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.

Reconnaissance enables attackers to understand system configurations and to find alternative ways to exploit a system. 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.

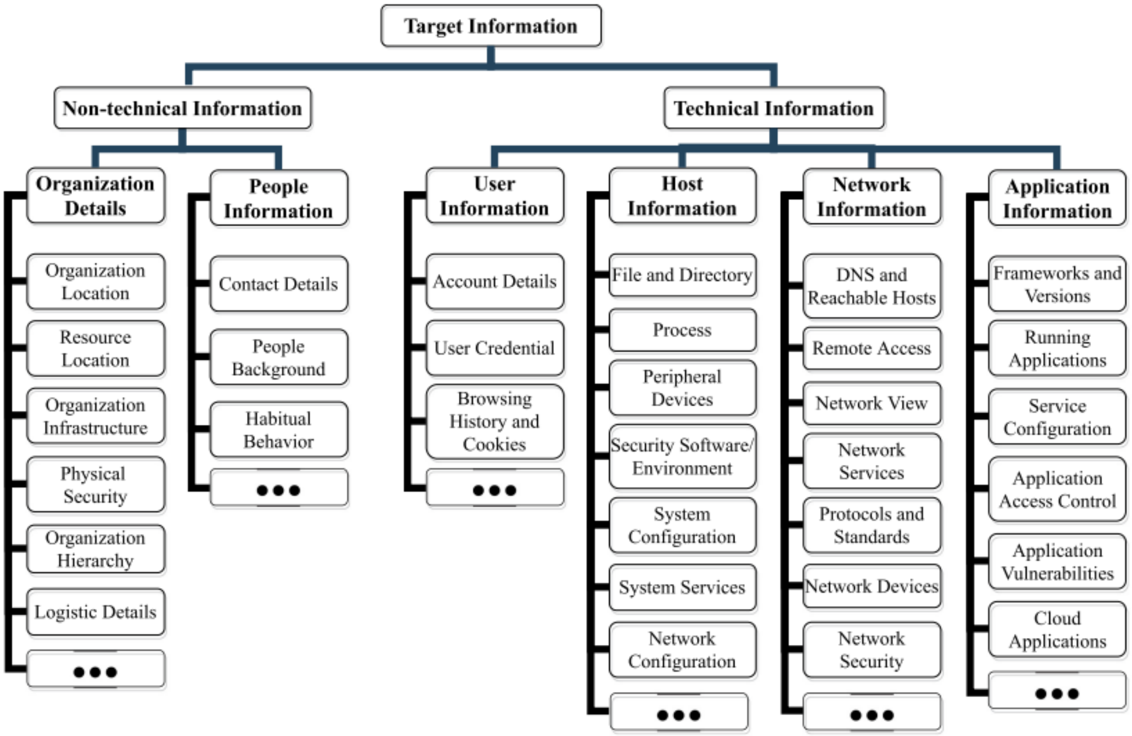


Fig. 1. Categories of target information for reconnaissance.

An adversary's **target information** can be divided into two main types: Non-technical 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 is often most useful for performing social engineering and initial access planning. Technical information is helpful for adversaries to find vulnerabilities to compromise specific systems.

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). **Organization information** includes the organization's background, resources, employee contacts and work details, physical access and security policies.

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. Adversaries can look for logistics information such as financial and business processes or intelligence, employee and management hierarchy, resource arrangement, and supply chain management data.

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. Adversaries can also join the organization and access confidential information as an insider.

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. Adversaries can collect contact details such as email addresses, phone numbers, and identity information. Information related to the technical or financial background of people is also useful. Adversaries can also attempt to learn their targets' habits.

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 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.

Adversaries look for **network-level information**, such as the network topology, network protocols, devices, and services, to understand the local network. Scanning and sniffing are highly effective approaches for obtaining target information at the network level.

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).

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 ICMP or communication protocols such as TCP or UDP. 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.

Network Devices: Adversaries can look for network device information such as hardware device manufacturer or vendor, operating systems and version, manufacturer settings, and networking configurations. Device information is useful for adversaries when employing exploits that target known vulnerabilities. Numerous tools are available for identifying device information, such as the manufacturing company.

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 breached. Adversaries can obtain network information using network or Internet foot printing, scanning or fingerprinting techniques and social engineering.

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, and application setup environment 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.

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.

System configuration: 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 and system manufacturer's value from the registry to identify the type of the machine 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) 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.

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, and the virtualization environment.

Files and Directories: Adversaries can look for directory contents and file lists. Particular directories containing configuration information or files with specific extensions 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.

Application-level security vulnerabilities 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 or Django) and environments (e.g., application run-time environments such as Java VM), which may have vulnerabilities. Misconfiguration is another possible weakness that creates loopholes and attract adversaries. 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) or by querying registry, and running processes.

Application or Package Configuration: Adversaries may also be interested in learning about the configuration of installed software and applications on a host. 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]. 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.

Database systems are prone to have misconfiguration and human errors that leave systems vulnerable to attacks. Adversaries can fingerprint versions of MySQL, PostgreSQL, Microsoft SQL Server, and Oracle Database by performing advanced queries. 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.

GUIs: Apart from this information, adversaries can also obtain data from the GUI windows of running applications. For example, they can collect window titles or text content and capture screenshots of them.

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.

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. Some APTs are also capable of querying information from account associated directories and enumerating local and domain users.

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 or utilizing spyware to collect user profile data or login information stored in a browser cache. Adversaries can also take advantage of web browser vulnerabilities to collect user-level information, e.g., by installing a malware extension.

Reconnaissance is present in different forms throughout the attack process and provides key information that is needed to execute subsequent phases. There are **two reconnaissance 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.

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, while remaining undetected.

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. The attack process starts with target selection and planning. The adversary begins collecting information about the target organization using 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. Once adversaries have access to the internal network, they seek more information about the network to engage in lateral movement and compromise other resources.

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. 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. 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.

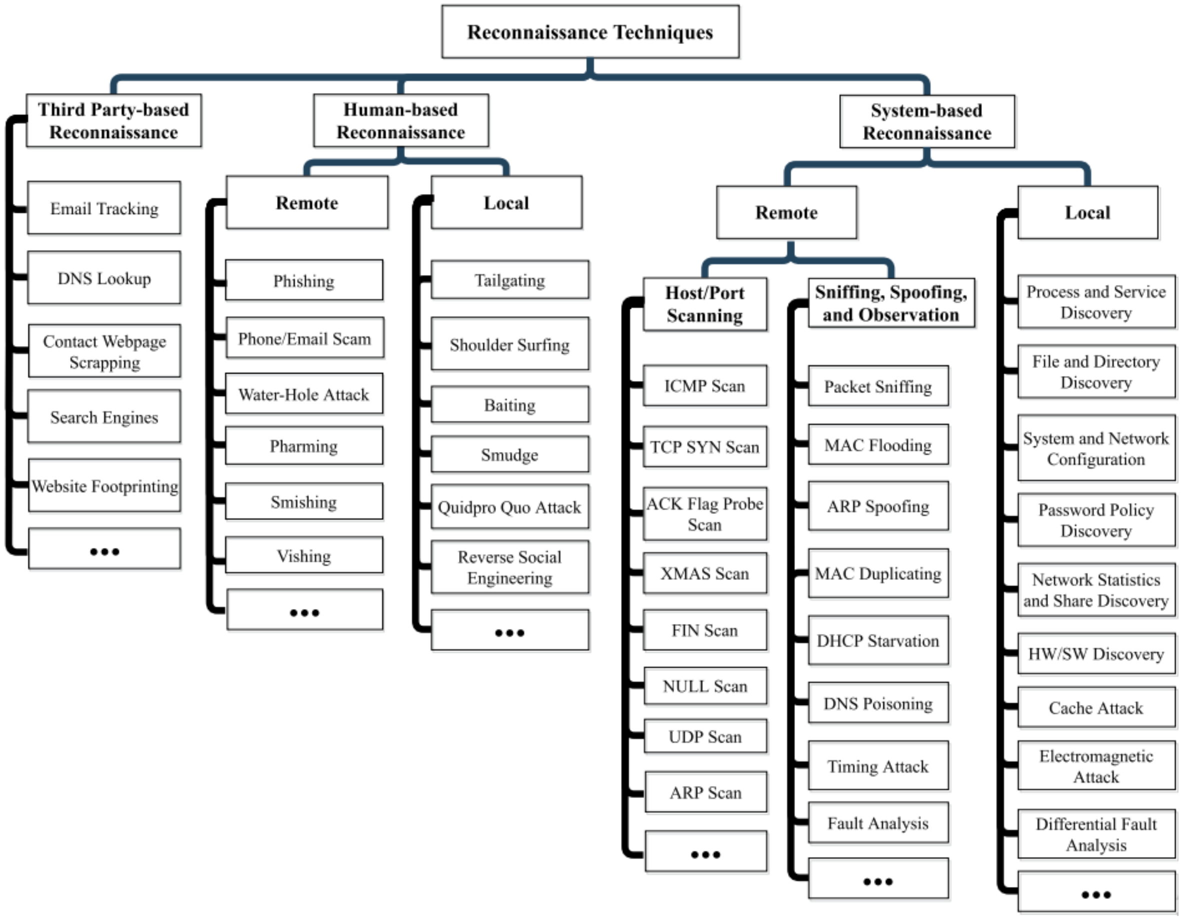
Reconnaissance techniques can be categorized based on the source of the information: third-party-based reconnaissance techniques, human-based reconnaissance techniques, and system-based reconnaissance techniques

Third-party source-based reconnaissance techniques: Extracting information from third parties. Third parties include websites, search engines, dark web, or personnel who are not involved with the target organization.

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.



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.

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. WHOIS information is usually stored in WHOIS databases and is maintained by regional Internet registries.

DNS interrogation tools are used to search for hosts in a network to obtain an internal 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. NSLookup 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 foot printing. Tools like WebExtractor can collect contact information such as phone numbers, email addresses, and fax numbers. Other tools, such as Website Watcher, are capable of monitoring web updates. Backdated site information can also be obtained from the Internet Archive.

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. Adversaries can follow the online activities of a person and learn about the person's habits, psychological state, and preferences 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). This enables adversaries to learn about their targets' email reading times and associated habits, which they can exploit in social engineering.

Search engines (such as Google, Yahoo, and Bing) can find background information (e.g., financial, technical, or business process reports) about an organization. Google hacking database and advanced search queries can help adversaries use advanced features of Google search to find more details.

Typically, fooling a human is significantly easier than fooling firewalls, honeypots, or intrusion detection/prevention systems. **Social engineering** has been recognized as one of the most common techniques employed by cyberattacks that result in high-profile data breaches. Social engineering is based on using deception to gain information through methods like baiting, pretexting, phishing, and spear-phishing.

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. **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).

Phishing has proven to be a very effective technique for stealing user credentials. The primary media of phishing are: the Internet, short messaging service (SMS), eFax, instant messaging, social networking, and telephone services.

A **watering hole** attack typically compromises a victim's machine by installing malicious code from a malicious website. 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.

Pretexting and vishing refer to impersonation through text messages or voice calls (vishing) and convincing targets to give access to particular resources. 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.

Pharming is often performed through DNS poisoning, which enables redirecting victim users to malicious sites even if they attempt to visit only legitimate sites. Therefore, regardless of the security measures taken by a user they may still fall victim to visiting malicious content.

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. 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.

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.

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.

Shoulder Surfing: An attacker can watch the target person entering a username, passwords, credit information, or other sensitive information by standing near them.

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. 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.

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.

Network scanning includes uniform scanning (probing random hosts within an 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. Botnet-based stealthy scanning is useful for discovering and compromising network infrastructure while minimizing detection by scanning from many hosts over multiple days.

Scanning techniques can also be categorized as horizontal scans, vertical scans, and coordinated/ distributed scans. A **vertical scan** involves an adversary targeting multiple ports on a single IP address. 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).

The most common low-level (i.e., network or transport layer) scanning techniques, emphasizing the network packet attributes, are:

TCP connect scan establishes a full three-way handshake with hosts within the target IP range. 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.

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. A SYN scan can be performed quickly within a given range of 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. 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).

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.

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. In some cases, it is possible to detect versions of services and operating systems as well.

ARP scan is a network discovery technique that works by broadcasting an ARP packet in the network and checking which hosts respond. 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 can also vary the attributes of network scans, including the speed, distribution, and destination of scanning. Depending on their motivations and on the defenses of the networks, adversaries may prefer a slow scan approach to avoid detection.

Attackers can also perform **application-level scanning techniques**, such as banner grabbing, operating system and application fingerprinting. The most common techniques are:

Banner grabbing is a vulnerability scanning techniques that uses application banner information, including name and version. 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.

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 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. Information can also be obtained by examining error message responses.

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 involves flooding a switch with abundant mapping requests so that the switch overflows at some point. 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 technique, 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.

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. 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.

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. 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. On Unix-based systems, the built-in command “ps” is available for checking running processes.

Service Discovery: Adversaries can collect information about running services on the Windows platform using system commands like “net start”, “tasklist” or “sc query”. 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” and “ifconfig” 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” or “nbtscan” to query NetBIOS name resolution information and to find vulnerabilities. ARP information can be obtained using the command “arp -a”.

File and Directory Discovery: Adversaries can list directory items on a Windows-based system by running “dir” or “tree” command. 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.

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; 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, 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. 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’ key-strokes and information, such as passwords, habits, or financial information. For example, terminal commands or application names typed in by a user can reveal further details of a system, used applications, and services.

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.

Many techniques have been proposed in the literature to use deception and information hiding to mitigate reconnaissance, including honey-pots, honey tokens, honey passwords, honey permissions and parameters. Techniques such as dynamic host address translation, route alteration, and IP randomization can lower the success of passive reconnaissance. Additional methods including database decoys, OS obfuscation, source code decoys, forging fake traffic, topology deception, hyperlinks decoys, simulation deception, and code embedding deception can mitigate both passive and active reconnaissance. Employee training, security awareness and best practices can mitigate social engineering tactics to some extent.