Cyber Security Information Indexing

COMP9313: Big Data Management

What is Cyber Security?

Wikipedia:

"Computer security, cybersecurity or information technology security (IT security) is the protection of computer systems from the theft of or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide"

Shatz et al. (2017)*:

"The approach and actions associated with security risk management processes followed by organizations and states to protect confidentiality, integrity and availability of data and assets used in cyber space. The concept includes guidelines, policies and collections of safeguards, technologies, tools and training to provide the best protection for the state of the cyber environment and its users"

* Schatz, Daniel; Bashroush, Rabih; and Wall, Julie (2017) "Towards a More Representative Definition of Cyber Security," Journal of Digital Forensics, Security and Law: Vol. 12 : No. 2 , Article 8

Security Vulnerabilities

- A flaw that arises in the design, implementation or maintenance of hardware and software
- Highly undesirable -> Can be exploited by malicious parties to gain unauthorized access to resources / assets
- Can cause financial loses, reputation damage and private data leakage
- Common examples of security vulnerabilities:
  - SQL Injection
  - Cross-site Scripting (XSS)
  - Remote Code Execution (RCE)

SQL Injection

- Occurs when a (possibly malicious) party constructs a SQL query that is executed through legitimate inputs of the system

- Allows for activities such as:
  - Reading/writing data to/from the database
  - Modifying data
  - Delete data
  - Execute admin operations

SQL Injection Example

```
// Get parameters from HTTP request
user_name = request.get("username")
password = request.get("password")

// A SQL statement vulnerable to SQL Injection
sql = "SELECT id FROM users WHERE username = '" +
        user_name + "' AND password='" + password + "'"

//Execute the SQL statement
my_database.execute(sql)
```

- Send the text below in the "password" field of the HTTP request:

  password' OR 1=1

- SQL statement actually executed

  SELECT id FROM users WHERE username='username' AND password='**password' OR 1=1**'

Common Weakness Enumeration (CWE)

- Standard to categorize software weaknesses and vulnerabilities
- Website: https://cwe.mitre.org
- Sponsored by National Cybersecurity FFRDC, which is own by The MITRE Corporation*
- Currently maintains a total of 808 different software weaknesses (CWE List Version 3.3)
- SQL Injection's CWE ID:
  - CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

(*) https://en.wikipedia.org/wiki/Common_Weakness_Enumeration

National Vulnerability Database (NVD)

- U.S. government repository of standard-based vulnerability management data
- Originally created in 2000 (as the Internet Categorization of Attacks Toolkit or ICAT)
- Includes:
  - Database of security checklist references
  - Security-related software flaws
  - Misconfigurations
  - Product names
  - Impact metrics
- Vulnerabilities are identified by its CVE (Common Vulnerability and Exposures)

https://nvd.nist.gov/

Example: Heart bleed (CVE-2014-0160)

QUICK INFO

CVE Dictionary Entry:
CVE-2014-0160
NVD Published Date:
04/07/2014
NVD Last Modified:
03/25/2019

"The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information..."

Impact

CVSS v2.0 Severity and Metrics:

Base Score: 5.0 MEDIUM
Vector: (AV:N/AC:L/Au:N/C:P/I:N/A:N) (V2 legend)
Impact Subscore: 2.9
Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): None
Availability (A): None
Additional Information:
Allows unauthorized disclosure of information

https://nvd.nist.gov/vuln/detail/CVE-2014-0160

Some Stats About Security Vulnerabilities

11,460

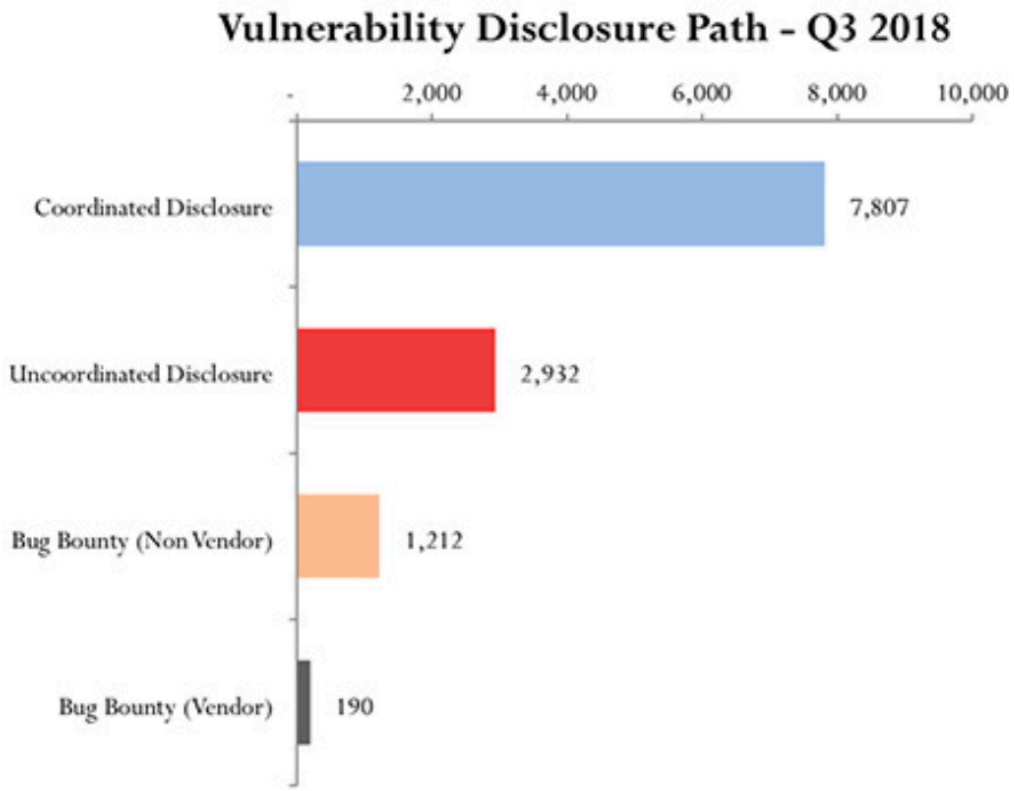
Vulnerabilities YTD 2019

69,291

Vulnerabilities Missing From CVE

207,289

Vulnerabilities All Time



As of 13/07/2019

Sources:
https://vulndb.cyberriskanalytics.com/#statistics
https://www.helpnetsecurity.com/2018/11/20/2018-q3-vulndb-quickview-report/

Why do we have security vulnerabilities?

- Producing software that's free of vulnerabilities is uncommon and extremely difficult specially in complex software systems

- Situation gets worst in scenarios with high requirements volatility

- No (functional) system is 100% secure*
  - We should at least focus on trying to discover vulnerabilities and fix them

(*) https://hbr.org/2017/12/you-cant-secure-100-of-your-data-100-of-the-time
http://dodccrp.org/events/12th_ICCRTS/CD/html/papers/108.pdf

Vulnerability Discovery Techniques

- Expert-based vulnerability discovery
  - Involves experts in security vulnerability
  - In-house or outsourced

- Examples:
  - Manual code review
    - Manual inspection of software code
    - Highly depends on knowledge and experience of code reviewer
  - Manual penetration testing
    - Simulation of an unauthorized attack to a system
    - More of an art, rather than a science
    - Requires not only expertise in security vulnerability but also creativity

Vulnerability Discovery Techniques

- Machine-based vulnerability discovery
  - Relies on automation for discovering vulnerabilities
  - Data Mining, Machine Learning, AI, etc.

- Examples:
  - Static Analysis
    - Analysis of code without running it
    - Large number of false-positives
  - Black-box vulnerability scanners
    - No need to access source code of software
    - Analyzes input vectors (test them with specially-crafted inputs)
    - Observes behaviour and raises an alert if the input triggers a vulnerability

Vulnerability Discovery Techniques

- Crowd-based vulnerability discovery
  - Relies on the crowd (Security Professionals)
  - Organizations call for crowd collaboration to find vulnerabilities

- Examples:
  - Organization-managed programs:
    - Mozilla's Security Bug Bounty Program (website)
    - Rewards (monetary and non-monetary)
  - Bug Bounty platforms
    - HackerOne
    - BugCrowd

Security Vulnerability Information Resources

Security Vulnerability Indexing and Searching

Security Vulnerability Information Resources

Issues:

- Multiple (lots) of security vulnerability sources
- Heterogenous data (e.g. HTML, JSON, plain text)
- Distributed, information silos
- Lack of information integration
- Multiple, heterogenous query interfaces (keyword search, REST API calls, DSLs)

Why all this is not enough?

- Many of these information sources are rather “old” (born in the 90s)
- Scattered vulnerability information: Each product manage its own vulnerability data
- Much of the information is not suitable for vulnerability scanners
- Efforts to standardize vulnerability information is not enough (Mitre, NVD, Oval, etc.)
- Comprehensive search for vulnerabilities? No, use Google.

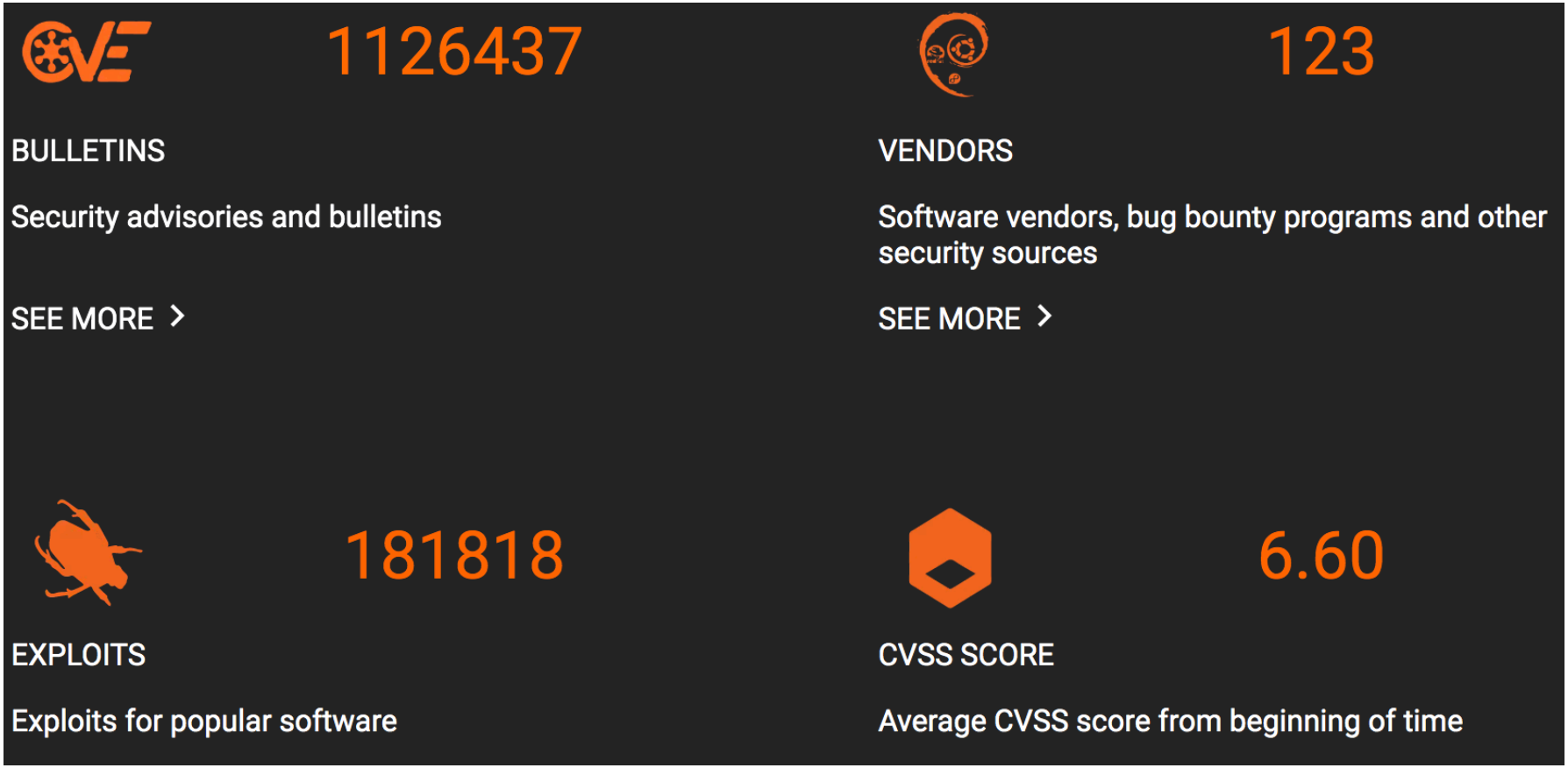
https://www.slideshare.net/AlexanderLeonov2/vulnerability-intelligence-and-assessment-with-vulnerscom

Vulners

- Vulnerability database and search engine with Open API
- Aggregates vulnerability data
- Developed by security experts, for security experts
- Fast search engine (ElasticSearch)
- Machine readable data (JSON)
- API-driven development

https://www.slideshare.net/AlexanderLeonov2/vulnerability-intelligence-and-assessment-with-vulnerscom

Some stats...

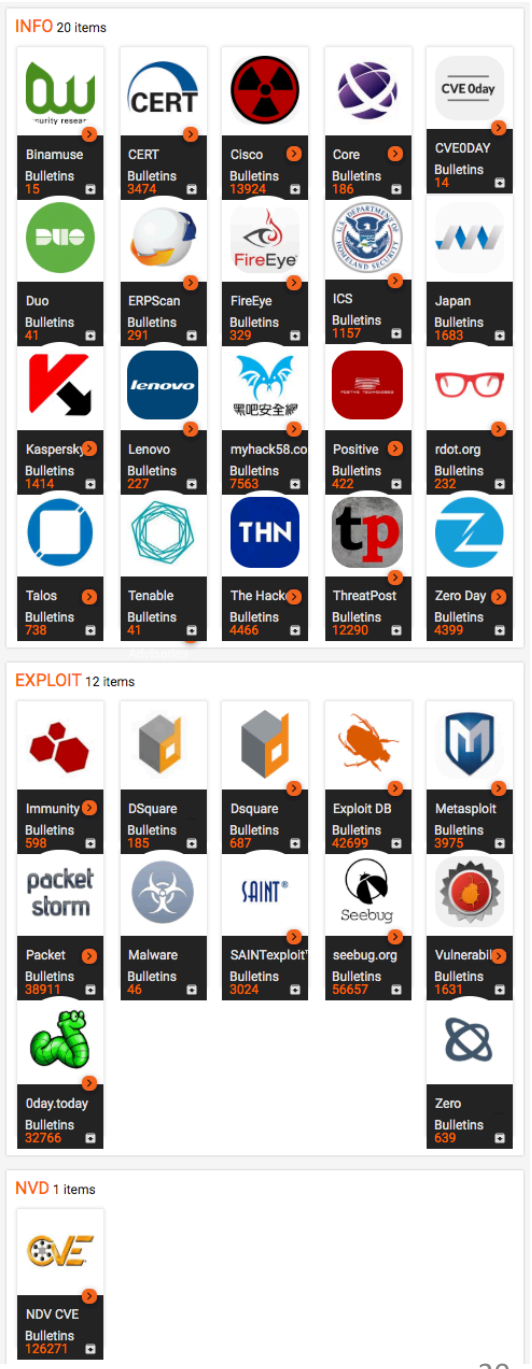


(as of 14/07/2019)

https://vulners.com/stats

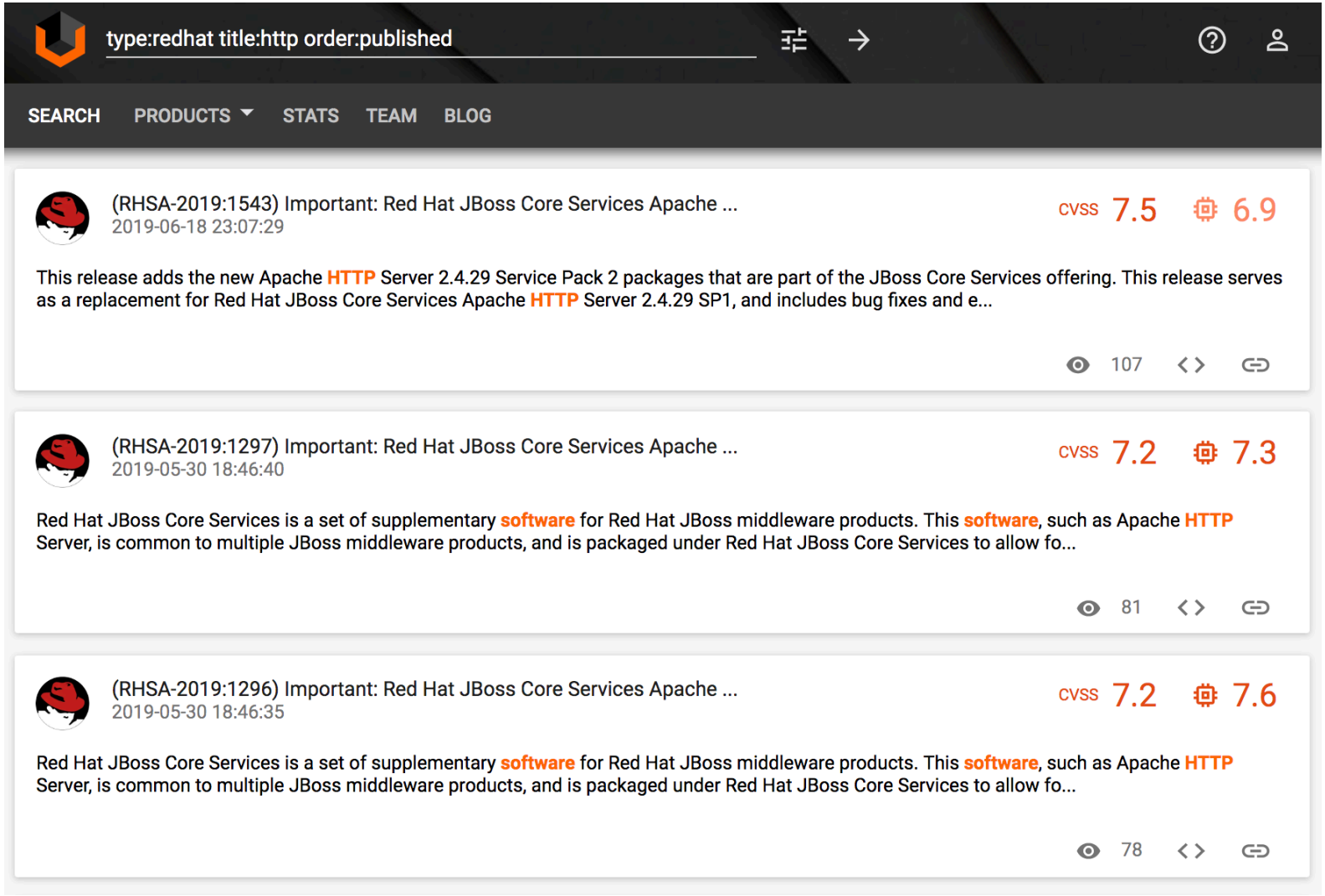
Bulletin families and vendors

- UNIX
  - IBM AIX
  - Amazon Linux AMI
  - RedHat Linux
  - …
- Exploit
  - ExploitDB
  - Metasploit
  - Dsquare
  - …
- Scanner
  - Tenable Nessus
  - NMAP
  - OpenVAS
  - …
- *and more…*



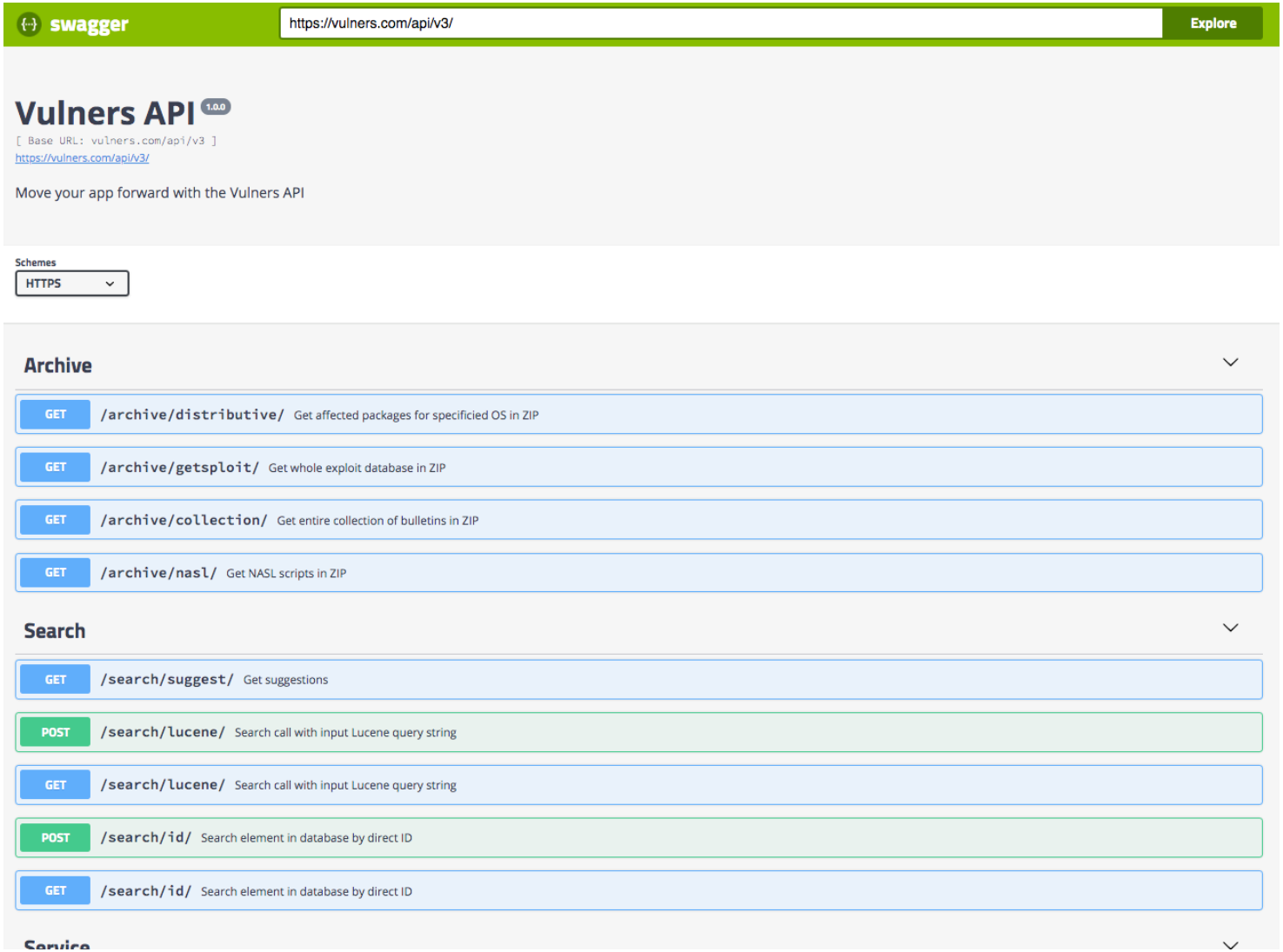
https://vulners.com/stats

Bulletin families and vendors



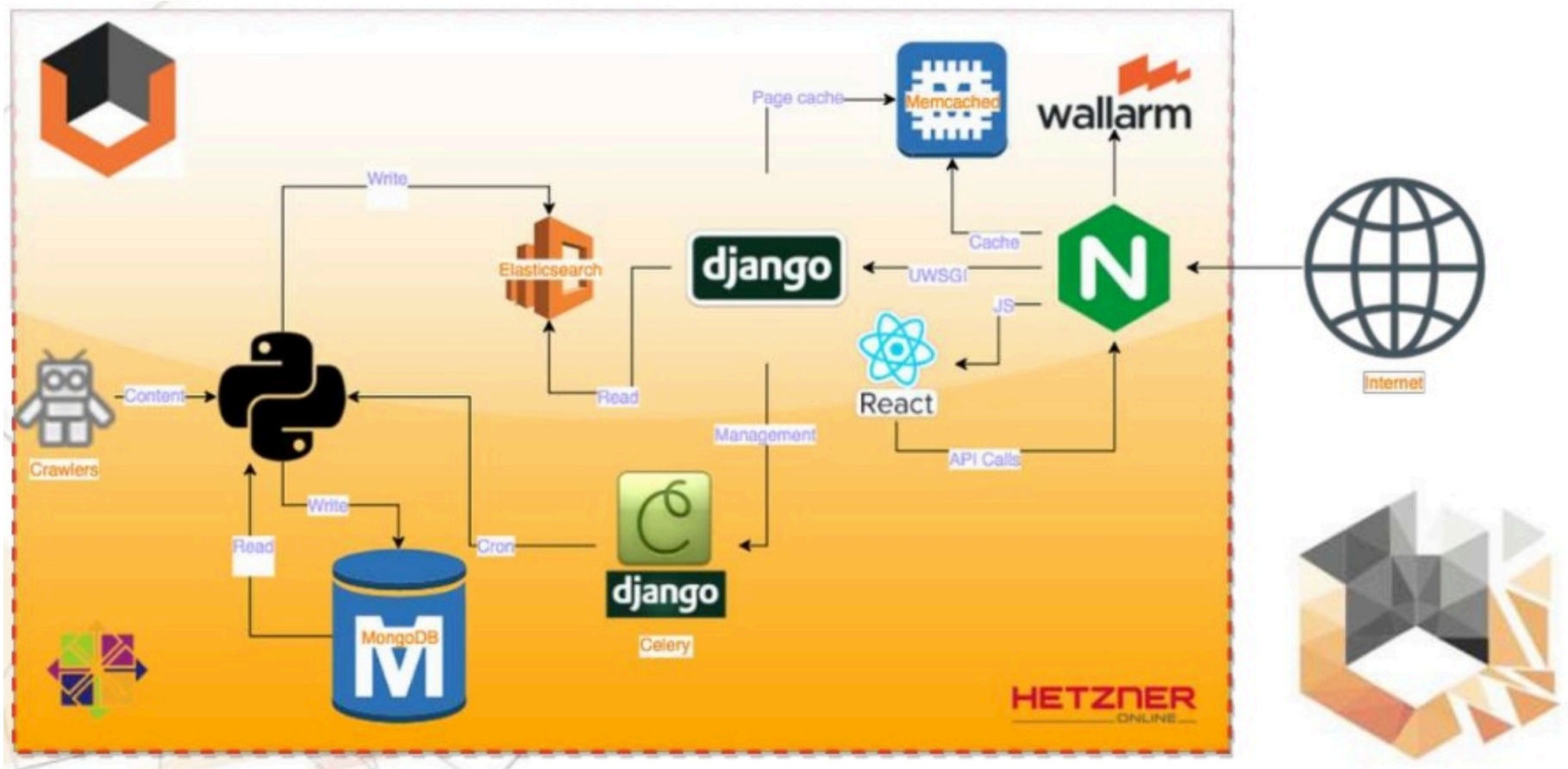
https://vulners.com/stats

Vulners APIs



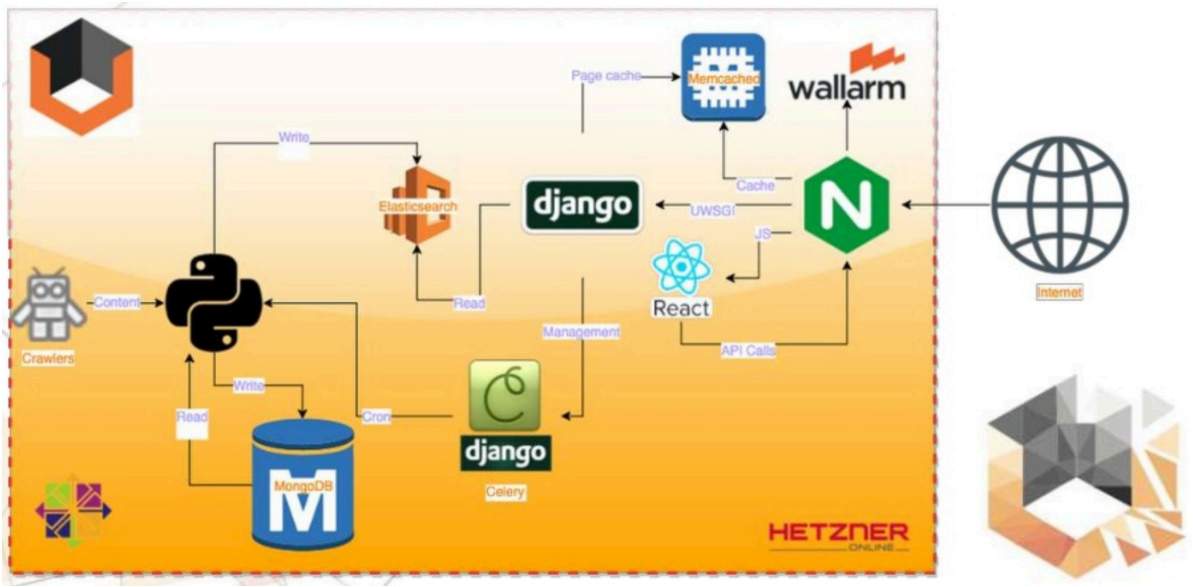
https://vulners.com/docs

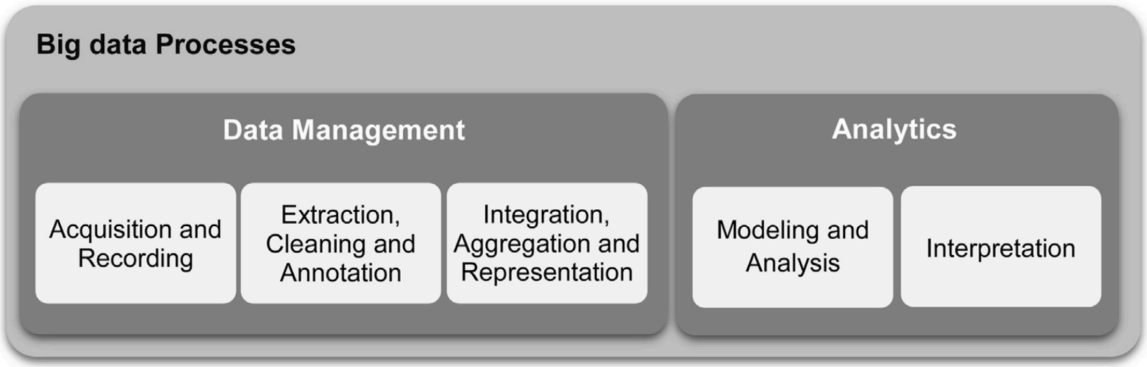
Architecture and Technologies



https://www.slideshare.net/AlexanderLeonov2/vulnerability-intelligence-and-assessment-with-vulnerscom

Mapping to Big Data Processes





https://www.slideshare.net/AlexanderLeonov2/vulnerability-intelligence-and-assessment-with-vulnerscom
https://www.sciencedirect.com/science/article/pii/S0268401214001066

Thanks