

ENTRY // Xiongmair Security Research #7 // Brute Force

[\[- \]](#) [\[X \]](#)

Date: 2019-09-04 Author: Ian Rash Index: #7

Looking at the hashing format, we know it's going to have collisions. To find out the plain-text password to the backdoor user account, we are going to need to take some time and brute force the hash to find the plain text password. I just want to point out, this step is mostly unnecessary, since the hash is as good as the plain text password when logging into the camera. Regardless, it would still be a good idea to crack it, just in case.

```
require "./dahua_hash"

module Brute
  def self.run(hash : String, start = "a") : String
    current = start
    counter = 0
    success = false

    start_time = Time.now
    until success
      if Dahua.digest(current) == hash
        puts "SUCCESS!!!"
        success = true
        break
      end

      counter += 1
      current = current.succ

      if counter % 1_000_000 == 0
        puts " @ #{current} : #{Time.now - start_time}"
      elsif counter % 10_000 == 0
        print '.'
      end
    end
    end_time = Time.now

    puts "Time: #{end_time - start_time}"
    puts "Result: #{current} : #{Dahua.digest(current)}"
    current
  end
end
```

We know the details of the "user" account, so all we need to do is plug it in and BAM!

```
Brute.run("OxhlwSG8")
```

We end up getting back the string "tluafed", or "default" backwards, after about 16 or so hours.

Looking up this string provides an [interesting article](#) which describes a method of testing to see if the camera is a Xiongmai, by going to a specific htm page, err.htm.



So now we know for certain that the camera is actually a Xiongmai product, not Besder.

TAGGED SKILLS:

🛠️ Exploitation 🛡️ Security Research </> C