OpenKAT – Indemnification statements (‘Vrijwaringsverklaring’)

What are indemnity statements?

Indemnity statements are a way of agreement between two parties that specific activities are allowed to be performed, that would otherwise be illegal. This agreement is often made in environments that require to be tested, such as penetration testing. The party performing the ‘illegal’ activities will not be sued or held responsible for damages if they adhere to the agreements made.

Why are indemnity statements relevant for security assessments?

By Dutch law is it unlawful to perform scanning activities or perform security tests without authorisation by the owner of the system[1][2]. In order to perform these activities in a lawful way it is required to agree to certain conditions between the owner of the system and the responsible party that will perform the activities. These conditions need to be written down and signed by both parties. Next to a pentest waiver, usually the following terms are also discussed:

- Time of day for performing testing activities
- Depth of activities, is only (automated) scanning allowed, or is it also allowed to perform manual checks. In case a (potential) vulnerability has been identified, is it allowed to further determine the impact of this vulnerability?
- Is it allowed to perform more destructive testing activities?
- Who is/are the owner(s) of the environment?

After the terms are agreed, both parties sign a penetration testing waiver and determine when the testing activities will take place. If anything unexpected happens during the testing activities, the security tester is waived from any liabilities (as long as it was not intentional).

External hosting providers

A very common scenario is for organisations to rent servers from third parties, like hosting providers or cloud solutions like like AWS and Azure. In these cases it is important to have permission from both the third party as well as the organisation that wishes to be tested for vulnerabilities. The reason for this is that the third party is the owner of the actual hardware against which the testing activities are performed. These activities might impact their other business activities, so it’s mandatory for them to be aware of this beforehand.

Regulations outside of the Netherlands

The scenario described above is applicable to the Netherlands, most countries have a similar law, to allow security testing to be performed without performing illegal activities. Before

---

[1] “Is een poortscan strafbaar? – Ius Mentis”, May 2nd 2014, https://blog.iusmentis.com/2014/05/02/een-portscan-strafbaar/

[2] “De security scan als strafbare poging tot computervredebreuk – Ius Mentis”, December 22nd 2014, https://blog.iusmentis.com/2014/12/22/de-security-scan-als-strafbare-poging-tot-computervredebreuk/

attempting any scanning or testing activities it is important to verify your local rules and regulations with regard to penetration testing and scanning of systems and ensure that you have the required waivers before starting.

Indemnity levels in OpenKAT

Indemnity levels can be used to indicate what level is accepted or suitable for gathering intelligence and/or performing security assessments. The following levels are defined:

- **L0** – Do not touch: no permissions are given to perform testing activities, or the system is very sensitive (embedded IoT systems)
- **L1** – Passive data collection using public sources on the internet. The goals is to gain insights in what public information could be a risk.
- **L2** – Perform targeted scans with limited intrusion. These scans are not designed to crash services and may or may not be detectable through monitoring.
- **L3** – Perform more intrusive scanning activities: connect to services to find out versions, log in with default credentials, automated testing of vulnerabilities, guessing of URLS and intensive crawling of web pages.
- **L4** – Intensive scanning (aka. 'Exploding-Cats-Mode'): where anything and everything is allowed, generally not recommended for production systems and networks, as this might break systems or services.

The scanning levels are not related to providing user credentials to perform the scanning activities.

Scanning rules

During the configuration of a scan the OpenKAT interface asks what scan level can be applied to the added scope. This is called a user declaration and means that the user has specified that a specific IP is allowed to be scanned with the specified level. These scan levels can differ between the added hosts.

When initiating a scan OpenKAT will determine, based on the given user declaration, whether or not to scan certain objects and with what scanning level (intensity) this will be done. These objects can be anything from IP-addresses, subnets, host names, ports, services/protocols and more. Each of these objects is given a specific scanning level in relation to other objects (either given or received through inheritance). This will help determine whether an object should (not) be scanned.

These scanning rules can be defined in two ways:

- Object A **gives** a maximum scanning level **to** object B.
- Object A **receives** a maximum scanning level **from** object B (through inheritance).

Example:

Scanning activities generally start by entering one or more IP addresses. Two common scenarios when scanning an IP address are:

- Scan a single IP address.
- Scan a subnet with multiple IP addresses.

Both the single IP address and the subnet are seen as objects within OpenKAT. The following two relations are defined for these objects:

- When scanning a single IP address you do not want to scan other IP addresses in a subnet.
- When scanning a single IP address you could inherit a scanning level from the subnet this IP address in.

In the first example you only wish to scan the single IP address, but not other IP addresses in a subnet. In the second example a subnet is scanned which contains multiple IP addresses. In a visual representation this looks as follows:

- IP address —> subnet: Do not scan the subnet.
- Subnet —> IP address: Scan the IP.

For both objects these relations are defined as follows within OpenKAT. If you want to scan an IP-address (`IPAddressV4` or `IPAddressV6`) a relation is present to the subnet object (`IPV4Netblock` or `IPV6Netblock`). In the code the following relations are defined:

```
- IPAddressV4 -> IPV4NetBlock, max_issue_level = 0, max_inherit_level=4
- IPAddressV6 -> IPV6NetBlock, max_issue_level = 0, max_inherit_level=4
```

This means that for each IP address object it will **give**, by default, level 0 (“Do not scan”) as the maximum level to the subnet (NetBlock). You do not want to initiate scanning activities on other IP addresses, as you might not have indemnity statements for these IP addresses, or they don’t even belong to your environment.

In return the IP address will **receive** a maximum scan level from the netblock of level 4 (“intensive scanning”). When entering a subnet of IP addresses, you are supposed to have taken care of all required indemnity statements, thus scanning all IP addresses at level 4 should not be an issue.

There are default configurations. If a user redefines these levels for a specific object this will always prioritise over the default configuration. If multiple users configured different scan levels for an object, the maximum inherited level is chosen as the scan level for the target.

Why is this done?

In some situations you want to be able to define a relation in two ways. IP addresses and subnets are an excellent way to show why this is relevant. In some cases only a single IP address is within your scanning scope, thus you’d only enter the IP address you wish to scan. It would be strange if automated scans would automatically scan all IP addresses that are in this subnet. This is what the `max_issue_level` is used for, it says: only scan this IP address and nothing else I have a relation to.

In other situations you’d wish to scan a whole subnet, which means all IP addresses that belong to this subnet. Since a subnet exists of multiple IP addresses you’d need to define the scan level for all IPs in the subnet. By way of inheritance this can be done automatically. After entering an IP subnet, it will automatically distribute scanning rights to all IPs in this subnet. In the example of:

```
- IPAddressV4 -> IPV4NetBlock, max_issue_level = 0, max_inherit_level=4
- IPAddressV6 -> IPV6NetBlock, max_issue_level = 0, max_inherit_level=4
```

You see that the `max_inherit_level` is specified as L4, which means that each IP address will **receive** a maximum scan level L4 from the subnet. The actual scan level it inherits is always limited to what the giving object itself has.

This is also specified from the subnet perspective as shown below:

- `IPV4NetBlock -> IPAddressV4, max_issue_level = 4, max_inherit_level = none`
- `IPV6NetBlock -> IPAddressV6, max_issue_level = 4, max_inherit_level = none`

Here it specifies that the subnet gives a max_issue_level of L4 to IP addresses. The `max_inherit_level` is not specified, meaning that a subnet will receive no scan level from IP addresses.

Inheritance table

Below gives an overview of all inheritance rights as defined in version 1.8 of OpenKAT. This table is used to describe the current situation.

How do you interpret this table?

- The first column (OOI type) defines a broad category under which type the object of interest falls.
- The second column is the OOI itself (object A)
- The third column is the OOI to which is referenced from the second column OOI (object B)
- The fourth column is the maximum issued scan level (`max_issue_scan_level`), which means that: as Object A I **give** the following maximum level to Object B.
- The fifth column is the maximum inherited scan level (`max_inherit_scan_level`), which states that: Object A will **receive** (inherit) the following maximum level from Object B.

OOI type	OOI (Object "A")	OOI (Object "B")	Maximum issued scan level	Maximum inherited scan level
DNS	DNSRecord	Hostname	0	2
	DNSNSRecord	Hostname	1	0
	DNSZone	Hostname	2	1
	DNSZone	DNSZone	0	1
	Hostname	DNSZone	1	2
	ResolvedHostname	Hostname	0	4
	ResolvedHostname	IPAddress	4	0
Certificates	X509Certificate	X509Certificate	1	0
	SubjectAlternativeNameHostname	Hostname	1	0
EmailSecurity	DNSSPFRecord	DNSTXTRecord	-	1
	DNSSPFMechanism	DNSSPFRecord	-	1
Network	IPAddressV4	IPV4NetBlock	0	4
	IPAddressV6	IPV6NetBlock	0	4
	IPPort	IPAddress	0	4
	IPV6NetBlock	IPAddressV6	4	-
	IPV4NetBlock	IPAddressV4	4	-
Web	Website	IPService	0	4
	Website	Hostname	-	4
	Website	X509Certificate	1	-
	HostnameHTTPURL	Hostname	2	4
	IPAddressHTTPURL	IPAddress	1	4
	HTTPResource	Website	0	4

	HTTPResource	WebURL	1	4
	HTTPHeader	HTTPResource	0	4
	URL	WebURL	2	-
	HTTPHeaderURL	HTTPHeader	0	1
	HTTPHeaderURL	URL	1	0
	HTTHeaderHostname	HTTPHeader	0	1
	HTTPHeaderHostname	Hostname	1	0
	ImageMetadata	HTTPResource	0	4
Service	IPService	IPPort	0	4
	IPService	Service	1	0
Software	SoftwareInstance	OOI	0	1
	SoftwareInstance	Software	1	0

Receiving an indemnity level

As a user you will receive a certain rights level from the administrator of your KAT environment. As a user you can either accept the given rights, or decide to only accept a lower level. This will be the level for which you can define indemnations (‘vrijwaringen’) on objects. If you are given L4, but are not comfortable with applying this level to (new) objects, you can decide to only accept a lower level, such as L3. This can be useful in situations where you are a junior, or are unfamiliar with either the technology or environment and do not wish to break things by accident. Whether you accept the given rights from the administrator, or decide to only accept a lower level, all objects that you’ll set, will only receive this maximum comfort level you have accepted. The scanner will know which scans to run with which rights.

How to determine the scan level

When choosing a scan level it is important to keep the following things in mind. When you are unsure about the answer, always go for a conservative approach and lower your scan level. If the exact

- Am I allowed to scan this object? Eg. IP, hostname, port, service, etc.
    - If no – Lower your scan level.
- Is the object within scope?
    - If no – Lower your scan level.
- Is this a sensitive or critical object?
    - If yes – Lower your scan level.
- Is the object running on system or environment owned by me?
    - If no – Lower your scan level and/or get permission to scan the target.
- Are other people and/or organisations using this object as well?
    - If yes – Lower your scan level.
- Does the object make connections to other (external) systems, services or hosts?
    - If yes – Lower your scan level (for inherited objects).