SECURE DATA GI

RELEASE DATE: 19 JANUARY 2015
CONFIDENTIAL

THREAT NAME:

CVE 2015-1427

AFFECTS:

ELASTIC SEARCH

SEVERITY RATING:

4

HIGH

RESPONSE TIME:

IMMEDIATE ACTION REQUIRED

THREAT ADVISORY

SUMMARY

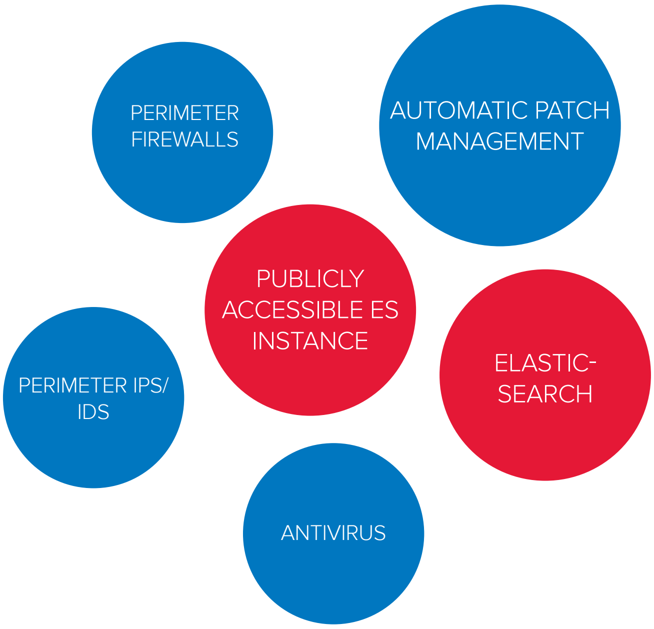
CVE-2015-1427 is a **vulnerability** in the **ElasticSearch** component of the wildly used ElasticSearch Kibana Logstash (EKL) data storage and analytics package. It **allows** a **remote attacker** to **execute code** via the **ElasticSearch web interface**. The vulnerability exists in the Groovy scripting engine and allows an attacker to **bypass the sandbox protection** mechanism. Without the sandbox protections, an attacker is able to execute server side code.

We are giving this a **High Severity Rating** as it can be **triggered remotely,** and allows arbitrary code execution on the server hosting the ElasticSearch instance. There is also a public **Metasploit exploit module available**. From an architectural point of view **ElasticSearch should not be exposed to the Internet** so the risk is slightly reduced, but the **default configuration** of ElasticSearch exposes the application to **all network interfaces** on the server, relying on the administrator to either change the configuration or place it behind a firewall.

A patch has been made available which mitigates this attack. Our recommendation is; an immediate application of the patch, to remove the ElasticSearch instance from unnecessary network interfaces and to explicitly define allowed source connections through a firewall.

TRIAGE CHART

This section is a quick lookup of what controls or technologies that, if in place, **limit the threats effectiveness**; that **cannot be used to influence the threat**, or **that the threat targets.** The size of each items shows how effective each one is. **These common systems are not affected**.



PERIMETER FIREWALLS

AUTOMATIC PATCH MANAGEMENT

PUBLICLY ACCESSIBLE ES INSTANCE

ELASTIC-SEARCH

PERIMETER IPS/ IDS

ANTIVIRUS

ATTACK PROCESS

This attack relies on a feature of ElasticSearch which allows clients to execute Groovy code (giving users an extra layer of intelligence in their searches) as part of the _search API. An attacker starts by submitting Groovy code to the ElasticSearch instance via a web call (see Appendix 1), which is executed by the server as part of its native scripting capabilities. The supplied code is executed in a sandbox environment, checking for any attempts to execute dangerous code, and returns a result back to the client. As there are no authentication controls built into ElasticSearch, if the attacker manages to bypass the sandbox protection she should be able to force the server to execute other commands that would allow her to take control of the server by downloading and installing malware.

The sandbox first checks what functions and classes are allowed to be executed through the scripting interface. It does this via the GroovySandboxExpressionChecker.java class, which uses two checks. First, it checks against a blacklist of method calls by calling methodBlacklist. Next, it ensures that the method is not null or uses a GStringExpression. In addition to these two checks, the sandbox also restricts packages that can have their methods called through the use of a whitelist defined by defaultReceiverWhitelist.

Chaining these methods together, and with the help of a technique known as reflection, an attacker is able to take a benign package class like java.lang.Math, and use it to load a reference to a potentially malicious class, such as java.lang.Runtime. The latter class can be used to execute system commands on the underlying operating system. This effectively functions as a pseudo-shell to the operating system's command line directly from ElasticSearch, which can be used to control the server or download more capable malware for example.

ATTACK PATH

1. Identify ElasticSearch instances that are reachable from the internet
2. Prepare a crafted query that bypasses the sandbox and downloads malware
3. Send the query to the ElasticSearch instance's web interface, causing it to download and execute the malware
4. With access to the underlying operating system, use privilege escalation methods to gain a higher level of access
5. All data and resources of the server are under the attackers control

SEVERITY

This vulnerability gives an attacker the ability to execute their own code on a vulnerable server, which will allow them to take control of the server. ElasticSearch should strictly not expose its web API to the Internet, limiting the vulnerabilities exploitation. However, ElasticSearch binds to all network interfaces by default, which increases the chance that some installations are accidentally vulnerable. We are giving CVE-2015-1472 a High Severity Rating because it is simply and reliably exploitable, and there is a public Metasploit exploit module available for it.

DETECTION

A patch has been issued, and all versions of ElasticSearch prior to 1.4.3 are vulnerable. You can check the version of an ElasticSearch instance with the following command:

```
curl 'http://localhost:9200/?pretty'
```

The "version" variable discloses what version of ElasticSearch the cluster is running.

Alternatively you can use the web query in Appendix 1 to see if the instance will execute commands.

By default ElasticSearch does not log the queries that a client executes through the instance, leaving no way to audit for exploitation attempts. Other egress filters (such as firewalls, transparent proxy servers and DNS servers) could be used to check the external

REMEDIATION AND PREVENTION

Remediating this vulnerability is difficult because an attacker could have established enough of a foothold on the server that they could have covered their tracks in the logs. If a positive confirmation of exploitation of a server has been found, we advise reinstalling the server to limit the change of persistent infections. All instances that may have been exposed, or continue to be exposed, should be closely monitored for abnormal traffic or suspicious activity.

A patch has been issued for this vulnerability, making its application the easiest and suggested method of prevention. If this is not possible, or in addition to applying the patch, access to the ElasticSearch API should be limited by a control such as a firewall.

APPENDIX

1. ELASTICSEARCH INSTANCE EXECUTION CHECK

The following script will try to execute code through the ElasticSearch instance in the same way as described above. If the output returns the myscript field in the same way as below then the instance is vulnerable.

```
curl http://localhost:9200/_search?pretty -XPOST -d ‘{
 “script_fields”: {
   “myscript”: {
     “script”: “java.lang.Math.class.forName(\”java.lang.Runtime\”)”
   }
  }
}’

//////OUTPUT//////

{
  <snip>
  “hits” : {
    “total” : 8,
    “max_score” : 1.0,
    “hits” : [ {
      <snip>
      “fields” : {
        “myscript” : [ “class java.lang.Runtime” ]
      }
    }
}}
```