北京大学
PEKING UNIVERSITY

口令强度检测

2018级信息安全工程小组报告

小组成员：胡志伟、卿山、郁星遥

北京大学

目录：

1.密码强度检测对用户设置密码的影响

2.口令强度检测算法现状及评估

3.口令强度检测方法研究

4.演示系统展示

北京大学

第一部分

密码强度检测对用户设置密码的影响

密码强度检测对用户设置密码的影响

引入介绍

- 密码强度检测通常在用户创造一个新账户时出现。现在很多的网站都会在用户创造或改变密码时为用户提供密码强度检测。

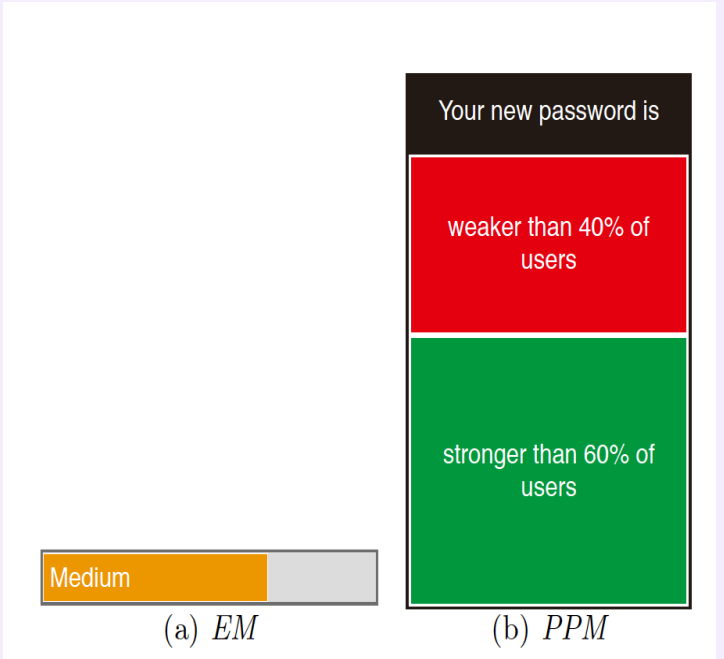
- 这些密码强度检测实时更新来展示你的密码是强的还是弱的。

- 但是密码强度检测有一个假设的前提是强密码是每个用户都希望得到的，那些选择了弱密码的用户也是这样的，他们选择弱密码的原因是因为他们没有意识到他们的密码是弱的。当他们通过密码测量器意识到他们的密码是弱的时，他们将想要重新选择一个强密码。

密码强度检测对用户设置密码的影响

实验

- 主要进行了两个实验：第一个是实验室实验，第二个是场景实验。
- 在第一个实验中只是单纯的评估密码强度检测对用户密码设置的影响。
- 在第二个实验探索了两个不同的使用案例：密码被用于保护敏感账户和密码被用于保护不重要的账户。
- 我们则测试了两个类型的密码强度测量：传统的显示强和弱密码测量器，和我们开发的一种新型密码强度检测，用于显示相对于系统上其他用户的密码强度

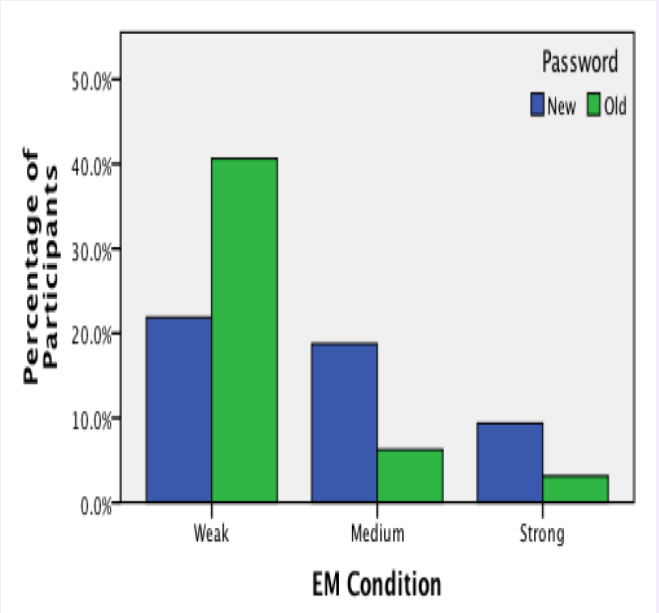


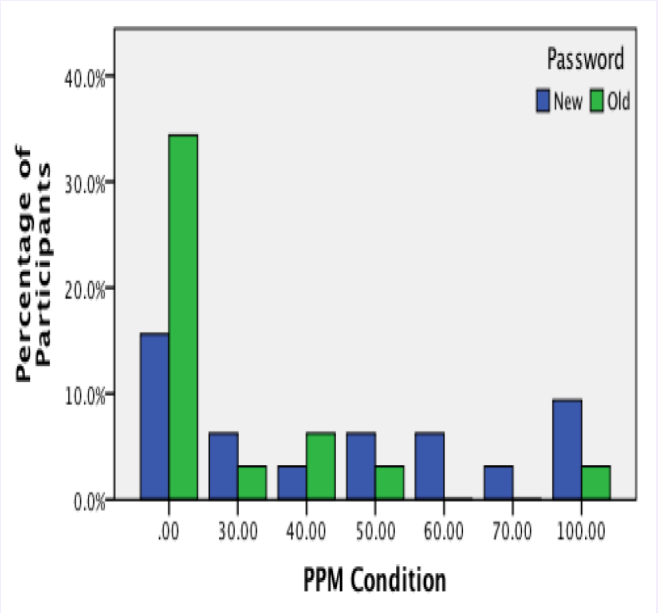
(a) *EM* (b) *PPM*

密码强度检测对用户设置密码的影响

实验结果

Bit Strength (x)	PPM	EM
x<=53.41	0%	Weak
53.41<x<=56.53	30%	Weak
56.53<x<=59.83	40%	Medium
59.83<x<=64.26	50%	Medium
64.26<x<=71.09	60%	Medium
71.09<x<=77.21	70%	Strong
77.21<x<=82.27	80%	Strong
82.27<x<=83.30	90%	Strong
83.30<x	100%	Strong





- 第一个实验中的密码强度测量都对用户产生了如图所示的相对积极影响。
- 第二个实验中当参与者为不重要的帐户创建密码时，我们没有观察到可能归因于密码强度检测产生的影响。但是当参与者创建重要的账户中的密码时，他们选择了显著更强的密码。

密码强度检测对用户设置密码的影响

结论

当前一些制造密码强度检测的动机似乎是相信用户不能理解他们的密码是弱的。而我们研究的结果显示是这个假设是有问题的。在很多情况下，我们发现了参与者知道他们选择了弱的密码。至少在一些不重要的账户，他们知道他们使用的密码是经常使用的，而且弱的。他们知不知道密码的强弱不是一个问题，他们知道后不想去改变才是一个问题。

北京大学

第二部分

口令强度检测算法现状及评估

设计一个可靠的检测计所面临的挑战

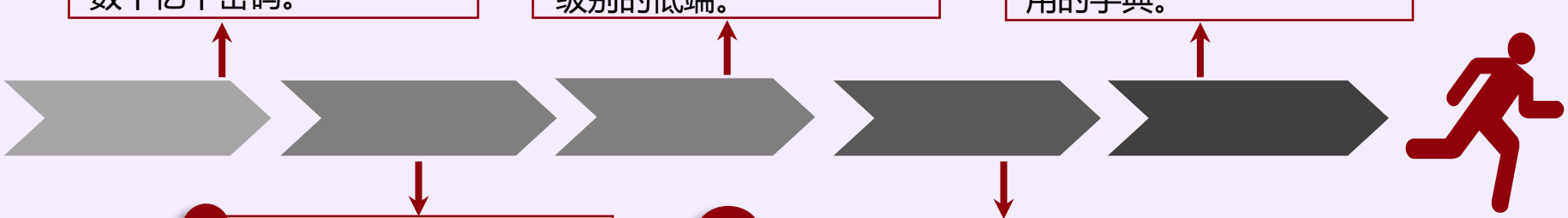
1. **在线攻击，离线攻击**：在一次在线攻击中每秒可以猜测出的密码很少，而在一次离线攻击中，每秒可以测试出数十亿个密码。

2. **真实世界的密码泄露**：大量用户使用的强泄漏密码，最有可能被集成攻击字典中，因此应该被禁止，或者至少应该被赋予较低的分数。

3. **密码**:只要不遵循简单的语法结构，密码短语就可以为用户提供更好的熵.然而大多数用户将密码排列在强度级别的低端。

4. **字典的相对性能**：一个更好设计的目标字典可能被证明对检测表更有效。即使我们的基本的字典被分配了较低的分数，用户也会尝试用其他简单的模式创造性地绕过这些限制。

5. **对检测器的影响**：一些因素可能会影响理想密码检查器的设计，包括:用户选择中的固有模式、破解工具中使用的字典。



数据来源：From Very Weak to Very Strong- Analyzing Password-Strength Meters

密码强度检测考虑的因素

- 字符集和长度要求。
- 强度刻度和标签。（强-弱，强-中-弱等）
- 用户信息。（考虑与用户相关的环境参数，例如用户的真实/帐户名称或电子邮件地址。）
- 检测位置的类型。（客户端检测，服务器端检测，两者结合）
- 多样性。（不同网站采用不用的密码检测程序）
- 熵估计和黑名单。
- ……

不同的检测算法

算法	特征
Heuristic/NIST	基于长度和遵从熵估计组合策略的启发式算法，还考虑了密码通过普通字典检查的额外好处。
Markov Model/OMEN	2012年Castelluccia等人提出对服务密码的n维 Markov模型进行训练，以提供准确的强度估计。因此，估计是基于密码组成的n维的概率。
Heuristic/Comp8	一种评分算法，这个计分函数被用来估计密码的强度。
Heuristics/zxcvbn	基于高级启发式，通过包含字典、考虑网络用语转换、键盘遍历等扩展LUDS方法。
PCFG/fuzzyPSM	对基础词典进行篡改规则修改，以匹配更强密码的训练分布。
Heuristic/Eleven	使用与NIST相似的强度度量。与LUDS方法类似，该仪表考虑字符集大小和长度。
RNN/DD-PSM	使用递归神经网络进行密码概率建模。作者还描述了一种允许使用特殊编码和Bloom过滤器的客户端实现的方法。
Heuristic/LPSE	它将密码转换为LUDS向量，并使用上述相似性度量将其与标准化的强密码向量进行比较。

PCFG：上下文概率无关法　RNN：神经网络

总结

总结

- 可用性VS安全性：用户可能会被迫使用更严格的密码，这可能会招致用户的不满。一种明显更好的方法是向用户提供关于他们所选密码质量的适当反馈，希望这种反馈能够影响用户自愿选择更好的密码。对于这种方法，密码强度检测计在提供反馈方面起着关键作用，应该以一致的方式提供反馈，以避免可能的用户混淆。在我们的大规模实证分析中，很明显，常用的密码强度计高度不一致，无法对用户的选择提供一致的反馈，有时还会提供明显具有误导性的强度测量。
- 我们发现基于马尔可夫模型、PCFGs和RNNs的理论PSM方案表现最好。我们还发现一些网站和密码管理器有相当准确的强度计。然而，实际使用的强度计不如学术建议准确，与5年前相比，强度计的精度没有明显提高。高精度是影响密码强度计安全性的一个重要方面。

北京大学

第三部分

口令强度检测方案研究

- 方案研究
- 设计思路

目前主要的口令强度检测方法

1、基于攻击的方法

这类方法根据特定攻击(或一组攻击)破解口令的时间来衡量口令的强度。攻击需要的时间越长，口令安全性就越强。

- 优点：按照口令破解的思路设计，检测结果真实性更好。
- 缺点：评估结果是针对某种特定攻击；无法直接测试攻击时间，只能用间接方法估计口令强度。

目前主要的口令强度检测方法

2、基于启发式的方法

这类方法根据一个**基于启发式的口令复杂性度量规则**来评估，一般采用NIST发布的SP800-63标准。

根据口令的长度和所使用的字符类型(例如小写、大写或数字)等简单的规则，提出了用“熵比特”来度量口令强度。对于这类方法，长度越长、字符类型越多的口令安全性更强。

这类方法也被称作LUDS(lower- and uppercase letters, digits and symbols)，是目前使用最广泛的口令强度检测方法。

- 优点：方法简单，基于专家经验指定的规则；能有效检测出易受暴力破解的弱口令
- 缺点：过于侧重规则，机械测试，有局限性。例如：口令“david-1982”的评估结果高于“rpitsga”；“1234a!”被评为强口令。

目前主要的口令强度检测方法

3、基于概率的方法

这些方法大多是基于马尔可夫模型。将**人类使用口令的概率**来估计破解口令的可能性来衡量口令强度。以往的做法是根据现有的公共数据库来获得口令的概率。

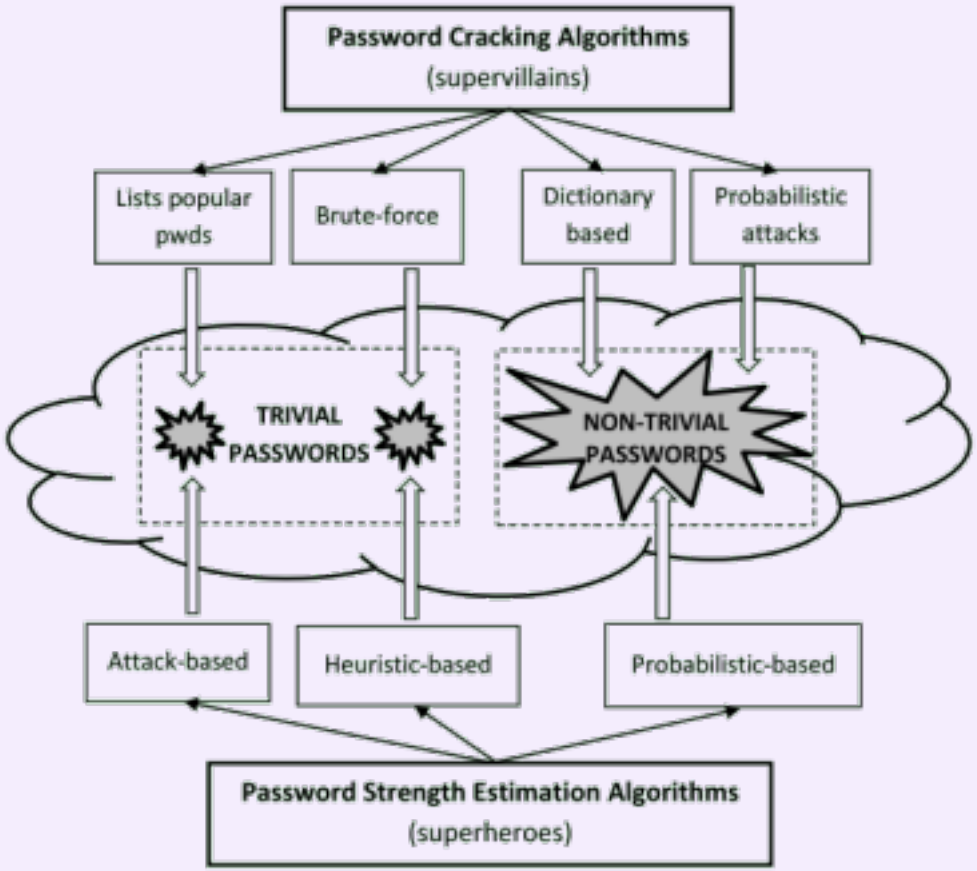
但大部分未出现在数据库中的口令无法可靠估计出现的概率，目前的做法是使用神经网络等技术构建统计模型，用一套大样本的口令训练数据，生成一个统计模型，能够尽可能表示所有可能的口令出现的概率。

- 优点：擅长检测不常用的口令的强度估计。
- 缺点：方法复杂；需要在通用性(即正确估计新的未见密码的强度)和准确性之间达成一个困难的折衷方案。

设计思路-想法

想法一

- 对于大多数情况，口令强度估计并**没有唯一有效的**解决方案。
- 对于不同口令，没有一种方法是普遍优于其他方法的，现有的口令强度检测方法都有各自的优缺点，没有一种方法是完全无用或可丢弃的。
- **将不同方法结合起来，优势互补**，从而生成一种更普遍、更准确和更可靠的整体方法。



设计思路-想法

想法二

- 同一个口令的强度可能随着环境不同而不同。

  例如，在一家使用西班牙语的公司中，使用匈牙利单词作为口令登录计算机可能是一个相当强的口令。然而，在匈牙利背景下使用的相同口令很可能被认为是弱口令。

- 口令强度检测算法不应该是不变的。它应该能够**适应不同的环境或者场景**，以便给出更准确的强度值(例如，根据不同的语言、不同的地区、家用公用等有所变化)。

- 举例：基于攻击的检测方法zxcvbn只适用于母语为英语的欧美国家。在zxcvbn中，口令“mima”强度远大于“secret”。

设计思路-想法

想法三

- 目前大部分web注册页面都没有在检测口令强度时，给出弱口令的改进建议。
- 口令强度检测的目的是为了帮助用户设置安全性更强的口令，所以在评估安全强度的同时，应该给出一些提高安全性的建议。
- 例如：

检测到用户使用重复字符（如：aaa、111）时，提示用户减少重复字符的使用。

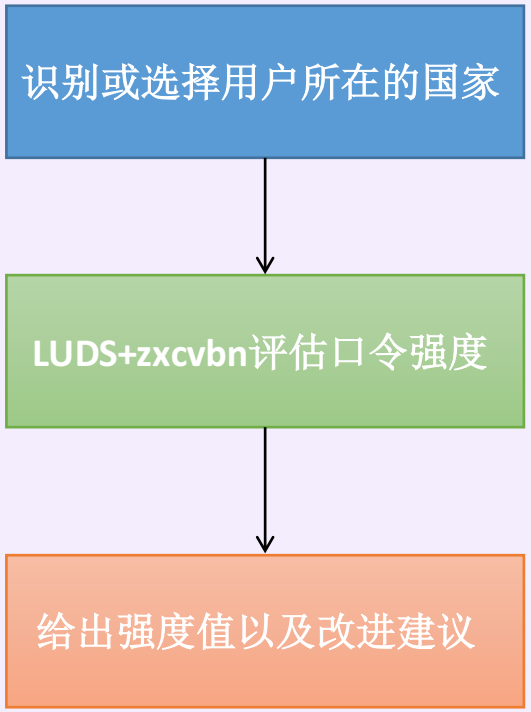
检测到用户使用连续字符（如：123456），提示用户口令容易被破解，建议使用无规律的字符。

设计思路

- 用户在输入口令之前，识别或者选择自己所在的国家。口令强度检测器会根据不同的国家和语言，采用不同的库对其进行检测。

- 口令检测进行两轮，综合使用基于启发式的方法和基于攻击的方法。
  第一轮使用LUDS给出评分a（满分100）；
  第二轮使用zxcvbn给出评分b（满分4）

- 结果：
  ①当a<60或b<2时,为弱口令，提示口令强度弱；
  ②其余情况强度值=a/100+b/4
  ③根据口令提出建议



设计思路

zxcvbn

- zxcvbn是基于攻击的方法，它针对人类的多数口令进行了某些假设。通过模式匹配和猜测估计，大概可以识别大约30K左右的常用密码。主要基于美国人口普查数据，维基百科，美国电影，电视流行词以及其它一些常用口令模式（如：日期，重复字符，序列字符，键盘模式等）。

- zxcvbn主要由三个部分组成：匹配、估计和搜索。（假设用“lenovo1111”作为口令输入）

（1）匹配阶段

输入口令，模式匹配阶段将查找一组与口令部分重叠的匹配模式。

可能返回lenovo (password token), eno (English “one” backwards), no (English), no (English), “on” backwards), 1111 (repeat pattern), and 1111 (Date pattern, 1/1/2011)

（2）估计阶段

估计阶段分别为每个匹配分配一个猜测估计值。

例如：如果lenovo是我们的一本密码词典中最常见的11007个密码，那么它将被分配到11007，因为攻击者按使用最多的顺序遍历该字典，在到达它之前需要这么多猜测。

（3）搜索阶段

最后阶段是搜索从中提取的不重叠的相邻匹配的序列，并且使总猜测数最小。

例如在本例中，搜索步骤将返回[lenovo (token), 1111(repeat)]，其中对字符串“1111”放弃需要更多猜测的日期模式（date），而选择字符重复模式（repeat）。

设计思路

zxcvbn具体算法原理

（1）匹配

匹配阶段有以下几种模式，分别会有对应的匹配器（函数）进行检测

pattern	examples
token	logitech l0giT3CH ain't parliamentarian 1232323q
reversed	DrowssaP
sequence	123 2468 jklm ywusq
repeat	zzz ababab l0giT3CHl0giT3CH
keyboard	qwertyuio qAzxcde3 diueoa
date	7/8/1947 8.7.47 781947 4778 7-21-2011 72111 11.7.21
bruteforce	x$JQhMzt

- token matcher将输入的密码转换成小写字母，并检查各子字符串在频率排序的字典中的类别。此外，在检查之前，还会根据l33t表将@映射到a，并将l映射到i或l，则它将通过分段[@->a，1->i]和[@->a，1->l]来尝试两个额外的匹配。
- sequence matching查找每个字符都是从最后一个字符到固定的Unicode码点距离的序列。
- repeat matcher搜索一个或多个字符的重复块，它只匹配单个字符重复。
- keyboard matcher 线性地遍历口令，根据每个键盘邻接图查找相邻键链。
- date matching日期匹配考虑4到8个字符的数字区域，检查表以查找可能的拆分，并尝试对每个拆分进行一天月年映射，这样，年不是在中间，月份在1到12之间，日是1到31包含在内。
- 未被上述匹配器检测到的归为暴力破解（bruteforce）

设计思路

zxcvbn具体算法原理

（2）评估

确定每个匹配的猜测次数，猜测次数估计的依据如下：

1、对于token模式，使用频率等级作为估计，因为攻击者根据口令使用频繁程度猜测token至少需要多次尝试。对于（反向）reversed的token，尝试次数加倍，因为攻击者随后需要尝试对每个token进行两次猜测(正常和反向)。计算公式：$\frac{1}{2}\sum_{i=1}^{min(U,L)}\binom{U+L}{i}$ 其中，U和L是token中大写字母和小写字母的个数。

2、对于keyboard模式，估计的猜测数计算公式：$\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{min(T,i-i)}\binom{i-1}{j-1}SD^{j}$ 其中L是模式的长度，T是项的数目，D是每个键的平均邻居数，S是键盘上的键数。

3、对于repeat模式，重复匹配对象由重复n次的基组成，其中递归匹配估计搜索步骤先前为基分配了许多猜测g。然后估计重复猜测尝试数为**g•n**。例如nownownow需要126次猜测，因为now在 Wiktionary集中排名42位，now重复3次。

4、对于Sequences模式，依据**s•n•|d|**打分。其中s是可能的起始字符数，n是长度，d是编码点增量(例如，9753中的-2)。对于1和z这种第一字符，s一般设置为一个小常数4，对于其他数字，s设置为10，其他字符，s设置为26.

5、对于date模式，我们假设猜测者从2016年开始，并逐步猜测更早或更晚的日期，得出大约**365•|2016-年份|**次的猜测。

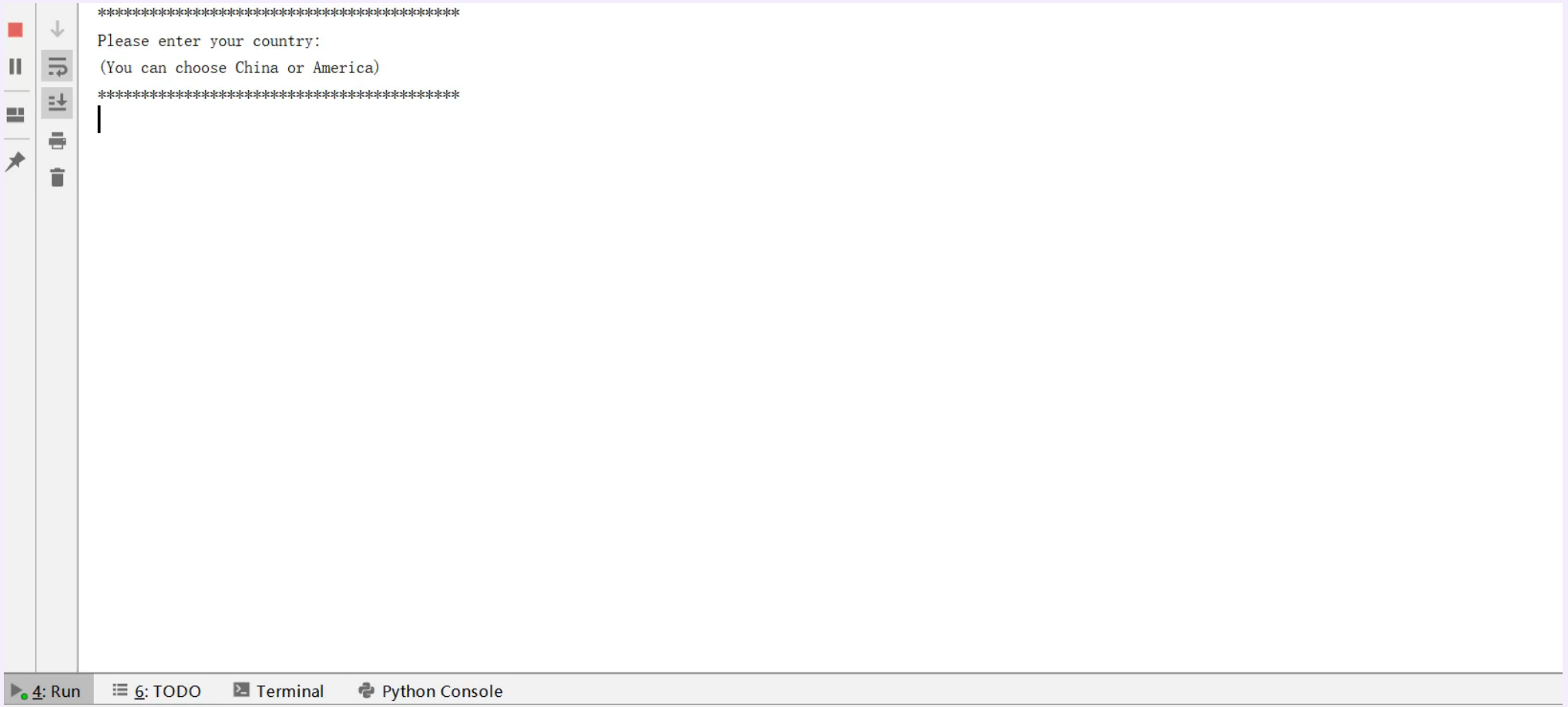
6、对于bruteforce，每个字符默认猜测次数C=10次，字符长度L需要猜测次数**C的L次方**。如果知道是字母，则C=26，如果是数字则C=10，其他未知字符则C=33.

北京大学

第四部分

演示系统展示

演示系统


****
Please enter your country:
(You can choose China or America)
****
4: Run
6: TODO
Terminal
Python Console

参考文献

- Javier Galbally, Iwen Coisel, and Ignacio Sanchez ,*A New Multimodal Approach for Password Strength Estimation—Part I: Theory and Algorithms*
- Daniel Lowe Wheeler,*zxcvbn: Low-Budget Password Strength Estimation*
- Blase Ur, Sean M. Segreti, Lujo Bauer, *Measuring Real-World Accuracies and Biases in Modeling Password Guessability*
- *De Carnavalet X D C, Mannan M. From Very Weak to Very Strong: Analyzing Password-Strength Meters[C]//NDSS. 2014, 14: 23-26.*
- *Golla M, Dürmuth M. On the Accuracy of Password Strength Meters[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018: 1567-1582.*
- *Ur B, Kelley P G, Komanduri S, et al. How does your password measure up? The effect of strength meters on password creation[C]//USENIX Security Symposium. 2012: 65-80.*
- *Egelman S, Sotirakopoulos A, Muslukhov I, et al. Does my password go up to eleven?: the impact of password meters on password selection[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013: 2379-2388.*
- *Lyastani S G, Schilling M, Fahl S, et al. Studying the Impact of Managers on Password Strength and Reuse[J]. arXiv preprint arXiv:1712.08940, 2017.*

北京大学
PEKING UNIVERSITY

Thanks for listening !