initMAX

Using Wazuh for Forensic Analysis

all our microphones are muted

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

use Chat for discussion, networking or applause

Agenda

1 Digital Forensics

2 Preparations

3 Evidence Collection

1

Digital Forensics

initMAX

What is Digital Forensics

- Science of collecting, preserving, analyzing, and presenting digital evidence.
- Critical in responding to cyber incidents (breaches, malware, insider threats).
- Focus on identifying anomalies and understanding incident scope, impact, and prevention.
- Emphasizes evidence integrity and chain of custody.

initMAX

Role of SIEM in Forensic Investigations

- SIEM systems aggregate and correlate logs across systems.
- Enable timeline reconstruction and threat detection.
- Wazuh adds real-time alerting, log analysis, and compliance tools.
- Key for identifying IOCs and tracking attacker behavior.

initMAX

Centralized Logging

- Centralizes logs from endpoints, servers, network devices, and applications.
- Simplifies correlation and anomaly detection.
- Essential log types:
  - OS logs
  - Authentication
  - Firewall
  - IDS/IPS
  - Application logs
  - EDR
- Ensures completeness, accuracy, and long-term retention of data.

initMAX

System Activity Tracking

- auditd or eBPF (Linux) and sysmon (Windows) capture deep system activity.
- Key logs: file access, process creation, registry changes, network events.
- Enables detection of lateral movement and persistence mechanisms.
- Integration enriches logs for easier forensic correlation.

initMAX

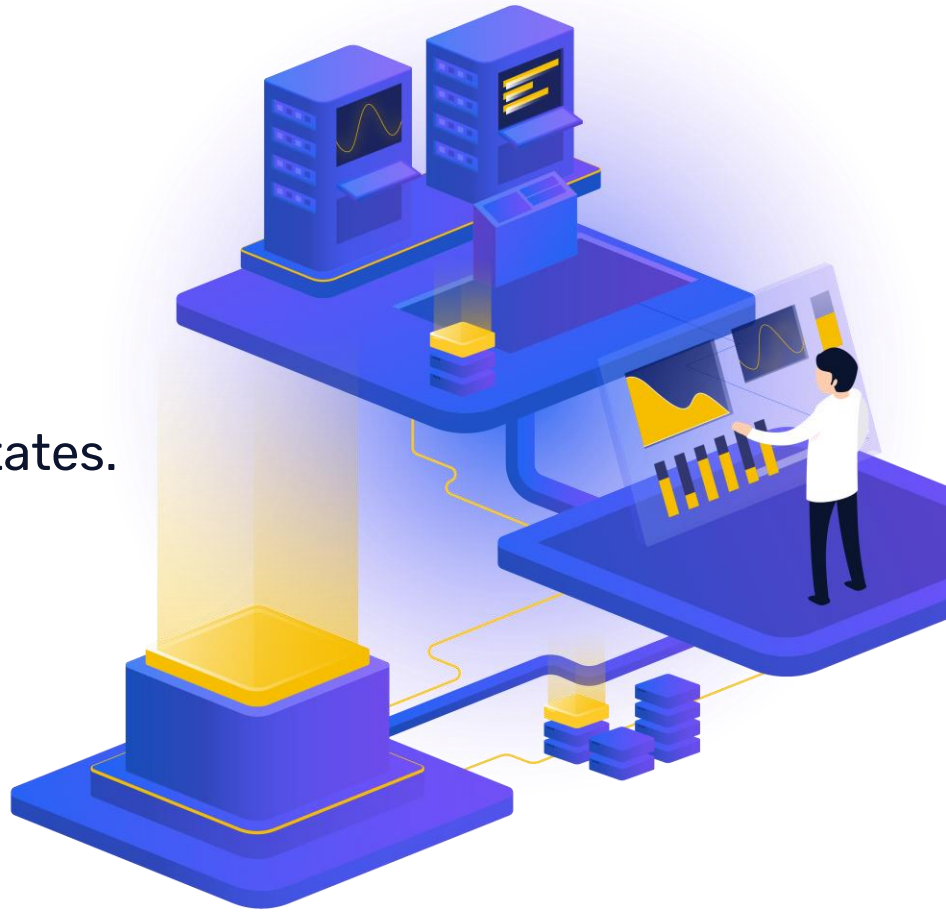
Evidence Preservation (FIM)

- Detects unauthorized changes to critical files and configs.
- Tracks who changed what, when, and how.
- Establishes a reliable baseline for forensic comparison (hash).
- Real-time monitoring with alerting for abnormal behavior.

initMAX

Indicators of Compromise (IOCs)

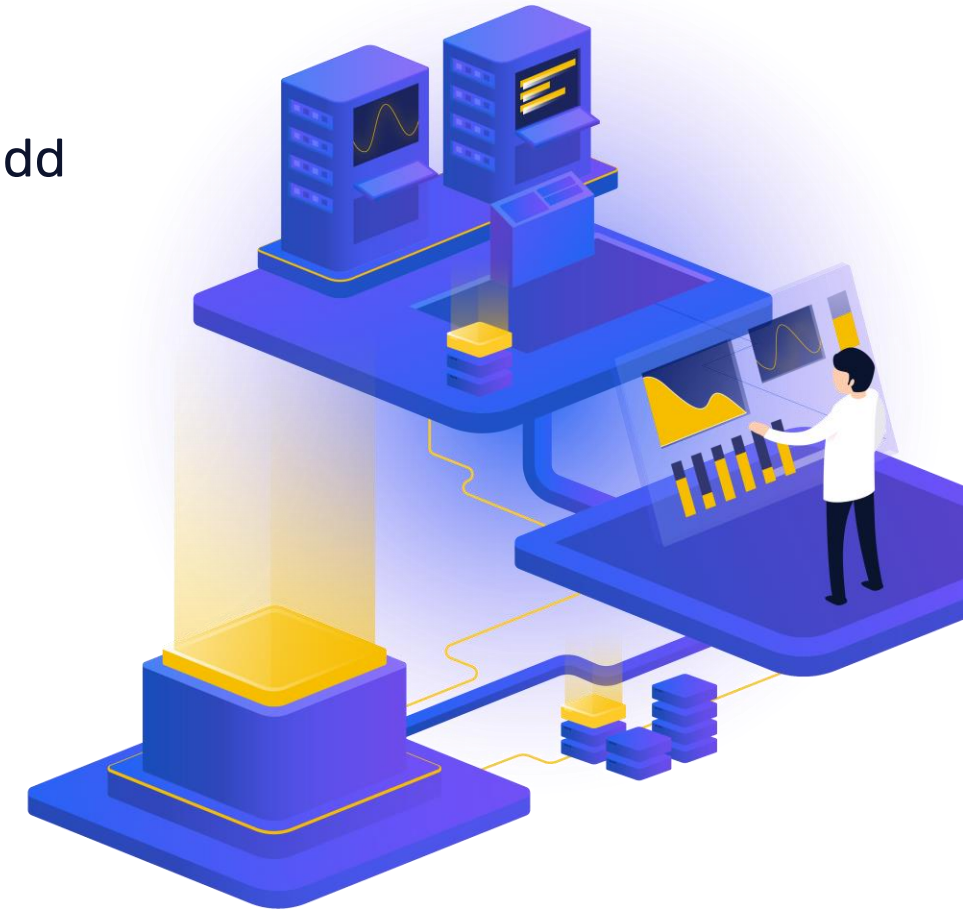
- IOCs: IPs, hashes, domains, registry keys linked to attacks.
- Understanding Event logs and metadata.
- Logs contain timestamps, user IDs, source IPs, and system states.
- Metadata adds context: who, when, where, what.
- Parsing and normalization make logs searchable and correlatable.
- Foundation for accurate forensic analysis.



initMAX

Detecting Suspicious Behavior

- Focus on deviations from normal patterns: failed logins, odd processes, etc.
- Behavioral rules detect anomalies and trigger alerts.
- Wazuh decoders and rulesets help flag high-risk activity.
- Alerts guide deeper forensic investigation.



initMAX

Building Incident Timeline

- Alerts are time-stamped and categorized for event reconstruction.
- Correlating alerts reveals attack paths, methods, and targets.
- Helps confirm hypotheses and identify compromised systems.
- Essential for post-incident reporting and response.

initMAX

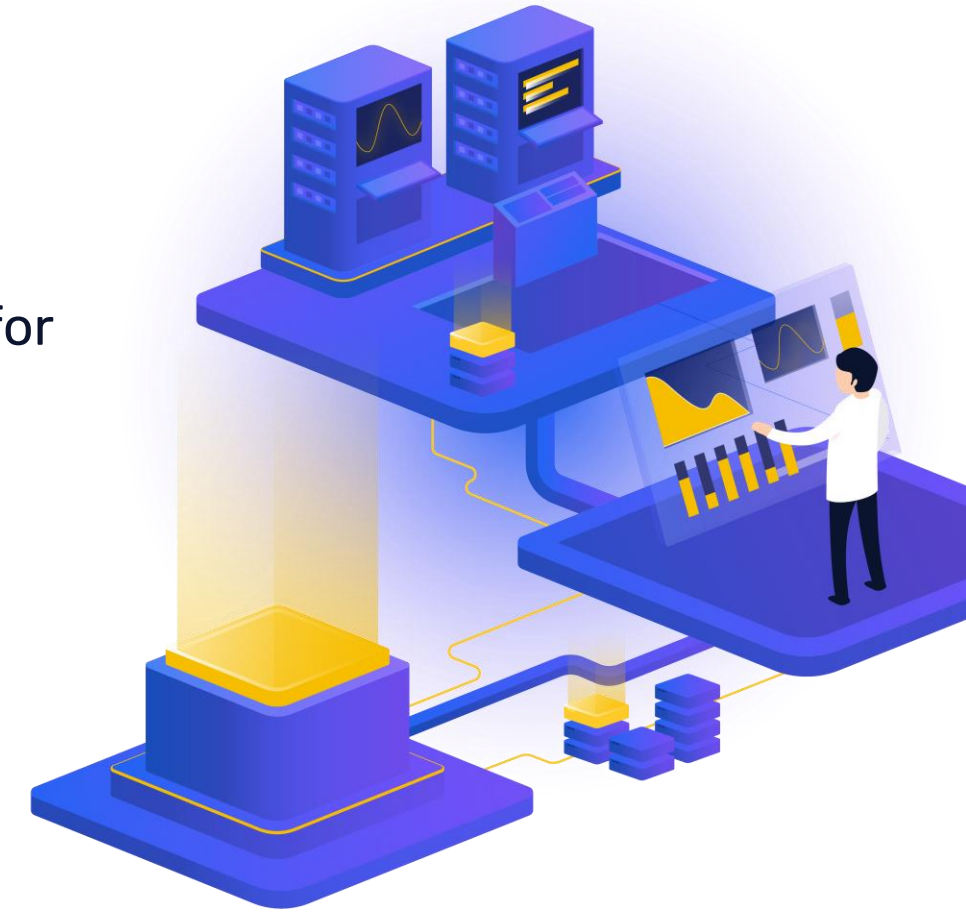
Proactive Hunting vs. Reactive Analysis

- **Proactive:** Hunt threats before alerts (hypothesis-driven).
- **Reactive:** Respond to alerts or known incidents.
- Both approaches are essential for complete coverage.
- Wazuh supports both via log search, rule tuning, and alerting.

initMAX

Legal Chain of Custody

- Export logs in standardized formats (e.g., JSON, CSV).
- Use WORM (Write Once Read Many) storage and hashing for tamper-proof preservation.
- Document collection and access steps for audit trail.
- Ensure compliance with legal and evidentiary standards.



initMAX

Automation of Forensic Tasks

- Wazuh API enables log searches, alert retrieval, and report generation.
- Automates repetitive forensic processes (IOC extraction, reporting).
- Can be integrated with SOAR and scripting tools.
- Enhances speed, consistency, and scalability.

initMAX

Case Management and Reporting

- Organize investigations with case tracking and alert tagging.
- Integrate with platforms like TheHive or DFIR-IRIS for collaboration.
- Generate forensic reports for legal, compliance, and executive use (eg. internal reviews, post-mortems, external auditors).
- Clear documentation streamlines response and audits.

2

Preparations

initMAX

System Activity Tracking

- Use **auditd** or **eBPF** (Linux) and **sysmon** (Windows) to capture deep system activity.
- **Key logs:** file access, process creation, registry changes, network events.
- This enables detection of lateral movement and persistence mechanisms.
- Integration enriches logs for easier forensic correlation.

initMAX

Setting up auditd

/etc/audit/rules.d/99-custom.rules

Log admin changes and root commands:

```
# log admin
-w /bin/su -p x -k audit-wazuh-x
-w /usr/bin/sudo -p x -k audit-wazuh-x
-w /etc/sudoers -p rw -k audit-wazuh-x

# log root commands
-a exit,always -F arch=b32 -F euid=0 -S execve -k audit-wazuh-c
-a exit,always -F arch=b64 -F euid=0 -S execve -k audit-wazuh-c
```

initMAX

Setting up auditd

/etc/audit/rules.d/99-custom.rules

Log permission changes:

```
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b32 -S removexattr -S lremovexattr -S fremovexattr -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
-a always,exit -F arch=b64 -S removexattr -S lremovexattr -S fremovexattr -F auid>=500 -F auid!=4294967295 -k audit-wazuh-a
```

initMAX

Setting up auditd

Do NOT forget self-auditing:

```
-w /var/log/audit/ -k audit-wazuh-w
-w /etc/audit/ -p wa -k audit-wazuh-w
-w /etc/libaudit.conf -p wa -k audit-wazuh-w
-w /etc/audisp/ -p wa -k audit-wazuh-w
-w /sbin/auditctl -p x -k audit-wazuh-w
-w /sbin/auditd -p x -k audit-wazuh-w
```

Do NOT forget to protect the configuration (when tested successfully):

```
# immutable
-e 2
```

initMAX

Setting up auditd

Load auditd rules:

```
auditctl -R /etc/audit/rules.d/audit.rules
```

List auditd rules:

```
auditctl -l
```

/var/ossec/etc/ossec.conf (if not present):

```
<localfile>
    <location>/var/log/audit/audit.log</location>
    <log_format>audit</log_format>
</localfile>
```

initMAX

Setting up auditd

/etc/ossec/lists/audit-keys

```
audit-wazuh-w:write
audit-wazuh-r:read
audit-wazuh-a:attribute
audit-wazuh-x:execute
audit-wazuh-c:command
```

initMAX

Setting up eBPF (Wazuh 4.12+)

Install following packages:

```
bpfcc-tools                     # bpf tools
linux-headers-$(uname -r)       # kernel headers
bpftrace                        # optional but usefull (scriptable)
```

Example manual run (as root):

```
execsnoop                                                                   # trace new process execution
opensnoop                                                                   # trace open() syscalls
bpftrace -e 'tracepoint:syscalls:sys_enter_execve { printf("%s\n", comm); }'   # trace new command execution (syscall)
```

Example Wazuh configuration (falls back to auditd or inotify):

```
<syscheck>
  <directories whodata="yes">/home/*/.ssh</directories>
  <whodata>
    <provider>ebpf</provider>
  </whodata>
</syscheck>
```

initMAX

Evidence Preservation (FIM)

- Detects unauthorized changes to critical files and configs.
- Tracks who changed what, when, and how.
- Establishes a reliable baseline for forensic comparison.
- Real-time monitoring with alerting for abnormal behavior.

initMAX

Evidence Preservation (FIM)

Example syscheck module configuration (Linux):

```
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/boot,/etc/grub.d</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/inittab,/etc/securetty,/etc/fstab</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/sysconfig,/etc/default</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/init.d,/etc/rc.d</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/systemd/system,/usr/lib/systemd/system</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/networks,/etc/netconfig,/etc/iproute2</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/passwd,/etc/group,/etc/shadow,/etc/gshadow</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/nsswitch.conf,/etc/sssd,/etc/krb5.conf</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/exports,/etc/idmapd.conf</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/sudoers,/etc/sudoers.d</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/etc/pam.d,/etc/security</directories>
<directories check_all="no" check_md5sum="yes" check_mtime="yes">/bin,/sbin,/usr/bin,/usr/sbin</directories>
```

initMAX

Evidence Preservation (FIM)

Example syscheck module configuration (Windows):

```
<directories recursion_level="0" restrict="regedit.exe$|system.ini$|win.ini$">%WINDIR%</directories>

<directories recursion_level="0"
restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe$|reged
t32.exe|regsvr32.exe|runas.exe|sc.exe|schtasks.exe|sethc.exe|subst.exe$|winrm.vbs$">%WINDIR%\SysWOW64</directories>

<directories recursion_level="0">%WINDIR%\SysWOW64\drivers\etc</directories>
<directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\SysWOW64\wbem</directories>
<directories recursion_level="0" restrict="powershell.exe$">%WINDIR%\SysWOW64\WindowsPowerShell\v1.0</directories>
<directories recursion_level="0">%WINDIR%\System32\drivers\etc</directories>
<directories recursion_level="0" restrict="WMIC.exe$">%WINDIR%\System32\wbem</directories>
<directories recursion_level="0" restrict="powershell.exe$">%WINDIR%\System32\WindowsPowerShell\v1.0</directories>

 <windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\batfile</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\cmdfile</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\comfile</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\exefile</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\piffile</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\AllFilesystemObjects</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\Directory</windows_registry>
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\Folder</windows_registry>
```

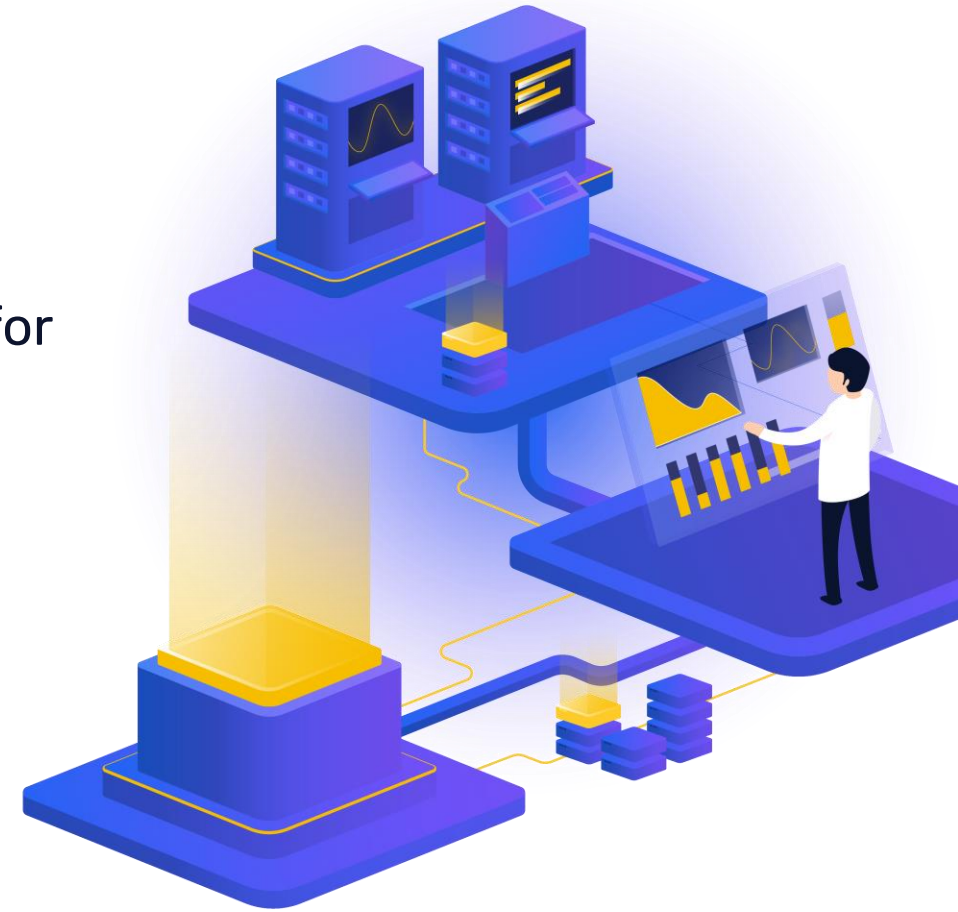
3

Evidence Collection

initMAX

Legal Chain of Custody

- Export logs in **standardized** formats (e.g., JSON, CSV).
- Use **WORM** (Write Once Read Many) storage and hashing for tamper-proof preservation.
- Document **collection** and **access** steps for audit trail.
- Ensure **compliance** with legal and evidentiary standards.



initMAX

Legal Chain of Custody

Log all!

```
<global>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
</global>
```

Export:

```
cat /var/ossec/logs/alerts/alerts.json | grep "$(date +%Y-%m-%d)" > case-001_wazuh_file_alerts.json
```

Filter?

```
jq 'select(.timestamp >= "2025-05-20T00:00:00Z" and .timestamp < "2025-05-21T00:00:00Z")' \
   /var/ossec/logs/alerts/alerts.json > case-001_wazuh_file_alerts_filtered.json
```

initMAX

Legal Chain of Custody

Export using Wazuh API:

```
curl -u wazuh_user:wazuh_pass -k \
  "https://<WAZUH-IP>:55000/alerts?from=2025-05-20T00:00:00Z&to=2025-05-21T00:00:00Z" \
  -o case-001_wazuh_api_alerts.json
```

Export from Wazuh Indexer (OpenSearch):

```
curl -X POST "http://localhost:9200/wazuh-alerts-*/_search" -H 'Content-Type: application/json' -d '{
  "query": {
    "range": {
      "@timestamp": {
        "gte": "now-1d/d",
        "lt": "now/d"
      }
    }
  }}' > case-001_wazuh_indexer_alerts.json
```

initMAX

Legal Chain of Custody

Use WORM! (requires root)

```
# Set file as immutable (write-once)
chattr +i /path/to/case-001_wazuh_*.json
```

Enterprise storage solutions that supports WORM concept:

- NetApp SnapLock
- Dell EMC Centera
- Hitachi Content Platform
- Amazon S3 Object Lock
- Azure Immutability Policy

initMAX

Legal Chain of Custody

Use hashes!

```
sha256sum /path/to/case-001_wazuh_*.json
```

And store them in WORM!

```
sha256sum /path/to/case-001_wazuh_file_alerts.json > /path/to/case-001_file_alerts.json.sha256
```

initMAX

Legal Chain of Custody

Amazon S3 example.

Create a bucket with Object Lock enabled:

```
aws s3api create-bucket \
  --bucket case-001_wazuh \
  --region eu-central-1 \
  --object-lock-enabled-for-bucket
```

Enable compliance mode and retention period:

```
aws s3api put-object-lock-configuration \
  --bucket case-001_wazuh \
  --object-lock-configuration '{
    "ObjectLockEnabled": "Enabled",
    "Rule": {
      "DefaultRetention": {
        "Mode": "COMPLIANCE",
        "Days": 365
      }
    }
  }'
```

initMAX

Legal Chain of Custody

Azure example.

Create a container:

```
az storage container create \
  --account-name <yourStorageAccount> \
  --name case-001_wazuh \
  --auth-mode login
```

Set an Immutability Policy (Legal Hold):

```
az storage container legal-hold set \
  --account-name <yourStorageAccount> \
  --container-name case-001_wazuh \
  --tags "Forensics" "case-001"
```

Lock the Immutability Policy:

```
az storage container immutability-policy lock \
  --account-name <yourStorageAccount> \
  --container-name case-001_wazuh
```

initMAX

Legal Chain of Custody

Best practices for WORM configuration:

- Always **timestamp** logs (Wazuh supports this).
- Use **cryptographic hashes** to verify integrity.
- Log **access to logs** for full audit trail.
- Use **RBAC** to restrict who can manage retention or access logs.
- Document your WORM policy for legal/compliance use.
- If you're **backing up** Wazuh logs using third-party systems (e.g., Veeam, Commvault, Veritas), ensure WORM options are enabled in their policies!

initMAX

Questions?

initMAX

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