Endpoint security

Traditional endpoint security encompasses numerous security tools, such as anti-malware software, anti-spyware software, personal firewalls, host-based intrusion prevention systems (HIPSs), and mobile device management (MDM) software. Endpoint security also requires implementation of effective endpoint security best practices, including patch management and configuration management.

Endpoint security basics

Endpoint security begins with a standard ("golden") image that ensures consistent configuration of devices across the organization, including:

- Disabling or removing operating system features and services that are not needed ("hardening")
- Installing current security updates
- Installing core applications

In practice, an organization will deploy numerous golden images, to, for example, support different device types, workgroups or departments, and user types (such as standard users and power users).

Most organizations deploy several security products to protect their endpoints, including personal firewalls, host-based intrusion prevention systems (HIPSs), mobile device management (MDM), mobile application management (MAM), DLP, and antivirus software. Nevertheless, cyber breaches continue to increase in frequency, variety, and sophistication. Additionally, the numbers and types of endpoints – including mobile and IoT devices – has grown exponentially and increased the attack surface. New variants of the Gafgyt, Mirai, and Muhstik botnets, among others, specifically target IoT devices, and new search engines, such as Shodan (Shodan.io), can automate the search for vulnerable internet-connected endpoints. Faced with the rapidly changing threat landscape, traditional endpoint security solutions and antivirus can no longer prevent security breaches on the endpoint.

Endpoint security is an essential element of cybersecurity because the network firewall cannot completely protect hosts from zero-day exploits. Zero-day exploits target unknown vulnerabilities in operating system and application software on host machines. Network firewalls may not be able to block an attacker's delivery of a zero-day exploit until a new signature identifying the zero-day attack has been developed and delivered to the firewall.

Note

With the proliferation of advanced malware such as remote access Trojans (RATs), anti-AV, and rootkits/bootkits, security vendors have largely rebranded their antivirus solutions as "anti-malware" and expanded their malware protections to encompass the broader malware classifications.

Network firewalls also may be restricted from decrypting all traffic because of regulations and laws. This restriction provides a window of opportunity for attackers to bypass a firewall's protection and exploit a host machine, necessitating endpoint security protection. Endpoint security protection is provided by an application that runs on the host machine. Effective endpoint security must be able to stop malware, exploits, and ransomware before they can compromise the host; provide protection while endpoints are online and offline; and detect threats and automate containment to minimize impact.

Malware protection

Malware protection – more specifically, antivirus software – has been one of the first and most basic tenets of information security since the early 1980s. Unfortunately, all of this hard-earned experience doesn't necessarily mean that the war is being won. For example, Trustwave's 2019 *Global Security Report* found that infection to detection of malware "in the wild" takes an average of 55 days.[1] Interestingly, web-based zero-day attacks, on average, remain "in the wild" up to four times longer than email-based threats because of factors that include user awareness of email-borne threats, availability and use of email security solutions (such as anti-spam and antivirus), and preferred use of the web as a threat vector by malware developers.

This poor "catch rate" is because of several factors. Some malware can mutate or can be updated to avoid detection by traditional anti-malware signatures. Also, advanced malware is increasingly specialized to the point where an attacker can develop customized malware that is targeted against a specific individual or organization.

Traditional anti-malware software uses various approaches to detect and respond to malware threats, including signature-based, container-based, application whitelisting, and anomaly-based techniques.

Signature-based anti-malware software

Signature-based antivirus (or anti-malware) software is the oldest and most commonly used approach for detecting and identifying malware on endpoints. This approach requires security vendors to continuously collect malware samples, create matching signature files for those samples, and distribute those signature files as updates for their endpoint security products to all of their customers.

Deployment of signature-based antivirus software requires installing an engine that typically has kernel-level access to an endpoint's system resources. Signature-based antivirus software scans an endpoint's hard drive and memory, based on a predefined schedule and in real time when a file is accessed. If a known malware signature is detected, the software performs a predefined action, such as:

Quarantine. Isolates the infected file so that it cannot infect the endpoint or other files

Delete. Removes the infected file

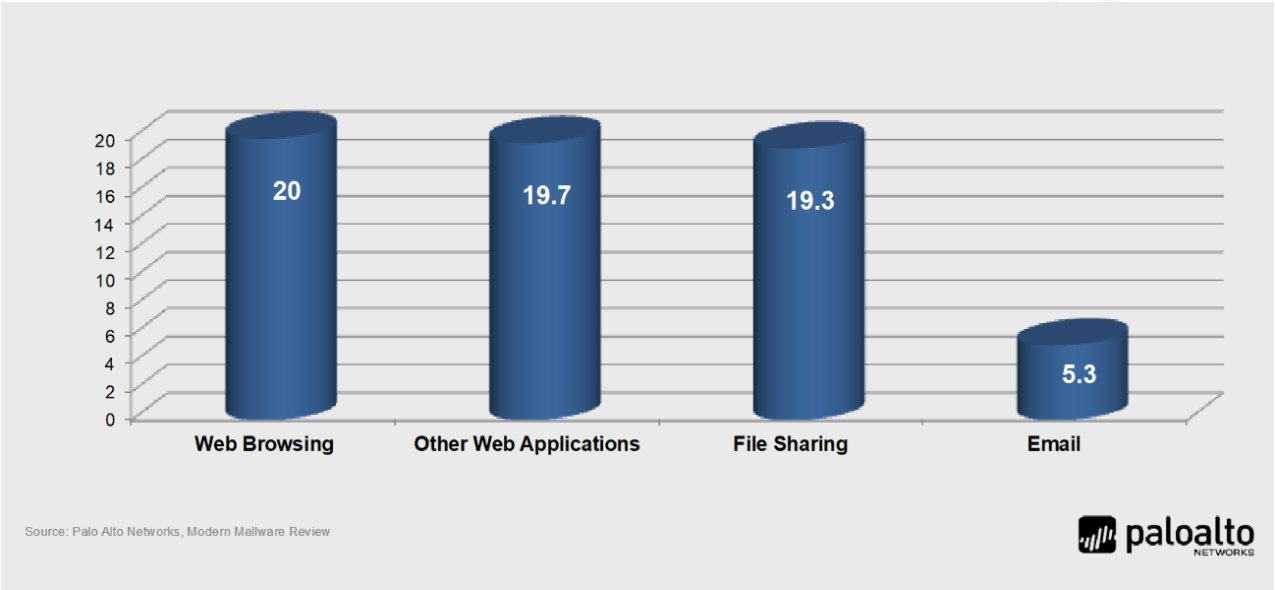
Alert. Notifies the user (and/or system administrator) that malware has been detected

[1] "2019 Trustwave Global Security Report." Trustwave. 2019. https://www.trustwave.com/en-us/resources/library/documents/2019-trustwave-global-security-report/.

Updated signatures must be regularly and frequently downloaded from the security vendor and installed on the organization's endpoints. Downloading and processing signature files in this manner can cause noticeable performance degradations on the networks and endpoints on which they are running.

Although the signature-based approach is very popular, its effectiveness is limited. By design, it is a reactive countermeasure because a signature file for new malware can't be created and delivered until the malware is already "in the wild," during which time networks and endpoints are blind to the threat – the notorious zero-day threat (or attack). The "zero-day" label is misleading, however, because the number of days from release to detection averages 5 to 20 days (see Figure 2-3).

Figure 2-3 *Average time to detection by application vector*



A sample of new or unknown suspicious traffic must first be captured and identified before a detection signature can be created by security vendors. The new signature must then be downloaded and installed on an organization's endpoints to provide protection.

This process means that some users and networks will be successfully breached by new malware until a new detection signature is created, downloaded, and installed. This reactive model creates a window of opportunity for attackers, leaving endpoints vulnerable – sometimes for weeks or even months – until new malware is suspected, collected, analyzed, and identified. During this time, attackers can infect networks and endpoints.

Another challenge for the signature-based approach is that millions of new malware variations are created each year (on average about 20,000 new forms daily), for which unique signatures must be written, tested, and deployed – after the new malware variation is discovered and sampled. Despite the fact that 70 percent of these millions of malware variations are based on a relatively limited number of malware "families" – numbering just seven in 2005 and increasing to only 20 over the past decade[2] – this reactive approach is not effective for protecting endpoints against modern malware threats.

[2] Ibid.

Also, advanced malware uses techniques such as metamorphism and polymorphism to take advantage of the inherent weaknesses of signature-based detection to avoid being discovered in the wild and to circumvent signatures that have already been created. These techniques are so commonly used that "70 to 90 percent of malware samples [collected] today are unique to a single organization."[3]

Container-based endpoint protection

Container-based endpoint protection wraps a protective virtual barrier around vulnerable processes while they're running. If a process is malicious, the container detects it and shuts it down, preventing it from damaging other legitimate processes or files on the endpoint.

However, the container-based approach typically requires a significant amount of computing resource overhead, and attacks have been demonstrated that circumvent or disable container-based protection. This approach also requires knowledge of the applications that need to be protected and how they interact with other software components. So, a containerization tool will be developed to support certain common applications but will not be capable of protecting most proprietary or industry-specific software. Even web browser plugins and the like can have problems operating correctly within a container-based environment.

Application whitelisting

Application whitelisting is another endpoint protection technique that is commonly used to prevent end users from running unauthorized applications – including malware – on their endpoints.

Application whitelisting requires a positive control model in which no applications are permitted to run on the endpoint unless they're explicitly permitted by the whitelist policy. In practice, application whitelisting requires a large administrative effort to establish and maintain a list of approved applications. This approach is based on the premise that if you create a list of applications that are specifically allowed and then prevent any other file from executing, you can protect the endpoint. Although this basic functionality can be useful to reduce the attack surface, it is not a comprehensive approach to endpoint security.

Modern trends such as cloud and mobile computing, consumerization, and bring your own device (BYOD) and bring your own access (BYOA) make application whitelisting extremely difficult to enforce in the enterprise. Also, after an application is whitelisted, it is permitted to run – even if the application has a vulnerability that can be exploited. An attacker can then simply exploit a whitelisted application and have complete control of the target endpoint regardless of the whitelisting. After the application has been successfully exploited, the attacker can run malicious code while keeping all of the activity in memory. Since no new files are created and no new executables attempt to run, whitelisting software is rendered ineffective against this type of attack.

[3] Ibid.

Anomaly detection

Endpoint security approaches that use mathematical algorithms to detect unusual activity on an endpoint are known as heuristics-based, behavior-based, or anomaly-detection solutions. This approach relies on first establishing an accurate baseline of what is considered "normal" activity. This approach has been around for many years and requires a very large dataset to reduce the number of false positives.

Key Terms

In anti-malware, a *false positive* incorrectly identifies a legitimate file or application as malware. A *false negative* incorrectly identifies malware as a legitimate file or application. In intrusion detection, a false positive incorrectly identifies legitimate traffic as a threat, and a false negative incorrectly identifies a threat as legitimate traffic.

Anti-spyware software

Anti-spyware software is very similar to traditional antivirus software because it uses signatures to look for other forms of malware beyond viruses, such as adware, malicious web application components, and other malicious tools, which share user behaviors without their permission.

Personal firewalls

Network firewalls protect an enterprise network against threats from an external network, such as the internet. However, most traditional port-based network firewalls do little to protect endpoints inside the enterprise network from threats that originate from within the network, such as another device that has been compromised by malware and is propagating throughout the network.

Personal (or host-based) firewalls are commonly installed and configured on laptop and desktop PCs. Personal firewalls typically operate as Layer 7 (Application layer) firewalls that allow or block traffic based on an individual (or group) security policy. Personal firewalls are particularly helpful on laptops used by remote or traveling users who connect their laptop computers directly to the internet (for example, over a public Wi-Fi connection). Also, a personal firewall can control outbound traffic from the endpoint to help prevent the spread of malware from that endpoint. However, note that disabling or otherwise bypassing a personal firewall is a common and basic objective in most advanced malware today.

Windows Firewall is an example of a personal firewall that is installed as part of the Windows desktop or mobile operating system. A personal firewall protects only the endpoint device that it is installed on, but it provides an extra layer of protection inside the network.

Host-based intrusion prevention systems

HIPS is another approach to endpoint protection that relies on an agent installed on the endpoint to detect malware. A HIPS can be either signature-based or anomaly-based, and it's therefore susceptible to the same issues as other signature and anomaly-based endpoint protection approaches.

Also, HIPS software often causes significant performance degradation on endpoints. A recent Palo Alto Networks survey found that 25 percent of respondents indicated HIPS solutions "caused significant end user performance impact."

Mobile device management

Mobile device management (MDM) software provides endpoint security for mobile devices such as smartphones and tablets. Centralized management capabilities for mobile devices provided by MDM include:

Data loss prevention (DLP). Restrict what type of data can be stored on or transmitted from the device.

Policy enforcement. Require passcodes, enable encryption, lock down security settings, and prevent *jailbreaking* or *rooting*, for example.

Malware protection. Detect and prevent mobile malware.

Software distribution. Remotely install software, including patches and updates over a cellular or Wi-Fi network.

Remote erase/wipe. Securely and remotely delete the complete contents of a lost or stolen device.

Geofencing and location services. Restrict specific functionality in the device based on its physical location.

Key Terms

Jailbreaking refers to hacking an Apple iOS device to gain root-level access to the device. Jailbreaking is sometimes done by end users to allow them to download and install mobile apps without paying for them, from sources other than the App Store that are not sanctioned and/or controlled by Apple. Jailbreaking bypasses the security features of the device by replacing the firmware's operating system with a similar, albeit counterfeit version, which makes it vulnerable to malware and exploits. Jailbreaking is known as *rooting* on Google Android devices.