CGILua session.lua Predictable Session ID

Discovery by Felipe Daragon

Presented by James Mouat

Who Discovered This?

Felipe Daragon

- Based in Rio de Janeiro, Brazil
- Founder and CEO of Syhunt
- Syhunt has researched over 29,000 Web Vulnerabilities since 2003
- Security researcher and self-taught genius
- Also likes Piña coladas & breaking software

About the Language

Lua (/ˈluːə/ ***LOO****-ə*, from Portuguese: *lua* [ˈlu.(w)ɐ] meaning *moon*)

- lightweight multi-paradigm language
- designed as a scripting language
- extensible semantics as a primary goal
- cross-platform since it is written in ANSI C
- has a relatively simple C API

- Source Wikipedia (http://en.wikipedia.org/wiki/Lua_(programming_language)

About the CGILua

CGILua

- Lua program executed (parsed) by webserver
- Conventional mark-up language with tags
- Allows for separation of concerns
- Comes with a plethora of libraries, including one for handling sessions… kinda…

- Source Project Maintainers GitHub page (https://github.com/keplerproject/cgilua)
- Source Project Maintainers project page (http://keplerproject.github.io/cgilua/)

Vulnerability Overview

CGILua Session Handling

- Library handing sessions, generates a weak SID
- Source code is freely available on GitHub for an attacker to quickly identify this mechanism
- Possible to guess/predict/steal valid Session ID through brute force attacks
- Affects versions: CGILua 5.0.x, CGILua 5.1.x., CGILua 5.2 alpha 1 & CGILua 5.2 alpha 2

- Syhunt Advisory (http://www.syhunt.com/?n=Advisories.cgilua-weaksessionid)
- Full Discosure (http://seclists.org/fulldisclosure/2014/Apr/318)

Timeline

- 27th March – Felipe emailed maintainers about the need of hardening CGILua

- 2nd April – Felipe emailed the maintainer once again.

- 2nd April – Maintainer responds to Felipe with the belief that:

"…session IDs generated by CGILua 5.0 and 5.2 are not insecure in its current form and that enhancing the randomness of the SID would not make it more secure."

Timeline - Continued

- 4th April – Felipe, frustrated with maintainers, reaches out to James and details his findings.
- 4th April – Felipe sends maintainer information about recommended Session ID length and entropy: https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Session_ID_Length
- 7th April – Felipe sends James a working copy of the latest libraries which he is testing against.
- 10th April – James agrees with Felipe's findings; shakes head.

Timeline - Continued

- 13th April – Felipe sends developer details of demonstration tool that is able to guess CGILua session IDs.
- 16th April – Felipe reserves a CVE allocation
- 30th April – Felipe still yet to received a response from maintainers to emails sent on April 4th & 13th.
- 30th April – Public disclosure as CVE-2014-2875
  - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-2875
  - http://seclists.org/fulldisclosure/2014/Apr/318

Timeline – Profit?

- 4th May – Felipe sends James an email about an unconfirmed security review found cgilua in use by Brazilian voting system.

- 5th May – Felipe forwards me a link to a story hitting the news –

‘...failure in Brazilian ballot during a safety test...”

http://g1.globo.com/tecnologia/blog/seguranca-digital/post/falha-na-urna-brasileira-reproduzia-fielmente-erro-de-1995-diz-professor.html

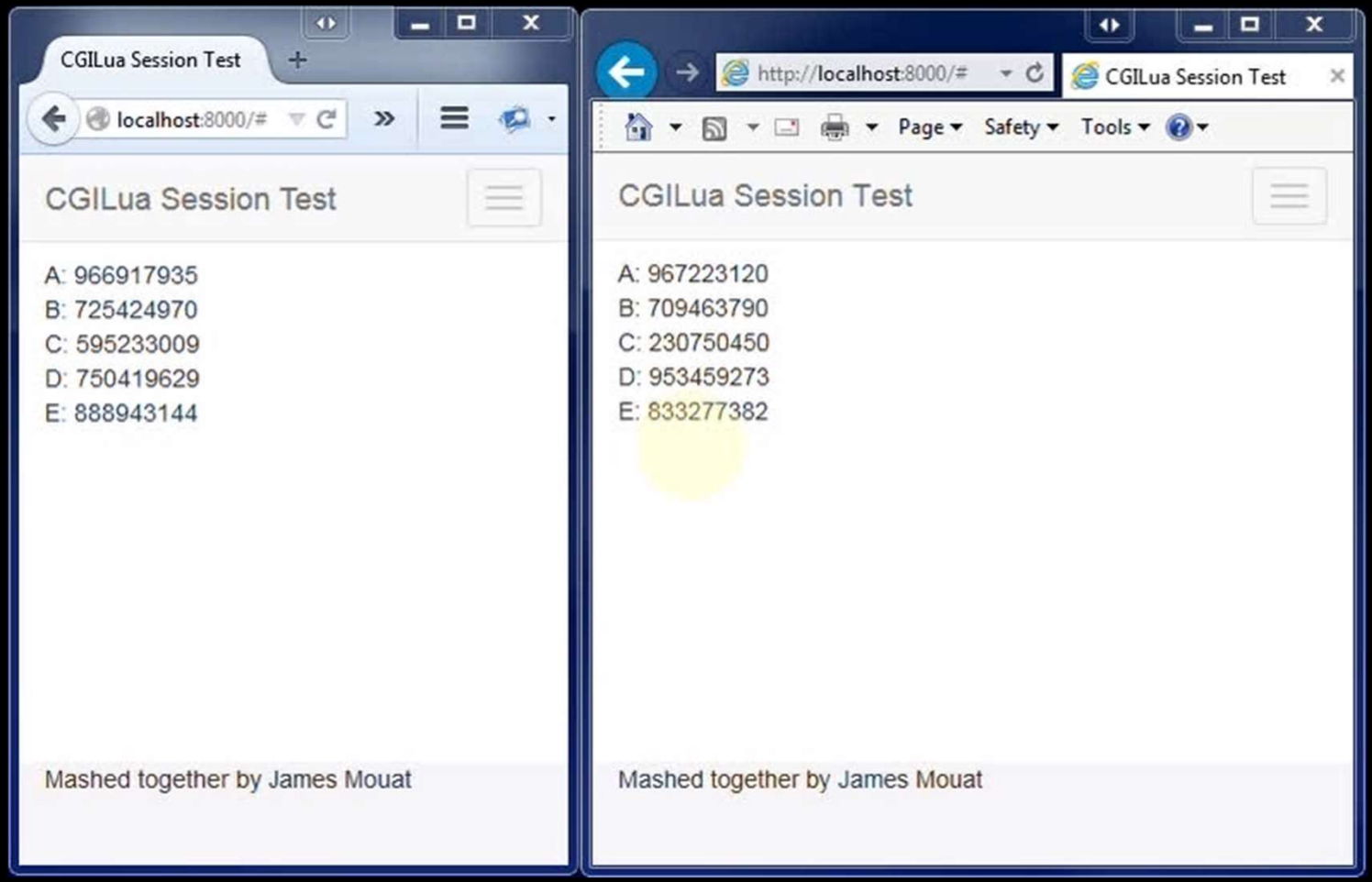
Under the Hood

Create New Session (cgilua\session.lua)

```
70 --¶
71 -- Creates a new identifier.¶
72 -- @return New identifier.¶
73 --¶
74 local function new_id ()¶
75   » return rand (RANGE)¶
76 end¶
77 ¶
78 ----------------------------------------------------------------------------¶
79 -- Creates a new session identifier.¶
80 -- @return Session identification.¶
81 ----------------------------------------------------------------------------¶
82 function M.new ()¶
83   » local id = new_id ()¶
84   » if find (id..".lua") then¶
85   »   » randseed (mod (time (), RANGE))¶
86   »   » repeat¶
87   »   »   » id = new_id (id)¶
88   »   » until not find (id..".lua")¶
89   » end¶
90   » return id¶
91 end¶
92 ¶
93 ----------------------------------------------------------------------------¶
```

Amusing Example

Fourfox vs iExploiter



How Pervasive ?

https://www.google.com/#q=inurl:cgilua.exe