Icinga 2.12.5 + 2.11.10: Security Releases

by **Alexander Klimov** | Jul 15, 2021

Today we are releasing Icinga 2.12.5 and 2.11.10, including two security fixes that may lead to privilege escalation for authenticated API users. **Depending on your setup, manual intervention beyond installing the new versions may be required, so please read the section on security carefully.** Other improvements in these releases include several bugfixes related to downtimes, downtime notifications, and more reliable connection handling.

One of the security fixes affects an interface used by Icinga Director. If you use it to configure Icinga agents, you have to update it to the **recently released version 1.8.1** for it to continue working.

Security

Both releases fix two issues related to information being exposed through the API inadvertently. You are affected if your cluster has API users configured that have permissions to query certain object types but are not trusted to know the information that was exposed.

- **CVE-2021-32739: PKI ticket salt exposed via API, potentially allowing privilege escalation for authenticated API users** (**Advisory**)API responses for `ApiListener` objects contained the attribute `ticket_salt` which is used to generate and verify PKI tickets. These tickets are bound to a single common name (CN) and allow requesting certificates with that particular CN signed by the Icinga CA. Knowing the ticket salt allows to create valid tickets for arbitrary CNs. Certificates obtained with this ticket in turn allow authenticating as API users (if they have `client_cn` set) or as any Icinga 2 node within the cluster.You are affected if your cluster has API users with the `objects/query/ApiListener` permission, possibly via some wildcard permission like `objects/query/*`, and you do not fully trust these users. You can test if this information is exposed by querying the API at `https://icinga-master:5665/v1/objects/apilisteners` using the credentials of a suspected API user.To mitigate this issue, you have to upgrade Icinga 2 to one of the fixed versions on nodes that have the ticket salt configured (typically only one or both masters that sign certificate requests) and replace the ticket salt with a new one if you suspect it was exposed. To do so, **follow the instructions at the end of this post**. Note that if have stored tickets in other systems like any automation tools, these have to be regenerated with the new ticket salt as well.Changing the ticket salt only prevents obtaining new certificates. If you suspect that an attacker might already have obtained a malicious certificate, you should replace the entire Icinga CA. To do so, **follow the instructions below**.

- **CVE-2021-32743: Passwords used to access external services inadvertently exposed through API** (**Advisory**)API responses for `IdoMysqlConnection`, `IdoPgsqlConnection`, `IcingaDB` and `ElasticsearchWriter` objects contained the attribute `password` exposing the password used to authenticate against these services.You are affected if your cluster has API users with the one of the following permissions that you do not trust knowing these credentials and if objects of the corresponding type exist:

  ```
  objects/query/IdoMysqlConnection
  objects/query/IdoPgsqlConnection
  objects/query/IcingaDB
  objects/query/ElasticsearchWriter
  ```

  These permissions may also be granted by some wildcard like `objects/query/*`. You can test if this information is exposed by querying the API at the following URLs using the credentials of a suspected API user:

  ```
  https://icinga-master:5665/v1/objects/idomysqlconnections
  https://icinga-master:5665/v1/objects/idopgsqlconnections
  https://icinga-master:5665/v1/objects/icingadbs
  https://icinga-master:5665/v1/objects/elasticsearchwriters
  ```

  An attacker who obtained these credentials can impersonate Icinga to these services and add, modify and delete information there. If credentials with more permissions are in use, this increases the impact accordingly.

  To mitigate the issue, you have to upgrade Icinga 2 to one of the fixed version on the nodes making use of these features (usually this is only the case on master nodes) and change these passwords afterwards if you suspect they could have been exposed to untrusted entities.

- Update OpenSSL version bundled on Windows to 1.1.1k (2.12.5: **#8885** / 2.11.10: **#8888**)

The first two issues can be worked around without upgrading by ensuring that none of these permissions mentioned above is given to affected API users. This can be done for example by only allowing to query fewer object types or by adding a filter to existing permissions, see the following two examples. However, the other mitigation steps still have to be applied.

```
object ApiUser "restrictive-example" {
  password = "secret"
  permissions = [ "objects/query/Host", "objects/query/Service" ]
}

object ApiUser "filter-example" {
  password = "secret"
  permissions = [ {
    permission = "objects/query/*"
    filter = {{ ! ["ApiListener", "IdoMysqlConnection", "IdoPgsqlConnection", "IcingaDB",
"ElasticsearchWriter"].contains(obj.type) }}
  } ]
}
```

Bugfixes

- Don't send Downtime end notification if Downtime hasn't started (2.12.5: **#8877** / 2.11.10: **#8878**)
- Don't let a failed Downtime creation block the others (2.12.5: **#8863** / 2.11.10: **#8871**)
- Support Downtimes and Comments for checkables with long names (2.12.5: **#8864** / 2.11.10: **#8870**)
- Trigger fixed downtimes immediately if the current time matches (instead of waiting for the timer) (2.12.5: **#8889** / 2.11.10: **#8891**)
- Add configurable timeout for full connection handshake (2.12.5: **#8866** / 2.11.10: **#8872**)

Enhancements

- Replace existing Downtimes on ScheduledDowntime change (2.12.5: **#8879** / 2.11.10: **#8880**)
- Improve crashlog (2.12.5: **#8865** / 2.11.10: **#8869**)

Appendix

The following sections contain more detailed instructions on manual steps that may have to be taken after installing this security update.

Changing the ticket salt

01. Generate a fresh ticket salt using the following OpenSSL command:

```
openssl rand -hex 16
```

02. Identify all Icinga nodes that sign certificates using tickets. You can identify those by the presence of a value for `TicketSalt` in `/etc/icinga2/constants.conf`. Typically, this is either one or both master nodes.

03. On all these nodes, change the value of `TicketSalt` in `/etc/icinga2/constants.conf` to the one generated in the first step. If there are multiple nodes, they all have to use the same value.

04. Restart the `icinga2` service.

Replacing the Icinga CA

Before beginning with this task, please be aware that this is task involes issuing and deploying new certificates for every single node in the cluster. Parts of the cluster using the new CA won't communicate with parts using the old CA and vice versa. So while performing this maintenance task, your monitoring will be interrupted.

01. On the primary certificate signing node (typically the primary master), remove the existing CA and generate a new one:

```
systemctl stop icinga2.service
mv /var/lib/icinga2/ca /var/lib/icinga2/ca.old
icinga2 pki new-ca
cp /var/lib/icinga2/ca/ca.crt /var/lib/icinga2/certs/ca.crt
```

02. Next, issue a new certificate for this node using the new CA (replace `icinga-master-1` with the actual name of that node):

```
icinga2 pki new-cert \
  --cn icinga-master-1 \
  --key /var/lib/icinga2/certs/icinga-master-1.key \
  --csr /var/lib/icinga2/certs/icinga-master-1.csr \
  --cert /var/lib/icinga2/certs/icinga-master-1.crt

icinga2 pki sign-csr \
  --csr /var/lib/icinga2/certs/icinga-master-1.csr \
  --cert /var/lib/icinga2/certs/icinga-master-1.crt

systemctl start icinga2.service
```

03. Now all other node certificates have to be replaced. Perform the following steps for each node in your cluster starting at the top of the hierarchy. So first do it on the second master if one exists, then on the satellites (also from top to bottom if multiple layers exist) and finally on the agents.

    01. Obtain a new PKI ticket by executing this command on your master (replacing `mynode42` with the actual name of the secondary master, satellite, or agent):

    ```
    icinga2 pki ticket --cn mynode42
    ```

    02. Then connect to that node and issue a new certificate using these commands.

    ```
    systemctl stop icinga2.service
    icinga2 pki new-cert --cn mynode42 \
      --key /var/lib/icinga2/certs/mynode42.key \
      --csr /var/lib/icinga2/certs/mynode42.csr \
      --cert /var/lib/icinga2/certs/mynode42.crt
    ```

    ```
    Stop-Service -Name icinga2
    cd "${Env:ProgramData}\icinga2\var\lib\icinga2"
    $Env:Path += ";${Env:ProgramFiles}\ICINGA2\sbin"
    & icinga2.exe pki new-cert --cn mynode42 `
      --key certs/mynode42.key `
      --csr certs/mynode42.csr `
      --cert certs/mynode42.crt
    ```

    03. Fetch the certificate of the parent node. Replace `icinga-master-1` / `192.0.2.56` / `5665` with the name / address / port of the parent node. The parent may also be a satellite if the current node is an agent below a satellite zone. You have to ensure that this certificate actually matches the parent instance's certificate in order to avoid man-in-the-middle attacks!

    ```
    icinga2 pki save-cert \
      --trustedcert /var/lib/icinga2/certs/icinga-master-1.crt \
      --host 192.0.2.56 \
      --port 5665
    ```

    ```
    & icinga2.exe pki save-cert `
      --trustedcert certs/icinga-master-1.crt `
      --host 192.0.2.56 `
      --port 5665
    ```

    04. Issue a new certificate. Additionally, replace `0011...ff` with the ticket obtained previously.

    ```
    icinga2 pki request \
      --key /var/lib/icinga2/certs/mynode42.key \
      --cert /var/lib/icinga2/certs/mynode42.crt \
      --ca /var/lib/icinga2/certs/ca.crt \
      --trustedcert /var/lib/icinga2/certs/icinga-master-1.crt \
      --host 192.0.2.56 \
      --port 5665 \
      --ticket 00112233445566778899aabbccddeeff
    ```

    ```
    & icinga2.exe pki request `
      --key certs/mynode42.key `
    ```

```
    --cert certs/mynode42.crt `
    --ca certs/ca.crt `
    --trustedcert certs/icinga-master-1.crt `
    --host 192.0.2.56 `
    --port 5665 `
    --ticket 00112233445566778899aabbccddeeff
```

05. **Only if this node is supposed to sign certificate requests** (you can identify those by checking whether the file `/var/lib/icinga2/ca/ca.key` exists), copy the new `/var/lib/icinga2/ca` directory from the primary certificate signing node to this one.

06. Finally, start Icinga:

```
systemctl start icinga2.service
```

```
Start-Service -Name icinga2
```

You May Also Like…

Icinga Camp Berlin 2023

It's time to spread some monitoring love! We're super happy to announce our next Icinga Camp in Berlin, on May 17th, 2023

Call for Papers

Register Now

Releasing Icinga Web v2.9.8, v2.10.4 and v2.11.3

Dec 14, 2022

Today we're announcing the general availability of Icinga Web v2.9.8, v2.10.4 and v2.11.3. These are accompanied by...

Updates for Icinga Web Graphite and GenericTTS Integrations

Nov 30, 2022

Today we're announcing the general availability of Icinga Web GenericTTS Integration v2.1.0 and Icinga Web Graphite...

Releasing Icinga DB v1.1

Nov 14, 2022

We are happy to announce the release of Icinga DB version 1.1 today. The main feature of this release is the addition...

Subscribe to our Newsletter

A monthly digest of the latest Icinga news, releases, articles and community topics.

Email Address

Subscribe