PatrOwl

Security Operations Orchestration

The next stage of proactive and efficient threat management

June 2018

Contact **getsupport**@patrowl.io for more

Cyber-Security challenges

Trends

Facts & Challenges

1. Cyber-security **mediatisation** causes high visibility of vulnerabilities and easiness of attacks
2. Poor **visibility** on Cyber-exposure risks
3. Security **tools** exists, largely adopted but ineffective without proper strategy, expertise and processes
4. Need to monitor a large, diversified, unmanaged and complex **scope**, even others assets
5. Scarceness of efficient **resources** in cyber-security
6. Tool capacity-based **approach** rather a business threats-based approach

Cyber-Exposure and risks are continuously growing and fastly changing

Cyber-Security challenges

Security Incidents

Precursores (may occur)

Indicators (have occurred or is happening now)

Events monitoring reveals vulnerabilities and suspicious changes

Infosec KB updates

- CVE, CVSS, CPE updates
- Unsecure configuration
- Exploit releasing
- New detection method: scanner update, new tool released, policy updates, infosec researches
- IOC published

Assets updates

- Application or system updates
- Infrastructure changes: open/closed ports, new subdomain detection
- IP or domain assignment

Ext. resource updates

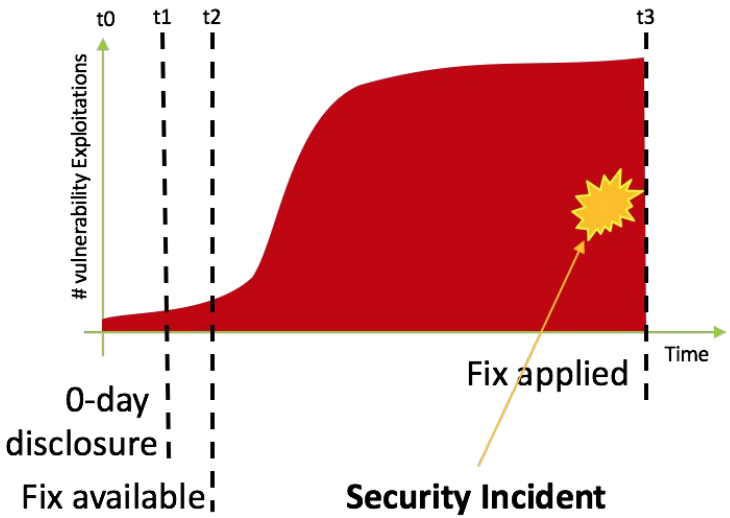
- Data leaks detection
- Fraud detection: IP or DNS blacklists, Malware analysis, Typoquating, ...
- Phishing reporting
- Changes on potential attackers' assets
- Attacks announcements
- Suspicious activities (SIEM)

Our vision

Legend:
Risk exposure window = t3 - t0
Time-to-Patch = t3 – t2
Red area = threat occurrence

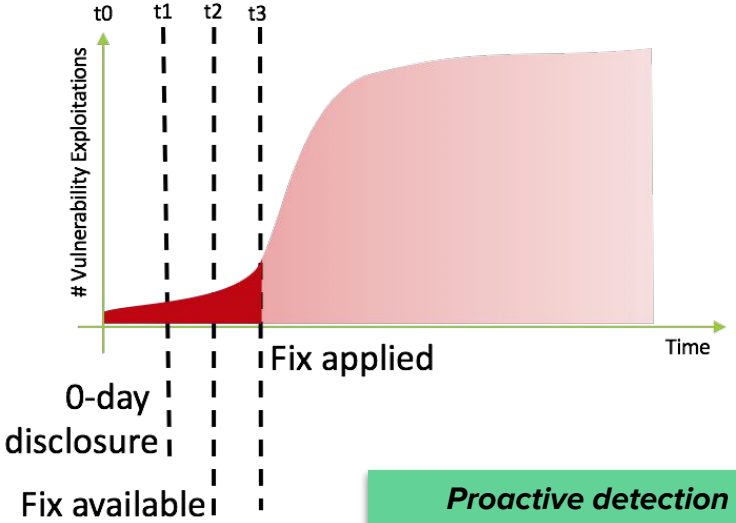
Incident-Based Process

Limited and inefficient strategy



Managed Security Process

Efficient strategy



Proactive detection
+ Alert notification
= Early fixing = Safe earlier

Our vision

We need to efficiently moving from a proactive to a predictive security posture

Thinking and acting like hackers	Using their mindset (tools, tactics and procedures), full-stack targeting
Security automation and orchestration	Enable to continuously scan an organisation's environment for any changes that might indicate a potential threat
Best-of-breed and custom tools	Unique cockpit and rationalized use of best-of-breed and custom tools to support the cyber-threat monitoring strategy and remediation workflow

Cyber Exposure assessment objectives:

- Identify the vulnerabilities before attackers
- Identify the risk exposure as seen by 3rd parties
- Identify early warning signs of threat scenarios
- Identify compromising of assets or data leaks ASAP

Monitoring scope:

- Company's known and unknown assets
- External resources (ex: Threat intelligence feeds)
- Attackers' assets

Our vision


Security Control assessment
Penetration Testing
Vulnerability Management
Security Operations Orchestration
Code Review
Cyber-Threat Intelligence
Compliance
SOC & DFIR Operations

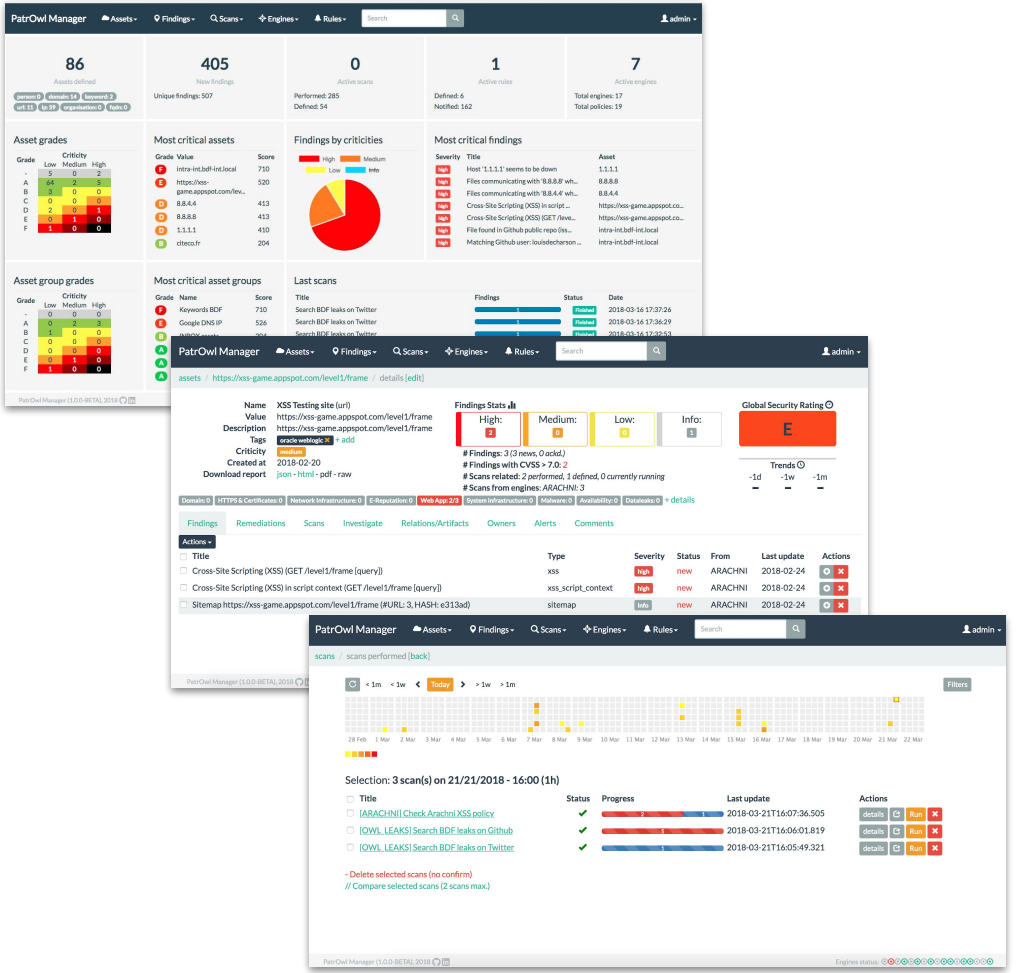
PatrOwl

Provide a centralized platform to :

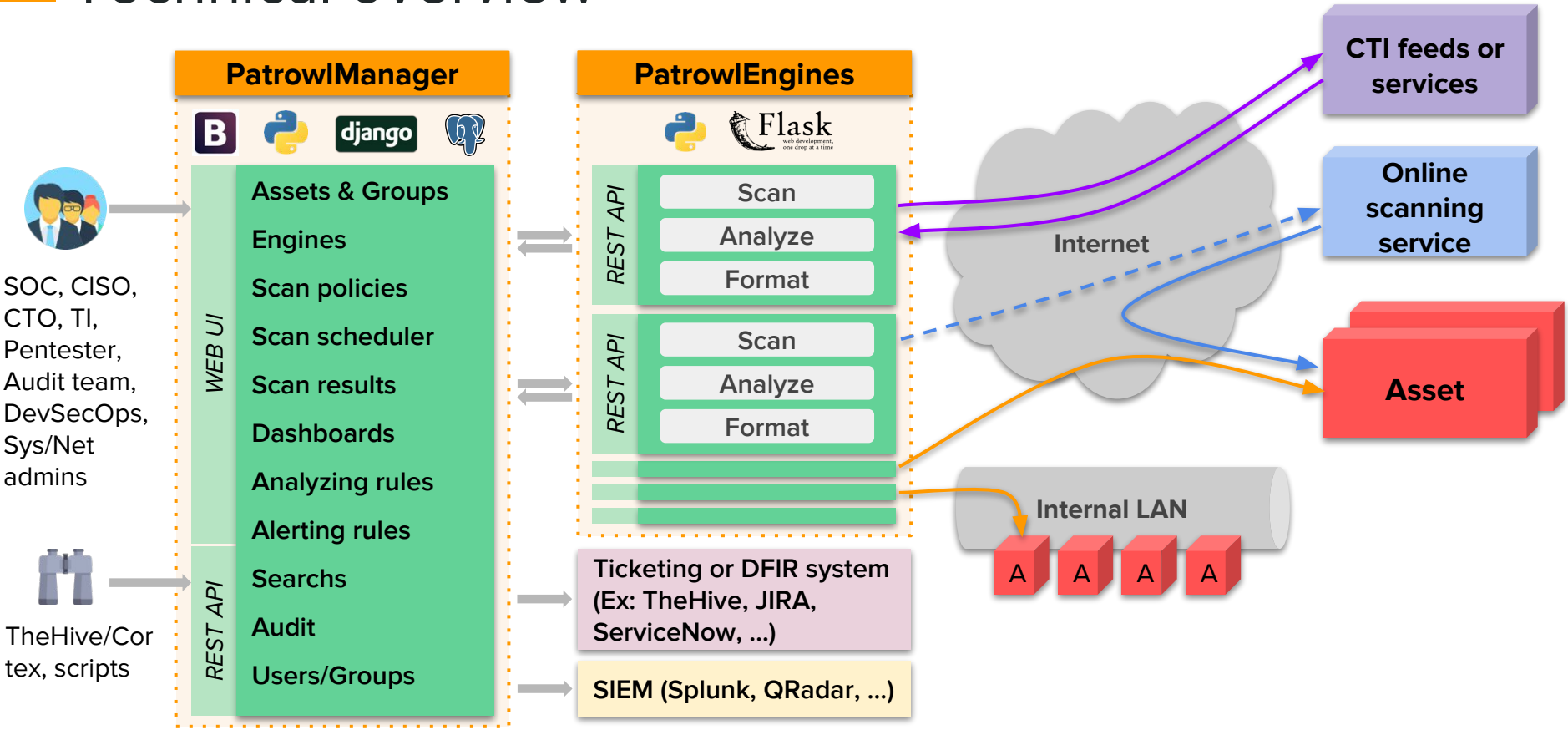
- Full-stack security overview (IP to Data)
- Define threat intelligence & vulnerability assessment scans policies
- Orchestrate scans using tailor-made engines
- Collect & aggregate findings
- Contextualize, tracks, prioritize findings
- Check remediation effectiveness

Users:

- CERT/SOC, CTO, CISO, Risk Manager, Audit teams, Penetration testers, Webmasters, Network and system engineers, Q&A teams, Business & App owners



Technical overview



Supported Engines (June 2018)

Tool	Description	System infra.	Network infra.	Domains	HTTPS & Certificates	E-Reputation	Data leaks	Malware	Web Applications
NMAP	Network scanner	✖	✖						
Tenable Nessus	Vulnerability assessment (Nessus Scanner only)	✖	✖		✖				
Arachni	Web Vulnerability scanner								✖
Censys	Internet-wide data scanner			✖	✖				
VirusTotal	Online malware and artefact analyzer			✖		✖		✖	
UrlVoid	Website Reputation Checker					✖			✖
Qualys SSL-Labs	TLS/SSL configuration scanner				✖	✖			

Supported Engines (June 2018)

Tool	Description	System infra.	Network infra.	Domains	HTTPS & Certificates	E-Reputation	Data leaks	Malware	Web Applications
OwlDNS	(Sub-)Domain analyzer			✖		✖			
OwlLeaks	Data scrapper on Google, GitHub, Twitter, ...						✖		
OwlCode	Retire.js and OWASP-DC analyzer								✖
Cortex	TheHive companion (30+ analyzers)	✖	✖	✖	✖	✖	✖	✖	✖

PatrOwl Engines ?

- An engine uses local binaries, scripts or remote services
- Data analysis are performed on the results, then findings are formatted in a generic format
- Custom engines can be connected to the back-end:
  - JSON REST API with strictly formatted inputs and outputs and a strict (but simple) workflow
  - A meta-engine is provided
  - Full documentation is in progress
  - Token and Basic authentication features will be soon supported
- ≃ 1 day needed for writing a simple engine
- All submitted engines by the community is be tested by SurvivOwl' engineers before being officially released

Use cases

Monitoring Internet-faced 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, paste, 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 provided scan templates

Vulnerability assessment of internal systems
Orchestrate regular scans on a fixed perimeter, check changes (asset, vulnerability, criticality)

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

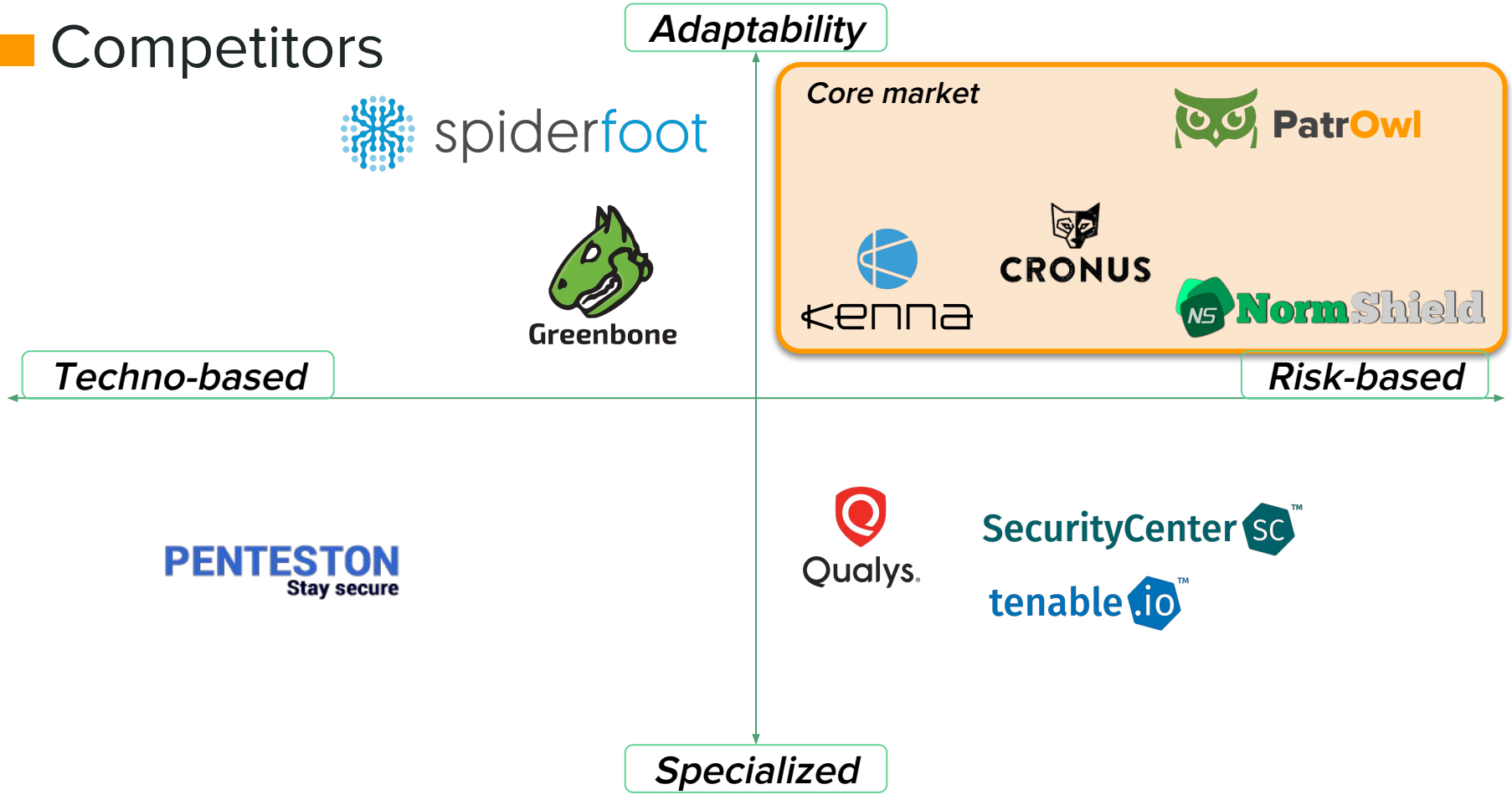
Attacker assets monitoring
Ensure readiness of teams by identifying attackers' assets and tracking changes of their IP, domaines, WEB applications

Continuous Integration / Continuous Delivery
Automation of static code analysis, external resources assessment and web application vulnerability scans

Business Model

	Free	Paying (Contact GreenLock Advisory)		
Products	**Open-source release** Github repository **Marketplace** Engines, policies, AI rules, dashboards	**Premium release - SaaS services** Shared or dedicated servers	**Premium release - On-Premise** Appliance or Docker	**Private Threat Intelligence feeds** **Risk Scorecards** **Marketplace** Advanced AI rules, policies, dashboards
Services	**Community services** Documentation Support Bug fixes + features	**Premium Support** Private ticketing, chats, phone Documentations + Trainings	**R&D** Custom developments, Threat Intelligence services	**Consulting** SOC/CTI Strategy, product integration or review, security audits, investigations

Competitors



Competitors

	SaaS	On-Premise	Open-Source
PatrOwl	✓	✓	✓
spiderfoot	✓	✓	✓
kenna	✓	✗	✗
NormShield	✓	✗	✗
Greenbone Sustainable Resilience	✗	✓	✓
Qualys	✓	✓	✗
SecurityCenter SC	✗	✓	✗
tenable.io	✓	✗	✗

Competitive advantages

Cost-Effective

Rationalize tools integration, product licenses and skills

Time-To-Value

Ease of use and deployment, default policies and engines policies

Adaptability & Scalability

REST API, Open-Source connectors, adaptable to organisation maturity level

360° overview

Full cyber-Exposure assessment in real-time with relevant data

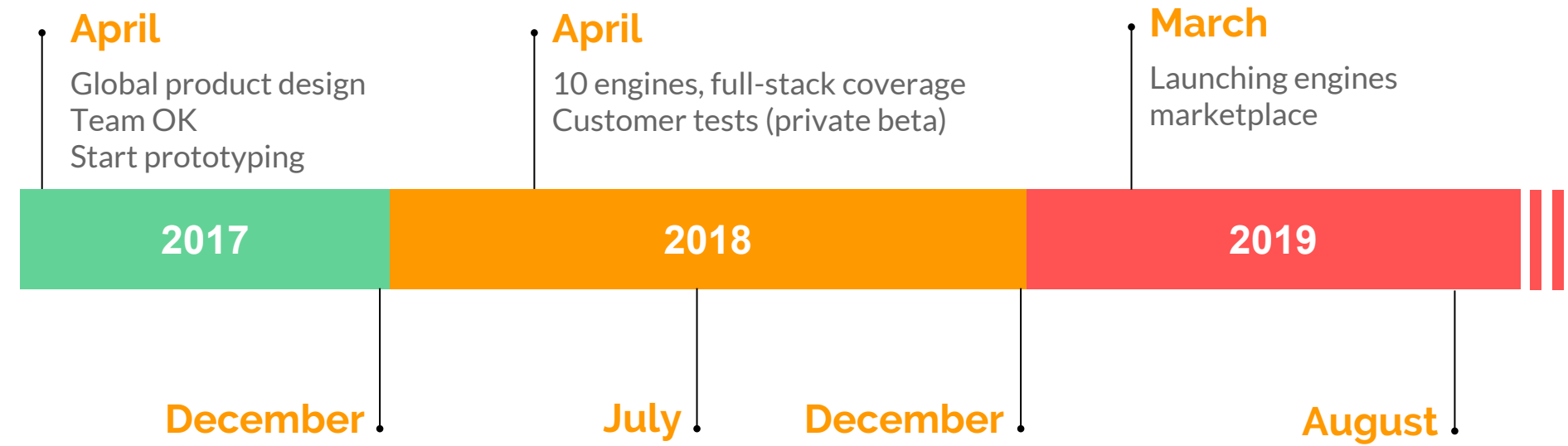
Always updated

Vulnerability KB, detection methods, threat scenario

Made by experts

Our team members are A+ security engineers

(Very) Big milestones



Contacts

More details ? Requesting a demo ? Meet us ?

Find us everywhere (no excuses !)

- Email: **getsupport**@patrowl.io
- Website: https://**www.patrowl.io**
- Twitter: **@patrowl_io**
- GitHub: **@Patrowl**

Logos

PatrOwl

Continuous Threat Intelligence

#76A144

#616161

#FF9900

PatrOwl overview

PatrOwl Manager (Backend)

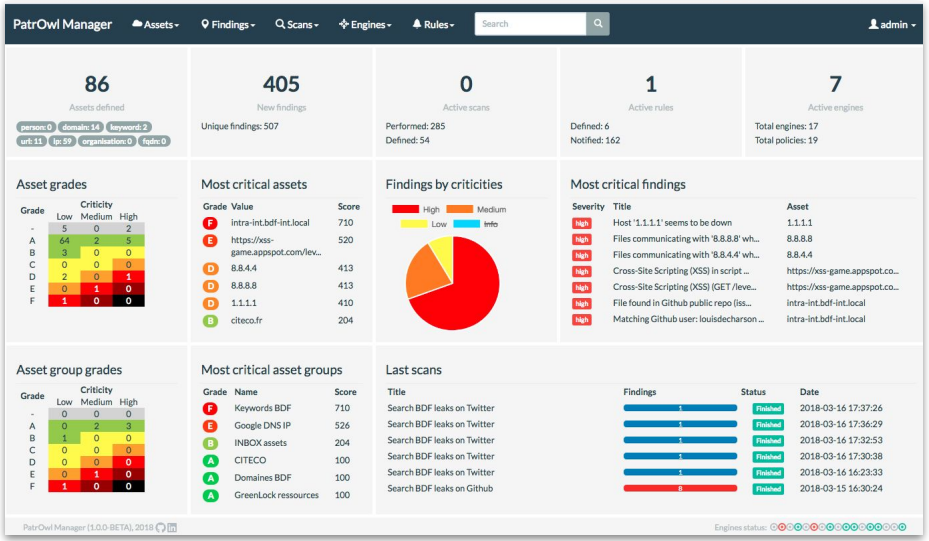
- Unified platform for managing assets, threats, scans, findings and engines
- Orchestrate scans started on engines

PatrOwl Engines

- REST API
- Perform the scans using locally installed or remote online tools

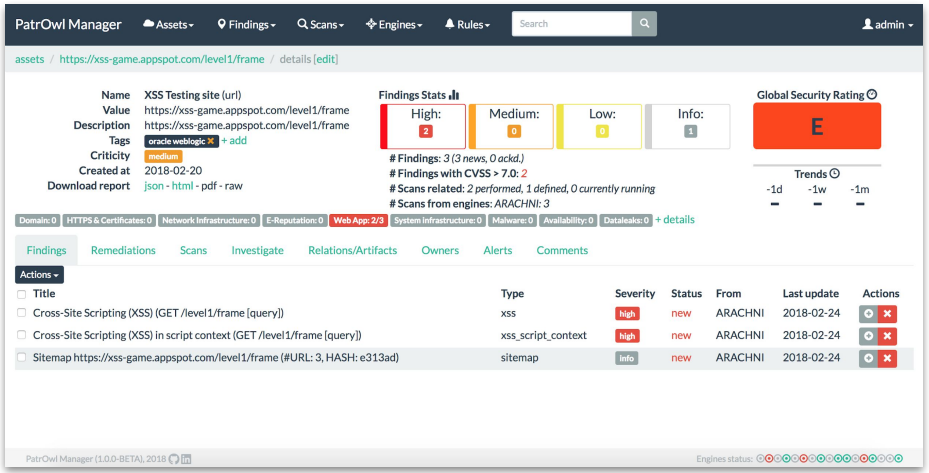
PatrOwl Manager - Dashboard

- 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 criticity
- Last scans status and results
- Top CVSS Score / Findings
- Top CVE, CWE, CPE, ...



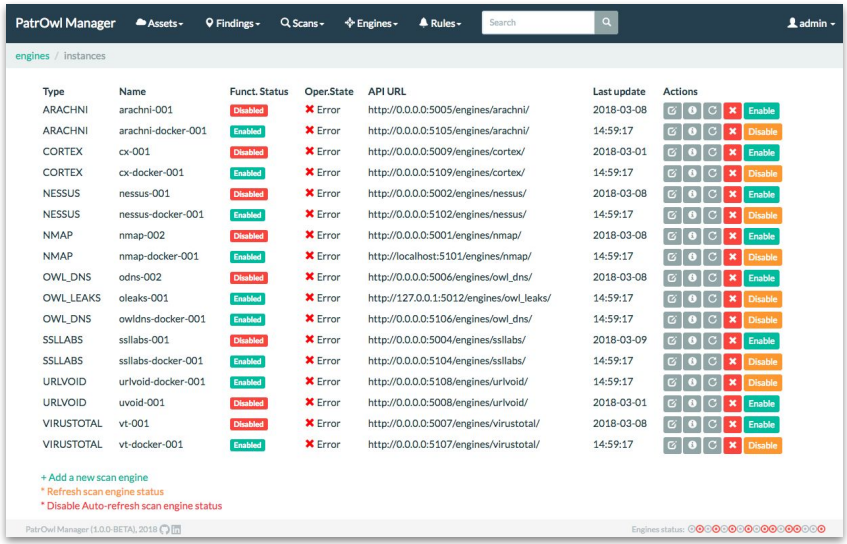
PatrOwl Manager - Asset detailed view

- Current finding counters and 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
- Report to HTML or JSON
  - @todo: PDF



PatrOwl Manager - Engine management view

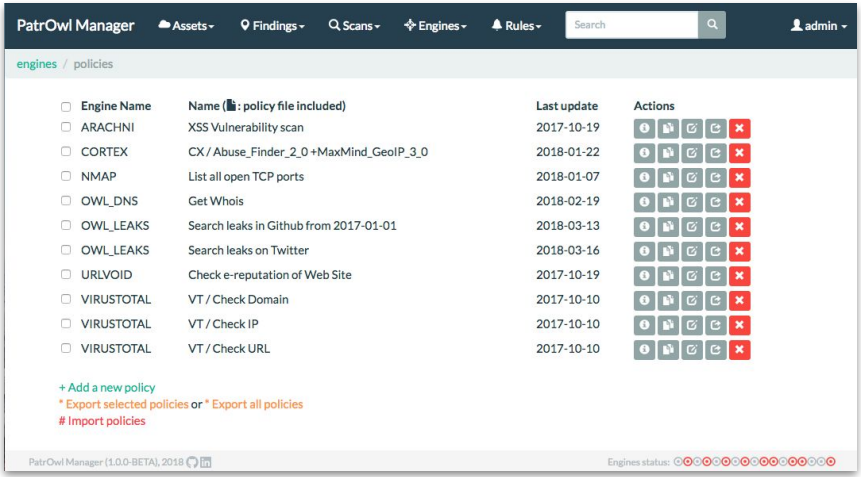
- Create, modify or delete engines
- Change functional state
- View engine info, including current scans performed
- Refresh engines states
- Enable/Disable the auto-refresh



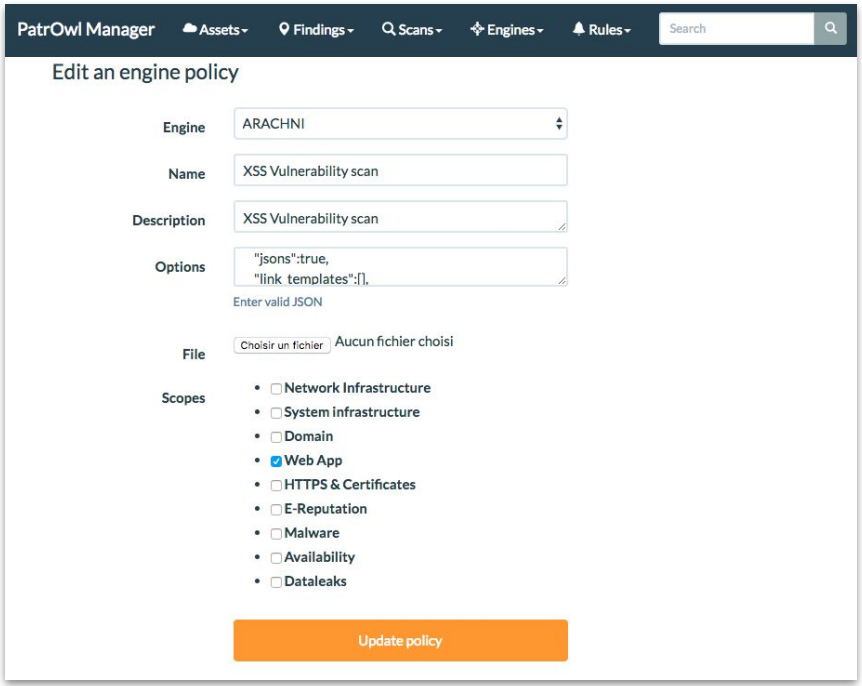
- Engines states are regularly updated and always shown in the footer:

PatrOwl Manager - Engine policy views

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

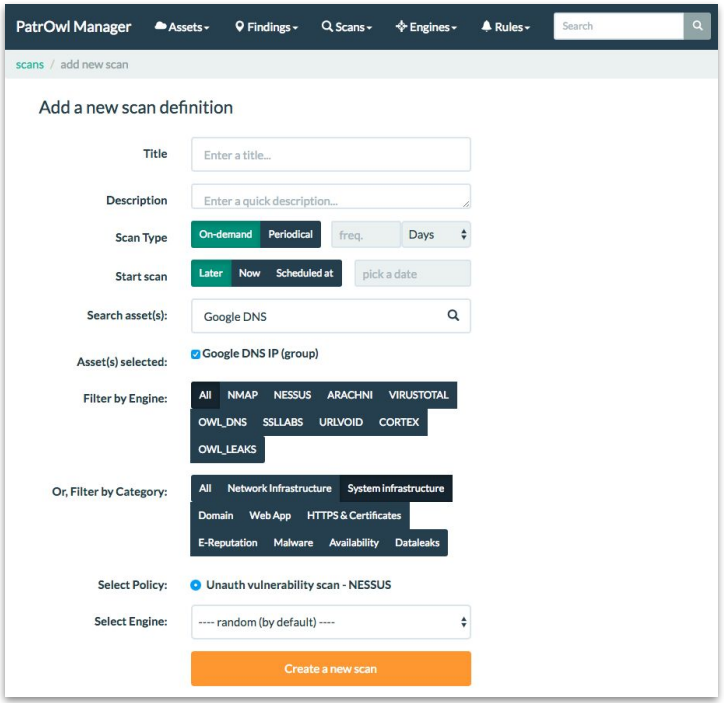


- Engine policy details:



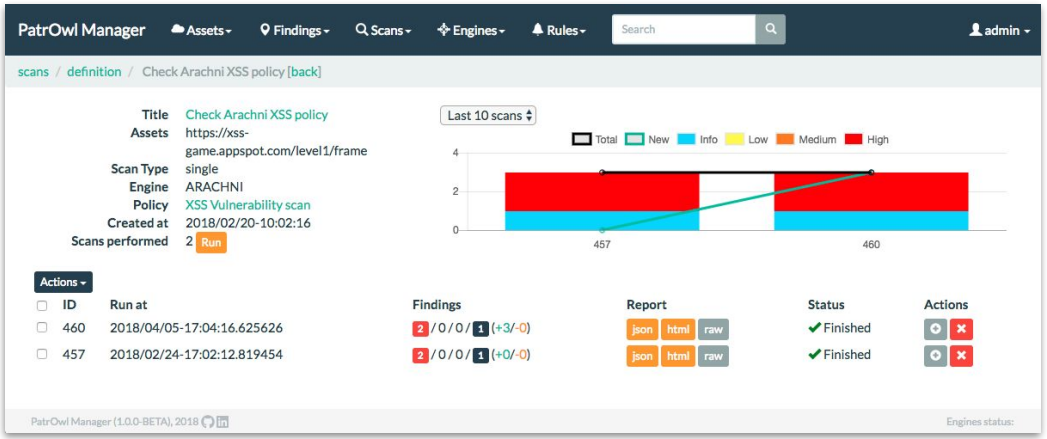
PatrOwl Manager - Scan definition creation view

- Search and select asset and asset group on theirs values or names
- 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



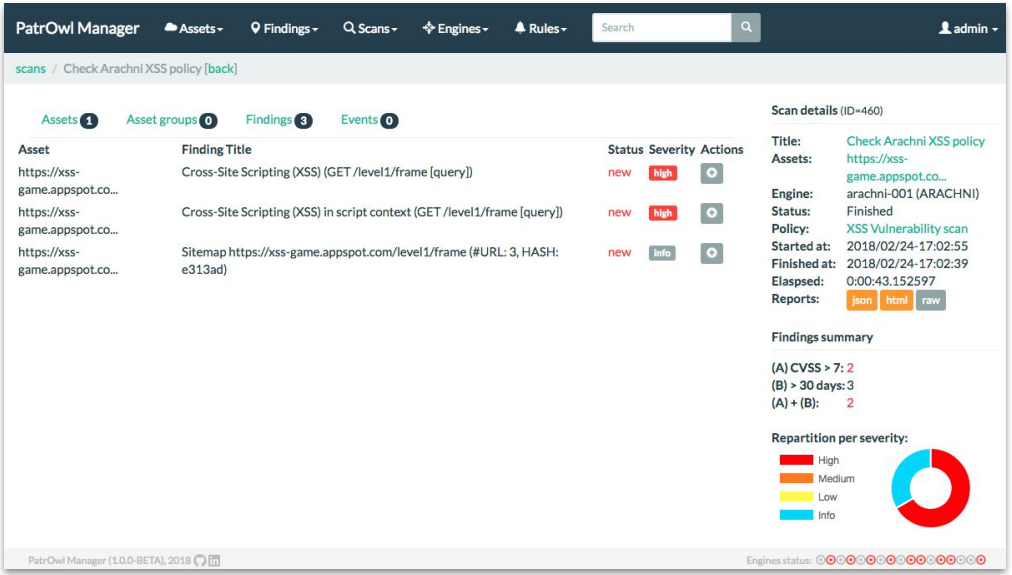
PatrOwl Manager - Scan definition view

- 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



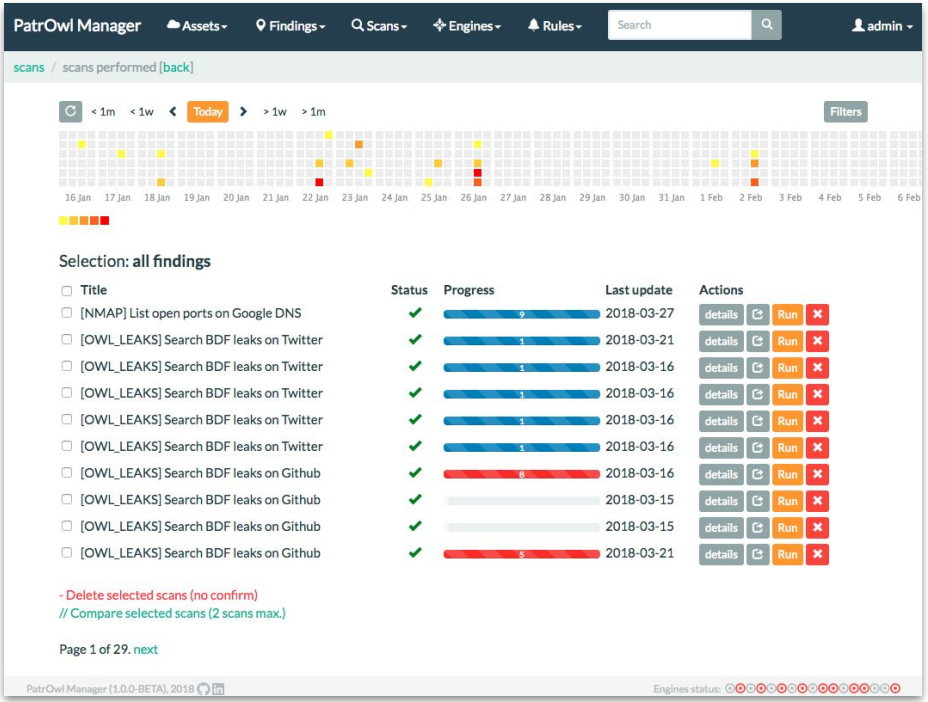
PatrOwl Manager - Scan performed view

- 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



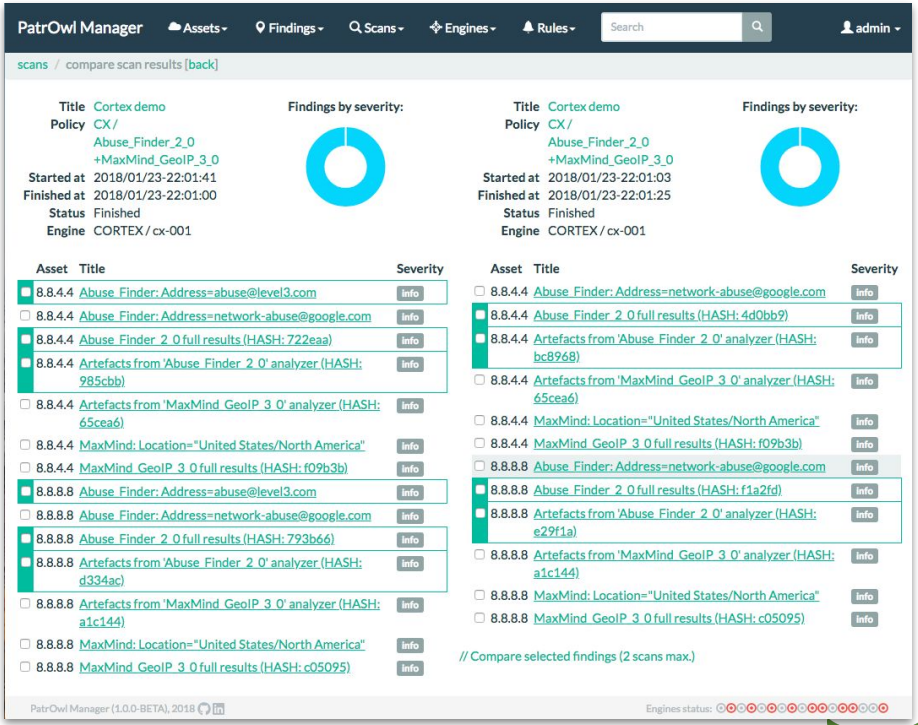
PatrOwl Manager - Scan performed view

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



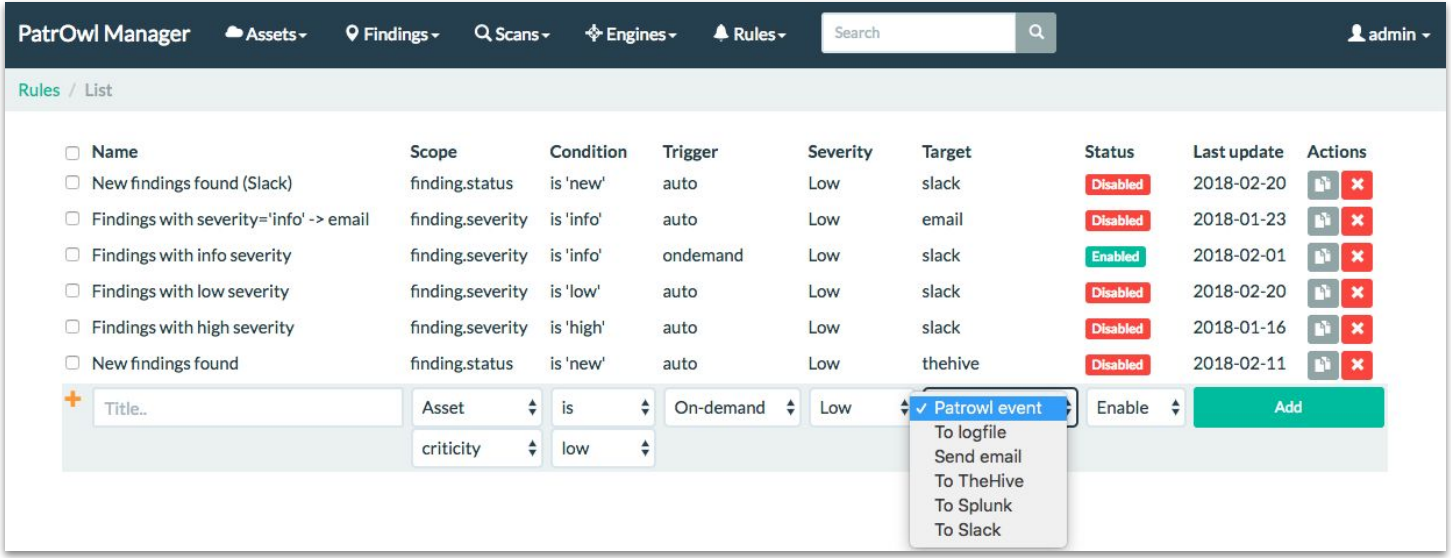
PatrOwl Manager - Scan compare view

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



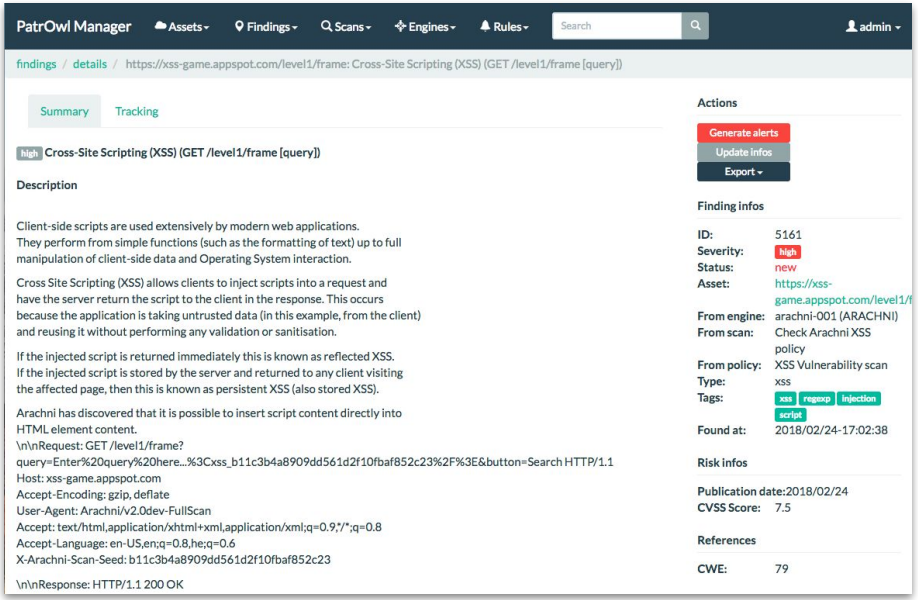
PatrOwl Manager - Alerting rules management view

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



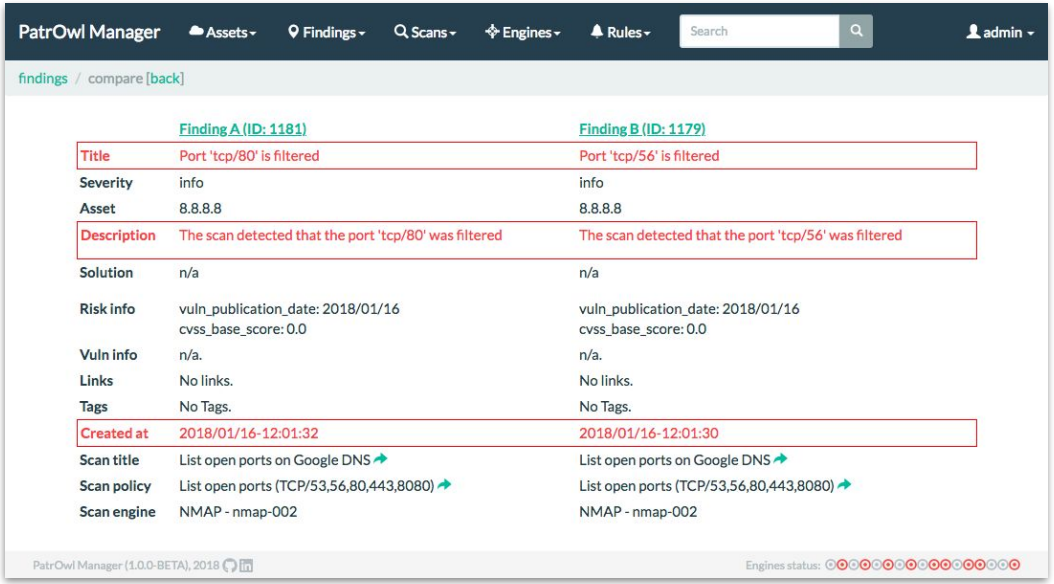
PatrOwl Manager - Finding view

- 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 finding differences



PatrOwl Engines

Features

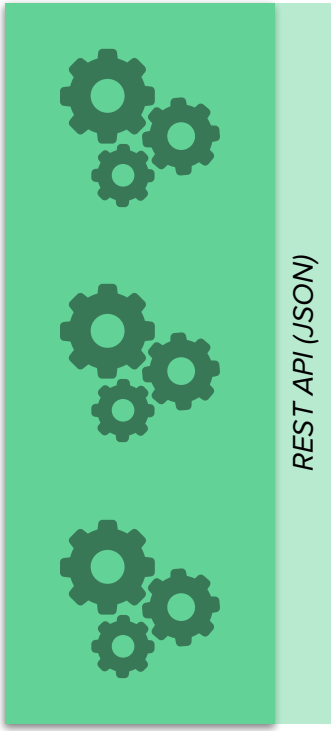
- REST API application written in Flask (Python 2.7)
- Multi-{scans, threads, assets}
- Support local or online scanners:
  - Nmap, Nessus, Cortex, Censys, Arachni, SSL-Labs, URLVoid and VirusTotal
  - owl_leaks: Keyword searches in Github and Twitter
  - owl_dns: DNS info, Subdomain listing, typosquatted domains
- Scan results (findings) are parsed, analyzed and formated
- @todo: support Basic & Token authentications
- @todo: full documentation ☐ 👌 🦃 🙏

- Meta-engine available
- Testing scripts available
- Dockerized

Metrics

- ~1 day to write a simple engine
- ~750 LoC per engine

PatrOwl Engines



Key functions

- **info():** returns engine metadata like version, name, description
- **status():** returns engine status
- **reloadconfig():** reload the config file
- **start():** checks parameters and start the scan
- **stop(*<scan_id>*):** stop the scan
- **status(*<scan_id>*):** returns the current scan status
  - FINISHED → PatrOwl will call getfindings()
  - ERROR → PatrOwl will stop the scan and raise and error
  - SCANNING → PatrOwl will retry later
- **getfindings(*<scan_id>*):** return the findings and a summary
- **getreport(*<scan_id>*):** return the raw report file(s)
- **clean(*<scan_id>*):** delete all scan-related objects