2

SOURCES OF INFORMATION

To have a successful vulnerability management program, you need information from several data sources. This chapter introduces you to each of these sources. In the next chapter, you'll see how they all come together to give you a useful vulnerability landscape for your organization.

Asset Information

Despite the importance of asset information, many organizations, large and small, don't have a full—or even fragmentary—understanding of what is on their networks. Perhaps you use a spreadsheet that you pass among network administrators and update intermittently. Or maybe you have a database of Windows desktops using a CMDB or endpoint management product. But to perform vulnerability management, you need a complete inventory of IP-connected devices and any additional data that you can glean about each

host. Non-networked devices, although important to an overall risk assessment, are outside the scope of an automated vulnerability management program.

Obtaining a list of hosts—and even a wealth of additional information—is straightforward. You can use a network-scanning tool, like Nmap, or a vulnerability scanner, like Nessus or Qualys (which you'll need anyway to collect vulnerability data), to do a network sweep and find live hosts. But these scans can be obtrusive and might cause application or even OS crashes. So, you need to carefully plan for an information-gathering scan.

New devices are added to networks all the time, and although most organizations have a change management policy in place, this is no guarantee that changes aren't made without following policy. To have updated and trustworthy asset information, you must perform discovery scans on a regular basis across the entire network.

Ideally, you would run these scans on a schedule. But the risks of regular scanning (which you'll learn about in the next section) might mean that your organization isn't comfortable having these scans done without a human monitoring them, ready to stop the scans if any issues crop up. If this is the case, you'll need to run fewer discovery scans, on a manual basis, and import the resulting data into your datastore.

CHANGE MANAGEMENT

Any organization with risk management in place will have a change management system to ensure that systems and networks remain in a stable state and to document any changes to that state. These systems can range from an email chain for change requests, approvals, and coordination to a full commercial change management system encompassing ticketing and configuration management.

Change control alone must not, and cannot, be the only control in place for IT changes. There are always ways around it. Administrators apply—or fail to apply—patches, add and change network routes to troubleshoot issues, buy and connect new network devices, or enable new services to fulfill a perceived business need without creating the necessary change control paper trail. Thus, you can't trust what a change management system says about the IT infrastructure's state.

Vulnerability Information

Once you have a complete account of all devices on the network, you'll configure a vulnerability scanner to do a deep scan of each device and discover any known host vulnerabilities. For example, a scanner might determine

that a Windows server is running a version of the IIS web server that is vulnerable to a directory traversal attack: the consequence is possible information disclosure.

When configuring and scheduling scans, carefully look at the available scanner options and tailor the settings to your environment and risk tolerance. The same goes for scheduling the times and scopes of scans. For example, you can scan some of your network sections, such as endpoint segments, every day. The reasons are that the risk of downtime is limited and the consequences aren't severe if a user's workstation is briefly offline. But scanning your critical systems, such as core production databases, might be too risky to do outside of scheduled maintenance windows. You need to understand the trade-off between getting fresh data and risking downtime.

By their very nature, network vulnerability scanners will only find vulnerabilities that are discoverable over a network connection. If a locally exploitable vulnerability is in a desktop application on a Windows endpoint, a network scan won't find it. For example, a network scanner won't find CVE-2018-0862—a vulnerability in Microsoft Equation Editor that an attacker can only exploit by opening a crafted Word or WordPad document. The reason is that Microsoft Office applications in general aren't detectable via a network scan.

To plug this hole, you could use an endpoint scanner (for example, the Qualys "scanless agent") or a *software configuration management (SCM)* tool or CMDB to gather a list of deployed software versions and determine known vulnerabilities by checking against a vulnerability database. Despite these limitations, having an accurate account of just network-discoverable vulnerabilities is an excellent start.

I'll cover vulnerability scanners in more detail in Chapter 3.

Exploit Data

Although a lot of information is available on a per-vulnerability basis, you can do more by combining data sources. The lowest-hanging fruit is exploit data. Information about publicly available exploits is widely accessible and often searchable. For example, the Exploit Database website (*https://www.exploit-db.com/*) has a searchable index of public exploits. Also, Metasploit, which I'll discuss in Chapter 14, has a large archive of usable exploits and a command line tool to easily deploy these exploits against target systems. Most exploits are associated with a particular vulnerability—a specific CVE ID. You can use the CVE ID to correlate exploit information with vulnerability information that you already possess.

Addressing an exploitable vulnerability is likely a higher priority to your organization than a vulnerability that isn't yet known to be exploitable. But not all exploits are equal. For instance, an exploit that enables arbitrary code execution is more severe than one that causes DoS or even one that permits reading arbitrary data. Knowing the consequences of an exploit is very useful for prioritizing exploits with more granularity.

CVE IDS

The CVE database is an attempt, led by the MITRE Corporation, to systematize and catalog all known information security vulnerabilities. Every newly discovered vulnerability is assigned a CVE ID in the form *CVE-yyyy-xxxx*, where *yyyy* is the current year and *xxxx* is a four (or more) digit number. The database is available online at *https://cve.mitre.org/*.

The CVE record includes a description of the vulnerability and links to data sources, such as official vulnerability announcements from the vendor, third-party notifications, and even exploit announcements. For an example of an exhaustively documented vulnerability, go to the CVE website and search for "CVE-2014-0160." This is the identifier for the *Heartbleed* vulnerability, a particularly nasty information leakage vulnerability that affected nearly every web server in existence. Its CVE page contains more than 100 references, from mailing list posts, to testing tools, to patch announcements from dozens of separate vendors.

Advanced Data Sources

The following list contains a few specialized and advanced data sources. Although largely outside the scope of this book, they're valuable references.

Threat intelligence feeds These feeds include information about the current threat landscape: threat actors and groups, the exploits currently being used in exploit kits, and the vulnerabilities with privately available exploits that aren't yet public knowledge. Use this information to determine which vulnerabilities are currently a higher risk to your organization. Because these threat feeds contain fresh data, you should use the feed data as soon as it comes in to get a timely assessment of your exposure to newly discovered threats. Numerous free and paid threat feeds are available, such as iSight Threat Intelligence, iDefense Threat Intelligence, and industry-specific threat feeds, like the one provided by FS-ISAC.

Proprietary exploits Although it's expensive, adding proprietary exploit data (sometimes known as exploit kits) to the publicly available information from Exploit Database and Metasploit broadens the range of exploits that you can match against your vulnerability data. Sources range from commercial threat intelligence sources that commission their own exploit research to decidedly gray- or black-market options, such as independent researchers selling newly discovered vulnerability and exploit information to the highest bidder. Whatever the source, proprietary exploit information will help you better prioritize your own vulnerability data based on exploits you would otherwise be unaware of.

Network configurations Use network configurations from routing devices like routers, firewalls, and managed switches to create a model of your network. By combining this information (which subnets route to which, which ports are accessible from where) with vulnerability and exploit data, you get a deep understanding of your network attack surface. For example, if a Tomcat exploit exists for an internal web application server but your router configuration indicates that this server is accessible only to a limited list of source IP addresses, it might be of less concern to you than if it were accessible to the internet at large. You might already have network configuration information, especially if you have a centralized configuration repository, such as SolarWinds. On the downside, it takes significant work to integrate this data with your existing vulnerability data. Some commercial vulnerability management products contain built-in functionality to ingest network configurations.

Summary

Each of the data sources discussed in this chapter contributes an important set of data to your vulnerability management system. Table 2-1 breaks down the data you can glean from each of these sources.

Table 2-1: Data Sources for Vulnerability Management

Data source	Important data
Host/port scanner (Nmap)	IP address MAC address Hostname Open ports (TCP and UDP) Service and OS fingerprinting
Network vulnerability scanner	(Same as above) Additional service fingerprinting and version detection Network vulnerabilities Local vulnerabilities (authenticated scans only)
Host-based vulnerability scanner	Local vulnerabilities
CMDB/SCM	OS details Deployed software details Configuration details Owner of the device Criticality of the device and application
Exploit databases	Exploit information Vulnerability mapping to exploits
Threat intelligence	Attacker and targeted industry intelligence Newly discovered, escalating, or widespread exploits
Exploit kits	Proprietary exploit information
Network configurations	Network topology and potential attack paths

In the next chapter, you'll take an in-depth look at vulnerability scanning.