



Running a Bug Bounty Program  
for your registry or registrar

Gavin Brown, CTO, CentralNic  
Group PLC

ICANN 63, Barcelona

# Introduction

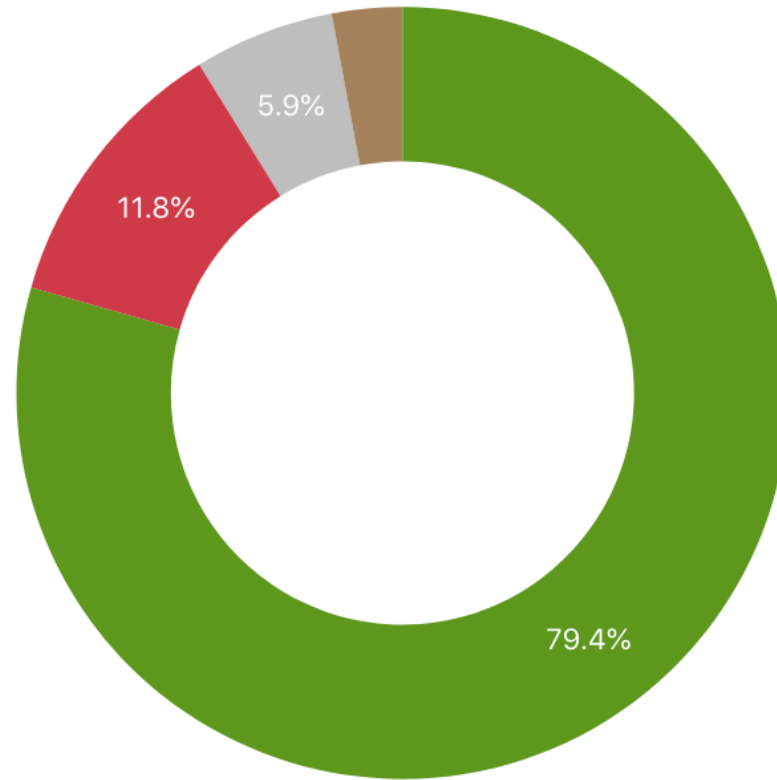
- ⌘ CentralNic Group PLC includes 4 registries and 5 registrars:
  - ⌘ CentralNic
  - ⌘ OpenRegistry
  - ⌘ KSRegistry
  - ⌘ SK-NIC
  - ⌘ TLD Registrar Solutions
  - ⌘ Internet.bs
  - ⌘ Instra Corporation
  - ⌘ Key-Systems
- ⌘ ISO 27001 certification for some parts of the business
- ⌘ Strong focus on security, especially in the registries
- ⌘ Active Bug Bounty Programs for 3 companies: CentralNic, Instra, Internet.bs

## What is a Bug Bounty Program?

- ⌘ A bug bounty program is a **continuous, crowd-sourced black-box penetration test**
- ⌘ Independent security researchers (i.e. hackers) test your systems, submit reports, and receive payment (bounties) for them
- ⌘ Used by Google, Facebook, Microsoft, IBM, Uber, Slack, Twitter, PayPal, and many others
- ⌘ Often managed through third-parties, e.g. HackerOne or Bugcrowd

# CentralNic's Bug Bounty Program

- ☞ Opened in December 2015
- ☞ Runs on HackerOne
- ☞ 451 invited hackers
- ☞ 345 reports received from 36 hackers
  - ☞ 206 legitimate reports
  - ☞ 88 out of scope/no impact
  - ☞ 34 informative
  - ☞ 15 duplicates
  - ☞ 0 spam



☞ [hackerone.com/centralnic](https://hackerone.com/centralnic)

☑ ● Resolved    ☑ ● Informative    ☑ ● Duplicate    ☑ ● Not applicable    ☑ ● Spam

# CentralNic's Bug Bounty Program

## Program Statistics

**> \$20,000**

Total bounties paid

**\$100**

Average bounty

**\$200 - \$700**

Top bounty range

**206**

Reports resolved

**37**

Hackers thanked

## Response Efficiency

**about 1 day**

Average time to first response

**about 1 day**

Average time to triage

**6 days**

Average time to bounty

**8 days**

Average time to resolution

**● 97% of reports**

Meet response standards

Based on last 90 days

## Most common reports

- ⌘ TLS config – HTTP, SMTP, XMPP, IMAP, etc – accepting weak ciphers
- ⌘ Session management
  - ⌘ Lack of rate limiting
  - ⌘ Session invalidation
- ⌘ Missing hardening headers
- ⌘ Missing XSS and XSRF mitigation
- ⌘ “information disclosure” – version numbers
- ⌘ “text injection”

## Benefits and Drawbacks

- ⌘ PRO: Many small payments spread out over time rather than a single large payment
- ⌘ PRO: Continuous testing – a point-in-time pen test report is obsolete as soon as it's written
- ⌘ PRO: Generates goodwill within the hacker/infosec community
- ⌘ PRO: Can help nudge greyhats towards becoming whitehats
  
- ⌘ CON: You're potentially "airing your dirty laundry in public"
- ⌘ CON: Paints a target on your business
- ⌘ CON: Not as widely recognised as traditional point-in-time pen tests when third parties review your security
- ⌘ CON: Needs careful management to be successful

## Starting your own program

- ⌘ Pick the provider that works best for you (support, costs, features, etc)
- ⌘ Define your scope – make it small at first, then expand
- ⌘ Make exclusions explicit – read other program’s scope for ideas
- ⌘ Make your bounty calculation rules transparent and fair
- ⌘ Use incentives to direct hackers towards areas that need more attention
- ⌘ Start private, but aim to go public
- ⌘ Provide test accounts when asked



## Dealing with hackers

- ⌘ Self-employed security professionals
- ⌘ Often starting out in their infosec careers
- ⌘ Want to leverage their hacking experience into jobs at “serious” infosec companies
- ⌘ Most are not native English speakers
- ⌘ Not always l337, might be n00bs, be patient and polite!

## Dealing with hackers

- ⌘ False positive rate is much lower than automated pen tests, but some noise still gets through.  
Always ask for evidence/PoCs if not provided
- ⌘ Set SLAs for first response, triage, resolution and bounty payment
- ⌘ Program must be supported by robust change management and QA processes
- ⌘ Try to fix everywhere to avoid the same issue being reported repeatedly
  - ⌘ Encourages best practices in operations, e.g. config management

## Calculating bounties

- ⌘ Score = Complexity x Severity
- ⌘ Complexity = how hard was it to find this bug?
- ⌘ Severity = how valuable is the asset it targets? How easy to exploit?
- ⌘ Score translates to \$\$\$
- ⌘ Always be prepared to justify your calculation and consider changing your policy in response to feedback

## What to Expect

- ⌘ Spam
- ⌘ False Positives
- ⌘ Issues with low/zero impact
- ⌘ Reports targeting web applications only
- ⌘ Reports derived from automated scanning tools
- ⌘ Your program can be tuned to improve the SNR
- ⌘ Continuous improvement of your organisation's security



CentralNic  
Group PLC

---

[gavin.brown@centralnic.com](mailto:gavin.brown@centralnic.com)

Signal: +447548243029

[hackerone.com/centralnic](https://hackerone.com/centralnic)

[centralnic.com/registry/security](https://centralnic.com/registry/security)