ATIX

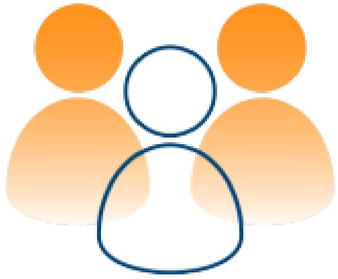
The Linux & Open Source Company

CVE scanning of Foreman hosts

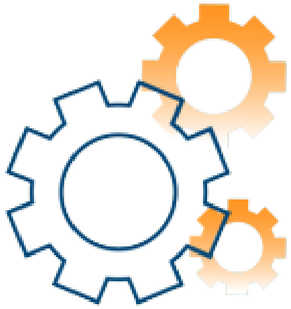
Config Management Camp 2024 – Bernhard Suttner

ATIX

The Linux & Open Source Company



Consulting



Engineering

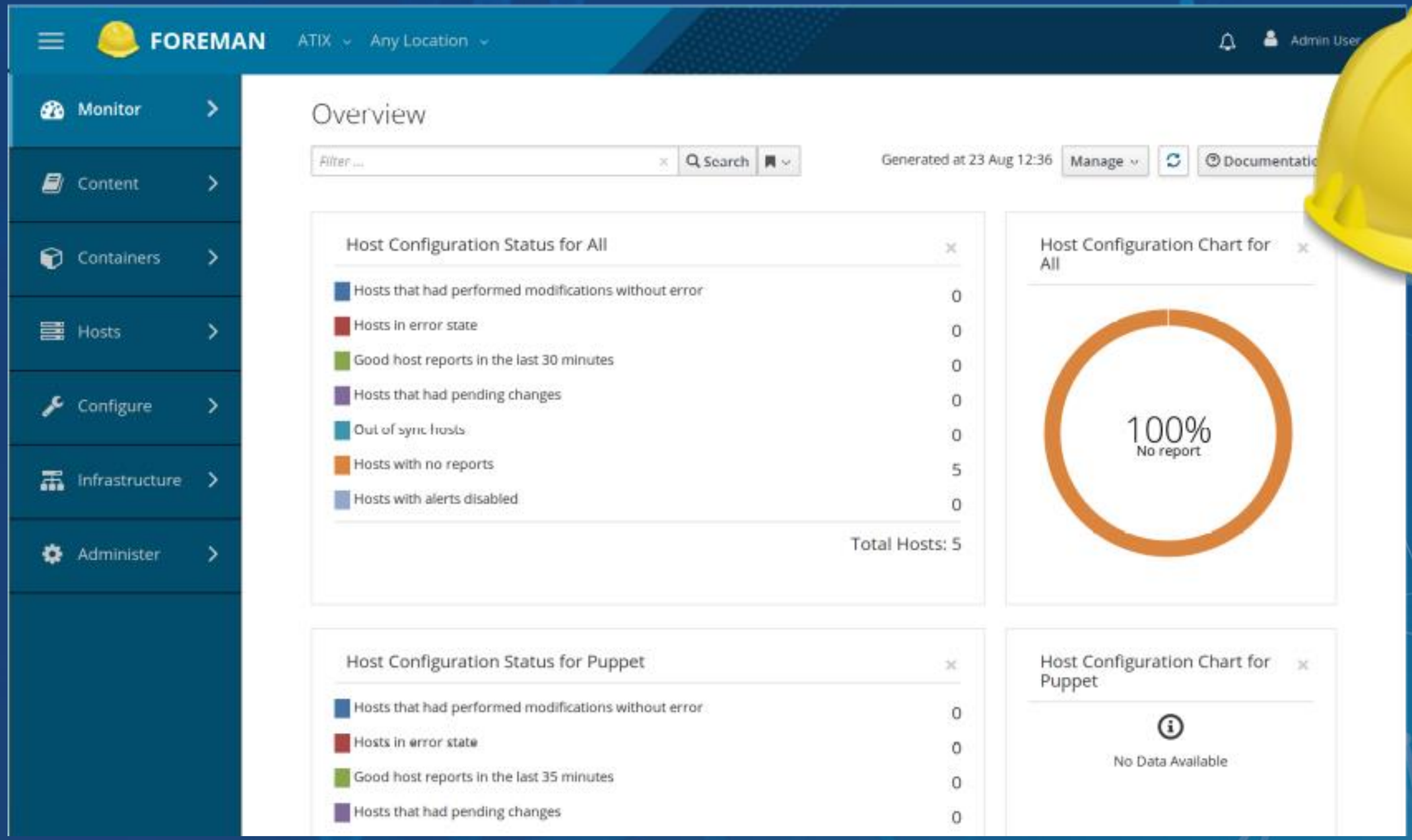
Support

Training

AlmaLinux · amazon web services · ANSIBLE · Azure · CentOS · debian · docker · FOREMAN · git · Google Cloud Platform

Istio · kafka · KATELLO · kubernetes · KVM · ORACLE Linux · oVirt · PROXMOX · puppet · RANCHER

Red Hat · RED HAT OPENSHIFT · Rocky Linux · SALTSTACK · SUSE · Terraform · Ubuntu · vmware vsphere · Windows


FOREMAN
ATIX
Any Location
Admin User
Monitor
Content
Containers
Hosts
Configure
Infrastructure
Administer
Overview
Filter ...
Search
Generated at 23 Aug 12:36
Manage
Documentatio
Host Configuration Status for All
Hosts that had performed modifications without error 0
Hosts in error state 0
Good host reports in the last 30 minutes 0
Hosts that had pending changes 0
Out of sync hosts 0
Hosts with no reports 5
Hosts with alerts disabled 0
Total Hosts: 5
Host Configuration Chart for All
100%
No report
Host Configuration Status for Puppet
Hosts that had performed modifications without error 0
Hosts in error state 0
Good host reports in the last 35 minutes 0
Hosts that had pending changes 0
Host Configuration Chart for Puppet
No Data Available

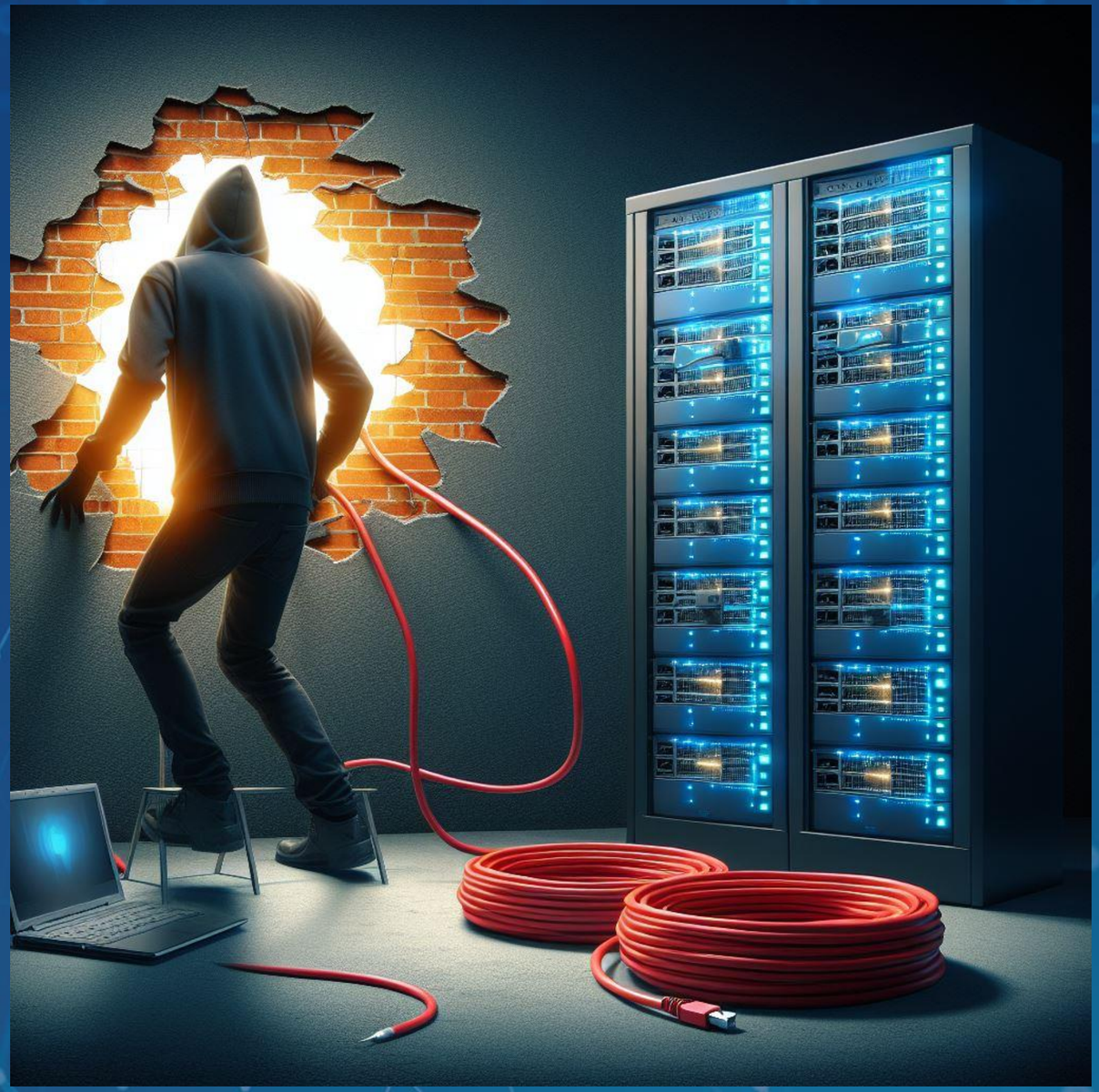
Target

Expectation

- Hardend operating system
- Well configured services
- Only necessary users and services have access
- Only necessary applications are installed
- Installed packages are up to date
- All security issues are closed – in time

Reality

…. you know what I mean.



Current situation

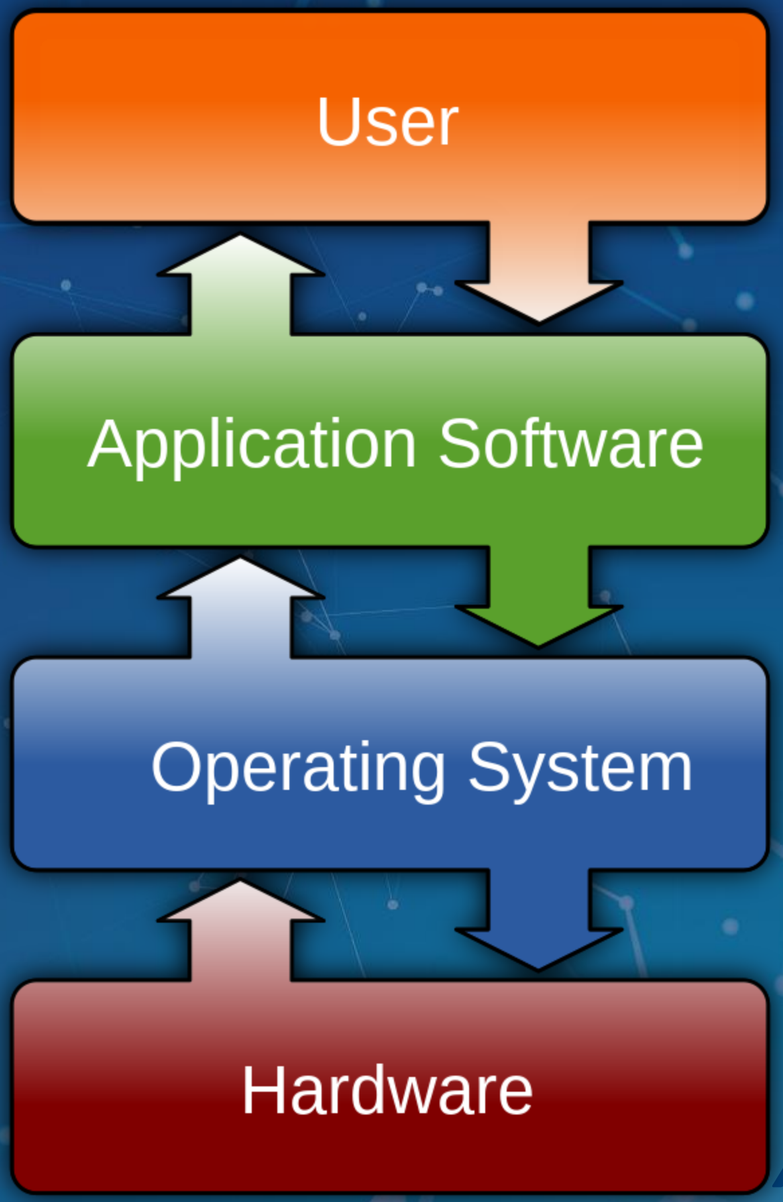
- Katello with Errata support
  - Recognize security issues in repositories
  - Install fixes to close the security gap

- Foreman OpenSCAP
  - Rules how the system – especially the operating system should look like

Software packages

- Hopefully: delivered by Katello.
- Maybe: installed by, 'tar archive', 'rpm –ivh', 'git clone', 'scp', …

- Delivered by the OS vendor
- Delivered by Katello
- Errata exists. Update possible via Katello



Secure application software

- Open questions:
  - Delivery and installation method?
  - Maintenance, Updates?
- Possible improvements
  - Package it with RPM / DEB
  - Create a repositoriy
  - Deliver it with Katello (yes!)
  - Upgrade it
- What else can we do?
  - Scan them if they have security issues.

Vulnerability* scanners

grype

aqua trivy

- Targets:

  Filesystem, Docker container image

  git, virtual machine image, AWS, Kubernetes

- How they work
  - Get vulnerability database
  - Generate SBOM for given source
  - Scan the SBOM

(* trivy: vuln,secret,misconfig)

Goal

Foreman "integration" light

- Installation of the security scanner
- Run the scanner once
- Scheduled job
- Collect results

How: Remote Exeuction jobs

Installation

```
<%#
name: Install security scanners
snippet: false
template_inputs:
- name: trivy_version
  default: 0.48.1
- name: grype_version
  default: 0.73.5
model: JobTemplate
job_category: Security
provider_type: script
kind: job_template
%>

# Install security scanners from github

<%
trivy_version = input('trivy_version')
grype_version = input('grype_version')

trivy_url = "https://github.com/aquasecurity/trivy/releases/download/v#{trivy_version}/trivy_#{trivy_version}_Linux-64bit.rpm"
grype_url = "https://github.com/anchore/grype/releases/download/v#{grype_version}/grype_#{grype_version}_linux_amd64.rpm"
-%>

yum install --assumeyes <%= trivy_url %> <%= grype_url %>
```

Installation / results

Jobs > Install security scanners with inputs trivy_version="0.48.1" grype_version="0.73.5"

Create Report | Rerun | Rerun failed | Job Task | Cancel Job | Abort Job

Overview | Preview templates

Providers and templates

Install security scanners - Script

van-holbach.sandbox.dev.atix

```
# Install security scanners from github


yum install --assumeyes https://github.com/aquasecurity/trivy/releases/download/v0.48.1/trivy_0.48.1_Linux-64bit.rpm https://github.com/anchore/grype/releases/download/v0.73.5/grype_0.73.5_linux_amd64.rpm
```

User input	Value
trivy_version	0.48.1
grype_version	0.73.5

Run

```
<%#
name: Run security scanner
template_inputs:
- name: target
  options: "filesystem\r\ndocker"
- name: path
- name: scanner
  options: "trivy\r\ngrype"
  default: trivy
model: JobTemplate
job_category: Security
provider_type: script
kind: job_template
%>
<%
scanner = input('scanner')
target = input('target').to_sym
path = input('path')
options = input('options')

scanners = {
    trivy: {
        docker: 'image',
        filesystem: 'filesystem'
    },
    grype: {
        docker: 'docker',
        filesystem: 'dir'
    }
}

if scanner == 'trivy'
    cmd = "#{scanners[:trivy][target]} #{path}"
    options += " --scanners vuln"
elsif scanner == 'grype'
    cmd = "#{scanners[:grype][target]}:#{path}"
    options += " --quiet"
end
exec_command = "#{scanner} #{cmd} #{options}"
-%>
<% if cmd.nil? -%>
echo "Unsupported security scanner #{scanner}"
exit 1
<% else -%>
<%= exec_command %>
<% end -%>
```

Run / results

Back to Job | Rerun | Toggle command | Toggle STDERR | Toggle STDOUT | Toggle DEBUG | Task Details | Cancel Job | Abort Job

Target: van-holbach.sandbox.dev.atix using Smart Proxy or.sandbox.dev.atix

```
 1:  2024-01-08T10:20:34.077Z        INFO    Vulnerability scanning is enabled                                    Scroll to bottom
 2:  2024-01-08T10:20:34.479Z        INFO    Detected OS: alpine
 3:  2024-01-08T10:20:34.479Z        INFO    Detecting Alpine vulnerabilities...
 4:  2024-01-08T10:20:34.626Z        INFO    Number of language-specific files: 2
 5:  2024-01-08T10:20:34.626Z        INFO    Detecting gobinary vulnerabilities...
 6:  2024-01-08T10:20:34.695Z        INFO    Detecting composer vulnerabilities...
 7:
 8:  nextcloud/all-in-one:20231220_153200-latest (alpine 3.18.5)
 9:  ============================================================
10:  Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
11:
12:
13:  usr/bin/caddy (gobinary)
14:  ========================
15:  Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 1, HIGH: 0, CRITICAL: 0)
16:
17:  ┌──────────────────┬────────────────┬──────────┬────────┬───────────────────┬───────────────┬──────────────────────────────────────────────────────┐
18:  │     Library      │ Vulnerability  │ Severity │ Status │ Installed Version │ Fixed Version │                        Title                         │
19:  ├──────────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────┤
20:  │ golang.org/x/cry │ CVE-2023-48795 │ MEDIUM   │ fixed  │ v0.14.0           │ 0.17.0        │ ssh: Prefix truncation attack on Binary Packet Proto │
21:  │ pto              │                │          │        │                   │               │ col (BPP)                                            │
22:  │                  │                │          │        │                   │               │ https://avd.aquasec.com/nvd/cve-2023-48795           │
23:  └──────────────────┴────────────────┴──────────┴────────┴───────────────────┴───────────────┴──────────────────────────────────────────────────────┘
24:  Exit status: 0                                                                                          Scroll to top
```


Run / results

Job invocations › Run security scanner with inputs target="docker" options="" path="nextcloud/all-in-one:20231220_153200-latest" scanner="grype"

› Template Invocation for van-holbach.sandbox.dev.atix ⇄

Back to Job | Rerun | Toggle command | Toggle STDERR | Toggle STDOUT | Toggle DEBUG | Task Details | Cancel Job | Abort Job

Target: van-holbach.sandbox.dev.atix using Smart Proxy or.sandbox.dev.atix

Scroll to bottom

```
  1: NAME                INSTALLED  FIXED-IN  TYPE       VULNERABILITY        SEVERITY
  2: apache2             2.4.58-r0            apk        CVE-2007-0086        High
  3: apache2             2.4.58-r0            apk        CVE-1999-1237        High
  4: apache2             2.4.58-r0            apk        CVE-1999-0236        High
  5: apache2             2.4.58-r0            apk        CVE-2007-0450        Medium
  6: apache2             2.4.58-r0            apk        CVE-1999-1412        Medium
  7: busybox             1.36.1-r5            apk        CVE-2023-42366       Medium
  8: busybox             1.36.1-r5            apk        CVE-2023-42365       Medium
  9: busybox             1.36.1-r5            apk        CVE-2023-42364       Medium
 10: busybox             1.36.1-r5            apk        CVE-2023-42363       Medium
 11: busybox-binsh       1.36.1-r5            apk        CVE-2023-42366       Medium
 12: busybox-binsh       1.36.1-r5            apk        CVE-2023-42365       Medium
 13: busybox-binsh       1.36.1-r5            apk        CVE-2023-42364       Medium
 14: busybox-binsh       1.36.1-r5            apk        CVE-2023-42363       Medium
 15: golang.org/x/crypto v0.14.0    0.17.0    go-module  GHSA-45x7-px36-x8w8  Medium
 16: php-cli             8.2.13               binary     CVE-2007-4596        High
 17: php-cli             8.2.13               binary     CVE-2022-4900        Medium
 18: php-cli             8.2.13               binary     CVE-2007-3205        Medium
 19: php-cli             8.2.13               binary     CVE-2007-2728        Medium
 20: php-fpm             8.2.13               binary     CVE-2007-4596        High
 21: php-fpm             8.2.13               binary     CVE-2022-4900        Medium
 22: php-fpm             8.2.13               binary     CVE-2015-3211        Medium
 23: php-fpm             8.2.13               binary     CVE-2007-3205        Medium
 24: php-fpm             8.2.13               binary     CVE-2007-2728        Medium
 25: sqlite-libs         3.41.2-r2            apk        CVE-2023-7104        Critical
 26: ssl_client          1.36.1-r5            apk        CVE-2023-42366       Medium
 27: ssl_client          1.36.1-r5            apk        CVE-2023-42365       Medium
 28: ssl_client          1.36.1-r5            apk        CVE-2023-42364       Medium
 29: ssl_client          1.36.1-r5            apk        CVE-2023-42363       Medium
 30: stdlib              go1.20.10            go-module  CVE-2023-45285       High
 31: stdlib              go1.20.10            go-module  CVE-2023-39326       Medium
 32: Exit status: 0
```

Scroll to top

Improve integration

Thoughts

- Write own plugin?
- Scheduled job?
- How to display the results?
- Re-use existing methods?

Solution

- REX job
- Write script on host
- Run the script
  - Executes scanner
  - Output as JSON
  - Read JSON -> write fact
  - Run subscription-manger upload facts

Run and collect facts

```
RUN_SCRIPT=$(mktemp)

cat <<EOF >$RUN_SCRIPT
#!/usr/bin/ruby

require 'json'
require 'tempfile'

def run_scanner(scanner, target, path, outfile, options)
  scanners = {
      trivy: {
          docker: 'image',
          filesystem: 'filesystem'
      },
      grype: {
          docker: 'docker',
          filesystem: 'dir'
      }
  }
  target = target.to_sym

  if scanner == 'trivy'
      puts "Will use trivy scanner."
      cmd = "#{scanners[:trivy][target]} #{path}"
      options += " --scanners vuln --quiet --format json --output #{outfile.path}"
  elsif scanner == 'grype'
      puts "Will use grype scanner."
      cmd = "#{scanners[:grype][target]}:#{path}"
      options += " --quiet  --output json --file #{outfile.path}"
  end
  exec_command = "#{scanner} #{cmd} #{options}"

  puts "Run scanner..."
  system(exec_command)
end

def write_rhsm_entry(f, facts_name, id, key, value)
    f.write("\"#{facts_name}.#{id}.#{key}\": \"#{value}\"")
end
```

```
def generate_unified_vuls(outfile)
  file = File.read(outfile)
  j = JSON.parse(file)

  puts "Generate vulnerability list."
  vuls = {}
  if j.has_key?('matches')
    j['matches'].each do |m|
        simple = {}
        simple['name'] = m['artifact']['name']
        simple['version'] = m['artifact']['version']
        simple['title'] = m['vulnerability']['description'].gsub(/[\[\]"\\\]/, "")
        simple['severity'] = m['vulnerability']['severity']
        simple['url'] = m['vulnerability']['dataSource']
        vuls[m['vulnerability']['id']] = simple
    end
  elsif j.has_key?('Results')
    j['Results'].each do |r|
      next unless r.has_key? 'Vulnerabilities'
      r['Vulnerabilities'].each do |v|
        simple = {}
        simple['name'] = v['PkgName']
        simple['installed'] = v['InstalledVersion']
        simple['fixed'] = v['FixedVersion'] || 'open'
        simple['status'] = v['Status']
        simple['title'] = v['Title'].gsub(/[\[\]"\\\]/, "")
        simple['severity'] = v['Severity']
        simple['url'] = v['PrimaryURL']
        simple['published'] = v['PublishedDate'] if v.has_key?('PublishedDate')
        vuls[v['VulnerabilityID']] = simple
      end
    end
  else
    raise Exception.new("Wrong format")
  end

  vuls
end

def write_file(vuls, facts_name, facts_file)
  puts "Write custom facts to #{facts_file}."
  File.open(facts_file, "w") do |f|
    f.write("{\n")
    vuls.each do |v_id, v|
      f.write(",") unless vuls.first[0] == v_id
      f.write("\n") unless vuls.first[0] == v_id
      v.each do |key, value|
        f.write(",\n") unless v.first[0] == key
        write_rhsm_entry(f, facts_name, v_id, key, value)
      end
    end
    f.write("\n}")
  end
end
```

```
###### MAIN ######

scanner = "<%= input('scanner') %>"
target = "<%= input('target') %>"
path = "<%= input('path') %>"
facts_name = "<%= input('facts_name') %>"
facts_file = "/etc/rhsm/facts/<%= input('facts_name') %>.facts"
options = "<%= input('options') %>"

outfile = Tempfile.new('security-scan')
if run_scanner(scanner, target, path, outfile, options)
  begin
    vuls = generate_unified_vuls(outfile)
    write_file(vuls, facts_name, facts_file)
  rescue => e
    puts "JSON wasn't created by trivy or grype!"
    puts e
  end
else
  puts "Failed to run scanner"
end
outfile.unlink
EOF

chmod +x $RUN_SCRIPT
ruby $RUN_SCRIPT
RES=$?
rm $RUN_SCRIPT

<% if input('run_upload_facts') -%>
if [ "$RES" -eq 0 ]; then
  echo "Upload facts."
  subscription-manager facts --update
fi
<% end -%>

exit $RES
```

Run and collect facts / job

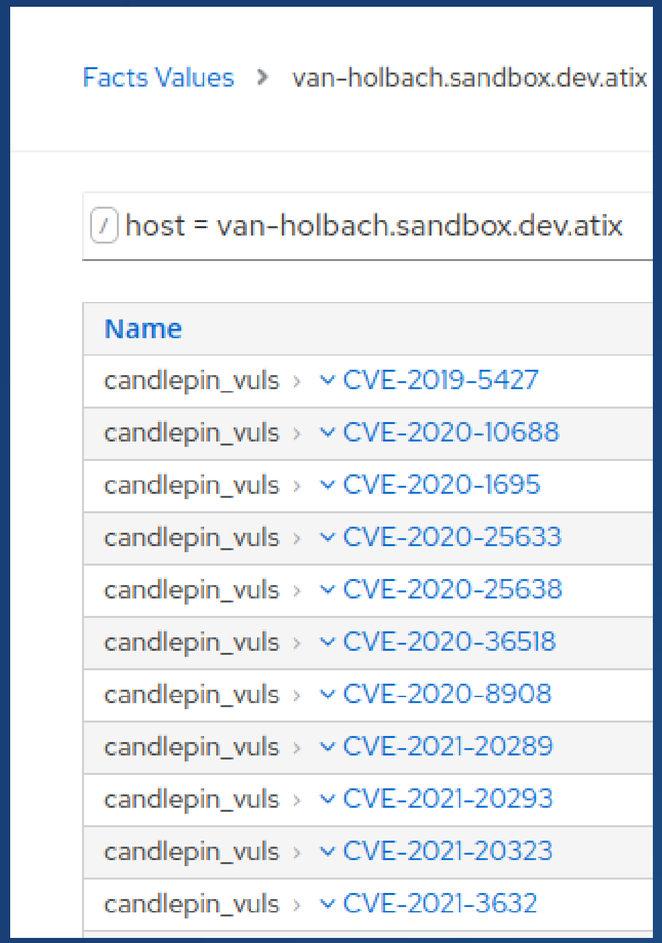
Job invocations › Run security scanner and write facts with inputs options="" scanner="trivy" target="file

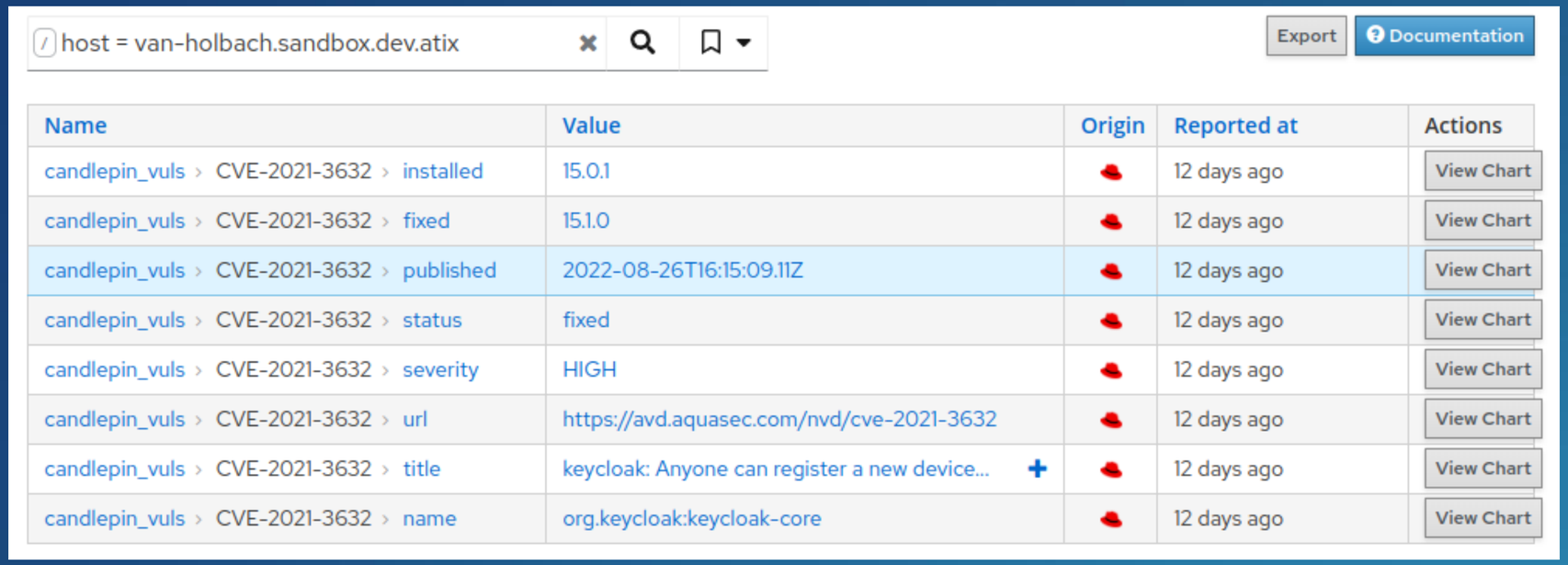
› Template Invocation for van-holbach.sandbox.dev.atix

Target: van-holbach.sandbox.dev.atix using Smart Proxy or.sandbox.dev.atix

```
1:  Will use trivy scanner.
2:  Run scanner...
3:  Generate vulnerability list.
4:  Write custom facts to /etc/rhsm/facts/candlepin_vuls.facts.
5:  Upload facts.
6:  Successfully updated the system facts.
7:  Exit status: 0
```

Run and collect facts / show facts





Future?

- Does it have a future at all? Is there a use-case?

- If yes:
  - Use it as it is right now?
  - Own plugin?
  - Generic "collects data from host, displays certain metrics and alerts me when certain events occur" plugin?

Links

- https://github.com/aquasecurity/trivy
- https://github.com/anchore/grype
- https://github.com/theforeman/foreman
- https://orcharhino.com

ATIX

The Linux & Open Source Company

Contact

sbernhard @ #theforeman-dev

https://github.com/sbernhard

suttner@atix.de