



Wazuh: Threat detection and active protection

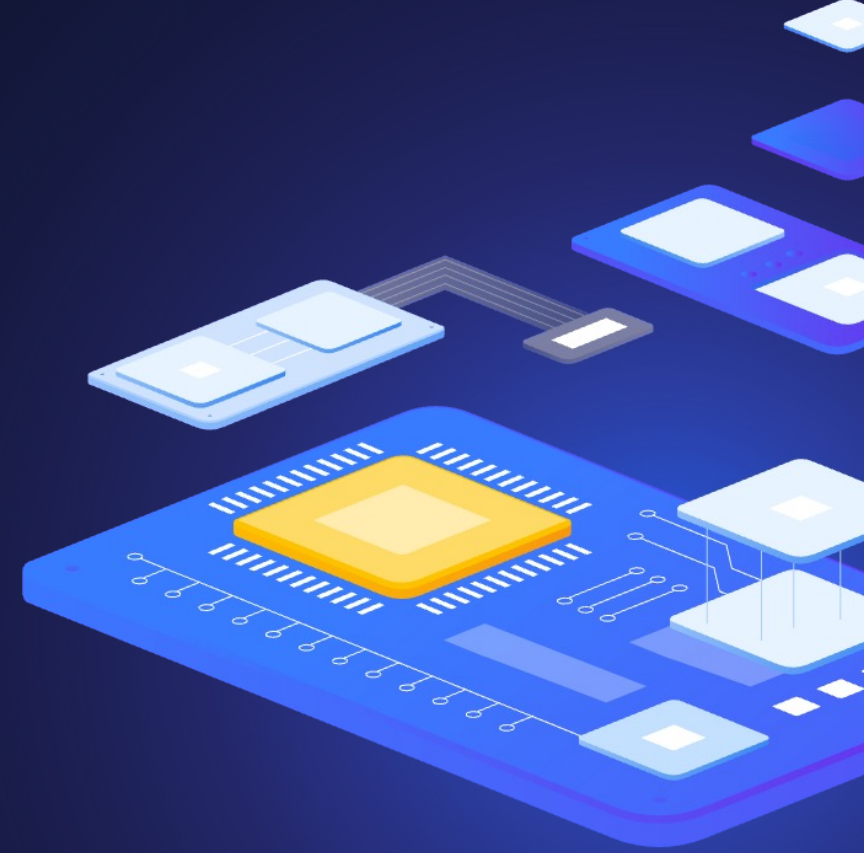
all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

Agenda

- 1 Intro
- 2 File Integrity Monitoring (FIM)
- 3 Malware detection with VirusTotal
- 4 Security Configuration Assessment (SCA)
- 5 Demo



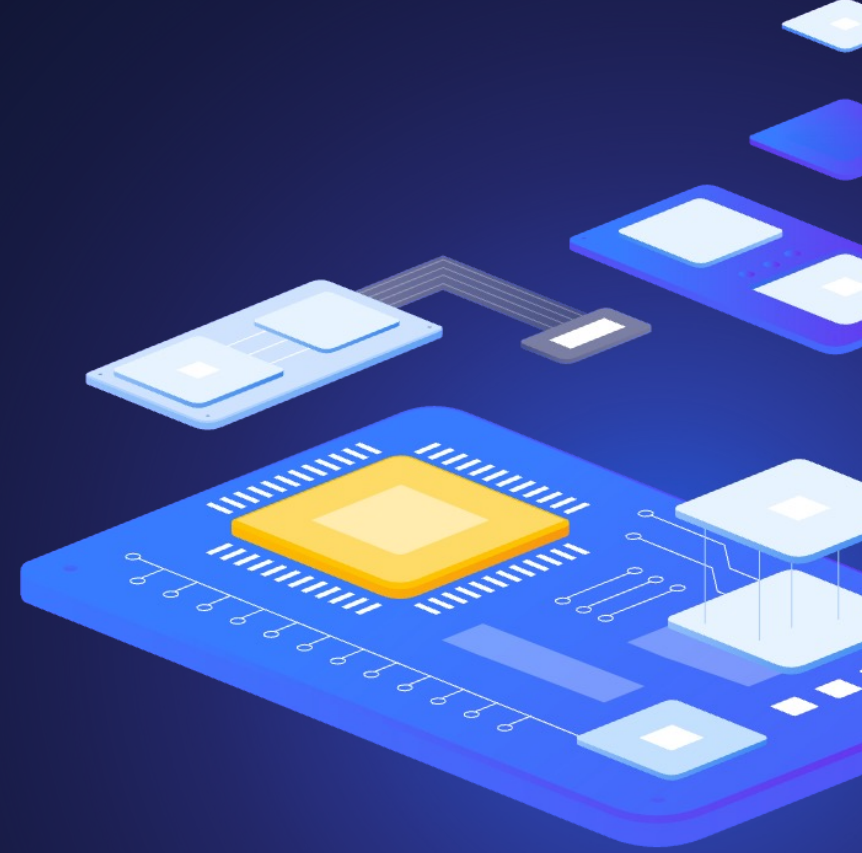
1

Intro



2

File Integrity Monitoring (FIM)



Wazuh: Threat detection and active protection

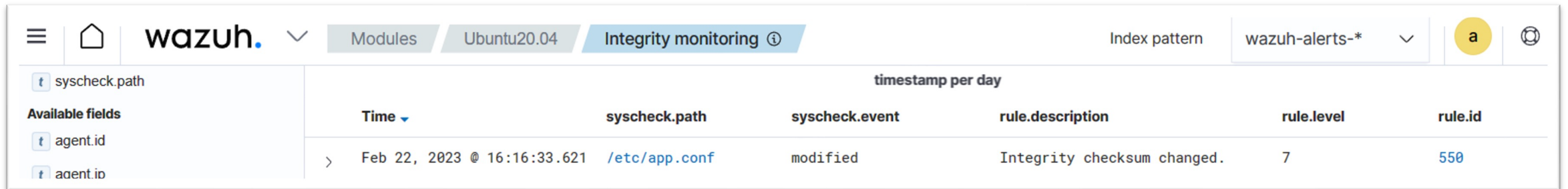
File integrity monitoring (FIM)

- ▶ Watches selected files or Windows registry and triggers alerts when these files are modified, including changes, additions and deletions
- ▶ Stores the checksums and other attributes of files
- ▶ Regularly compares received information against the historical for those files
- ▶ Supports near real-time file integrity monitoring
- ▶ Provides information on who made the changes to the monitored files and the name of the program or process used to make the changes



Wazuh: Threat detection and active protection

File integrity monitoring (FIM)



The screenshot shows the Wazuh web interface for File Integrity Monitoring (FIM). The breadcrumb navigation includes 'Modules', 'Ubuntu20.04', and 'Integrity monitoring'. The index pattern is set to 'wazuh-alerts-*'. The table displays a single alert entry with the following details:

syscheck.path	timestamp per day	syscheck.event	rule.description	rule.level	rule.id
/etc/app.conf	Feb 22, 2023 @ 16:16:33.621	modified	Integrity checksum changed.	7	550

Wazuh: Threat detection and active protection

File integrity monitoring (FIM)



The screenshot shows the Wazuh web interface for File Integrity Monitoring (FIM) on an Ubuntu20.04 system. The interface is divided into a left sidebar with a list of system check attributes and a main content area displaying a table of these attributes and their values. Several attributes are highlighted with blue boxes.

Attribute	Value
syscheck.md5_before	
syscheck.mode	
syscheck.mtime_after	
syscheck.mtime_before	
syscheck.perm_after	
syscheck.perm_before	
syscheck.sha1_after	
syscheck.sha1_before	
syscheck.sha256_after	
syscheck.sha256_before	
syscheck.size_after	
syscheck.size_before	
syscheck.uid_after	
syscheck.uname_after	
syscheck.win_perm_after	
timestamp	
syscheck.audit.group.id	0
syscheck.audit.group.name	root
syscheck.audit.login_user.id	1000
syscheck.audit.login_user.name	ubuntu
syscheck.audit.process.cwd	/
syscheck.audit.process.id	139877
syscheck.audit.process.name	/usr/bin/nano
syscheck.audit.process.parent_cwd	/
syscheck.audit.process.parent_name	/usr/bin/bash
syscheck.audit.process.ppid	105085
syscheck.audit.user.id	0
syscheck.audit.user.name	root
syscheck.changed_attributes	size, mtime, md5, sha1, sha256
syscheck.diff	0a1 > updated image to V2
syscheck.event	modified
syscheck.gid_after	0
syscheck.gname_after	root

3

Malware detection with VirusTotal



Malware detection with VirusTotal

- ▶ [VirusTotal](#) is an online service that analyzes files and URLs to detect viruses, worms, trojans, and other malicious content using antivirus engines and website scanners
- ▶ By sending the hash to the VirusTotal engine, you can know if VirusTotal has already scanned that specific file, and you can analyze its report
- ▶ VirusTotal also provides an API that allows access to the information generated by VirusTotal without needing to utilize the HTML website interface
- ▶ The VirusTotal public API is limited to 500 requests per day at a rate of 4 requests per minute
- ▶ [More informations about VirusTotal API](#)



Wazuh: Threat detection and active protection

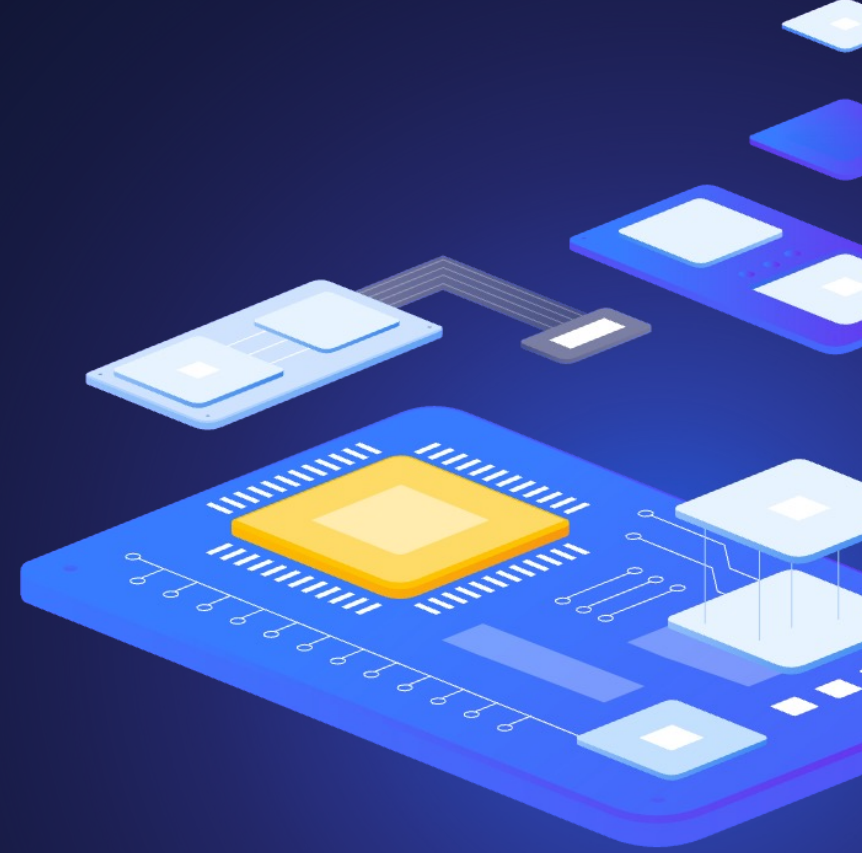
Malware detection with VirusTotal

- ▶ Wazuh FIM looks for any file addition, change, or deletion on the monitored folders
- ▶ Integration makes an HTTP POST request to the VirusTotal database using the VirusTotal API.
- ▶ This call sends the extracted file hash to compare it with the information in the VirusTotal database
- ▶ Integration receives a JSON response
- ▶ Wazuh logs the response
- ▶ [Wazuh integration with external APIs](#)



4

Security Configuration Assessment (SCA)



Security Configuration Assessment (SCA)

- ▶ Helps maintain a standard configuration through the monitored endpoints
- ▶ Use predefined checks based on the Center of Internet Security (CIS) or OS specific alternative
- ▶ Provides periodic scanning and reporting of misconfigurations in the monitored system
- ▶ [Policies for the SCA](#) scans are written in YAML format
- ▶ Policies can be extended or written completely new to fit organization needs
- ▶ For example, a rule can be used to look for the existence of a file, a directory, a Windows registry key, a running process and many others
- ▶ It is also possible to execute a command and check its output against a regular expression



Security Configuration Assessment (SCA)

```
- id: 2651
  title: "Ensure SSH HostbasedAuthentication is disabled"
  description: "The HostbasedAuthentication parameter specifies if authentication is allowed through trusted hosts via the user of .rhosts, or /etc/hosts.equiv, along with successful public key client host authentication. This option only applies to SSH Protocol Version 2."
  rationale: "Even though the .rhosts files are ineffective if support is disabled in /etc/pam.conf, disabling the ability to use .rhosts files in SSH provides an additional layer of protection."
  remediation: "Edit the /etc/ssh/sshd_config file to set the parameter as follows: HostbasedAuthentication no"
  compliance:
    - cis: ["5.2.9"]
    - cis_csc: ["16.3"]
    - pci_dss: ["4.1"]
    - hipaa: ["164.312.a.2.IV", "164.312.e.1", "164.312.e.2.I", "164.312.e.2.II"]
    - nist_800_53: ["SC.8"]
    - tsc: ["CC6.7"]
  condition: all
  rules:
    - 'c:sshd -T -> r:HostbasedAuthentication\s+no'
```

Security Configuration Assessment (SCA)

- › Check that a file exists:
 - › `f:/path/to/file`
- › Check file contents against regex:
 - › `d:/path/to/directory -> r:REGEX`
- › Check if a process is running
 - › `p:process_name`
- › Check the output of a command
 - › `c:command -> output`
- › Check the output of a command using regex
 - › `c:command -> r:REGEX`
- › Check if a registry exists
 - › `r:path/to/registry`
- › Check if a registry key exists
 - › `r:path/to/registry -> key`



Security Configuration Assessment (SCA)

- ▶ Check for file contents, whole line match:
 - ▶ `f:/proc/sys/net/ipv4/ip_forward -> 1`
- ▶ Check if a file exists:
 - ▶ `f:/proc/sys/net/ipv4/ip_forward`
- ▶ Check if a directory contains files:
 - ▶ `d:/var/lib/mysql -> r:^.mysql_history$`
- ▶ Check if a directory exists:
 - ▶ `d:/etc/mysql`
- ▶ Check the running configuration of sshd for the maximum authentication tries allowed:
 - ▶ `c:sshd -T -> !r:^\s*MaxAuthTries\s+3\s*$`
- ▶ Check if root is the only account with UID 0:
 - ▶ `f:/etc/passwd -> !r:^# && !r:^root: && r:^\\w+:\\w+:0:`



Demo time



File Integrity Monitoring (FIM)

- ▶ Detect creation and modification of cron jobs
- ▶ Wazuh by default has a set of rules to detect when changes are made to cron jobs.
- ▶ The rules are rules ID 2830, 2831, 2832, 2833, and 2834.

```
<rule id="2832" level="5">
  <if_sid>2830</if_sid>
  <match>REPLACE</match>
  <description>Crontab entry changed.</description>
  <group>pci_dss_10.2.7,pci_dss_10.6.1,gpg13_4.13,gdpr_IV_35.7.d,hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AU.6,tsc_CC6.8, ... </group>
</rule>

<rule id="2833" level="8">
  <if_sid>2832</if_sid>
  <match>REPLACE (root)</match>
  <description>Root's crontab entry changed.</description>
  <mitre>
    <id>T1053.003</id>
  </mitre>
  <group>pci_dss_10.2.7,pci_dss_10.6.1,pci_dss_10.2.2,gpg13_4.13,gdpr_IV_35.7.d,gdpr_IV_32.2,hipaa_164.312.b,nist_800_53_AU.14, ...</group>
</rule>
```

Wazuh: Threat detection and active protection

File Integrity Monitoring (FIM)

```
# Edit Wazuh Agent configuration in /var/ossec/etc/ossec.conf
<syscheck>
  <directories realtime="yes">/var/spool/cron/crontabs/</directories>
  <directories realtime="yes" report_changes="yes">/etc/cron.d/</directories>
  <directories realtime="no" report_changes="yes">/etc/crontab</directories>

  <directories check_all="no" check_md5sum="yes" realtime="yes" report_changes="yes">/opt/myapp/</directories>
  <nodiff>/opt/my_bad_app/passwords.txt</nodiff>
</syscheck>

# Restart Wazuh Agent service
systemctl restart wazuh-agent
```

Wazuh: Threat detection and active protection

File Integrity Monitoring (FIM)

```
# Edit Wazuh Manager local rules file in /var/ossec/etc/rules/local_rules.xml
<group name="Crontab check,">
  <rule id="100010" level="12">
    <if_sid>550, 554</if_sid>
    <field name="file" type="pcre2">^\var\spool\cron\crontabs</field>
    <description>Cron job has been modified for user "${uname}". </description>
    <mitre>
      <id>T1053.003</id>
    </mitre>
  </rule>
  <rule id="100011" level="12">
    <if_sid>550, 554</if_sid>
    <field name="file" type="pcre2">^\etc\crontab</field>
    <description>Crontab file /etc/crontab has been modified. </description>
    <mitre>
      <id>T1053.003</id>
    </mitre>
  </rule>
</group>

# Restart Wazuh Manager service
systemctl restart wazuh-manager
```

Malware detection with VirusTotal

```
# Edit Wazuh Manager configuration in /var/ossec/etc/ossec.conf
<integration>
  <name>virustotal</name>
  <api_key>{%API_KEY%}</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>

# Restart Wazuh Manager service
systemctl restart wazuh-manager

# Edit Wazuh Agent configuration in /var/ossec/etc/ossec.conf
<directories realtime="yes">/opt/myapp/download/</directories>

# Restart Wazuh Agent service
systemctl restart wazuh-agent

# Test it out
mkdir -pv /opt/myapp/download/
cd /opt/myapp/download/
curl -LO https://secure.eicar.org/eicar.com && ls -lAh eicar.com
```

Custom SCA policies

- ▶ This can either be done on Wazuh Manager server and then remotely distributed by it to the agents (this needs to be explicitly allowed on Wazuh Agent by the following command):

```
# On Wazuh Agent allow remote SCA configuration push from Wazuh Manager  
echo "sca.remote_commands=1" >> /var/ossec/etc/local_internal_options.conf
```

- ▶ Or it can be configured locally on individual agents (hopefully by automation)

```
# On Wazuh Agent create a directory for custom SCA files  
mkdir /var/ossec/etc/custom-sca-files  
  
# And open the policy file itself for editing  
vim /var/ossec/etc/custom-sca-files/myapp_check.yml
```

Custom SCA policies

```
---
policy:
  id: "myapp_check"
  file: "myapp_check.yml"
  name: "Wazuh: Threat detection and active protection - demo"
  description: "Wazuh: Threat detection and active protection - demo"
  references:
    - https://www.initmax.com/

requirements:
  title: "Check that the myapp configuration file exists on monitored endpoint."
  description: "Requirements for running the SCA scans against endpoint with 'myapp_check.yml' on them."
  condition: all
  rules:
    - 'f:/opt/myapp/config'

checks:
  - id: 10001
    title: "Ensure password is disabled in the myapp configuration file"
    description: "Password is disabled in the myapp configuration file."
    rationale: "Password is considered weak for running the application. Threat actors can brute-force your password."
    remediation: "Disable password usage by setting the value of the 'UsePassword' configuration directive to 'no'."
    condition: all
    regex_type: "osregex"
    rules:
      - 'f:/opt/myapp/config -> !r:^# && r:UsePassword && r:no$'
...

```

Custom SCA policies

```
# Change ownership of the SCA policies directory (and files)
chown wazuh:wazuh /var/ossec/etc/custom-sca-files -R

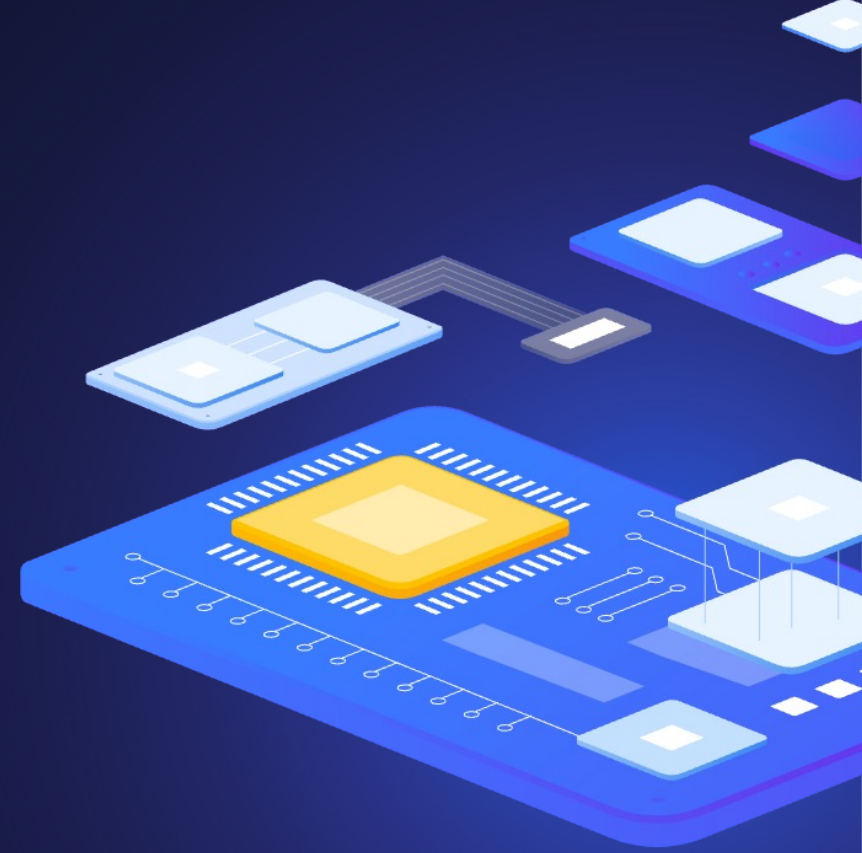
# Edit Wazuh Agent configuration in /var/ossec/etc/ossec.conf
<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>

  <policies>
    <policy enabled="yes">/var/ossec/etc/custom-sca-files/myapp_check.yml</policy>
    <policy enabled="no">etc/shared/myapp_check.yml</policy>
  </policies>
</sca>

# Restart Wazuh Agent service
systemctl restart wazuh-agent
```



Questions?



Wazuh: Threat detection and active protection

Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



tomas.hermanek@initmax.cz

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184