23rd International Conference on Network Protocols (ICNP'15)*. IEEE, 453–459.
[41] Jaime Blasco. 2011. Another Sykipot Sample Likely Targeting US Federal Agencies AT&T Alien Labs. AT&T. Retrieved from https://cybersecurity.att.com/blogs/labs-research/another-sykipot-sample-likely-targeting-us-federal-agencies.
[42] Elias Bou-Harb, Mourad Debbabi, and Chadi Assi. 2013. Cyber scanning: A comprehensive survey. *IEEE Commun. Surv. Tutor.* 16, 3 (2013), 1496–1519.
[43] Jarryd Boyd. 2015. Understanding Xmas Scans. Retrieved from https://www.plixer.com/blog/understanding-xmas-scans/.
[44] Sandor Boyson. 2014. Cyber supply chain risk management: Revolutionizing the strategic control of critical IT systems. *Technovation* 34, 7 (2014), 342–353.
[45] Matthew Brocker and Stephen Checkoway. 2014. iSeeYou: Disabling the MacBook webcam indicator LED. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security'14)*. 337–352.
[46] Richard G. Brody, Elizabeth Mulig, and Valerie Kimball. 2007. Phishing, pharming and identity theft. *Acad. Account. Financ. Stud. J.* 11, 3 (2007), 43–56.
[47] Nick Carr. 2017. Cyber Espionage Is Alive and Well: APT32 and the Threat to Global Corporations. FireEye Inc. Retrieved from https://www.fireeye.com/blog/threat-research/2017/05/cyber-espionage-apt32.html.
[48] Defense Use Case. 2016. Analysis of the cyber attack on the Ukrainian power grid. Electricity Information Sharing and Analysis Center (E-ISAC'16). SANS Industrial Control System.
[49] William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin. 2003. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Longman.
[50] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. 2018. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Syst. Appl.* 106 (2018), 1–20.
[51] Brenden Claypool. 2002. Stealth port scanning methods. *Glob. Inf. Assur. Cert. Pap.* 1, 4 (2002).
[52] Kyle Coffey, Richard Smith, Leandros Maglaras, and Helge Janicke. 2018. Vulnerability analysis of network scanning on SCADA systems. *Secur. Commun. Netw.* (2018).
[53] Assaf Dahan. 2017. Operation Cobalt Kitty. Cybereason. Retrieved from https://cdn2.hubspot.net/hubfs/3354902/Cybereason%20Labs%20Analysis%20Operation%20Cobalt%20Kitty.pdf.
[54] Alberto Dainotti, Alistair King, Kimberly Claffy, Ferdinando Papale, and Antonio Pescapé. 2014. Analysis of a "/0" stealth scan from a botnet. *IEEE/ACM Trans. Netw.* 23, 2 (2014), 341–354.
[55] Michael K. Daly. 2009. Advanced Persistent Threat (or Informationized Force Operations). *23rd Large Installation System Administration Conference (LISA'09)*.
[56] Usman Ali Dar and Arsalan Iqbal. 2018. The silent art of reconnaissance: The other side of the hill. *Int. J. Comput. Netw. Commun. Secur.* 6, 12 (2018), 250–263.
[57] Marco De Vivo, Eddy Carrasco, Germinal Isern, and Gabriela O. de Vivo. 1999. A review of port scanning techniques. *ACM SIGCOMM Comput. Commun. Rev.* 29, 2 (1999), 41–48.
[58] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. 2018. Investigating system operators' perspective on security misconfigurations. In *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security*. 1272–1289.
[59] Jerome Doaty and Garrett Primm. 2018. We're Seeing a Resurgence of the Demonic Astaroth WMIC Trojan. Cofense. Retrieved from https://cofense.com/seeing-resurgence-demonic-astaroth-wmic-trojan/.
[60] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. 2018. I never signed up for this! Privacy implications of email tracking. In *Proceedings of the 18th Privacy Enhancing Technologies Symposium (PETS'18)*, 109–126.
[61] ESET. 2019. MACHETE Just Got Sharper: Venezuelan Government Institutions Under Attack. Retrieved October 21, 2019 from https://www.welivesecurity.com/wp-content/uploads/2019/08/ESET_Machete.pdf.
[62] Birhanu Eshete, Adolfo Villafiorita, and Komminist Weldemariam. 2011. Early detection of security misconfiguration vulnerabilities in web applications. In *Proceedings of the 6th International Conference on Availability, Reliability and Security*. IEEE, 169–174.
[63] Robert Falcone. 2017. XAgentOSX: Sofacy's XAgent macOS Tool. Retrieved from https://unit42.paloaltonetworks.com/unit42-xagentosx-sofacys-xagent-macos-tool/.

[64] Robert Falcone and Bryan Lee. 2016. The OilRig Campaign: Attacks on Saudi Arabian Organizations Deliver Helminth Backdoor. Retrieved from https://unit42.paloaltonetworks.com/the-oilrig-campaign-attacks-on-saudi-arabian-organizations-deliver-helminth-backdoor/.
[65] Robert Falcone, Bryan Lee, and Riley Porter. 2018. OilRig Targets a Middle Eastern Government and Adds Evasion Techniques to OopsIE. paloalto networks. Retrieved October 27, 2019 from https://unit42.paloaltonetworks.com/unit42-oilrig-targets-middle-eastern-government-adds-evasion-techniques-oopsie/.
[66] Daniel Genkin, Adi Shamir, and Eran Tromer. 2014. RSA key extraction via low-bandwidth acoustic cryptanalysis. In *Annual Cryptology Conference*. Springer, 444–461.
[67] Daniel Genkin, Adi Shamir, and Eran Tromer. 2017. Acoustic cryptanalysis. *J. Cryptol.* 30, 2 (2017), 392–443.
[68] Michael Glassman and Min Ju Kang. 2012. Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT). *Comput. Hum. Behav.* 28, 2 (2012), 673–682.
[69] GReAT. 2014. The Epic Turla Operation. SecureList. Retrieved from https://securelist.com/the-epic-turla-operation/65545/.
[70] GReAT. 2015. Darkhotel's Attacks in 2015. Securelist. Retrieved from https://securelist.com/darkhotels-attacks-in-2015/71713/.
[71] Daniel Gruss, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. 2016. Flush+ Flush: A fast and stealthy cache attack. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 279–299.
[72] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. 2016. A literature survey on social engineering attacks: Phishing attack. In *Proceedings of the International Conference on Computing, Communication and Automation (ICCCA'16)*. IEEE, 537–540.
[73] Shuang Hao, Nick Feamster, and Ramakant Pandrangi. 2010. *An Internet-wide View into DNS Lookup Patterns*. Technical Report. School of Computer Science, Georgia Tech.
[74] Nihad A. Hassan. 2019. Gathering evidence from OSINT sources. In *Digital Forensics Basics*. Springer, 311–322.
[75] Nihad A. Hassan and Rami Hijazi. 2018. The evolution of open source intelligence. In *Open Source Intelligence Methods and Tools*. Springer, 1–20.
[76] Nihad A. Hassan and Rami Hijazi. 2018. Technical footprinting. In *Open Source Intelligence Methods and Tools*. Springer, 313–339.
[77] Joseph M. Hatfield. 2019. Virtuous human hacking: The ethics of social engineering in penetration-testing. *Comput. Secur.* 83 (2019), 354–366.
[78] Ryan Heartfield and George Loukas. 2015. A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks. *Comput. Surv.* 48, 3 (2015), 1–39.
[79] Benjamin Hettwer, Stefan Gehrer, and Tim Güneysu. 2020. Applications of machine learning techniques in side-channel attacks: A survey. *J. Cryptogr. Eng.* 10, 2 (2020), 135–162.
[80] Julie Andersen Hill. 2018. SWIFT bank heists and article 4A. *J. Consum. Commerc. Law* 22, 1 (2018).
[81] Hannes Holm, Teodor Sommestad, Jonas Almroth, and Mats Persson. 2011. A quantitative evaluation of vulnerability scanning. *Inf. Manage. Comput. Secur.* 19, 4 (2011), 231–247.
[82] Nazrul Hoque, Monowar H. Bhuyan, Ram Charan Baishya, Dhruba K. Bhattacharyya, and Jugal K. Kalita. 2014. Network attacks: Taxonomy, tools and systems. *J. Netw. Comput. Appl.* 40 (2014), 307–324.
[83] Threat Intelligence. 2019. Waterbug: Espionage Group Rolls Out Brand-New Toolset in Attacks Against Governments. Symantec. Retrieved from https://www.symantec.com/blogs/threat-intelligence/waterbug-espionage-governments.
[84] Anton Ivanov, Fedor Sinitsyn, and Orkhan Mamedov. 2018. SynAck Targeted Ransomware Uses the Doppelgänging Technique. Securelist. Retrieved from https://securelist.com/synack-targeted-ransomware-uses-the-doppelganging-technique/85431/.
[85] Koteswara Ivaturi and Lech Janczewski. 2011. A taxonomy for social engineering attacks. In *International Conference on Information Resources Management*. Centre for Information Technology, Organizations, and People, 1–12.
[86] Gregoire Jacob, Ralf Hund, Christopher Kruegel, and Thorsten Holz. 2011. JACKSTRAWS: Picking command and control connections from bot traffic. In *Proceedings of the USENIX Security Symposium*, Vol. 2011. San Francisco, CA.
[87] Sushil Jajodia, Noseong Park, Fabio Pierazzi, Andrea Pugliese, Edoardo Serra, Gerardo I. Simari, and V. S. Subrahmanian. 2017. A probabilistic logic of cyber deception. *IEEE Trans. Inf. Forens. Secur.* 12, 11 (2017), 2532–2544.
[88] Ari Juels and Ronald L. Rivest. 2013. Honeywords: Making password-cracking detectable. In *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*. ACM, 145–160.
[89] Parisa Kaghazgaran and Hassan Takabi. 2015. Toward an insider threat detection framework using honey permissions. *J. Internet Serv. Inf. Secur.* 5, 3 (2015), 19–36.
[90] Michael Kan. 2017. Chinese Hackers Go After Third-party IT Suppliers to Steal Data. Network World. Retrieved from https://www.networkworld.com/article/3187359/chinese-hackers-go-after-third-party-it-suppliers-to-steal-data.html.

[91] Miltiadis Kandias, Nikos Virvilis, and Dimitris Gritzalis. 2011. The insider threat in cloud computing. In *Proceedings of the 6th International Workshop on Critical Information Infrastructures Security (CRITIS'11)*. Springer, 93–103.
[92] Anna Kang, Jae Dong Lee, Won Min Kang, Leonard Barolli, and Jong Hyuk Park. 2014. Security considerations for smart phone smishing attacks. In *Advances in Computer Science and its Applications*. Springer, 467–473.
[93] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. 2020. A survey exploring open source Intelligence for smarter password cracking. *Forens. Sci. Int.: Digit. Invest.* 35 (2020), 301075.
[94] Rotem Kerner. 2015. *Reconnaissance: A Walkthrough of the "APT" Intelligence Gathering Process*. Technical Report. EMC Corporation.
[95] Ajoy Kumar Khan and Hridoy Jyoti Mahanta. 2014. Side channel attacks and their mitigation techniques. In *Proceedings of the 1st International Conference on Automation, Control, Energy and Systems (ACES'14)*. IEEE, 1–4.
[96] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. 2019. Spectre attacks: Exploiting speculative execution. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'19)*. IEEE, 1–19.
[97] Katherine Koleski. 2017. *The 13th Five-Year Plan*. Technical Report. U.S.-China Economic and Security Review Commission.
[98] Katharina Krombholz, Heidelinde Hobel, Markus Huber, and Edgar Weippl. 2015. Advanced social engineering attacks. *J. Inf. Secur. Appl.* 22 (2015), 113–122.
[99] Mikhail Kuzin and Sergey Zelensky. 2018. Calisto Trojan for macOS: The First Member of the Proton Malware Family? Securelist. Retrieved from https://securelist.com/calisto-trojan-for-macos/86543/.
[100] Denis Legezo. 2019. Chafer Used Remexi Malware to Spy on Iran-based Foreign Diplomatic Entities. SecureList. Retrieved from https://securelist.com/chafer-used-remexi-malware/89538/.
[101] Jack W. Lightfoot. 2016. *Law Enforcements' Perceptions and Preparedness to Address Child Exploitation Via Hacking*. Master's Thesis. Georgia Southern University.
[102] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, et al. 2018. Meltdown: Reading kernel memory from user space. In *Proceedings of the 27th {USENIX} Security Symposium ({USENIX} Security'18)*. 973–990.
[103] Suqi Liu, Ian Foster, Stefan Savage, Geoffrey M. Voelker, and Lawrence K. Saul. 2015. Who is. com? Learning to parse WHOIS records. In *Proceedings of the Internet Measurement Conference*. 369–380.
[104] Xin Robert Luo, Wei Zhang, Stephen Burd, and Alessandro Seazzu. 2013. Investigating phishing victimization with the heuristic–systematic model: A theoretical framework and an exploration. *Comput. Secur.* 38 (2013), 28–38.
[105] Gordon Fyodor Lyon. 2009. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.
[106] Yangdi Lyu and Prabhat Mishra. 2018. A survey of side-channel attacks on caches and countermeasures. *J. Hardw. Syst. Secur.* 2, 1 (2018), 33–50.
[107] Steve Mansfield-Devine. 2009. Simple website footprinting. *Netw. Secur.* 2009, 4 (2009), 7–9.
[108] Wojciech Mazurczyk and Luca Caviglione. 2021. Cyber reconnaissance techniques. *Commun. ACM* 64, 3 (2021), 86–95.
[109] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Fast, lean, and accurate: Modeling password guessability using neural networks. In *Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security'16)*. 175–191.
[110] Warren Mercer and Paul Rascagneres. 2018. Comprehensive Threat Intelligence: Olympic Destroyer Takes Aim At Winter Olympics. Cisco Talos Blog. Retrieved from https://blog.talosintelligence.com/2018/02/olympic-destroyer.html.
[111] Warren Mercer, Paul Rascagneres, and Jungsoo An. 2018. Korea in the Crosshairs. Cisco Talos Intelligence Group. Retrieved from https://blog.talosintelligence.com/2018/01/korea-in-crosshairs.html.
[112] Yuantian Miao, Chao Chen, Lei Pan, Qing-Long Han, Jun Zhang, and Yang Xiang. 2021. Machine learning based cyber attacks targeting on controlled information: A survey. arXiv:2102.07969. Retrieved from https://arxiv.org/abs/2102.07969.
[113] Kevin D. Mitnick and William L. Simon. 2011. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons.
[114] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. 2013. A survey of intrusion detection techniques in cloud. *J. Netw. Comput. Appl.* 36, 1 (2013), 42–57.
[115] Husameldin Mukhtar, Khaled Salah, and Youssef Iraqi. 2012. Mitigation of DHCP starvation attack. *Comput. Electr. Eng.* 38, 5 (2012), 1115–1128.
[116] Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh. 2018. Rendered insecure: Gpu side channel attacks are practical. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 2139–2153.

[117] Arvind Narayanan and Vitaly Shmatikov. 2005. Fast dictionary attacks on passwords using time-space tradeoff. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*. 364–372.
[118] Rick Nelson. 2001. Methods of Hacking: Social Engineering. Institute for Systems Research.
[119] Cybereason Nocturnus. 2019. Operation Soft Cell: A Worldwide Campaign Against Telecommunications Providers. Retrieved from https://www.cybereason.com/blog/operation-soft-cell-a-worldwide-campaign-against-telecommunications-providers.
[120] Nick Nykodym, Robert Taylor, and Julia Vilela. 2005. Criminal profiling and insider cyber crime. *Comput. Law Secur. Rev.* 21, 5 (2005), 408–414.
[121] Alexander Grigorievich Ostapenko, Sergei Sergeyevich Kulikov, Nikolai Nikolaevich Tolstykh, Yuri Gennadievich Pasternak, and Larisa Georgievna Popova. 2013. Denial of service in components of information telecommunication systems through the example of "network storm" attacks. *World Appl. Sci. J.* 25, 3 (2013), 404–409.
[122] Rodney Owens and Weichao Wang. 2011. Non-interactive OS fingerprinting through memory de-duplication technique in virtual machines. In *Proceedings of the 30th IEEE International Performance Computing and Communications Conference*. 1–8.
[123] Thomas Perianin, Sebastien Carré, Victor Dyseryn, Adrien Facon, and Sylvain Guilley. 2021. End-to-end automated cache-timing attack driven by machine learning. *J. Cryptogr. Eng.* 11, 2 (2021), 135–146.
[124] Luan Huy Pham, Massimiliano Albanese, Ritu Chadha, Cho-Yu J Chiang, Sridhar Venkatesan, Charles Kamhoua, and Nandi Leslie. 2020. A quantitative framework to model reconnaissance by stealthy attackers and support deception-based defenses. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS'20)*. IEEE, 1–9.
[125] Clifton Phua. 2009. Protecting organisations from personal data breaches. *Comput. Fraud Secur.* 2009, 1 (2009), 13–18. https://www.sciencedirect.com/science/article/abs/pii/S1361372309700119.
[126] Gilles Piret and Jean-Jacques Quisquater. 2003. A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. In *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 77–88.
[127] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. 2019. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* 34 (2019), 100199.
[128] Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. 2009. Statistical analysis of second order differential power analysis. *IEEE Trans. Comput.* 58, 6 (2009), 799–811.
[129] Bipin Kumar Rai, Ravi Verma, and Shiva Tiwari. 2021. Using open source intelligence as a tool for reliable web searching. *SN Comput. Sci.* 2, 5 (2021), 1–12.
[130] Vivek Ramachandran and Sukumar Nandi. 2005. Detecting ARP spoofing: An active technique. In *International Conference on Information Systems Security*. Springer, 239–250.
[131] Symantec Security Response. 2010. The Trojan.Hydraq Incident. Symantec. Retrieved from https://www.symantec.com/connect/blogs/trojanhydraq-incident.
[132] Neil C. Rowe. 2004. A model of deception during cyber-attacks on information systems. In *Proceedings of the 1st IEEE Symposium on Multi-Agent Security and Survivability*. IEEE, 21–30.
[133] Fatima Salahdine and Naima Kaabouch. 2019. Social engineering attacks: A survey. *Fut. Internet* 11, 4 (2019), 89.
[134] Malek Ben Salem and Salvatore J. Stolfo. 2011. Decoy document deployment for effective masquerade attack detection. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 35–54.
[135] H. P. Sanghvi and M. S. Dahiya. 2013. Cyber reconnaissance: An alarm before cyber attack. *Int. J. Comput. Appl.* 63, 6 (2013).
[136] Asanka Sayakkara, Nhien-An Le-Khac, and Mark Scanlon. 2019. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digit. Invest.* 29 (2019), 43–54.
[137] James Scott and Drew Spaniel. 2016. Know Your Enemies 2.0. Retrieved from https://www.covenantsec.com/wp-content/uploads/2016/03/ICIT-Brief-Know-Your-Enemies-2.0.pdf.
[138] Accenture Security. 2019. The Cost of Cybercrime. Ponemon Institute LLC. Retrieved from https://www.accenture.com/_acnmedia/pdf-96/accenture-2019-cost-of-cybercrime-study-final.pdf.
[139] Siraj A. Shaikh, Howard Chivers, Philip Nobles, John A. Clark, and Hao Chen. 2008. Network reconnaissance. *Netw. Secur.* 2008, 11 (2008), 12–16.
[140] Zain Shamsi, Ankur Nandwani, Derek Leonard, and Dmitri Loguinov. 2014. Hershel: Single-packet OS fingerprinting. *ACM SIGMETRICS Perf. Eval. Rev.* 42, 1 (2014), 195–206.
[141] Ryan Sherstobitoff. 2018. Hidden Cobra Targets Turkish Financial Sector with New Bankshot Implant. McAfee Blogs. Retrieved October 27, 2019 from https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/hidden-cobra-targets-turkish-financial-sector-new-bankshot-implant/.
[142] Ryan Sherstobitoff and Jessica Saavedra-Morales. 2018. Gold Dragon Widens Olympics Malware Attacks, Gains Permanent Presence on Victims' Systems. Retrieved from https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/gold-dragon-widens-olympics-malware-attacks-gains-permanent-presence-on-victims-systems/.

[143] Siddhant Shrivastava. 2016. BlackEnergy—Malware for Cyber-Physical Attacks. iTrust.
[144] Sudeep Singh and Yin Hong Chang. 2016. Targeted Attacks against Banks in the Middle East. FireEye Inc. Retrieved from https://www.fireeye.com/blog/threat-research/2016/05/targeted_attacksaga.html.
[145] Raphael Spreitzer, Veelasha Moonsamy, Thomas Korak, and Stefan Mangard. 2017. Systematic classification of side-channel attacks: A case study for mobile devices. *IEEE Commun. Surv. Tutor.* 20, 1 (2017), 465–488.
[146] Sid Stamm, Zulfikar Ramzan, and Markus Jakobsson. 2007. Drive-by pharming. In *Proceedings of the 9th International Conference on Information and Communications Security (ICICS'07)*. Springer, 495–506.
[147] Didier Stevens. 2011. Malicious PDF documents explained. *IEEE Secur. Priv.* 9, 1 (2011), 80–82.
[148] Branka Stojanović, Katharina Hofer-Schmitz, and Ulrike Kleb. 2020. APT datasets and attack modeling for automated detection methods: A review. *Comput. Secur.* 92 (May 2020), 101734.
[149] Shridatt Sugrim, Sridhar Venkatesan, Jason A. Youzwak, Cho-Yu J. Chiang, Ritu Chadha, Massimiliano Albanese, and Hasan Cam. 2018. Measuring the effectiveness of network deception. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI'18)*. IEEE, 142–147.
[150] Jianhua Sun and Kun Sun. 2016. DESIR: Decoy-enhanced seamless IP randomization. In *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM'16)*. IEEE, 1–9.
[151] Symantec Security Response. 2016. Buckeye Cyberespionage Group Shifts Gaze from US to Hong Kong. Symantec. Retrieved from https://www.symantec.com/connect/blogs/buckeye-cyberespionage-group-shifts-gaze-us-hong-kong.
[152] Fahimeh Tabatabaei and Douglas Wells. 2016. OSINT in the context of cyber-security. In *Open Source Intelligence Investigation*. Springer, 213–231.
[153] Team CIRCL. 2013. Analysis of a PlugX Malware Variant Used for Targeted Attacks. Computer Incident Response Center, Luxembourg. Retrieved from http://circl.lu/assets/files/tr-12/tr-12-circl-plugx-analysis-v1.pdf.
[154] Mike Ter Louw, Jin Soon Lim, and Venkat N. Venkatakrishnan. 2008. Enhancing web browser security against malware extensions. *J. Comput. Virol.* 4, 3 (2008), 179–195.
[155] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security'16)*. 601–618.
[156] Yair Tsarfaty. 2018. Micropsia Malware. Radware Blog. Retrieved from https://blog.radware.com/security/2018/07/micropsia-malware/.
[157] Andrea Tundis, Wojciech Mazurczyk, and Max Mühlhäuser. 2018. A review of network vulnerabilities scanning tools: Types, capabilities and functioning. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ACM, 65.
[158] Martin Ussath, David Jaeger, Feng Cheng, and Christoph Meinel. 2016. Advanced persistent threats: Behind the scenes. In *Proceedings of the 50th Annual Conference on Information Science and Systems (CISS'16)*. IEEE, 181–186.
[159] Nart Villeneuve, James T. Bennett, Ned Moran, Thoufique Haq, Mike Scott, and Kenneth Geers. 2014. Exposing Attacks on Foreign Affairs Ministries. FireEye, Inc. Retrieved from https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-operation-ke3chang.pdf.
[160] Arun Vishwanath. 2014. Habitual Facebook use and its impact on getting deceived on social media. *J. Comput.-Mediat. Commun.* 20, 1 (2014), 83–98.
[161] Cliff Wang and Zhuo Lu. 2018. Cyber deception: Overview and the road ahead. *IEEE Secur. Priv.* 16, 2 (2018), 80–85.
[162] Yong Wang, Kevin Streff, and Sonell Raman. 2012. Smartphone security challenges. *Computer* 12 (2012), 52–58.
[163] Yien Wang and Jianhua Yang. 2017. Ethical hacking and network defense: Choose your best network vulnerability scanning tool. In *Proceedings of the 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA'17)*. IEEE, 110–113.
[164] Yi-Min Wang and Douglas Beck. 2017. Honey Monkey Network Exploration. US Patent 9,596,255.
[165] Chris Wren, Denis Reilly, and Tom Berry. 2010. Footprinting: A methodology for auditing esystem vulnerabilities. In *Proceedings of the Developments in E-systems Engineering*. IEEE, 263–267.
[166] Tarun Yadav and Arvind Mallari Rao. 2015. Technical aspects of cyber kill chain. In *International Symposium on Security in Computing and Communication*. Springer, 438–452.
[167] Mengjia Yan, Read Sprabery, Bhargava Gopireddy, Christopher Fletcher, Roy Campbell, and Josep Torrellas. 2019. Attack directories, not caches: Side channel attacks in a non-inclusive world. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'19)*. IEEE, 888–904.
[168] Chao Yang, Yimin Song, and Guofei Gu. 2012. Active user-side evil twin access point detection using statistical techniques. *IEEE Trans. Inf. Forens. Secur.* 7, 5 (2012), 1638–1651.
[169] Ezer Osei Yeboah-Boateng and Priscilla Mateko Amanor. 2014. Phishing, SMiShing & Vishing: An assessment of threats against mobile devices. *J. Emerg. Trends Comput. Inf. Sci.* 5, 4 (2014), 297–307.

[170] Hossein Rouhani Zeidanloo and Azizah Abdul Manaf. 2009. Botnet command and control mechanisms. In *Proceedings of the 2nd International Conference on Computer and Electrical Engineering*, Vol. 1. IEEE, 564–568.

[171] Xiaopeng Zhang. 2017. In-Depth Analysis of A New Variant of .NET Malware AgentTesla. Retrieved from https://www.fortinet.com/blog/threat-research/in-depth-analysis-of-net-malware-javaupdtr.html.

[172] Rui Zhao and Chuan Yue. 2013. All your browser-saved passwords could belong to us: A security analysis and a cloud-based new design. In *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*. 333–340.

Received 3 June 2020; revised 24 September 2021; accepted 6 April 2022