TLP: WHITE
Webinar 03/07/2020

PatrOwl

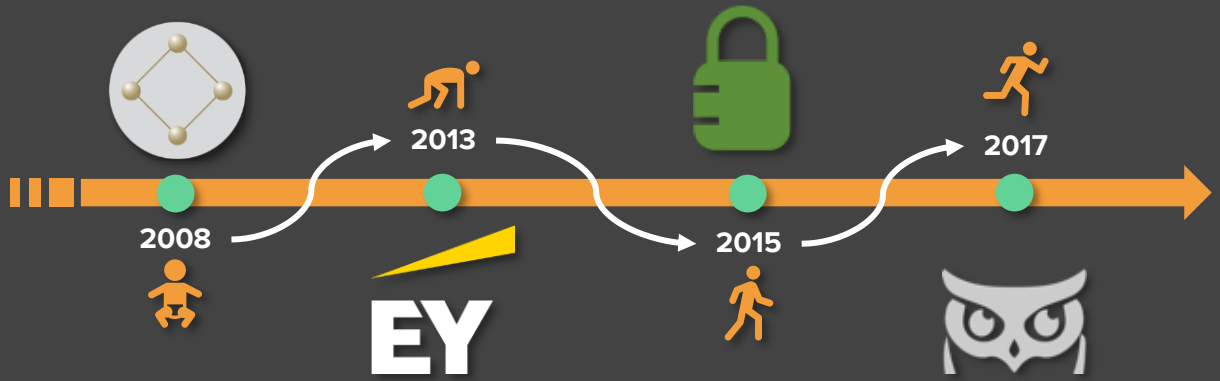
Anticipate Web threats with automation

SecOps orchestration with an open-source SOA(R) platform
#SOAR #SecOps #OpenSource #PreventiveSecurity

Contact getsupport@patrowl.io for more

Let me introduce myself

Nicolas MATTIOCCO
@MaKyOtOx
35 y/o FR

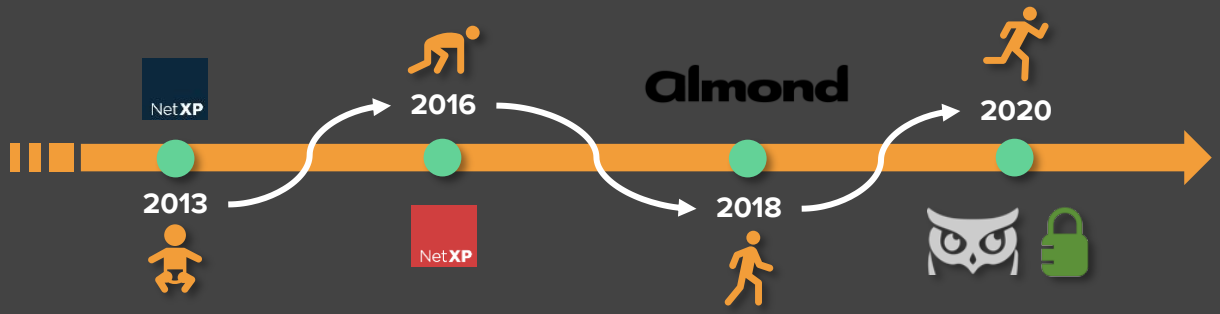


- Security auditor
- Currently onboarded in the **Red Team** of an internal CERT/CSIRT for a financial institution in France
- First-timer on an OSS project
- Proud dad (first-timer too)

You don't even ~~care~~ need to know more about me...

Let me introduce myself *I'm a man of wealth and taste*

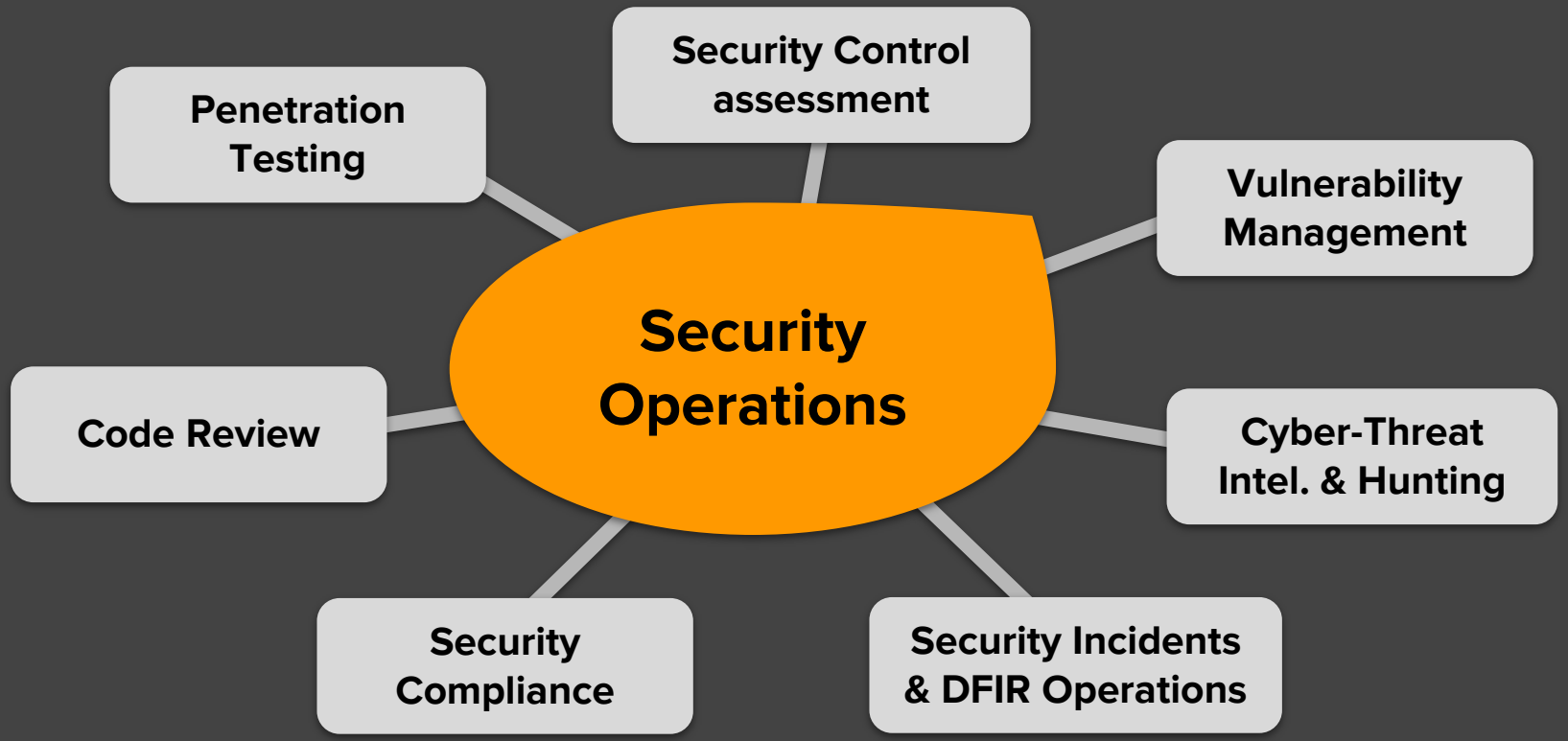
Florent MONTEL
@pepito_oh
30 y/o FR



- Pentester for 7 years
- Co-Founder of an offensive security teams (2 to 8 pentesters)
- Manager (of a 20 pentester team, always)
- Ok. I love pentest.

You don't even ~~care~~ need to know more about me…

Our own definition of *SecOps* ©



What is **PatrOwl** ?

Open source, unified, integrated and scalable platform for SecOps automation and orchestration:

- **Continuous** and **full-stack** security overview
- Define threat intelligence & vulnerability assessment scans policies
- Orchestrate scans using tailor-made engines
- Collect & aggregate findings
- Contextualize, track and prioritize findings
- Check fixes and remediation effectiveness

End-Users:

- CERT/CSIRT, SOC, CTI, DFIR, Penetration testers, Risk Manager, Internal Audit, CISO, Fusion Center
- CTO, Dev[Sec]Ops, Network and system engineers, QA Team, Developers
- M&A, Compliance teams
- InsurTech

What is **PatrOwl** for **professionals** ?

PatrOwl PRO Edition

PatrOwl
Continuous Threat Management

SaaS *and* On-Premise

- **C**ontinuous asset discovery
- **RBAC, multi-tenancy, Enterprise auth**
- **R**eal-time risk evaluation and **P**rioritization
- **P**ro engines

Cyber Rating

Integration and custom dev.

Support and trainings

Vulnerability Intelligence

PatrOwl Hears
Continuous Vulnerabilities Management and Scoring

SaaS *and* On-Premise

- **R**eal-time cybersecurity news (CVE, exploits, bulletins)
- **P**roduct and vendor monitoring
- **C**ontextualized vulnerability scoring
- **R**eal-time security alerts

The end. Thank you for the attention !

Questions ?

A new deadly tool ! But we missed some details…

Story **horse** ?

Facing current and future cyber-security challenges

Trends

Assets exposed ↗ + **Threats** Vulnerabilities | Attackers | Security incidents ↗ + **Business impacts** of security incidents ↗

Facts & Challenges

1. **Poor visibility** on Cyber-exposure risks: Need to monitor a large, diversified, unmanaged and complex scope, even others assets ;
2. **Scarcity** of skilled and efficient **resources** in cyber-security ;
3. **Windows of exposure problem:** Cyber-security mediatisation causes high visibility for vulnerabilities and easiness of attacks ;
4. **Tool capacity-based approach** rather a business threats-based approach. Our great security tools are ineffective without proper strategy, expertise and processes.

Cyber-Exposure and risks are continuously growing and quickly changing

Facing current and future cyber-security challenges

Detecting security incidents

Precursores (may occur)

Indicators (have occurred or is happening)

Events monitoring reveals vulnerabilities and suspicious changes

Asset updates

- Application, system or network updates
- Infrastructure changes: open/closed ports, new subdomain, IP or domain assignment
- **Shadow IT ?**

Infosec KB updates

- CVE, CVSS, CPE updates
- 0-days & misconfigs
- Exploit releasing
- New detection method: scanner update, new tool released, policy updates, infosec researches
- Publication of IOCs

Ext. resource updates

- Data leaks
- Fraud: IP or DNS blacklists, Malware analysis, Typosquatting, ...
- Phishing campaign
- Changes on potential attackers' assets
- Attacks announcements
- Suspicious activities (SIEM)

How to face these bigger, better, badder threats ?

What about **Automation** and **Orchestration** ?

How to face these bigger, better, badder threats ?

Me too !

Why automating SecOps ?

Do more checks

- Cover a larger and diversified scope
- Empower new capacities and improve cyber-security maturity level
- Get a better overview of cyber-exposure (full-stack)

Do it more often

- Continuously checking for vulnerabilities and suspicious changes
- Reduce delays in discovering and fixing a security incident (vulnerability or pwnage)
- Keep updated of your cyber-exposition risks

Do it more efficiently

- Reduce time to low value-adding tasks to focus on more complex security cases
- Reduce and manage costs
- Assess effectiveness of your SecOps activities through measurable KPIs

Do compliance and benchmarks

- Define and expedite controls
- Assess compliance level regarding corporate, regulatory and statutory standards
- Benchmark security level of assets using same control policies


AUTOMATION
PLEASE TAKE MY JOB

Of course, there are several known limits…

- It does not cover 100% of risks in itself (do not be so naïve… Black magic does not exist)

Number of alerts ?	**False-Positives ?**
Functional vulnerabilities ?	**Qualification & Contextualisation ?**
Total Costs of Ownership ?	**Cyber-Defence Strategy ?**

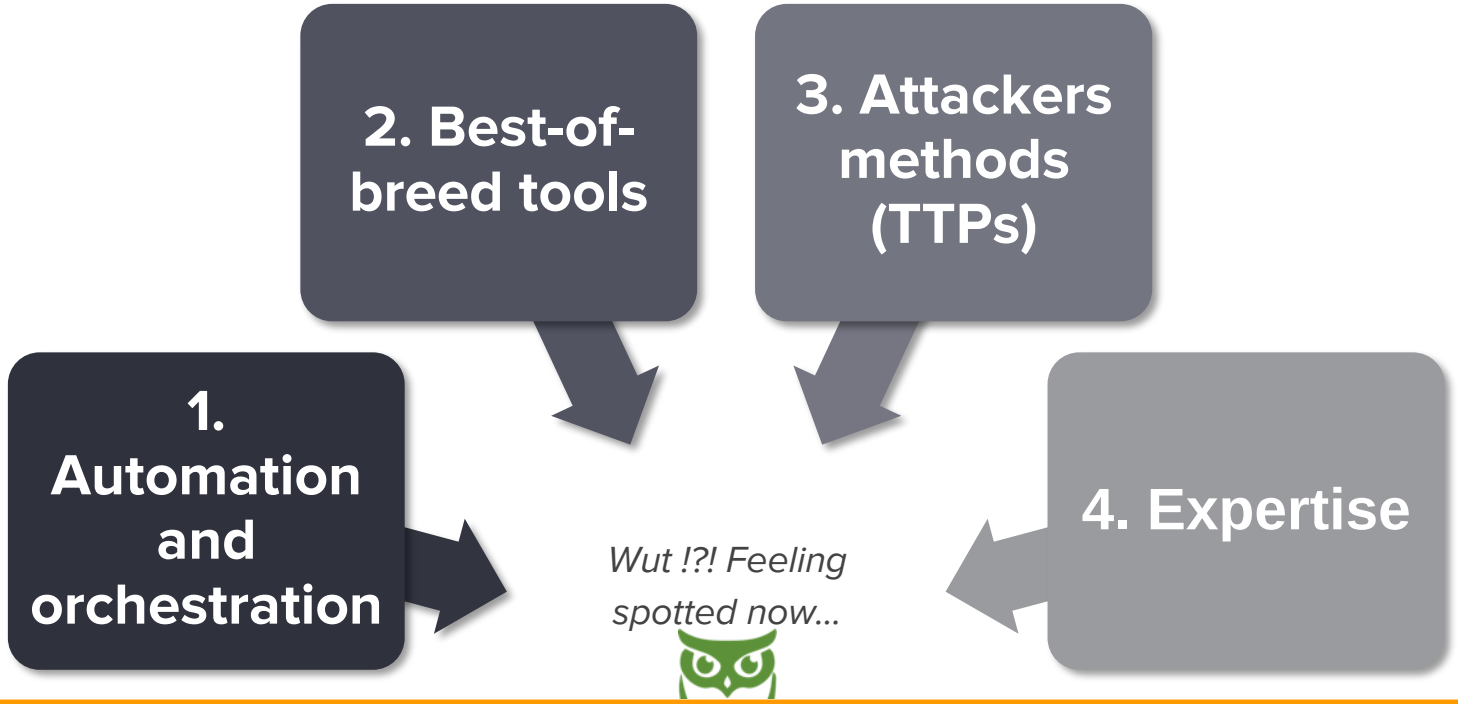
- … and probably all others generic downsides of automated systems …

SecOps automation as a new standard ?

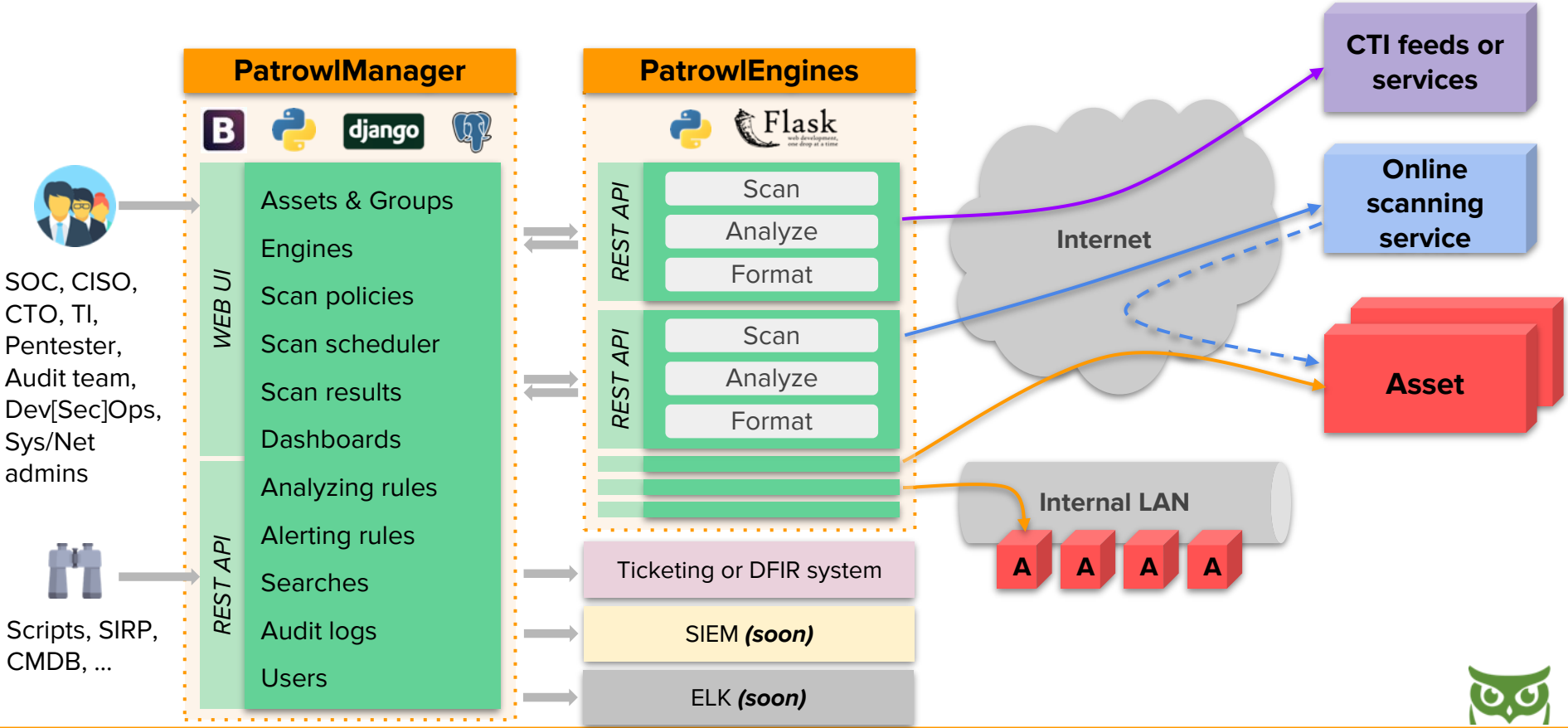
BTW, we built **PatrOwl** for automating and orchestrating **SecOps**

PatrOwl's incentives

Efficiently moving from a reactive to a more *predictive* security posture with:

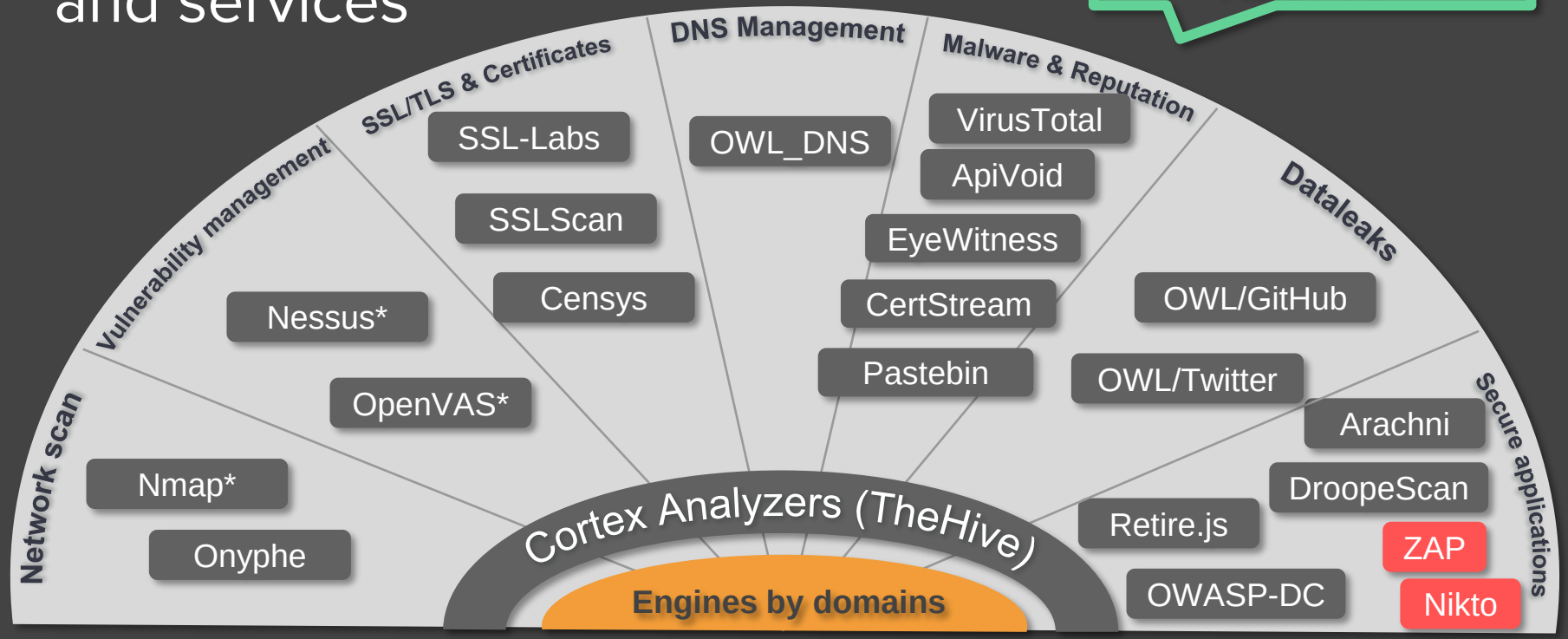


PatrOwl: Technical architecture



PatrowlEngines: Supported tools and services

Turnkey micro-apps: Docker images + REST-API



* Does not embedding scanner

Next engines in the pipeline (to be confirmed)

- **Vulnerability management**: Qualys, Rapid7 IVM/Nexpose
- **Pasties**: AIL-Framework
- **CTI**: MISP, Shodan
- **WEB**: Acunetix, Burp, WPScan
- **Containers**: AquaSec, CLAIR, Jfrog Xray, TwistLock
- **Dataleaks**: Git/truffleHog, gitGraber, Clouseau
- **Cloud**: ScoutSuite, CloudSploit
- **... Any other idea ?**

PatrOwl is a SOA(R) framework

Basic web **use case**

Why web?

Large majority of exposed applications

Large range of exploitation

Home made ☺

So many technologies

Always changing

New ways of deployment

It’s time to adapt our cyber strategy regarding web applications

Web development, a **classic** timeline

Business teams

Security teams

FYI: our new website is now published!

Sorry the new what?

Our new website developed by our web agency in Romania. It was a really urgent need and it's a **critical** website that why I though it was important to warn security experts!

@#!@&&é{}!

Classic problematics

- Security teams not involved in the development processes or orders
- Security always seen as **drag** for **efficiency**
- Needs always 'urgent' for business
- Outsourced development (cost)
- No control over contracts (and security requirements) with the providers

Web development, a **classic** timeline

not watched → watched → watched → watched

"The Talk"

Brand new functionalities ordered on our website! (of course, you don't know, would be to easy)

- Launch the very basic scanner that you have (Nessus) to comfort yourself a little bit
- Ask your favourite pentest teams if they can intervene tomorrow for testing the application
- Lucky you, they can (double price, another one bite in your security budget, yeah!) $$
- Kick-off, authorization, pentest on going. Critical vulnerabilities found
- Vulnerabilities transmitted, provider said that patch are applied
- Order a counter-audit $$
- It's not **patched!**
- Vulnerabilities transmitted, provider said that patch are applied
- Order a counter-audit $$
- It's **patched!**
- Critical application = now annually pentest! $$

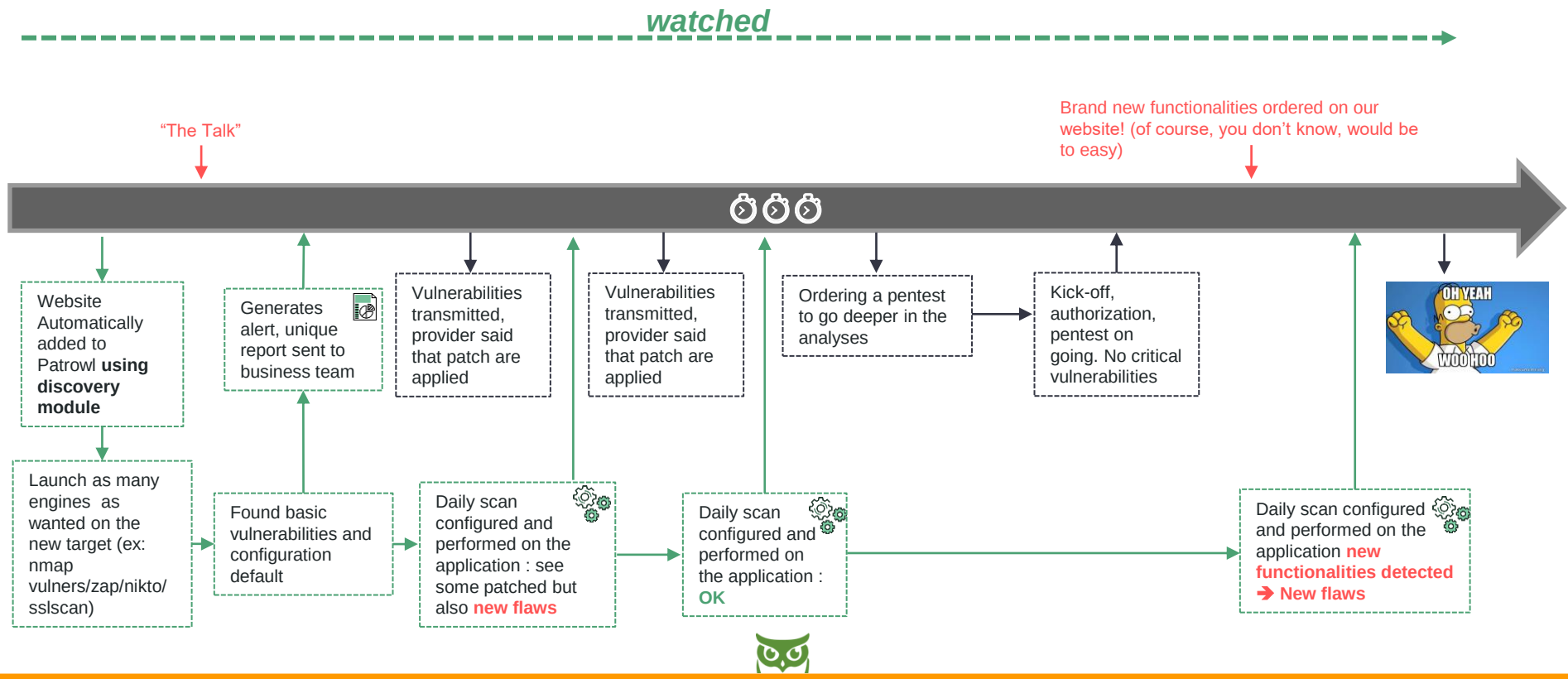
Web development, a **classic** timeline

Opportunist hacker **You**

Automatic recon scripts, large scanning, finding for easily exploitable vulnerabilities, persistence, ransomware, cryptomining…

Your custom security orchestration and processes

Web development, **automation** !



Web development, a **classic** timeline

Your custom security orchestration and processes

PatrOwl is a SOA(R) framework

Other use cases

Use case #2: SAST/DAST in a DevSecOps pipeline

Objectives

- Examining source code and running WEB applications for security defects

Basic strategy

1. On each commit || tag || merge detected on the code repository: clone the project and start a static code analysis of external libraries (SAST)
2. Once the WEB application is deployed on a staging environment, start an autonomous scan (DAST)

Added values

- Free/Open-source tools:
  - OWASP-DC → .Net, Java external libs (at least)
  - Retire.js → JS dependencies
  - Arachni, Zap, Nikto → Web application scanners
- Easy integration of other security tools (REST API + **Patrowl4py** client)
- Early detection of (basic) vulnerabilities
  - But potentially critical !
- Areas of improvement:
  - Docker images checks
  - More integrations are coming: Checkmarx, Acunetix, Burp ...

Use case #3: Phishing preparation scenario

Objectives

- Search for early signs of malicious domains/websites presence

Basic strategy

1. Search for suspicious domains
   - randorizSec.fr, randourisec.fr, ...
2. Monitor them, looking for changes
   - Still parked domains ?
   - Issued certificates ?
   - New exposed services ?

Added values

- Free/Open-source tools:
  - CertStream : Search for potential fraudulent domains/certificates
  - EyeWitness: Take screenshots
  - OWL_DNS & VirusTotal: Gather data on (sub-)domains from CTI feeds
- Continuous discovery and monitoring of suspicious impersonating domains
- Areas of improvement:
  - Typosquating vectors
  - Image recognition

Use case #4: Code leaks on GitHub

Objectives

- Search for leaked internal source code, API Keys, passwords, scripts, …

Basic strategy

1. List the text pattern your want to monitor (beware of false-positives !). Ex:
   - Internal server and application names
   - Internet domain names
   - Cloud API Keys
   - Sensitive email addresses
2. Search these patterns on public GitHub repositories

Added values

- Free/Open-source tools:
  - OWL/GitHub: Search text patterns on GitHub
- Continuous monitoring of public GitHub repositories, including history and development branches!
- Areas of improvement:
  - Search secrets on public and internal Git/SVN repositories (~truffleHog, gitGrabber, …)

Various use cases

Monitoring Internet-facing systems
Scan continuously websites, public IP, domains and subdomains for vulnerabilities, misconfigurations, ...

Data leaks
Monitor code leaks on GitHub, sharing platforms (Pasties), emails in dump leaks, open AWS buckets, ...

Phishing / APT scenario preparation
Monitor early signs of targeted attacks: new domain registration, suspicious Tweets, suspicious pasties, VirusTotal submissions, phishing reports, ...

Vulnerability and remediation tracking
Identify vulnerabilities, send a full report to ticketing system (TheHive, JIRA, ...) and rescan to check for remediation

Regulation and Compliance
Evaluate compliance gaps using tailor-made scan templates

Vulnerability assessment
Orchestrate regular scans on a fixed perimeter, check changes (asset, vulnerability, CVSS, available exploits)

Penetration tests
Perform the reconnaissance steps, the full-stack vulnerability assessment and the remediation checks

Monitoring attacker or suspicious assets
Ensure readiness of teams by identifying attackers' assets and tracking changes of their IP, domains, WEB applications

Securing the CI / CD pipeline
Automation of static code analysis, external resources assessment and web application vulnerability scans

PatrOwl produces findings. A lots of findings…

How to about **prioritization** ?

Once upon a time in a CERT/CSIRT

Morning routine

How to **prioritize** findings ?

- **Our morning routine when a new vulnerability is discovered:**

  Sources: Vulnerability Feeds, CTI, Bluez, Redz, 'Private channels' …

  - We need answers about our **exposure** and **compromising** statuses:
    - ✓ ~~Is it a named vulnerability, with a logo and and dedicated website? @All: We're screwed !~~
    - ✓ What is the CVSS Base Score ? @SOC: Tell us ! Classical communication only to known product owners if it is upper than 7.0 and continue if it's upper than 9.0.
    - ✓ Are we vulnerable ? @SOC+Redz: Confirm the versions, the running configurations and counter-measures in place on our assets, contact product owners !
    - ✓ Are we exposed from the Internet ? @SOC+CTI: Tell us !
    - ✓ Is the vulnerability identified on a critical asset ? @SOC: Tell us !
    - ✓ Are we aware of any functional exploit ? @Redz+CTI: Go find them and test it !
    - ✓ Is there any patch or compensation measure available ? @SOC+CTI: Tell us !
    - ✓ Are there any likelihood catalysts : exploited in the wild? Media hype level ? Exploited by relevant threat actors ? @CTI: Tell us !
    - ✓ Are we already p0wned ? @DFIR: Investigate and reassure us !
    - ✓ Are we able to detect exploitation ? @DFIR: Tell us and/or try to setup alerts !
    - ✓ OK folks, do we have enough data to initiate a CSIRT alert ? @CERT manager: yes / no !

How to **prioritize** findings ?

Wrap up

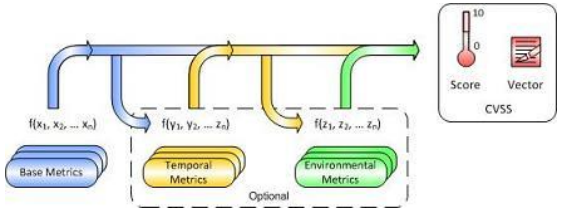
- It is a **teamwork**, not just within the CERT/CSIRT/SOC team
- CVSS Base Score as a primary criteria ? **Really ?**
- Vulnerability metadata **are not static**. They are continuously evolving over the time. Ex:
  - New patch available
  - New exploit released

Prioritize or die

Is the **CVSS Base Score** sufficiently **enough** to be a primary factor of discrimination in vulnerability management ?

Brief reminder of CVSS scoring

- Score ranging from **0.0** (low) to **10.0** (high/critical)
- Metrics:
  - **Base**: represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.
  - **Temporal**: represents the characteristics of a vulnerability that change over time but not among user environments.
  - **Environmental**: represents the characteristics of a vulnerability that are relevant and unique to a user's environment.
- Vector string: text representation of a set of CVSS metrics.



- Several versions: CVSSv1 (2005, NIAC/DHS), CVSSv2 (2007, NIST), CVSSv3.0 (2015, FIRST), CVSSv3.1 (2019, FIRST), *CVSSv4.0 (202x, FIRST)*

Pros	Cons
▪ THE standard ▪ Largely adopted ▪ Transparent ▪ Understandable from everyone	▪ Availability (v2 vs. v3 vs. nothing) ▪ Accuracy ▪ Completeness ▪ Updates ▪ Trust ▪ Equations ?!?

- Only the CVSS Base score is usually provided. Temporal and Environmental scores are on our behalf
- Other fun facts:
  - HeartBleed (CVE-2014-0160) was scored at 5.0
  - Spectre (CVE-2017-5753) was scored at 4.7

Again:

Is the **CVSS Base Score** sufficiently **enough** to be a primary factor of discrimination in vulnerability management ?

Criteria for prioritization

Vulnerability

- **CVSSv2 Impact & exposure**
  - Low (0.0 – 3.9)
  - Medium (4.0 – 6.9)
  - High (7.0 – 10.0)
- **Patch availability**
  - Official/Temporal fix /No/Unknown
- **Age of vulnerability**
  - Hot (0 – 14 days)
  - Recent (15 – 89 days)
  - Old (> 90 days)
- **Discovery ease**
  - ~impossible, difficult, easy
- **Detection ease**
  - ~impossible, difficult, easy

Threat

- **Exploit availability**
  - No known exploit available
  - A private exploit is available
  - A public exploit is available
- **Exploit maturity**
  - Trusting level: Tested, Validated, Shared by a trusted partner
- **Exploit ease**
  - Theoretical, difficult, easy, auto
- **Threat intensity**
  - Exploited in the wild (yes/no) ?
  - In the news (yes/no) ?
- **Threat relevancy**
  - Exploited by monitored threat actors (yes/no) ?

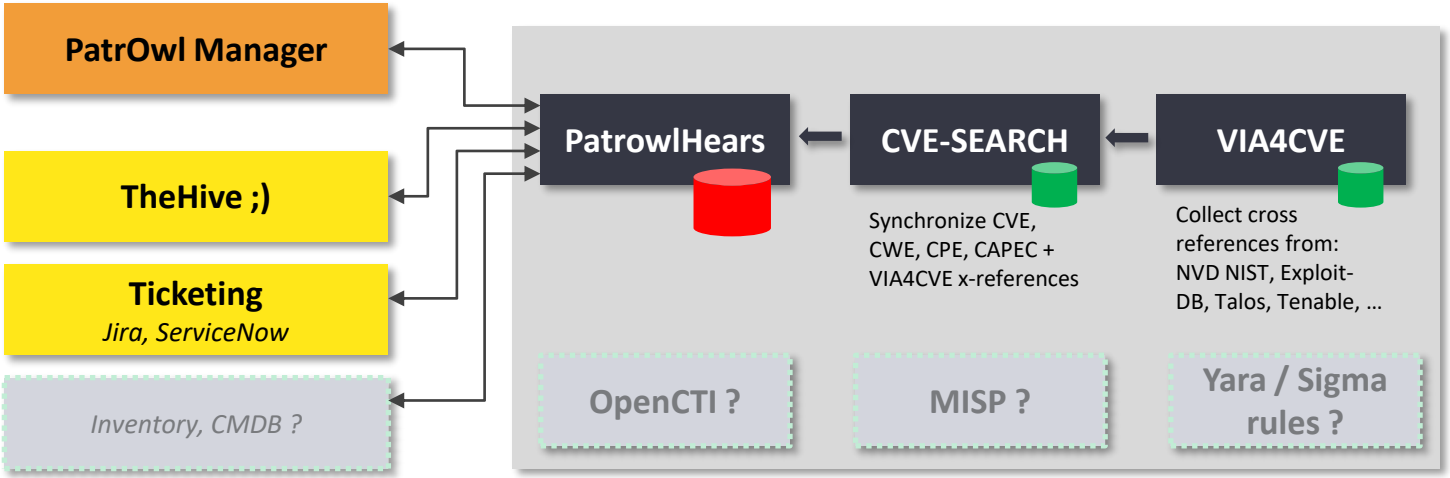
Asset

- **Criticality (ERM-based)**
  - Low
  - Medium
  - High
- **Vulnerable asset interface exposure**
  - Internet
  - Intranet
  - Restricted network
- **Distribution (number of occurrences)**
  - 0 < x assets ≤ 5
  - 6 < x assets ≤ 100
  - > 100 assets

4 suggested actions

- **1/ Now+**: Immediate correction + CSIRT crisis
- **2/ Now**: Immediate correction
- **3/ Next**: Apply fix in the next patching campaign
- **4/ Never**: Apply fix if possible (attention needed / possibly acceptable)

PatrowlHears architecture



- Collect and clean data: CVE, CPE, cross-references
- Create / Update vulnerability metadata from:
  - Collected data
  - User inputs
- Compute a vulnerability prioritization rating using:
  - Vulnerability metadata
  - Asset criticality and exposure
- Monitor vulnerabilities and Vendor/Products: track changes:
  - Track changes (CVSS, exploits, …)
  - Alert: TheHive, Email, Slack, Jira
- *Share feeds: public/private*

Take-away

Cost-Effective

Rationalize tools integration, product licenses and skills

Time-To-Value

Ease of use and deployment, templates for scan policies

Adaptability & Scalability

REST API, Open-Source connectors, adaptable to organisation's ecosystems

360° overview

Full-stack assessment of cyber-exposure, in real-time with relevant data

Always updated

Vulnerability KB, detection methods, threat scenarios

Made with ♥ by experts

Our team members are A+ security engineers

Roadmap

What’s **next** ?

We currently work on:

- **More integration with:** TheHive Cortex RUDDER
  - Security Incident Response
  - IT Automation and Continuous Configuration
- **Patrowl4py:** Python API client for PatrowlManager and PatrowlEngines
- **Testing various use cases**
- **Debugging and improving quality (endlessly) and security**
- **Documenting (endlessly too !) + scan templates**
- **Supporting MITRE ATT&CK & scenario-based tests**
- **Building an Enterprise solution (SaaS and on-premise). Stay tuned !**
  - *Pro features: LDAP/AD/SAML/OAuth authentication, Cloud security assessment engines, assets auto-discovery and synchronisation, awesome custom dashboards, risk-based controls, Jira/ServiceNow integration, ELK ...*

It’s an open-source project: Contribution is needed !!

Who’s up for:

- **Testing it** and giving us lots of **feedbacks !**
- **Contributing:**
  - New engines
  - Debug
  - Features ??
- **Joining the core team ?**
- **Support us ?**

Dev[Sec]Ops, Security engineer, Cloud Architect, UX/UI Designer, QA Tester, Wonder-Woman (Batman is tolerated too) …

Q&A

1. **We have lots of questions !?!**

2. **We want a demo !?!**
   -- Meet us at the bar !

3. **Enough ! Please stop talking bro !?!**
   -- Thanks for the attention !

Contacts

More details ? Meet us ? Contributing ? Want a demo ? Want an awesome sticker ? Share a beer ? Requesting a Cloud/SaaS demo account (BETA test) ? Join us (**we are hiring**) ?

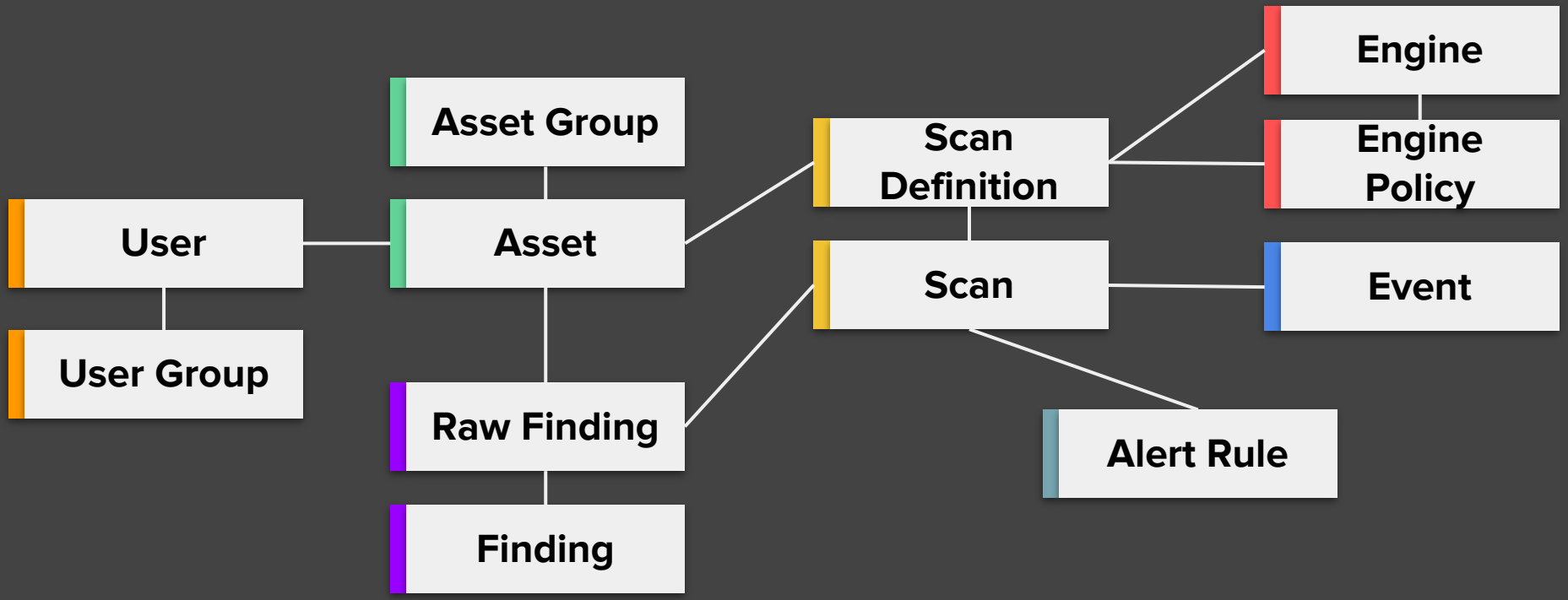
Find us everywhere on earth:

- **Now:** Just in front of you
- **Mail: getsupport@patrowl.io**
- **Web: https://patrowl.io**
- **Twitter: @patrowl_io** (Follow us !)
- **GitHub: @Patrowl** (Star and fork us !)

Before you ask: Why PatrOwl is named “PatrOwl” ?

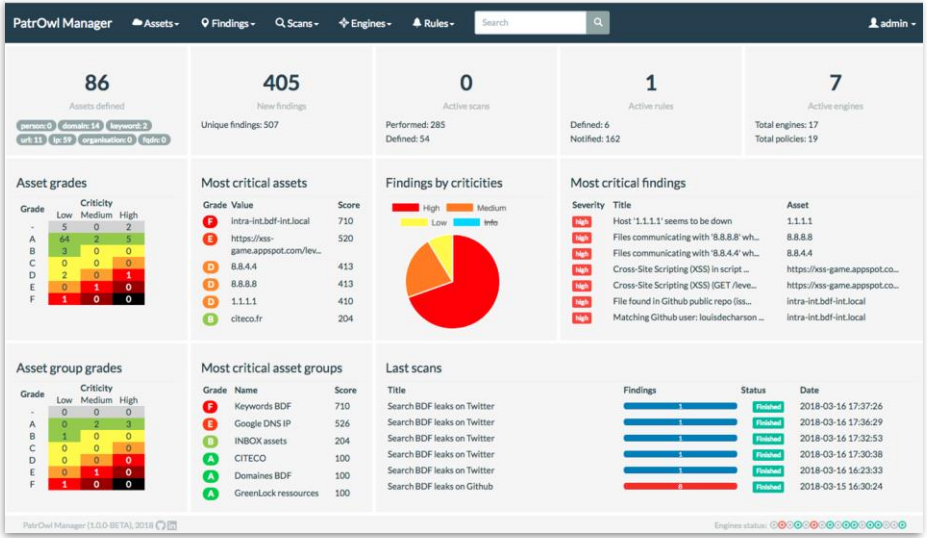
- The owl is able to see in the dark ~~deep web~~ with a large peripheral vision (almost 360°)
- The domain “patrowl.io” name was not already registered

Data Model (simplified)



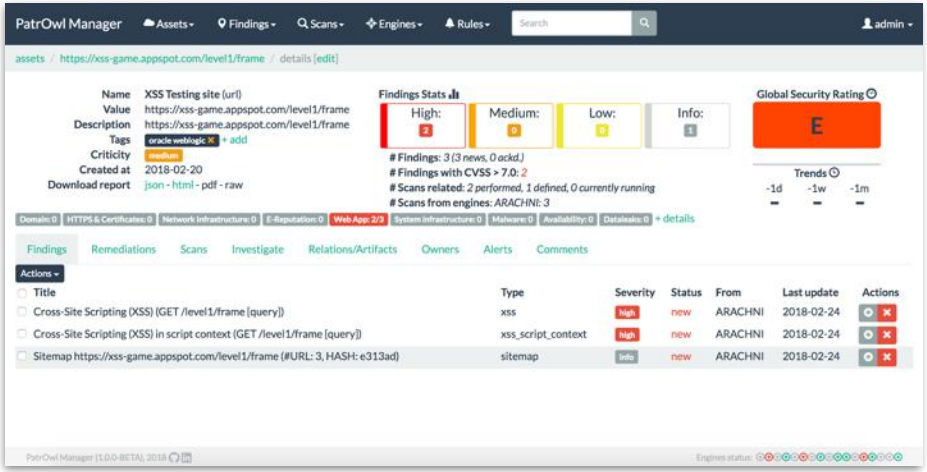
PatrOwl Manager - Dashboard

- Global indicators on assets, findings, scans, engines and rules
- Asset and asset group grades
- Most vulnerables assets and asset groups
- Most critical findings
- Findings repartition by severity
- Last scans status and results
- Top CVSS Score / Findings
- Top CVE, CWE, CPE, …



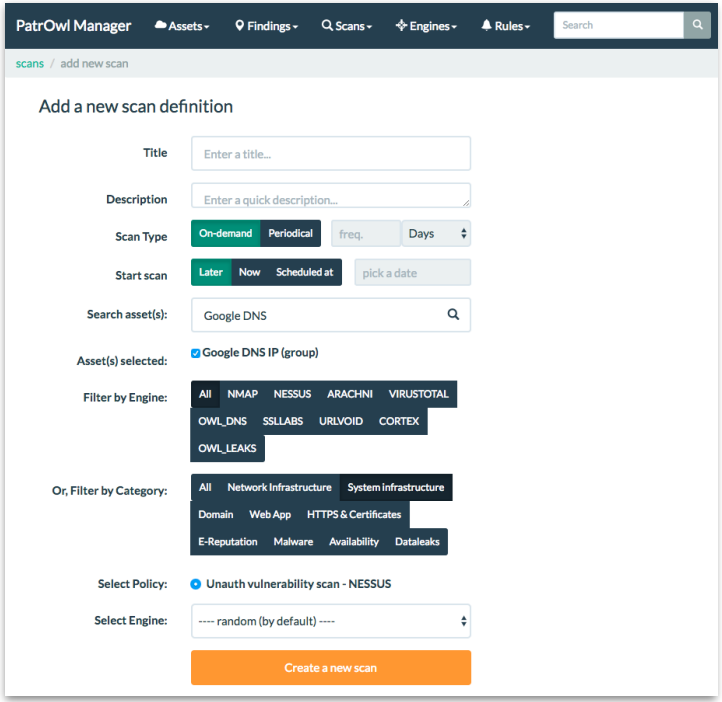
PatrOwl Manager - Asset detailed view

- Current finding counters, risk grade and trends (last week, months, ...)
- Findings by threat domains:
  - Domain, HTTPS & Certificate, Network infrastructure, System, Web App, Malware, E-Reputation, Data Leaks, Availability
- All findings and remediations tips
- Related scans and assets
- Investigation links
- Export HTML, CSV or JSON reports
- Custom tags



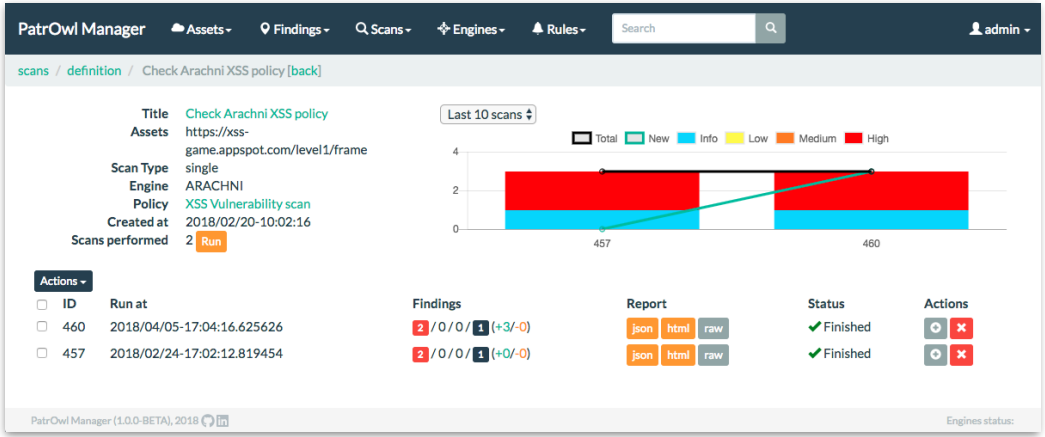
PatrOwl Manager - Scan definition creation view

- Search and select assets and asset groups on their value or name
- Filter policies by engine type or threat domain
- Select engine
  - If no engine is selected, an engine is randomly chosen in available engines for each scan



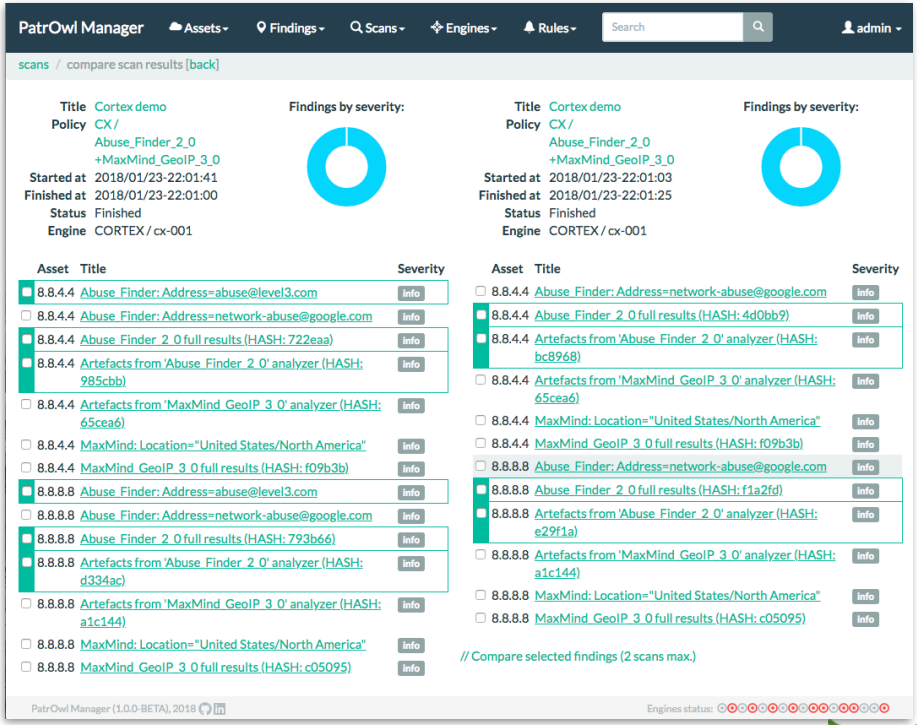
PatrOwl Manager - Scan definition view

- Related scan results overview
  - ID, starting datetime, finding counters by severities, status
- Quick run button
- Quick scan report (HTML or JSON), delete or show details



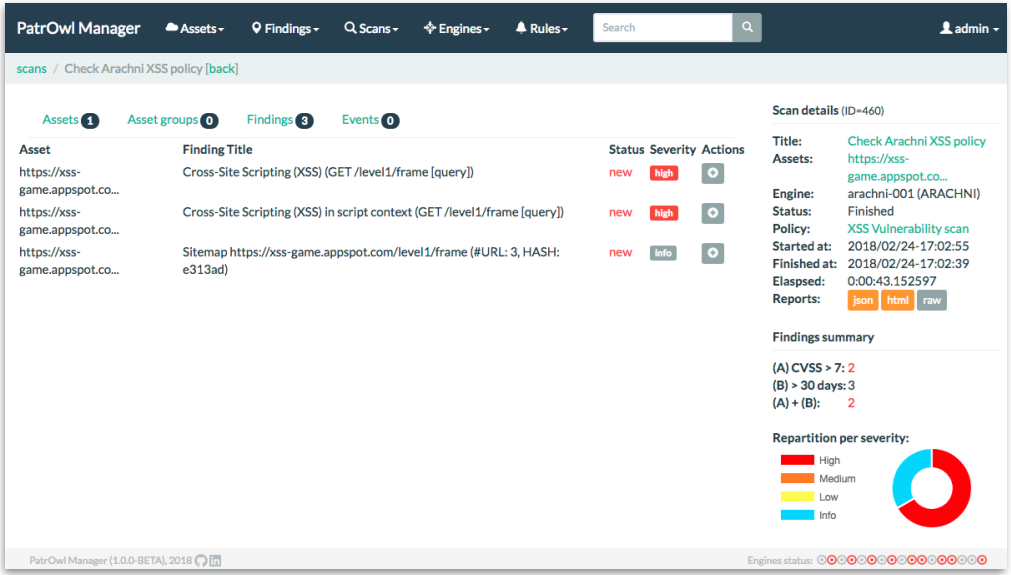
PatrOwl Manager - Scan compare view

- Highlighting differences:
  - new and missing findings
  - same finding type but different details
- Link to the findings comparison view



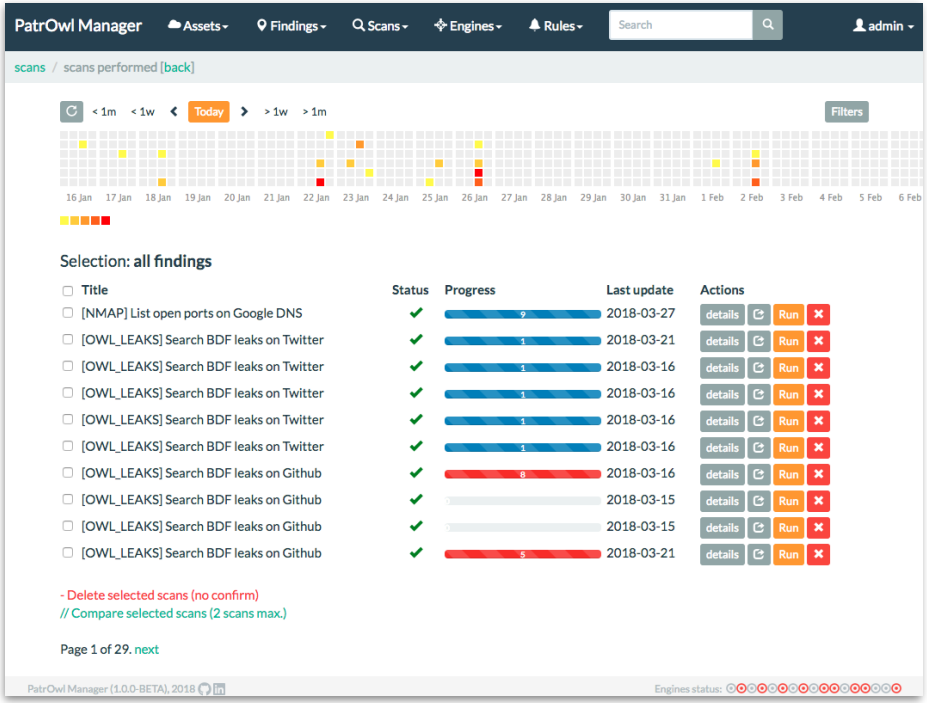
PatrOwl Manager - Scan results view

- Scans info: title, assets, status, policy, start/end dates
- Findings list + show details link
- Quick scan report (HTML or JSON)
- Findings summary on metrics
- Asset and asset group overview
- List of related events



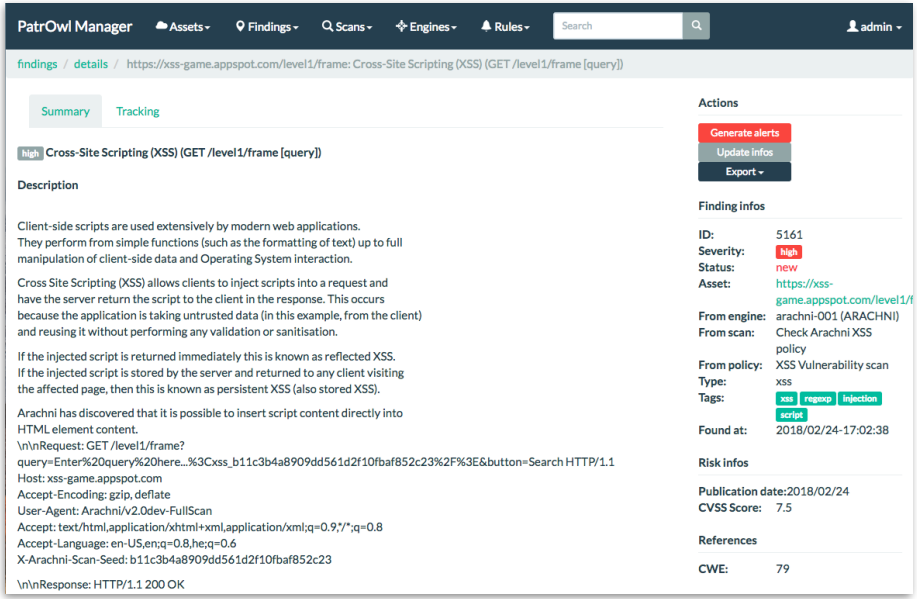
PatrOwl Manager - Scan performed view

- Scans heatmap over days, weeks and months
- Advanced filters
- Run or delete scans
- Show scan details
- Compare selected scans



PatrOwl Manager - Finding view

- Finding info
- Description, solution, links and hash
- Quick actions:
  - Generate alerts
  - Change metadata: severity, status, tags, CVSS
  - Export to file (JSON or STIX2 format)
- Show tracking info
  - Changes history
  - Matching scans



PatrOwl Manager - Finding compare view

- Highlighting differences between findings

PatrOwl Manager | Assets | Findings | Scans | Engines | Rules | Search | admin

findings / compare [back]

	Finding A (ID: 1181)	Finding B (ID: 1179)
Title	Port 'tcp/80' is filtered	Port 'tcp/56' is filtered
Severity	info	info
Asset	8.8.8.8	8.8.8.8
Description	The scan detected that the port 'tcp/80' was filtered	The scan detected that the port 'tcp/56' was filtered
Solution	n/a	n/a
Risk info	vuln_publication_date: 2018/01/16 cvss_base_score: 0.0	vuln_publication_date: 2018/01/16 cvss_base_score: 0.0
Vuln info	n/a.	n/a.
Links	No links.	No links.
Tags	No Tags.	No Tags.
Created at	2018/01/16-12:01:32	2018/01/16-12:01:30
Scan title	List open ports on Google DNS	List open ports on Google DNS
Scan policy	List open ports (TCP/53,56,80,443,8080)	List open ports (TCP/53,56,80,443,8080)
Scan engine	NMAP - nmap-002	NMAP - nmap-002

PatrOwl Manager (1.0.0-BETA), 2018 | Engines status:

PatrOwl Manager - Alerting rules management view

- Create, copy, modify or delete alerting rules
- Change functional status

PatrOwl Manager | Assets | Findings | Scans | Engines | Rules | Search | admin

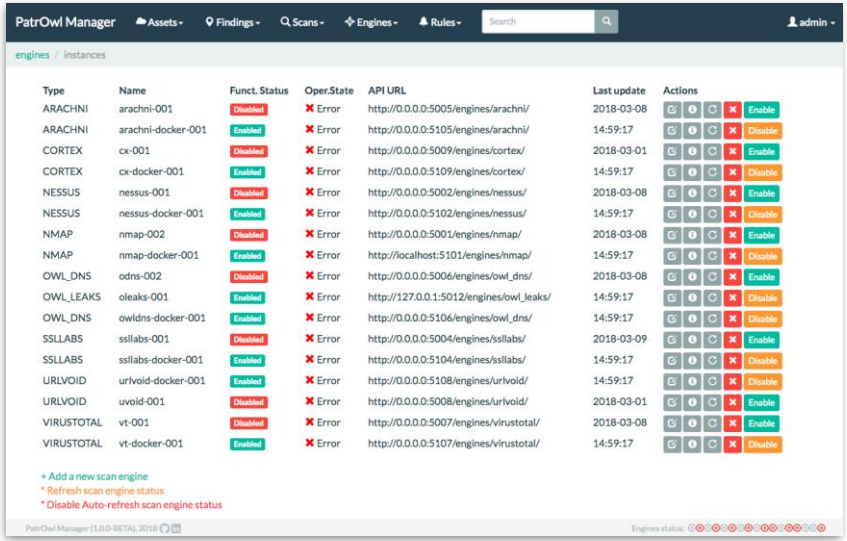
Rules / List

Name	Scope	Condition	Trigger	Severity	Target	Status	Last update	Actions
New findings found (Slack)	finding.status	is 'new'	auto	Low	slack	Disabled	2018-02-20	
Findings with severity='info' -> email	finding.severity	is 'info'	auto	Low	email	Disabled	2018-01-23	
Findings with info severity	finding.severity	is 'info'	ondemand	Low	slack	Enabled	2018-02-01	
Findings with low severity	finding.severity	is 'low'	auto	Low	slack	Disabled	2018-02-20	
Findings with high severity	finding.severity	is 'high'	auto	Low	slack	Disabled	2018-01-16	
New findings found	finding.status	is 'new'	auto	Low	thehive	Disabled	2018-02-11	
Title..	Asset	is	On-demand	Low	✓ Patrowl event	Enable	Add	
	criticity	low						

- To logfile
- Send email
- To TheHive
- To Splunk
- To Slack

PatrOwl Manager - Engine management view

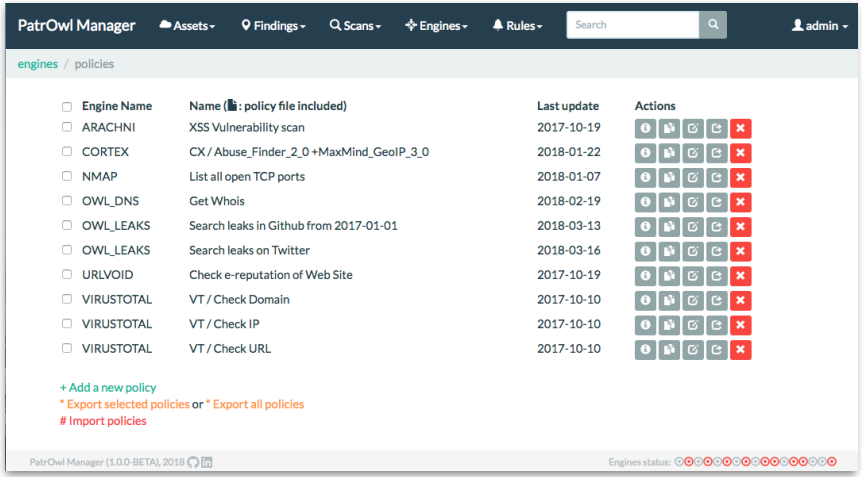
- Create, modify or delete engines
- Change functional state
- View engine info, including current scans performed
- Refresh engines states
- Enable/Disable the auto-refresh



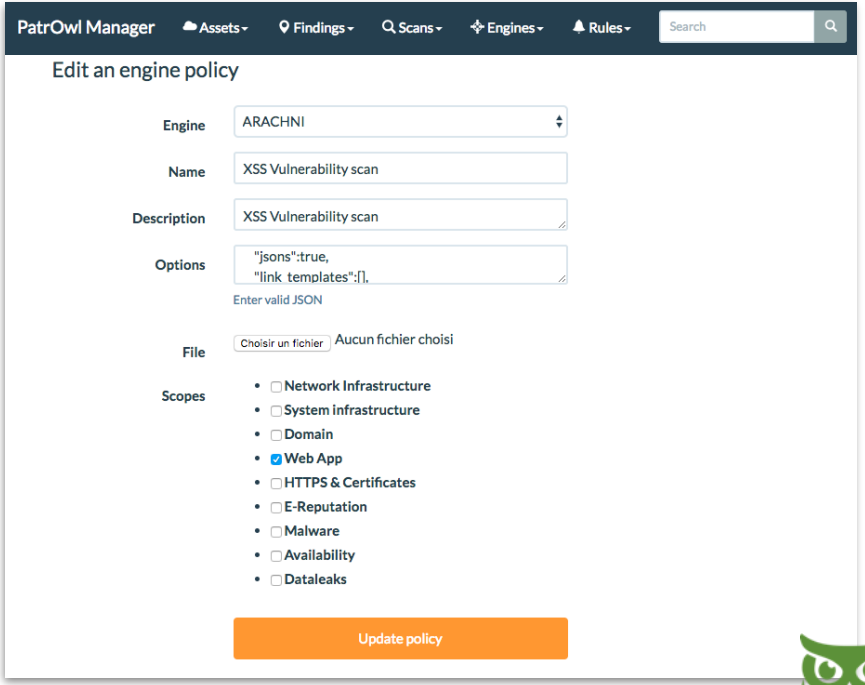
- Engines states are regularly updated and always shown in the footer:

PatrOwl Manager - Engine policy views

- Create, copy, modify or delete engine policies
- Quick policy info retrieving



- Engine policy details:



Contribution needed !!

Who's up for:

- **Test it** and give us **feedbacks !**
- **Contribute !**
  - New engines
  - Debug
  - Features ??

- **Joining the core team ?**
  - Dev[Sec]Ops, Security engineer, Cloud Architect, UX/UI Designer, QA Tester, Wonder-Woman (Batman is tolerated too) ...

Q&A

We have questions !?!

We want a demo !?!

Stop talking bro ! We want a break now !?!

Contacts

More details ? Meet us ? Contributing ? Want a demo ? Want an awesome sticker ? Share a beer ? Requesting an online demo account (BETA test) ?

Find us everywhere on earth:

- **Mail: getsupport@patrowl.io**
- **Web: https://patrowl.io**
- **Twitter: @patrowl_io** (Follow us !)
- **GitHub: @Patrowl** (Star and fork us !)