Traps: Advanced Endpoint Protection

TRAPS:

- **Prevents all vulnerability exploits**
- **Prevents all malware-driven attacks**
- **Provides Immediate forensics of prevented attacks**
- **Is scalable, lightweight and user friendly**
- **Integrates with the network and cloud security**

Palo Alto Networks® Traps provides Advanced Endpoint Protection that prevents sophisticated vulnerability exploits and malware-driven attacks. Traps accomplishes this through a highly scalable, lightweight agent that uses an innovative new approach for defeating attacks without requiring any prior knowledge of the threat itself. By doing so, Traps provides organizations with a powerful tool for protecting endpoints from virtually every targeted attack.

Palo Alto Networks Traps takes a unique approach to endpoint security, designed to provide complete security protection for the endpoint, including the prevention of both conventional attacks as well as advanced and targeted attacks that traditional solutions cannot prevent.

Instead of looking to identify the millions of individual attacks themselves, or detect malicious behavior that may be undetectable, Traps focuses on the core techniques that every attacker must link together in order to execute their attack. By setting up a series of exploit 'traps' into the process to mitigate these techniques, Traps can thwart the attack immediately before any malicious activity can successfully run.

This unique approach allows Traps to be agnostic to application, protecting all applications, including those developed by 3rd parties.

Exploit prevention

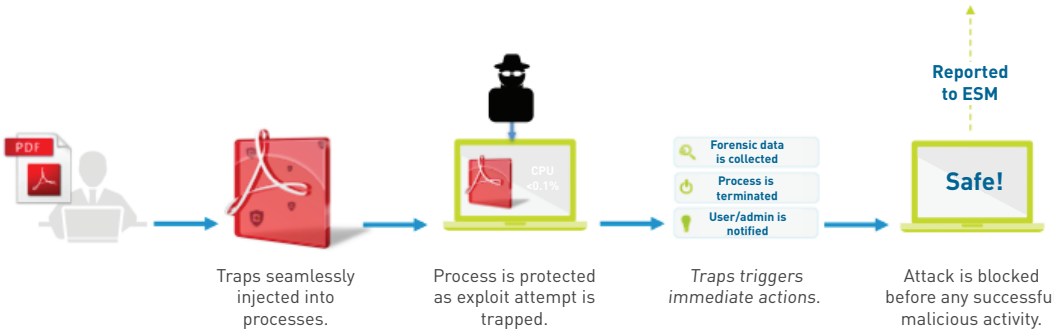
The actual process of exploiting a vulnerability on an endpoint requires execution of multiple advanced techniques operating in sequence. For example, in a typical attack the attacker will attempt to gain control of a system by first attempting to corrupt or bypass memory allocation or handlers. By using memory-corruption techniques such as buffer overflow or heap corruption the hacker can utilize weaknesses or vulnerabilities within the target software to execute their specific code. Once an attacker is able to execute custom code, he can download malware or completely control the system to his full advantage.

Regardless of the attack or its complexity—in order for the attack to be successful the attacker must execute a series exploit techniques in sequence. Some attacks may involve more steps, some may involve less, in all cases at least two or three techniques must be used in order to exploit the targeted endpoint.

How Exploit Prevention works

Traps employs a series of exploit prevention modules aimed at mitigating and blocking the different exploit techniques available to attackers. These modules operate like "traps", injected into the user processes and designed to trigger and block the attacker's exploit technique as soon as it's attempted. Whenever an application is opened Traps seamlessly injects prevention modules into the process as transparent, static "traps". Once a module is injected into the process, that process is then protected from any exploit. If an exploit attempt is made using one of the few available techniques, Traps will immediately block that

paloalto networks



How it works: Exploit prevention.

technique, terminate the process, and notify both the user and the admin that an attack was prevented. In addition, Traps will collect detailed forensics and report that information to the Endpoint Security Manager (ESM). Due to the chain-like nature of an exploit, preventing just one technique in the chain is all that is needed in order to block the entire attack.

If no attempt is made it's business as usual for that user and process. Given the minimal resource utilization of Traps, there will be no user experience implications of the preventative measures that were deployed behind the scenes.

By focusing on the exploit techniques and not the attack itself, Traps can prevent the attack without prior knowledge of the vulnerability, regardless of patches in place, and without signatures or software updates. It's important to note that Traps isn't scanning or monitoring for malicious activity, so there's a massive scalability benefit to this approach as very little CPU and memory are used.

Traps exploitation prevention is designed to prevent attacks on program vulnerabilities based on memory corruption or logic flaws. Examples of attacks that Traps can prevent, include:

- Memory corruption
- Java code from running in browsers, under certain conditions
- Executables from spawning child processes, under certain conditions
- Dynamic-link library (DLL) hijacking (replacing a legitimate DLL with a malicious one of the same name)
- Hijacking program control flow
- Inserting malicious code as an exception handler

Malware Prevention

Malicious executable files, known as malware, are often disguised as or embedded in non-malicious files. They can harm computers by attempting to gain control, gather sensitive information, or disrupt the normal operations of the system.

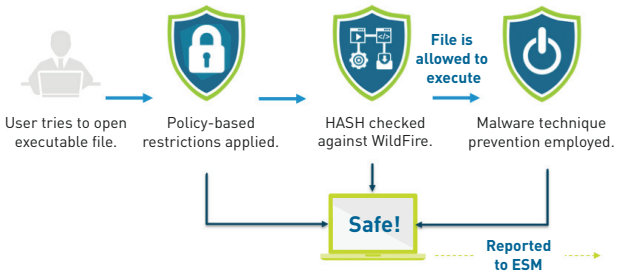
While advanced attackers are increasingly exploiting software vulnerabilities, attacks are also advancing with unknown or manipulated Malware (Executable files) and because these types of attacks generally don't have known signatures, known strings or previously known behavior, traditional endpoint security approaches are unable to prevent them.

In order to effectively prevent the execution of malware on the endpoint, Traps employs the following three components of malware prevention:

1. **Policy-Based Restrictions:** Policy restrictions provide organizations with the ability to set up policies restricting specific execution scenarios, and not the whitelisting or blacklisting of specific files. The attack surface can be greatly reduced by simply controlling the source of file installation. When a user attempts to open the executable, Traps will evaluate the execution restriction rules that may apply. Examples of common policy based restrictions;
   - Running executables from certain folders
   - Running executables from external media
   - Processes spawning child processes
   - Java processes run from browsers
   - Running unsigned processes
   - Thread Injection
2. **Wildfire™ Inspection:** For execution of files that are not limited to the policy restrictions set in place, Traps Endpoint Security Manager will query the WildFire threat cloud with a hash, to determine if the file is malicious, benign, or unknown within the global threat community. If WildFire confirms that a file is known malware, Traps will prevent the file from executing and will notify the ESM.
3. **Malware Techniques Mitigation:** Similar to Exploit techniques,

attackers utilize common, and identifiable techniques when trying to deploy their malware. In the event that the file execution is not restricted by policy or has not been matched by hash to a known attack in the Wildfire threat cloud, Traps will implement technique-based mitigations that limit or block; child processes, Java processes initiated in web browsers, remote thread and process creation, and unsigned processes execution—in order to prevent the attack entirely from executing.

Forensics



How it works: Malware prevention.

Whenever Traps prevents an attack, real-time forensic details about the event will be collected about; the file, what occurred, the memory state when it was prevented, etc. and report the logged information to the Endpoint Security Manager (ESM). Despite the fact that the attack was prevented, there is still a great amount of intelligence that can be gathered. By capturing all the forensics of the attempted attack, organizations can apply proactive defenses to other endpoints that may not be protected.

Traps Architecture

Traps provides a 3-tier management structure consisting of the Endpoint Security Manager, Endpoint Connection Server, and endpoint agents. This model allows for massive horizontal scalability while still maintaining a centralized configuration and database for policies, forensics, etc.

Endpoint Security Manager

The Endpoint Security Manager provides an administrative dashboard for managing security events, endpoint health, and policy rules. The ESM also handles the communication to WildFire when hashes are sent for inspection. The ESMs all-in-one management center covers:

- Configuration management
- Logging and DB query
- Admin dashboard and security overview
- Forensics captures
- Integration configuration

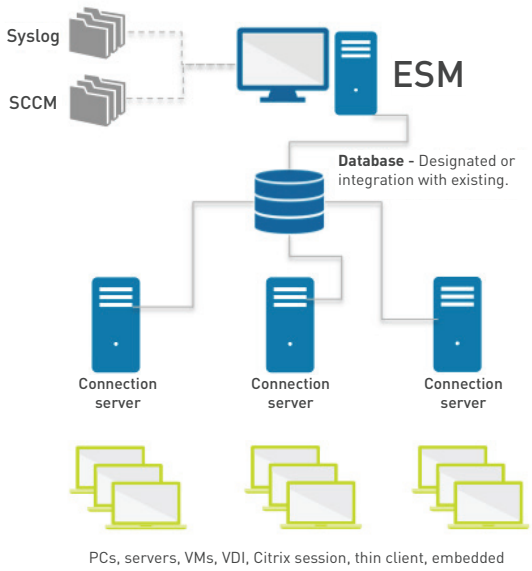
The Endpoint Security Manager includes a centralized database that stores administrative information, security policy rules, endpoint history, and additional information about security events. The database is managed over the MS-SQL platform.

The Endpoint Security Manager can write logs to an external logging platform, such as security information and event management (SIEM), Service Organization Controls (SOCs), or syslog, in addition to storing its logs internally. Specifying an external logging platform allows an aggregated view of logs from all Endpoint Servers.

Endpoint Server

The Endpoint Server regularly distributes the security policy to all agents and manages all the information related to security events.

- **Traps Status** – Notifications and health pages in the Endpoint Security Manager display the status for each endpoint.
- **Notifications** – Traps agent sends notification messages about changes in the agent, such as the start or stop of a service, to the Endpoint Server.
- **Prevention reports** – Traps reports all of the information pertaining to an event, to the Endpoint Server in real-time.



Coverage & Platform Support

Traps protects unpatched systems, requires no hardware and is supported across any platform that runs Microsoft Window; desktops, servers, industrial control systems, terminals, VDI, VMs and embedded systems etc.

Traps currently supports the following Windows-based operating systems:

WORKSTATIONS

- Windows XP SP3
- Windows 7
- Windows 8.1
- Windows Vista SP1

SERVERS

- Windows Server 2003
- Windows Server 2008 (+R2)
- Windows Server 2012 (+R2)

Specifications

With the unique approach taken, Traps operates in a somewhat static capacity and doesn't scan for malicious activity our resource utilization is very low:

TRAPS AGENT:

- CPU – Average utilization of 0.1%
- Memory Consumption – 25 MB
- Disk Space – 15 MB

paloalto networks
the enterprise security company™

4401 Great America Parkway
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com

Copyright ©2014, Palo Alto Networks, Inc. All rights reserved. Palo Alto Networks, the Palo Alto Networks Logo, PAN-OS, App-ID and Panorama are trademarks of Palo Alto Networks, Inc. All specifications are subject to change without notice. Palo Alto Networks assumes no responsibility for any inaccuracies in this document or for any obligation to update information in this document. Palo Alto Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. **PAN_DS_TRAPS_100814**