SECURITY-RELATED MONITORING **WITH ZABBIX**

Kaspars Mednis
ZABBIX Technical Support Engineer

ZABBIX SUMMIT '19

SECURITY MONITORING – WHY ?

Potential issues

- Software vulnerabilities
- Weak configurations
- Unnecessary open ports
- Physical intrusions

WHY ZABBIX ?

Zabbix is not a dedicated security monitoring tool....
but you can monitor the following

- configuration files
- log files
- SNMP traps
- and much more.....

CHECKSUM MONITORING

SECURITY-RELATED MONITORING
WITH ZABBIX

WHY MONITOR CHECKSUMS ?

Because it is the simplest way to detect changes to important files

- works out of box
- very simple to setup
- efficient

HOW TO MONITOR CHECKSUMS ?

Two types of checksums supported:

vfs.file.cksum[file] - calculates a 32-bit ckecksum (CRC-

vfs.file.cksum[/etc/passwd] = 1222364044

vfs.file.md5sum[file] - calculates a 128-bit MD5 hash

vfs.file.md5sum[/etc/passwd] [illegible]

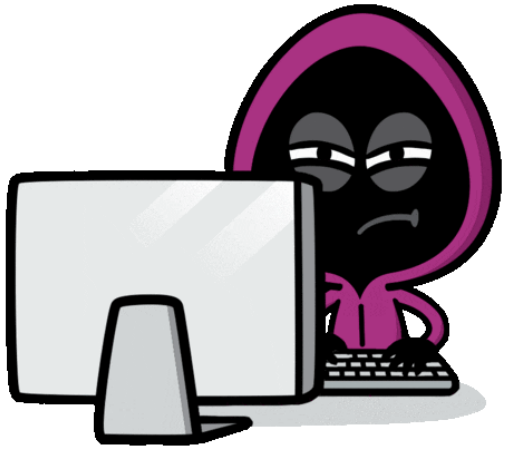
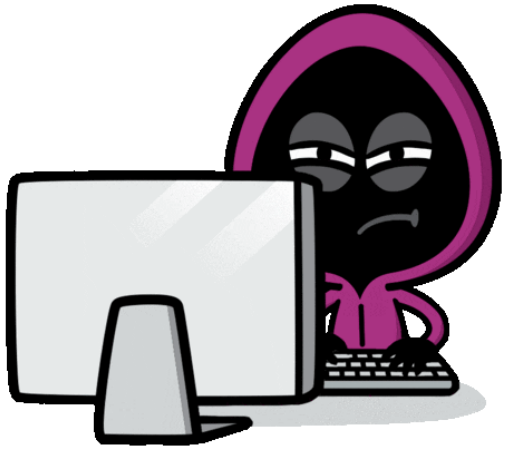
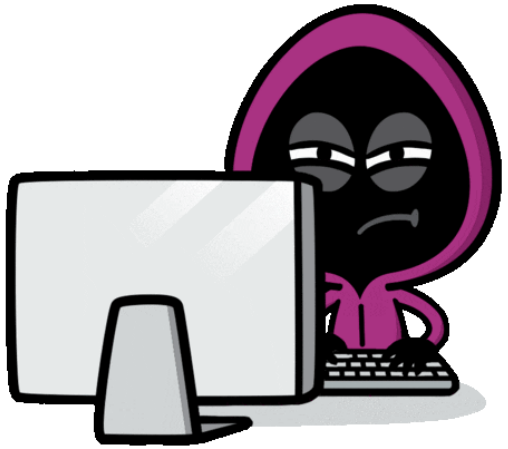
CONFIG FILES

SECURITY-RELATED MONITORING
WITH ZABBIX

CONFIGURATION ISSUES

Default configuration gives a lot of information
And while it is very useful for deployment and troubleshooting....

It can contain known weaknesses
It can also give very valuable information to potential hackers !!!

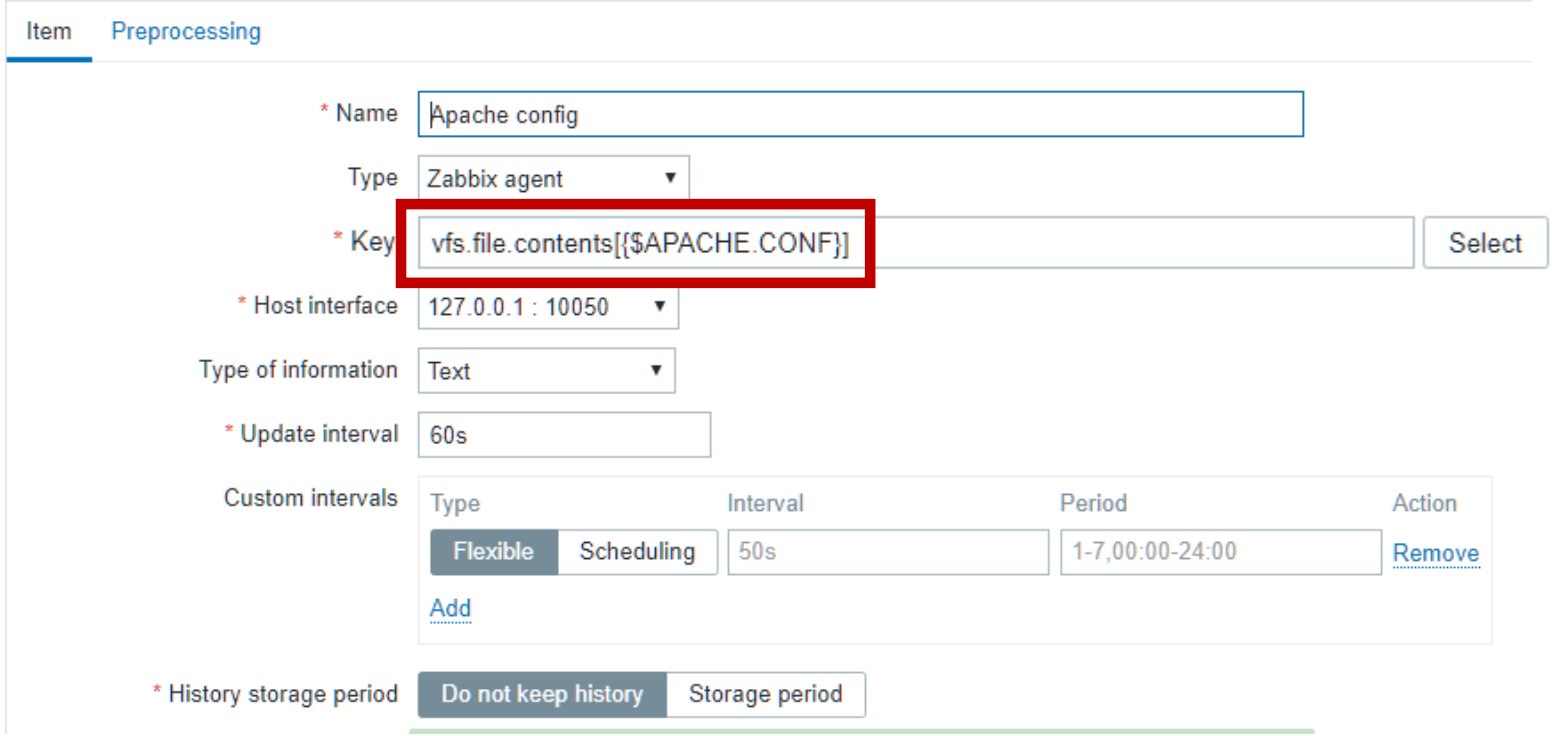




HOW TO MONITOR CONFIGURATION ?

You can monitor the content of a configuration file

vfs.file.contents[file] – returns back the content of a file

The most important parts of a configuration file can be monitored using dependent items.



ZABBIX EXAMPLE

Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type
•••	Apache config		vfs.file.contents[/etc/httpd/conf/httpd.conf]	60s	0		Zabbix agent
•••	Apache config: Apache server signature		apache.server.signature		90d		Dependent item
•••	Apache config: Apache server tokens		apache.server.tokens		90d		Dependent item

Item | Preprocessing

Preprocessing steps	Name	Parameters		Custom on fail	Actions
1:	Matches regular expression	ServerTokens		☑	Test Remove
Custom on fail	Discard value / Set value to / Set error to	ServerTokens Full			
2:	Regular expression	ServerTokens\s+(.+)	\1	☐	Test Remove
3:	Discard unchanged with heartbea	3600			Test Remove

Add

Test all steps

Update | Clone | Check now | Clear history and trends | Delete | Cancel

RESULTS

Name ▲	Inter…	Hist…	Tren…	Type	Last check	Last value
Apache (3 Items)						
Apache config vfs.file.contents[/etc/httpd/conf/ht…	60s	0		Zabbix ag…		
Apache server signature apache.server.signature		90d		Depende…	2019-10-10 14:39:19	On
Apache server tokens apache.server.tokens		90d		Depende…	2019-10-10 15:14:19	Full

Severity	Recovery time	Status	Info	Host	Problem
High		PROBLEM		Test	Apache server tokens is not "prod"

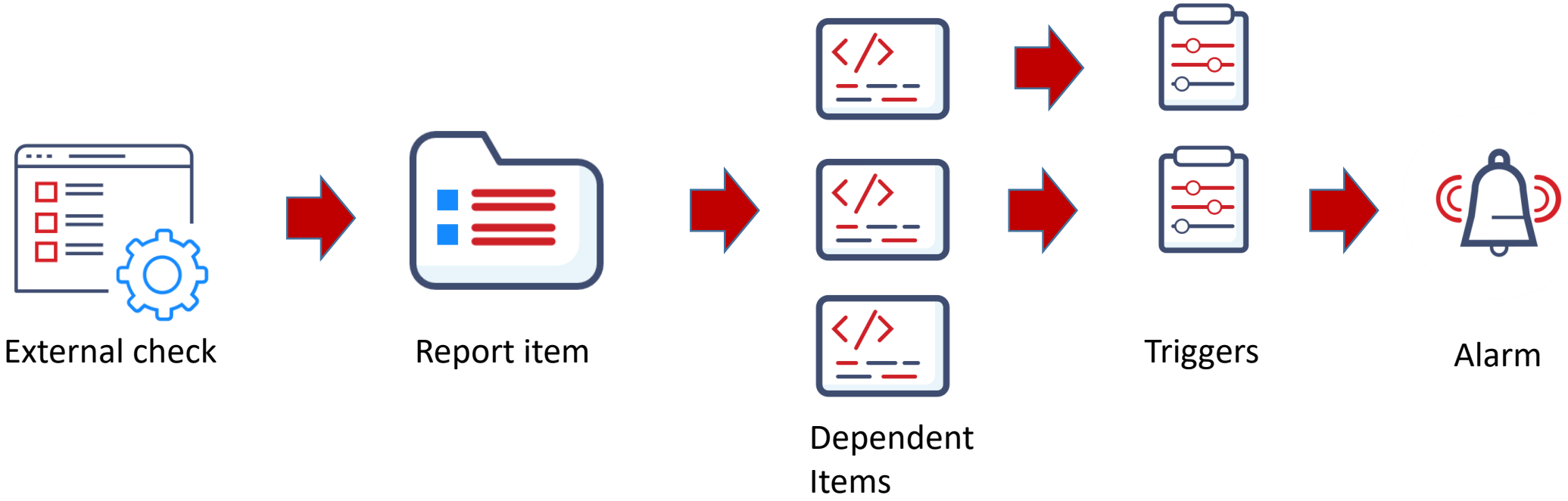
VULNERABILTY SCANS

SECURITY-RELATED MONITORING
WITH ZABBIX

WHAT IF YOU ARE NOT A SECURITY EXPERT ?

- External programs can be used to check vulnerabilities
- Output can be parsed, and useful information extracted
- Triggers can be created to send out alerts

HOW IT WORKS



MONITORING USING SCRIPTS

Example of a security report

```
[root@          home]# nikto -o report.xml -C all -Tuning 9 -h  http://127.0.0.1/
- ***** RFIURL is not defined in nikto.conf--no RFI tests will run *****
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    127.0.0.1
+ Target Port:        80
+ Start Time:         2019-10-02 18:45:26 (GMT3)
---------------------------------------------------------------------------
+ Server: Apache
+ Retrieved x-powered-by header: PHP/5.4.16
+ Server leaks inodes via ETags, header found with file        /robots.txt, fields: 0x3ce 0x593eb000cf4c0
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ 2121 requests: 0 error(s) and 5 item(s) reported on remote host
+ End Time:           2019-10-02 18:45:32 (GMT3) (6 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

HOW TO EXTRACT INFORMATION ?

Use Zabbix built – in preprocessing

- Regular expressions
- JSON PATH
- XML PATH
- CSV to JSON
- JavaScript

LLD PREPROCESSING POSSIBILITIES

Discovery rule | Preprocessing | LLD macros | Filters

Preprocessing steps | Name

1: Regular expression

Add

Add

- **Text**
  - Regular expression
- **Structured data**
  - XML XPath
  - JSONPath
  - CSV to JSON
- **Custom scripts**
  - JavaScript
- **Validation**
  - Does not match regular expression
  - Check for error in JSON
  - Check for error in XML
- **Throttling**
  - Discard unchanged with heartbeat
- **Prometheus**
  - Prometheus to JSON

DEPENDENT ITEMS EXAMPLE

Item | Preprocessing

Preprocessing steps

	Name	Parameters	
1:	Regular expression	(\d+(?= error))	\1

Add

Update | Clone | Check now | Clear history and trends | Delete | Cancel

Host	Name ▲	Interval	History	Trends	Type	Last check	Last value
Test	**report** (4 Items)						
	Scan errors scan.errors		90d	365d	Dependent item	2019-10-02 19:39:25	0 errors
	Scan reports scan.reports		90d	365d	Dependent item	2019-10-02 19:39:25	5 reports
	Scan requests scan.requests		90d	365d	Dependent item	2019-10-02 19:39:25	2121 requests
	Security report scan.sh["http://127.0.0.1/zabbix"]	30s	90d		External check	2019-10-02 19:39:25	- ***** RFIURL is not define...

ADVANCED VULNERABILTY SCANS

SECURITY-RELATED MONITORING
WITH ZABBIX

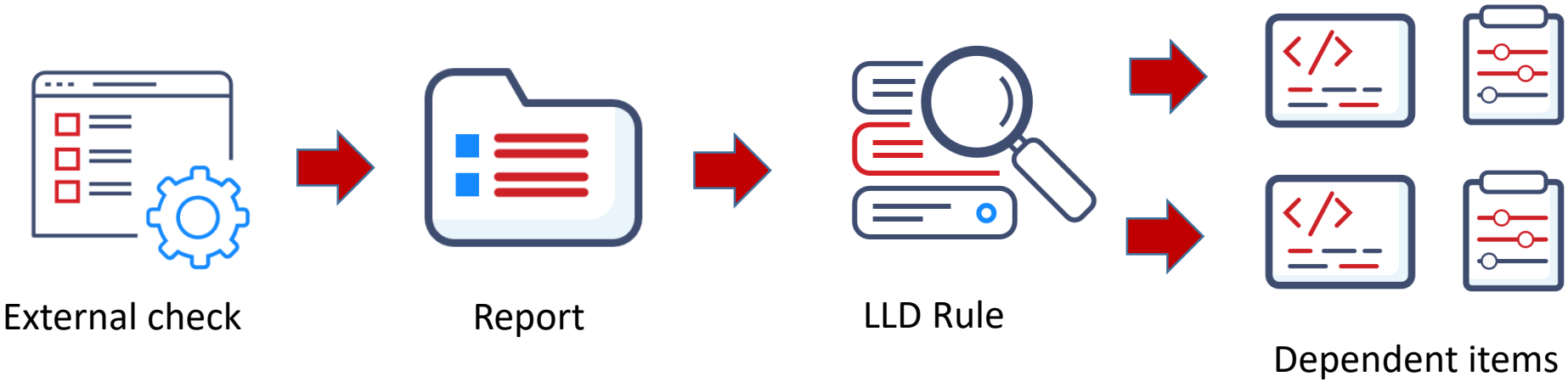
CAN YOU DO SOMETHING WITH THIS REPORT ?


```
{"niktoscan":{"nxmlversion":"1.2","options":"-o report.xml -C all -Tuning 9 -h http://127.0.0.1/myap
p","version":"2.1.6","scandetails":{"sitename":"http://127.0.0.1:80/myapp/","targetport":"80","check
s":"62","errors":"0","targethostname":"127.0.0.1","statistics":{"itemstested":"62","endtime":"2019-1
0-02 18:45:32","itemsfound":"5","elapsed":"6"},"item":{"999990":{"itemid":"999990","iplink":"http://
127.0.0.1:80/myapp/","namelink":"http://127.0.0.1:80/myapp/","description":"Allowed HTTP Methods: GE
T, HEAD, POST, OPTIONS, TRACE ","uri":"/myapp/","osvdbid":"0","method":"OPTIONS","osvdblink":"http:/
/osvdb.org/0"},"999986":{"itemid":"999986","iplink":"http://127.0.0.1:80/myapp/","namelink":"http://
127.0.0.1:80/myapp/","description":"Retrieved x-powered-by header: PHP/5.4.16","uri":"/myapp/","osvd
bid":"0","method":"GET","osvdblink":"http://osvdb.org/0"},"999967":{"itemid":"999967","iplink":"http
://127.0.0.1:80/myapp/","namelink":"http://127.0.0.1:80/myapp/","description":"Web Server returns a
valid response with junk HTTP methods, this may cause false positives.","uri":"/myapp/","osvdbid":"0
","method":"OFXMDQTR","osvdblink":"http://osvdb.org/0"},"999971":{"itemid":"999971","iplink":"http:/
/127.0.0.1:80/myapp/","namelink":"http://127.0.0.1:80/myapp/","description":"HTTP TRACE method is ac
tive, suggesting the host is vulnerable to XST","uri":"/myapp/","osvdbid":"877","method":"TRACE","os
vdblink":"http://osvdb.org/877"},"999984":{"itemid":"999984","iplink":"http://127.0.0.1:80/myapp/rob
ots.txt","namelink":"http://127.0.0.1:80/myapp/robots.txt","description":"Server leaks inodes via ET
ags, header found with file /myapp/robots.txt, fields: 0x3ce 0x593eb000cf4c0 ","uri":"/myapp/robots.
txt","osvdbid":"0","method":"GET","osvdblink":"http://osvdb.org/0"}},"targetbanner":"Apache","target
ip":"127.0.0.1","starttime":"2019-10-02 18:45:26","hostheader":"127.0.0.1","siteip":"http://127.0.0.
1:80/myapp/"},"hoststest":"0","scanstart":"Wed Oct  2 18:45:25 2019","scanelapsed":" seconds","scane
nd":"Thu Jan  1 03:00:00 1970"}}
```


ZABBIX SUMMIT '19

PROCESS THE REPORT USING LLD

Any JSON format data can be processed by LLD



LLD RULE DESIGN


```
"niktoscan": {
    "options":"-o report.xml -C all -Tuning 9 -h
    http://127.0.0.1/zabbix",
    "scandetails": {
        "targetport":"80",
        "checks":"62",
        "errors":"0",
        "targethostname":"127.0.0.1",
        "statistics": {
            "itemstested":"62",
            "endtime":"2019-10-02 18:45:32",
            "itemsfound":"5",
            "elapsed":"6"
        },
        "item": {
            "itemid":"999990",
            "namelink":"http://127.0.0.1:80/zabbix/",
            "description":"Allowed HTTP Methods: GET, HEAD,
            POST, OPTIONS, TRACE ",
            "uri":"/zabbix/",
            "osvdbid":"0",
            "method":"OPTIONS",
            "osvdblink":"http://osvdb.org/0"
        },
        "999986": {
            "itemid":"999986",
            "namelink":"http://127.0.0.1:80/zabbix/",
            "description":"Retrieved x-powered-by header:
            PHP/5.4.16",
            "uri":"/zabbix/",
            "osvdbid":"0",
            "method":"GET",
            "osvdblink":"http://osvdb.org/0"
        },
```


Discovery rule | Preprocessing | LLD macros | Filters

* Name	Vulnerability LLD
Type	External check
* Key	report.pl[lld]
* Host interface	127.0.0.1 : 10050
* Update interval	1h

Discovery rule | Preprocessing | LLD macros | Filters

Preprocessing steps

	Name	Parameters
1:	JSONPath	$.niktoscan.scandetails.item.*

Add

Update | Clone | Check now | Delete | Cancel

Discovery rule | Preprocessing | LLD macros | Filters

LLD macros

LLD macro	JSONPath	
{#ITEMID}	$.itemid	Remove

Add

Update | Clone | Check now | Delete | Cancel

LLD ITEM PROTOTYPES

Wizard	Name ▲	Key	Interval	History	Trends	Type
•••	Vulnerability data: Vulnerability {#ITEMID}	vulnerability[{#ITEMID}]		90d		Dependent item

Item prototype | Preprocessing

* Name: Vulnerability {#ITEMID}

Type: Dependent item

* Key: vulnerability[{#ITEMID}] [Select]

* Master item: Test: Vulnerability data [Select] [Select prototype]

Type of information: Character

Item prototype | Preprocessing

Preprocessing steps

	Name	Parameters
1:	JSONPath	$.niktoscan.scandetails.item.{#ITEMID}.description

Add

[Update] [Clone] [Delete] [Cancel]

ITEMS CREATED FROM THE REPORT

Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type
•••	Vulnerability LLD: Vulnerability data: Vulnerability 999967	Triggers 1	vulnerability[999967]		90d		Dependent item
•••	Vulnerability LLD: Vulnerability data: Vulnerability 999971	Triggers 1	vulnerability[999971]		90d		Dependent item
•••	Vulnerability LLD: Vulnerability data: Vulnerability 999984	Triggers 1	vulnerability[999984]		90d		Dependent item
•••	Vulnerability LLD: Vulnerability data: Vulnerability 999986	Triggers 1	vulnerability[999986]		90d		Dependent item
•••	Vulnerability LLD: Vulnerability data: Vulnerability 999990	Triggers 1	vulnerability[999990]		90d		Dependent item
•••	Vulnerability data		report.pl[data]	1h	0		External check

Time ▼	Severity	Info	Host	Problem
2019-10-02 19:02:20	Warning		Test	Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
2019-10-02 19:02:20	Warning		Test	Retrieved x-powered-by header: PHP/5.4.16
2019-10-02 19:02:20	Warning		Test	Web Server returns a valid response with junk HTTP methods, this may cause false positives.
2019-10-02 19:02:20	Warning		Test	Server leaks inodes via ETags, header found with file /zabbix/robots.txt, fields: 0x3ce 0x593eb000cf4c0
2019-10-02 19:02:20	Warning		Test	HTTP TRACE method is active, suggesting the host is vulnerable to XST

WHAT IF THE SCRIPT TAKES TOO LONG TO EXECUTE ?

Maximum execution time is 30s .....

00:30

In this case cron jobs or other scheduling mechanisms can be used

SERVICES MONITORING

SECURITY-RELATED MONITORING
WITH ZABBIX

CAN WE MONITOR SERVICES OUT OF BOX ?

Yes – using new Zabbix agent 2 GO

Two new item keys supported

systemd.unit.discovery[<type>] List of systemd units and their details.

type - *all*, *automount*, *device*, *mount*, *path*, *service* (default), *socket*, *swap*, *target*

systemd.unit.info[<unit name>,<property>,<interface>] Systemd unit information

unit name - unit name
property - unit property (e.g. ActiveState (default), LoadState, Description)
interface - unit interface type (e.g. Unit (default), Socket, Service)

SERVICES MONITORING EXAMPLE

Item | Preprocessing

* Name	Systemd firewall status
Type	Zabbix agent
* Key	systemd.unit.info[firewalld.service] Select
* Host interface	127.0.0.1 : 10050
Type of information	Character
* Update interval	1m

Name ▲	Interval	History	Trends	Type	Last check	Last value
- other - (1 Item)						
Systemd firewall status systemd.unit.info[firewalld.service]	1m	90d		Zabbix agent	2019-10-09 15:42:30	inactive

PORT MONITORING

SECURITY-RELATED MONITORING
WITH ZABBIX

CAN WE MONITOR OPEN PORTS ?

Yes, of course !

Zabbix can do it out of box

- check open ports using **net.tcp.port[]** simple check
- use discovery to scan your entire network for open ports

WHY WE NEED TO MONITOR OPEN PORTS ?

Why do you need this ?

- Applications with weak security (telnet, ftp)
- Unneeded applications with known vulnerabilities
- Less open ports – more secure system

SIMPLE NETWORK DISCOVERY RULE

* Name	Local network
Discovery by proxy	No proxy ▾
* IP range	192.168.0.1-254
* Update interval	1h
* Checks	Zabbix agent "system.uname" Edit Remove FTP Edit Remove Telnet Edit Remove SNMPv2 agent "1.3.6.1.2.1.1.5.0" Edit Remove New

Check type	SNMPv2 agent ▾
* Port range	161
* SNMP community	public
* SNMP OID	1.3.6.1.2.1.1.5.0

Update Cancel

UNSECURE WEB PAGES

SECURITY-RELATED MONITORING
WITH ZABBIX

HOW CAN WE FIND HTTP ENABLED PAGES ?

HTTPS is the recommended web protocol today

Open HTTP port does not mean the page is not redirected to HTTPS

How to check it ?

- Use Zabbix built in web scenarios
- check the response code

The HTTP response status code **301 Moved Permanently** is used for permanent URL redirection

WEB SCENARIO EXAMPLE

Step of web scenario

* Name: Zabbix HTTP page
* URL: http://www.zabbix.com — Parse

Query fields

Name		Value	
name	⇒	value	Remove

Add

Follow redirects: ☐

Retrieve mode: Body | Headers | Body and headers

* Timeout: 15s

Required string: pattern

Required status codes: 301

Update | Cancel

CAN WE FIND UNSECURE HTTPS PAGES ?

You can use a webscenario to authenticate the certificate

- verify that SSL certificate of the web server is valid (trusted by a known certificate authority, not expired etc.)
- verify that the *Common Name* field or the *Subject Alternate Name* field of the web server certificate matches the server name.

HTTPS CERTIFICATE VALIDATION

Scenario | Steps | Authentication

HTTP authentication	None
SSL verify peer	☑
SSL verify host	☑
SSL certificate file	my_secure_certificate.cer
SSL key file	ssl_key
SSL key password	supersecurepassword

Update | Clone | Clear history and trends | Delete | Cancel

EXPIRED CERTIFICATES

SECURITY-RELATED MONITORING
WITH ZABBIX

CAN WE FIND EXPIRED CERTIFICATES?

Community made externals cripts can be used to warn you about yourcertificate expiration

You can monitor (for example)

- time until expiration (if valid)
- expired days ago (if expired)

SNMP TRAPS

SECURITY-RELATED MONITORING
WITH ZABBIX

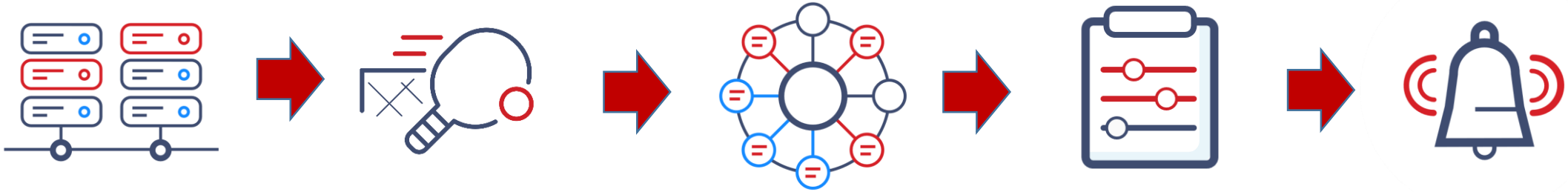
CAN WE MONITOR SNMP TRAPS ?

Yes, using Zabbix SNMP trapper item

What to monitor ?

- Administrative logins
- Ports status up/down
- New devices (MAC security)
- Thresholds reached (Network attacks)
- Any other security related checks

HOW SNMP TRAPS WORK ?

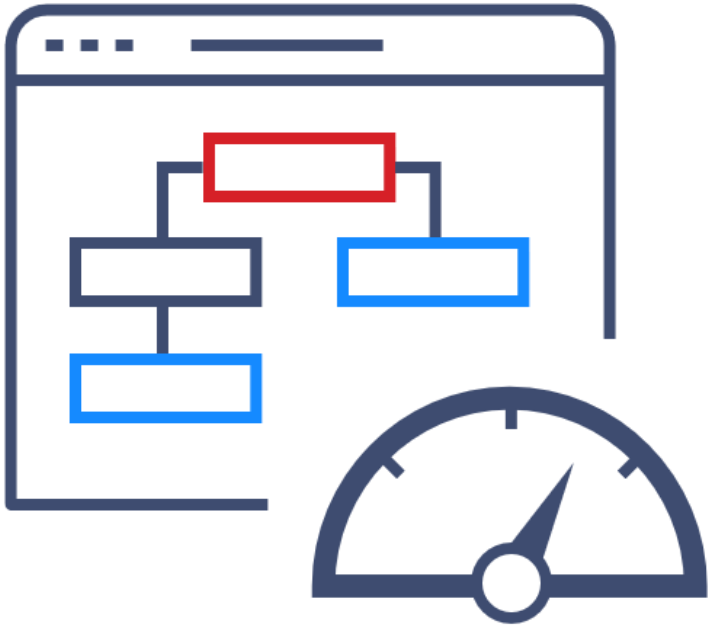


SENSOR MONITORING

SECURITY-RELATED MONITORING
WITH ZABBIX

MONITOR YOUR ENVIRONMENT WITH ZABBIX

- Temperature sensors
- Smoke sensors
- Humidity sensors
- Door sensors
- Motion detection sensors



LOG FILE MONITORING

SECURITY-RELATED MONITORING
WITH ZABBIX

WHY DO YOU NEED TO MONITOR LOGS ?

A lot of security related information can be found in log files

For example

- Unsuccessful logins
- Successful logins !
- Elevation of privileges

LOG FILE MONITORING

Log files can be parsed to find important information

Dependent items can be created from log items

Triggers can be created to alert about serious security issues

Information from log files can be extracted and used in trigger names and tags

```
Oct  3 08:18:05 zabbix42 sshd[8103]: Failed password for root from 127.0.0.1 port 52684 ssh2
Oct  3 08:18:05 zabbix42 sshd[8103]: Connection closed by 127.0.0.1 port 52684 [preauth]
Oct  3 08:18:05 zabbix42 sshd[8103]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=localhost  user=root
Oct  3 08:18:12 zabbix42 sshd[8108]: Failed password for kaspars from 127.0.0.1 port 52694 ssh2
Oct  3 08:18:13 zabbix42 sshd[8108]: Failed password for kaspars from 127.0.0.1 port 52694 ssh2
Oct  3 08:18:13 zabbix42 sshd[8108]: Connection closed by 127.0.0.1 port 52694 [preauth]
Oct  3 08:20:11 zabbix42 sshd[8337]: Failed password for root from 127.0.0.1 port 52926 ssh2
Oct  3 08:20:11 zabbix42 sshd[8337]: Failed password for root from 127.0.0.1 port 52926 ssh2
Oct  3 08:20:11 zabbix42 sshd[8337]: Connection closed by 127.0.0.1 port 52926 [preauth]
Oct  3 08:20:22 zabbix42 sshd[8369]: Failed password for kaspars from 127.0.0.1 port 52960 ssh2
Oct  3 08:20:22 zabbix42 sshd[8369]: Failed password for kaspars from 127.0.0.1 port 52960 ssh2
Oct  3 08:20:22 zabbix42 sshd[8369]: Connection closed by 127.0.0.1 port 52960 [preauth]
Oct  3 08:20:55 zabbix42 sshd[8437]: Accepted password for kaspars from 127.0.0.1 port 53034 ssh2
Oct  3 08:20:55 zabbix42 sshd[8437]: pam_unix(sshd:session): session opened for user kaspars by (uid=0)
Oct  3 08:20:57 zabbix42 sudo: pam_unix(sudo:auth): authentication failure; logname=kaspars uid=1000 euid=0 tty=/dev/pts/3 ruser=kaspars rhost=  user=kaspars
Oct  3 08:21:03 zabbix42 sudo: kaspars : user NOT in sudoers ; TTY=pts/3 ; PWD=/home/kaspars ; USER=root ; COMMAND=/bin/su
Oct  3 08:21:05 zabbix42 sudo: pam_unix(sudo:auth): authentication failure; logname=kaspars uid=1000 euid=0 tty=/dev/pts/3 ruser=kaspars rhost=  user=kaspars
Oct  3 08:21:09 zabbix42 sudo: kaspars : user NOT in sudoers ; TTY=pts/3 ; PWD=/home/kaspars ; USER=root ; COMMAND=/bin/su
Oct  3 08:23:09 zabbix42 sudo: kaspars : user NOT in sudoers ; TTY=pts/3 ; PWD=/home/kaspars ; USER=root ; COMMAND=/bin/su
Oct  3 08:26:59 zabbix42 sudo: kaspars : user NOT in sudoers ; TTY=pts/3 ; PWD=/home/kaspars ; USER=root ; COMMAND=/bin/su
```

MASTER LOG ITEM

Master item contains all important log information

Item | Preprocessing

* Name	secure log
Type	Zabbix agent (active)
* Key	log["/var/log/secure"] Select
Type of information	Log
* Update interval	1s
* History storage period	Do not keep history / Storage period 90d
Log time format	
New application	security
Applications	-None-

DEPENDENT LOG ITEMS

Dependent items extract information from the main log

Item | Preprocessing

* Name: sudo failed
Type: Dependent item
* Key: sudo.fail [Select]
* Master item: Zabbix42: secure log [Select]
Type of information: Log
* History storage period: Do not keep history | Storage period | 90d

Item | Preprocessing

Preprocessing steps

	Name	Parameters		Custom on fail
1:	Regular expression	(.+user NOT in sudoers.+)	\1	✓

Custom on fail: Discard value | Set value to | Set error to

Add

Update | Clone | Check now | Clear history and trends | Delete | Cancel

DEPENDENT LOG ITEMS

Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type
•••	secure log: login failed	Triggers 2	root.login.failed		90d		Dependent item
•••	secure log: login suceeded		root.login.success		90d		Dependent item
•••	secure log		log["/var/log/secure"]	1s	90d		Zabbix agent (active)
•••	secure log: sudo failed	Triggers 1	sudo.fail		90d		Dependent item

Host	Name ▲	Interval	History	Trends	Type	Last check	Last value
Zabbix42	- other - (4 Items)						
	login failed root.login.failed		90d		Dependent item	2019-10-03 08:20:23	Failed password for kaspar...
	login suceeded root.login.success		90d		Dependent item	2019-10-03 08:20:55	Accepted password for kas...
	secure log log["/var/log/secure"]	1s	90d		Zabbix agent ...	2019-10-03 08:43:10	Oct 3 08:43:09 zabbix42 su...
	sudo failed sudo.fail		90d		Dependent item	2019-10-03 08:43:10	Oct 3 08:43:09 zabbix42 su...

GATHERING USEFULL INFORMATION

Information can be extracted from the logs using function

regsub (<pattern>,<output>)

Extracted information can be used in

- Trigger names
- Trigger tags

LOG TRIGGERS

Log line:

sudo: **kaspars** : user NOT in sudoers ; TTY=pts/3 ; PWD=/home/kaspars ; USER=zabbix ; COMMAND**=/bin/ping**

Examples to extract user and executed command

{{ITEM.VALUE}.regsub("sudo: (.+) :", user: \1)}

{{ITEM.VALUE}.regsub("COMMAND=(.+)", command: \1)}

LOG BASED TRIGGER EXAMPLE

Trigger | Tags | Dependencies

* Name: Sudo attempt without permissions {{ITEM.VALUE}.regsub("sudo: (.+) :", user: \1)} {{

Severity: Not classified | Information | Warning | Average | **High** | Disaster

* Expression: {Zabbix42:sudo.fail.str(user NOT in sudoers)}=1 [Add]

Expression constructor

OK event generation: Expression | Recovery expression | **None**

PROBLEM event generation mode: Single | **Multiple**

Allow manual close: ☑

URL:

USE TAGS TO FILTER INFORMATON!

Trigger | Tags | Dependencies

Trigger tags | Inherited and trigger tags

Name	Value	Action
USER	{{ITEM.VALUE}.regsub("sudo: (.+) :",\1	Remove

Add

Update | Clone | Delete | Cancel

Time ▼		Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
14:00:46		High		PROBLEM		Zabbix42	sudo attempt by : user: kaspars command: /bin/ping	23s	No		USER: kaspars
14:00											

CAN WE MONITOR WINDOWS LOGS ?

Yes, a special key **eventlog** can be used

You can filter event logs by

- Source (Security, System etc…)
- Severity (*"Warning"*, *"Error"*, *"Critical"* etc…)
- Eventid

  4625 – Logon Failure

  4740 - A user account was locked out)

ZABBIX INTEGRATIONS

SECURITY-RELATED MONITORING
WITH ZABBIX

ZABBIX INTEGRATION

ZABBIX

PRODUCT | SOLUTIONS | SERVICES & SUPPORT | TRAINING | PARTNERS | COMMUNITY | ABOUT US | DOWNLOAD

All Categories | Official Templates | Agents | API | Applications | AWS | Backups | Business KPI | Caching | Clouds

Containers | CRM | DevOps | Databases | ERP | HA & Clusters | Helpdesks | Infrastructure | IoT | Java | Logfiles

Mail | Message brokers | Mobile | Monitoring systems | Network | Notifications & Alerting | Orchestration | Operation Systems

Printers | Search Engines | Security | Services | Servers | Storage | Telephony | Virtual Machines | Visualization | Web

Security monitoring

Antivirus

Barracuda

Blue Coat

Check Point

F5 Networks

Fortinet

Firewalls

Kaspersky

SELinux

HTTPS

OpenVPN

THANK YOU!

Kaspars Mednis
ZABBIX Technical Support Engineer

ZABBIX SUMMIT '19

zabbix